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

disusun oleh:
Kelompok 47 - Cepalta

Indah Novita Tangdililing	13523047
Aliya Husna Fayyaza	13523062
Bevinda Vivian	13523120

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

2024

Kata Pengantar

Kami ucapkan syukur kepada Tuhan Yang Maha Esa atas rahmat dan petunjuk-Nya yang memungkinkan kami menyelesaikan laporan ini tepat waktu. Dokumen ini merupakan sebuah laporan yang disusun untuk memenuhi tugas besar 1 dalam mata kuliah IF2123 Aljabar Linier dan Geometri. Selain itu, laporan ini juga bertujuan untuk meningkatkan pemahaman kami mengenai sistem persamaan linier, determinan, dan aplikasinya dalam bahasa pemrograman Java.

Terima kasih kami ucapkan kepada Bapak Dr. Ir. Rinaldi, M.T. yang telah menjadi dosen pengampu mata kuliah di kelas K-02 IF2123 Aljabar Linier dan Geometri. Selain itu kami juga ingin mengungkapkan terima kasih kepada Kak Ahmad Naufal Ramadan (13522005) yang senantiasa membantu kami sebagai Asisten dalam Tugas Besar Aljabar Linier dan Geometri 1 ini. Kami sadar bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, jika terdapat kesalahan dalam penulisan atau ketidaksesuaian dalam materi yang kami sampaikan dalam laporan ini, kami mohon maaf. Tentunya kami juga sangat mengharapkan saran dan kritik yang membangun untuk meningkatkan kualitas tugas besar ataupun laporan ini.

Bandung, 24 Oktober 2024

Kelompok 47
Cepalta

Daftar Isi

Daftar Tabel

Daftar Gambar

Bab 1

Deskripsi Masalah

Dalam Tugas Besar 1 Aljabar Linear dan Geometri, mahasiswa diminta untuk mengembangkan satu atau lebih library aljabar linier dalam Bahasa Java, yang mencakup fungsi-fungsi untuk menyelesaikan SPL melalui metode eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks balikan, serta kaidah Cramer. Selain itu, library ini juga harus mampu menghitung determinan dan invers matriks. Tugas ini mencakup penyelesaian persoalan interpolasi polinomial, di mana mahasiswa harus menentukan polinom yang melewati sejumlah titik data, serta regresi, di mana prediksi nilai dilakukan berdasarkan variabel yang tersedia. Program yang dikembangkan harus dapat menerima input dari keyboard maupun file teks, dan menghasilkan output yang sesuai dengan permasalahan yang dihadapi. Berikut ini deskripsi masalah lengkap yang telah diberikan kepada mahasiswa dalam Tugas Besar 1 Aljabar Linear dan Geometri:

1. 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.
2. 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).

Spesifikasi program 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 a_{ij} , dan b_i. Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

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

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

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

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

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

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

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

5. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x_{1i}, x_{2i}, …, x_{ni}, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
6. Untuk persoalan SPL, luaran program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).
7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

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

dan untuk regresi adalah

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

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

```
x1 = X
.
x3 = X²
.
x5 = 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 Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi linier dan kuadratik berganda
7. Interpolasi Gambar (Bonus)
8. Keluar
```

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

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

Begitu juga untuk pilihan menu nomor 2, 3, dan 6.

Bab 2

Teori Singkat

2.1 Sistem Persamaan Linier

Sistem Persamaan Linier (SPL) adalah sekumpulan persamaan linier yang terdiri atas beberapa variabel. SPL sendiri merupakan permasalahan yang sering ditemukan di bidang sains dan rekayasa. Solusi dari SPL dapat berupa satu (unik/tunggal), tidak ada, atau banyak (tidak berhingga) solusi. Metode untuk mendapatkan solusi dari SPL sendiri di antaranya terdiri dari metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan juga kaidah Cramer.

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

Gambar 1. Sistem Persamaan Linier yang Terdiri Atas m Buah Persamaan dan n Buah Peubah

2.2 Metode Eliminasi Gauss dan Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss adalah teknik yang digunakan untuk menyelesaikan sistem persamaan linier dengan mengubahnya menjadi bentuk eselon baris. Proses ini melibatkan penghapusan variabel secara bertahap untuk mempermudah pencarian solusi. Dalam metode ini, baris-baris dalam matriks augmented dimodifikasi dengan melakukan operasi baris elementer.

Operasi Baris Elementer (OBE) yang dimaksud merupakan operasi penjumlahan dan perkalian yang dilakukan terhadap matriks augmented untuk menghasilkan matriks eselon baris atau matriks eselon baris tereduksi, rincian operasi tersebut adalah:

1. Pertukaran dua buah baris.

2. Penjumlahan sebuah baris dengan kelipatan baris lainnya.
3. Perkalian sebuah baris dengan konstanta tidak nol.

Metode eliminasi Gauss-Jordan merupakan pengembangan dari metode eliminasi Gauss, yang tidak hanya menghasilkan bentuk eselon baris tetapi juga bentuk eselon baris tereduksi. Dalam metode ini, elemen pada suatu baris elemen pertamanya merupakan 1 maka pada kolom elemen tersebut yang lainnya harus bernilai 0, sehingga memungkinkan penentuan solusi secara langsung.

Terdapat sifat-sifat matriks eselon baris yang harus diterapkan untuk menggunakan metode eliminasi Gauss dan Gauss-Jordan, yaitu:

1. Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama).
2. Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
3. Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.

untuk matriks eselon baris tereduksi (metode eliminasi Gauss-Jordan) terdapat sifat tambahan yaitu:

4. Setiap kolom yang memiliki 1 utama memiliki nol di tempat lain.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & -5 & -15 \end{bmatrix} \overset{R3/(-5)}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Gambar 2. Metode Eliminasi Gauss

$$\begin{bmatrix} 1 & 0 & -1 & -3 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & \mathbf{0} & 0 & 1 & 0 \\ 0 & \mathbf{0} & 0 & -10 & 0 \end{bmatrix} \begin{matrix} R1+3R3 \\ \sim \\ R2-2R3 \\ R4+10R3 \end{matrix} \begin{bmatrix} 1 & 0 & -1 & \mathbf{0} & 0 \\ 0 & 1 & 1 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathbf{0} & 0 \end{bmatrix}$$

Gambar 3. Metode Eliminasi Gauss-Jordan

2.3 Matriks Kofaktor, Matriks Adjoin, dan Matriks Balikan

Matriks kofaktor adalah matriks yang dihasilkan dengan mengganti setiap elemen dalam matriks asal dengan kofaktornya. Kofaktor dari suatu elemen adalah hasil dari determinan submatriks yang diperoleh dengan menghapus baris dan kolom yang mengandung elemen tersebut, dikalikan dengan −1(i+j), di mana i dan j adalah indeks dari elemen. Matriks kofaktor digunakan dalam perhitungan determinan dan dalam menghitung invers matriks menggunakan metode adjoin.

$$A = \begin{bmatrix} 6 & -3 & 1 \\ 2 & 2 & -4 \\ 1 & 5 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} +\begin{vmatrix} 6 & 3 \\ -4 & 0 \end{vmatrix} & -\begin{vmatrix} 1 & 3 \\ 2 & 0 \end{vmatrix} & +\begin{vmatrix} 1 & 6 \\ 2 & -4 \end{vmatrix} \\ -\begin{vmatrix} 2 & -1 \\ -4 & 0 \end{vmatrix} & +\begin{vmatrix} 3 & -1 \\ 2 & 0 \end{vmatrix} & -\begin{vmatrix} 3 & 2 \\ 2 & -4 \end{vmatrix} \\ +\begin{vmatrix} 2 & -1 \\ 6 & 3 \end{vmatrix} & -\begin{vmatrix} 3 & -1 \\ 1 & 3 \end{vmatrix} & +\begin{vmatrix} 3 & 2 \\ 1 & 6 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

Gambar 4. Contoh Mencari Matriks Kofaktor

Matriks adjoin (atau adjugate yang sering disingkat dengan adj) adalah transpose dari matriks kofaktor. Transpose sendiri dimaksud dengan menukar elemen pada baris menjadi kolom atau elemen pada kolom menjadi baris. Matriks ini digunakan untuk menghitung invers matriks. Berdasarkan **Gambar 4**, matriks adjoin dari contoh tersebut adalah:

$$\mathrm{adj}(A) = \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$$

Gambar 5. Contoh Mencari Matriks Adjoin

Matriks balikan (invers matriks) adalah matriks yang, jika dikalikan dengan matriks asal, akan menghasilkan matriks identitas. Hanya matriks persegi yang memiliki invers. Untuk menghitung invers, metode yang umum digunakan adalah metode adjoin dan metode eliminasi Gauss. Invers matriks sangat penting dalam penyelesaian sistem persamaan linier, karena dapat digunakan untuk menemukan solusi dari persamaan Ax=b dengan rumus

$$x = A^{-1}b$$

Matriks balikan A dapat dihitung dengan menggunakan rumus:

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

2.4 Determinan

Determinan adalah nilai yang dihasilkan dari suatu matriks persegi dan memiliki berbagai aplikasi dalam aljabar linier, termasuk dalam menentukan keberadaan solusi dari sistem persamaan linier. Untuk matriks 2x2 dapat dicari determinan dengan cara: Misal A = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, maka determinannya adalah det(A) = ad – bc. Untuk matriks 3x3 dapat dihitung dengan metode lain yaitu metode ekspansi kofaktor. Misal A = $\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$, maka determinannya adalah:

$$C_{ij} = (-1)^{i+j} M_{ij} = \text{Kofaktor Entri } a_{ij}$$

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

Secara baris

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

Secara kolom

Gambar 6. Persamaan Untuk Mencari Determinan Menggunakan Ekspansi Kofaktor

Determinan yang tidak sama dengan nol menunjukkan bahwa sistem persamaan memiliki solusi unik, sedangkan determinan nol menunjukkan bahwa sistem tidak memiliki solusi atau memiliki solusi tak hingga.

2.5 Kaidah Cramer

Kaidah Cramer adalah metode untuk menyelesaikan sistem persamaan linier dengan menggunakan determinan. Metode ini berlaku untuk sistem dengan jumlah persamaan sama dengan jumlah variabel, dan solusi untuk setiap variabel dihitung dengan membagi determinan dari matriks koefisien dengan determinan dari matriks yang dibentuk dengan mengganti kolom yang sesuai.

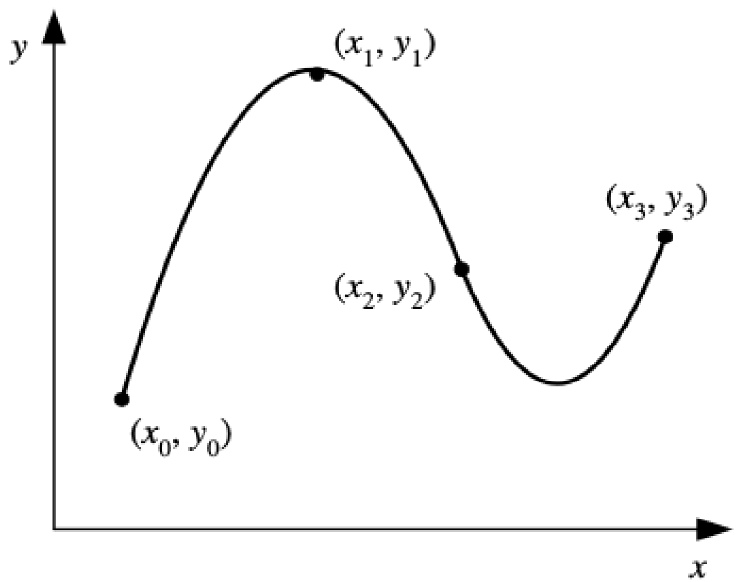
Jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variabel) sedemikian sehingga $det(A) \neq 0$, maka SPL tersebut memiliki solusi yang unik yaitu:

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

2.6 Interpolasi Polinom

Interpolasi polinom adalah proses menemukan polinom yang melewati sejumlah titik yang diketahui. Metode yang umum digunakan dalam interpolasi adalah interpolasi Lagrange dan interpolasi Newton. Polinom yang dihasilkan dapat digunakan untuk memperkirakan nilai di antara titik data, sehingga sangat berguna dalam aplikasi seperti pengolahan sinyal dan analisis data.

Polinom interpolasi derajat n yang melalui titik-titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ adalah $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$



Gambar 7. Ilustrasi Interpolasi Polinomial Untuk Beberapa Titik

2.7 Interpolasi Bicubic Spline

Interpolasi bicubic spline adalah metode interpolasi yang menggunakan polinom kubik untuk mengaproksimasi data pada grid dua dimensi. Dibandingkan dengan interpolasi bilinear yang hanya memperhitungkan 4 piksel (2x2), bicubic mempertimbangkan 16 piksel (4x4), sehingga hasilnya lebih halus dan detail. Metode ini sering digunakan dalam pemrosesan citra, terutama saat kualitas hasil lebih penting daripada kecepatan. Bicubic dapat diterapkan dengan menggunakan polinomial Lagrange, spline kubik, atau algoritma konvolusi kubik, dan pilihan nilai *b* dan *c* dapat memengaruhi artefak yang muncul pada gambar yang dihasilkan.

Permukaan yang diinterpolasi dapat memanfaatkan model turunan berarah dari kedua sumbu, baik sumbu x maupun sumbu y. Persamaan polinomial untuk membentuk sebuah matriks solusi X yang digunakan adalah:

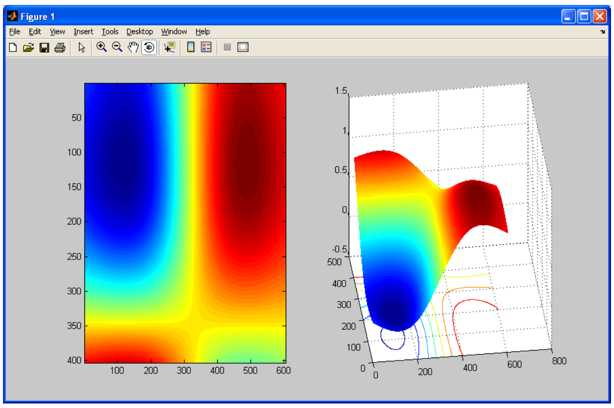
$$p(x,y) = \sum^{3} \sum^{3} a_{ij} x^i y^j .$$

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

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

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

Gambar 8. Persamaan Polinomial dalam Bicubic Spline



Gambar 9. Contoh Penerapan Bicubic Spline dalam Image Processing

2.8 Regresi Linier

Regresi linier adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen dan variabel dependen. Dalam regresi linier sederhana, model yang dihasilkan berbentuk garis lurus, sedangkan regresi linier berganda melibatkan beberapa variabel independen. Model ini digunakan untuk memprediksi nilai variabel dependen berdasarkan nilai variabel independen.

Secara umum model regresi linear dinyatakan dengan rumus berikut:

$$\bar{Y}_t = a + bX \qquad a = \frac{\sum Y \sum X^2 - \sum X \sum XY}{n \sum X^2 - (\sum X)^2} \qquad b = \frac{n \sum XY - \sum X \sum Y}{n \sum X^2 - (\sum X)^2}$$

Y = Variabel dependen
X = Variabel independen
a = Konstanta
b = koefisien regresi

Gambar 10. Rumus Umum Regresi Linier

2.9 Regresi Kuadratik Berganda

Regresi kuadratik berganda adalah bentuk dari regresi yang memungkinkan hubungan non-linier antara variabel independen dan variabel dependen dengan menyertakan suku kuadrat dari variabel independen. Model ini dapat menangkap pola yang lebih kompleks dibandingkan dengan regresi linier sederhana. Metode ini sering digunakan dalam analisis data yang menunjukkan perilaku non-linier.

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + e_i \text{ for } i = 1, 2, \ldots, n, \qquad \hat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T Y,$$

$$\begin{aligned} y &= \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{j=1}^{p} \sum_{k=j}^{p} \beta_{j,k} x_j x_k, \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \beta_{1,1} x_1^2 + \beta_{1,2} x_1 x_2 + \cdots + \beta_{1,p} x_1 x_p + \beta_{2,2} x_2^2 + \beta_{2,3} x_2 x_3 + \cdots \\ &\quad + \beta_{2,p} x_2 x_p + \cdots + \beta_{p-1,p-1} x_{p-1}^2 + \beta_{p-1,p} x_{p-1} x_p + \beta_{p,p} x_p^2, \end{aligned}$$

Gambar 11. Bentuk Umum Regresi Kuadratik Berganda

(y = model regresi linear berganda, b = vektor parameter regresi)

Bab 3

Implementasi Pustaka dan Program dalam Java

Di dalam tugas besar 1, kami membuat 5 buah package. Dengan package pertama bernama “algebra”, package kedua bernama “interpolation”, package ketiga bernama “menu”, package keempat bernama “regression”, dan package kelima bernama “utils”.

3.1 algebra

3.1.1 Determinan.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari determinan.

Tabel 3.1.1 Deskripsi Fungsi dan Prosedur Pada Class Determinan.java

Fungsi/Prosedur	Deskripsi
determinanEkspansiKofaktor(Matriks m)	Menghitung determinan matriks menggunakan metode ekspansi kofaktor pada baris pertama.
determinanReduksiBaris(Matriks m)	Menghitung determinan menggunakan eliminasi baris dan perkalian elemen diagonal utama matriks.
StringHasil(double det)	Menghasilkan output dalam bentuk string yang menyatakan nilai determinan matriks.

3.1.2 Inverse.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari invers matriks.

Tabel 3.1.2 Deskripsi Fungsi dan Prosedur Pada Class Inverse.java

Fungsi/Prosedur	Deskripsi
transpose(double[][] matriks)	Menghitung transpose dari matriks input dengan menukar baris menjadi kolom dan sebaliknya.
invers(Matriks m)	Menghitung invers dari matriks input dengan mencari adjoint dan membaginya dengan determinan matriks.
StringHasil(double[][] inv)	Menghasilkan output dalam bentuk string yang menampilkan elemen-elemen matriks invers.

3.1.3 Matriks.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk membuat sebuah matriks.

Tabel 3.1.3 Deskripsi Fungsi dan Prosedur Pada Class Matriks.java

Fungsi/Prosedur	Deskripsi
Matriks(String filePath)	Konstruktor untuk membuat matriks dari file, menggunakan FileHandler untuk membaca data matriks dari file.
Matriks(int row, int col, boolean isEmpty)	Konstruktor untuk membuat matriks kosong dengan ukuran tertentu (biasanya digunakan untuk membuat kofaktor).
Matriks(int row, int col, int code)	Konstruktor untuk membuat matriks dari input pengguna, dengan opsi memasukkan elemen xToPredict berdasarkan code.
tulisMatriks()	Menulis elemen-elemen matriks ke konsol.
kofaktor(int p, int q)	Membentuk submatriks dengan mengabaikan baris ke-p dan kolom

	ke-q, digunakan untuk perhitungan kofaktor.
submatriks(int rowExclude, int colExclude)	Membentuk submatriks dengan mengabaikan baris dan kolom tertentu, digunakan dalam perhitungan kofaktor.
getElement(int i, int j)	Mengembalikan elemen matriks pada baris i dan kolom j.
getRow()	Mengembalikan jumlah baris dari matriks.
getCol()	Mengembalikan jumlah kolom dari matriks.

3.1.4 SPL.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari sistem persamaan linier.

Tabel 3.1.4 Deskripsi Fungsi dan Prosedur Pada Class SPL.java

Fungsi/Prosedur	Deskripsi
gauss(double[][] A, double[] B)	Menggunakan eliminasi Gauss untuk mencari solusi SPL, dengan pengecekan untuk kasus tidak ada solusi atau variabel bebas.
getParametricSolution(double[][] A, double[] B)	Menghasilkan solusi parametrik untuk SPL jika terdapat variabel bebas.
ParametrikStringHasil(double[][] A, double[] B)	Menghasilkan representasi string untuk solusi parametrik SPL dalam bentuk variabel seperti s, t, atau u.
StringHasil(double[] solusi)	Menghasilkan representasi string untuk solusi SPL yang ditemukan.
gaussJordan(double[][] A, double[] B)	Menggunakan metode Gauss-Jordan untuk mencari solusi SPL, dengan pengecekan variabel bebas dan normalisasi pivot.

cramer(Matriks A, double[] B)	Menggunakan aturan Cramer untuk menyelesaikan SPL jika determinan matriks tidak nol.
splbalikan(Matriks A, double[] B)	Menyelesaikan SPL menggunakan metode invers matriks jika matriks berbentuk persegi dan memiliki solusi unik.

3.2 interpolation

3.2.1 BicubicSplineInt.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk menghitung interpolasi bicuib.

Tabel 3.2.1 Deskripsi Fungsi dan Prosedur Pada Class BicubicSplineInt.java

Fungsi/Prosedur	Deskripsi
bicubicInterpolation(double[][] inputMatrix, double a, double b)	Menghitung interpolasi bicubic untuk matriks input (4x4), dan menampilkan hasil interpolasi pada titik (a, b). Prosesnya melibatkan pembentukan matriks X (16x16), vektor B (16), penyelesaian sistem persamaan linear dengan metode Gauss, dan evaluasi hasil interpolasi.

3.2.2 PolynomialInt.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari invers matriks.

Tabel 3.2.2 Deskripsi Fungsi dan Prosedur Pada Class CubicSplineInt.java

Fungsi/Prosedur	Deskripsi
interpolasiGauss(double[][] A, double[] B, double titik)	Menggunakan eliminasi Gauss untuk menyelesaikan sistem persamaan linear dan membentuk persamaan

	polinomial interpolasi. Mengembalikan nilai polinom pada titik yang diestimasi.
interpolasiGaussJordan(double[][] A, double[] B, double titik)	Menggunakan eliminasi Gauss-Jordan untuk menyelesaikan sistem persamaan linear dan membentuk persamaan polinomial interpolasi. Mengembalikan nilai polinom pada titik yang diestimasi.

3.3 menu

3.3.1 Main.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk menjalankan program utama.

Tabel 3.3.1 Deskripsi Fungsi dan Prosedur Pada Class Main.java

Fungsi/Prosedur	Deskripsi
main(String[] args)	Menjalankan program utama dengan menampilkan menu pilihan yang memungkinkan pengguna untuk memilih berbagai operasi matriks seperti SPL, determinan, matriks balikan, interpolasi polinom, interpolasi bicubic spline, dan regresi. Program berlanjut sampai pengguna memilih untuk keluar.

3.3.2 SubMenu.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk menampilkan bagian dari menu.

Tabel 3.3.2 Deskripsi Fungsi dan Prosedur Pada Class SubMenu.java

Fungsi/Prosedur	Deskripsi
InputChoice()	Menampilkan pilihan input (dari file atau manual) dan membaca pilihan

	pengguna.
MatrixSource(int choice, int code)	Menghasilkan objek matriks berdasarkan pilihan input pengguna (file atau manual), dengan menangani kesalahan file.
SPL()	Menyediakan submenu untuk menyelesaikan Sistem Persamaan Linier (SPL) dengan berbagai metode (Gauss, Gauss-Jordan, Cramer, Invers).
Determinan()	Menyediakan submenu untuk menghitung determinan matriks menggunakan metode ekspansi kofaktor atau reduksi baris.
Inverse()	Menghitung invers dari sebuah matriks dan menampilkan hasilnya.
InterpolasiPol()	Menyediakan submenu untuk interpolasi polinomial (belum diimplementasikan lengkap dalam kode yang diberikan).
InterpolasiBic()	Menggunakan metode interpolasi bicubic untuk menghitung nilai interpolasi dan menampilkan hasilnya.
Regresi()	Menyediakan submenu untuk regresi (linear berganda atau kuadratik berganda) dan menampilkan hasil regresi.

3.4 regression

3.4.1 MultipleLinearRegression.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari regresi linier berganda.

Tabel 3.4.1 Deskripsi Fungsi dan Prosedur Pada Class MultipleLinearRegression.java

Fungsi/Prosedur	Deskripsi
ubahInputMenjadiMatriks(double[][] titikSampel)	Mengubah input titik sampel (x, y) menjadi matriks X yang siap untuk dihitung dalam regresi linear, dengan kolom pertama selalu 1 untuk koefisien konstan (β0).
pisahkanNilaiY(double[][] titikSampel)	Memisahkan nilai y dari sampel input untuk membentuk vektor Y.
regresiLinearGauss(double[][] titikSampel)	Menghitung koefisien regresi linear dengan eliminasi Gauss, mengembalikan koefisien yang merepresentasikan persamaan regresi.
prediksi(double[] beta, double xBaru)	Menghitung nilai prediksi y untuk nilai x baru berdasarkan koefisien regresi yang telah dihitung.
StringHasil(double[] hasilReg, double pred, double xBaru)	Menghasilkan representasi string dari persamaan regresi dan hasil prediksi untuk nilai x baru.

3.4.2 regresiKuadratikBerganda.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari regresi kuadratik berganda.

Tabel 3.4.2 Deskripsi Fungsi dan Prosedur Pada Class regresiKuadratikBerganda.java

Fungsi/Prosedur	Deskripsi
SampleToMatrix(Matriks file, int SampleNum)	Mengubah data sampel dari objek matriks menjadi matriks kuadratik yang mencakup variabel linear, kuadrat, dan interaksi.
SampleToYMatrix(Matriks file, int SampleNum)	Mengubah data sampel menjadi vektor Y untuk regresi kuadratik berganda berdasarkan nilai variabel yang diberikan.

toMatrixRKB(Matriks file)	Membentuk matriks X untuk regresi kuadratik berganda dengan memasukkan semua data sampel.
toMatrixYRKB(Matriks file)	Membentuk matriks Y untuk regresi kuadratik berganda dengan memasukkan semua data sampel.
gaussRKB(Matriks file)	Menggunakan eliminasi Gauss untuk menyelesaikan sistem persamaan linear dalam regresi kuadratik berganda dan memperoleh koefisien.
prediksi(double[] hasilSPL, double[] xk)	Menghitung prediksi hasil regresi kuadratik berganda untuk nilai variabel input xk berdasarkan koefisien yang diperoleh.
StringHasil(Matriks file)	Menghasilkan representasi string dari hasil regresi kuadratik berganda, termasuk koefisien variabel dan fungsi regresi yang diperoleh.

3.5 utils

3.5.1 FileHandler.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari determinan.

Tabel 3.5.1 Deskripsi Fungsi dan Prosedur Pada Class FileHandler.java

Fungsi/Prosedur	Deskripsi
FileHandler(String filePath)	Konstruktor untuk membaca data dari file dan menentukan ukuran matriks serta apakah ada baris tambahan

<table>
<tr><td></td><td>untuk prediksi (xToPredict). Membaca dan menyimpan data dalam matriks serta nilai prediksi jika ada.</td></tr>
<tr><td>FileWriter(String filePath, String string)</td><td>Metode statis untuk menulis string ke dalam file dengan jalur yang diberikan, dan memberikan pesan sukses jika penulisan berhasil.</td></tr>
<tr><td colspan="2">Keterangan Tambahan:
- Konstruktor FileHandler: Membaca file dan menghitung jumlah baris dan kolom, kemudian menyimpan data dalam bentuk matriks. Jika baris terakhir memiliki kolom lebih sedikit, diasumsikan sebagai data prediksi (xToPredict).
- Penanganan FileWriter: Metode untuk menulis data string ke file, biasanya digunakan untuk menyimpan hasil perhitungan.</td></tr>
</table>

3.5.2 MatrixUtils.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari invers matriks.

Tabel 3.5.2 Deskripsi Fungsi dan Prosedur Pada Class MatrixUtils.java

<table>
<tr><th>Fungsi/Prosedur</th><th>Deskripsi</th></tr>
<tr><td>swap_rows(double[][] mat, int row1, int row2)</td><td>Menukar dua baris dalam matriks mat antara row1 dan row2.</td></tr>
<tr><td>multiply_row(double[][] mat, int row, double scal)</td><td>Mengalikan semua elemen dalam baris row dari matriks mat dengan sebuah skalar scal.</td></tr>
<tr><td>add_other_row(double[][] mat, int row, int added_row, double scal)</td><td>Menambahkan elemen-elemen dari baris added_row yang telah dikalikan dengan scal ke baris row dari matriks mat.</td></tr>
<tr><td colspan="2">Keterangan Tambahan:
- swap_rows: Digunakan dalam metode eliminasi Gauss atau Gauss-Jordan untuk menukar baris matriks saat mencari elemen pivot.
- multiply_row: Berguna untuk menormalisasi baris matriks dengan membagi elemen baris oleh nilai elemen pivot.
- add_other_row: Digunakan untuk membuat elemen-elemen di bawah atau di atas pivot menjadi nol dalam metode eliminasi baris.</td></tr>
</table>

3.5.3 general.java

Pada class ini terdapat fungsi-fungsi yang digunakan untuk mencari invers matriks.

Tabel 3.5.2 Deskripsi Fungsi dan Prosedur Pada Class MatrixUtils.java

Fungsi/Prosedur	Deskripsi
factorial(int num)	Menghitung nilai faktorial dari num secara rekursif. Mengembalikan 1 jika num adalah 0, atau num * factorial(num-1) untuk nilai lainnya.
nCr(int n, int r)	Menghitung kombinasi (nCr) untuk memilih r elemen dari n elemen, dengan menggunakan rumus n! / (r! * (n-r)!). Mengembalikan 0 jika n kurang dari atau sama dengan 1.
nCrMember(int n, int r)	Menghasilkan array 2D yang berisi semua kombinasi dua elemen dari n elemen, dengan masing-masing kombinasi terdiri dari dua angka.
Keterangan Tambahan: - factorial(int num): Berguna untuk menghitung faktorial yang diperlukan dalam kombinasi. - nCr(int n, int r): Berguna untuk menghitung jumlah kombinasi dari n objek yang dipilih r sekaligus. - nCrMember(int n, int r): Menghasilkan semua pasangan unik dari elemen yang dapat dipilih, yang berguna dalam perhitungan interaksi dalam regresi.	

Bab 4

Eksperimen

4.1 Tampilan Antarmuka

```
=========================
|        CEPALTA        |
=========================
===========MAIN==========
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi
0. KELUAR
```

Gambar 13. Main Menu Program

```
Pilih metode penyelesaian SPL:
1. Metode Gauss
2. Metode Gauss-Jordan
3. Metode Cramer
4. Metode Matriks Balikan
```

Gambar 14. Tampilan Pilihan Sistem Persamaan Linier

```
Pilih metode penyelesaian determinan:
1. Metode Ekspansi Kofaktor
2. Metode Reduksi Baris
```

Gambar 15. Tampilan Pilihan Determinan

```
Pilih metode regresi:
1. Linear Berganda
2. Kuadratik Berganda
```

Gambar 16. Tampilan Pilihan Regresi

```
1. Input dari file
2. Input secara manual
2
Jumlah baris (m):
2
Jumlah kolom (n):
2
Masukkan elemen-elemen matriks:
Elemen [0][0]: 1
Elemen [0][1]: 1
Elemen [1][0]: 1
Elemen [1][1]: 1
```

Gambar 17. Tampilan Input

```
Elemen [3][2]: -4
Elemen [3][3]: 2
Elemen [3][4]: 6
Tidak ada solusi.
```

```
Elemen [3][3]: -3
Elemen [3][4]: -3
Solusi dalam bentuk parametrik:
x1 = -3.0 + (3.0) * s
x2 = 0.0 + (4.0) * s
x3 = s
x4 = t
```

Gambar 18. Tampilan Output

4.2 Studi dan Tes Kasus

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

a.

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

Solusi:

```
Pilih metode penyelesaian SPL:
1. Metode Gauss
2. Metode Gauss-Jordan
3. Metode Cramer
4. Metode Matriks Balikan
1
1. Input dari file
2. Input secara manual
2
Jumlah baris (m):
4
Jumlah kolom (n):
5
Masukkan elemen-elemen matriks:
Elemen [0][0]: 1
Elemen [0][1]: 1
Elemen [0][2]: -1
Elemen [0][3]: -1
Elemen [0][4]: 1
Elemen [1][0]: 2
Elemen [1][1]: 5
Elemen [1][2]: -7
Elemen [1][3]: -5
Elemen [1][4]: -2
Elemen [2][0]: 2
Elemen [2][1]: -1
Elemen [2][2]: 1
Elemen [2][3]: 3
Elemen [2][4]: 4
Elemen [3][0]: 5
Elemen [3][1]: 2
Elemen [3][2]: -4
Elemen [3][3]: 2
Elemen [3][4]: 6
Tidak ada solusi.
```

2. SPL berbentuk matriks *augmented*

a.

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

Solusi:

```
Masukkan elemen-elemen matriks:
Elemen [0][0]: 1
Elemen [0][1]: -1
Elemen [0][2]: 2
Elemen [0][3]: -1
Elemen [0][4]: -1
Elemen [1][0]: 2
Elemen [1][1]: 1
Elemen [1][2]: -2
Elemen [1][3]: -2
Elemen [1][4]: -2
Elemen [2][0]: -1
Elemen [2][1]: 2
Elemen [2][2]: -4
Elemen [2][3]: 1
Elemen [2][4]: 1
Elemen [3][0]: 3
Elemen [3][1]: 0
Elemen [3][2]: 0
Elemen [3][3]: -3
Elemen [3][4]: -3
Solusi dalam bentuk parametrik:
x1 = -3.0 + (3.0) * s
x2 = 0.0 + (4.0) * s
x3 = s
x4 = t
Apakah ingin menulis ke file?
1. Ya
2. Tidak
2
```

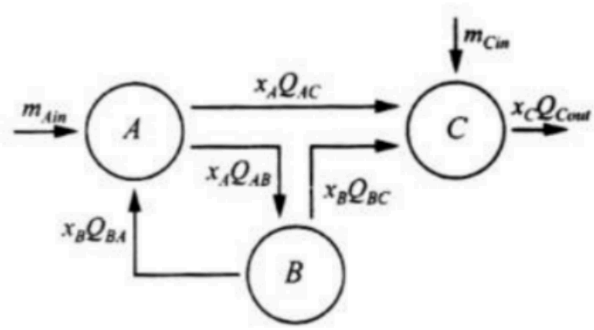
3. SPL berbentuk

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

Solusi:

```
Elemen [2][0]: 1
Elemen [2][1]: 3
Elemen [2][2]: 2
Elemen [2][3]: -1
Elemen [2][4]: 2
Elemen [3][0]: 1
Elemen [3][1]: 0
Elemen [3][2]: 6
Elemen [3][3]: 4
Elemen [3][4]: 3
Solusi SPL:
x1 = -0.2243243243243243
x2 = 0.18243243243243243
x3 = 0.7094594594594594
x4 = -0.2581081081081081
```

4. Lihatlah sistem reaktor pada gambar berikut.



Dengan laju volume Q dalam m^3/s dan input massa min dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

$$\begin{aligned}
\text{A:} &\quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0 \\
\text{B:} &\quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0 \\
\text{C:} &\quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0
\end{aligned}$$

Tentukan solusi x_A, x_B, x_C dengan menggunakan parameter berikut : Q_{AB} = 40, Q_{AC} = 80, Q_{BA} = 60, Q_{BC} = 20 dan Q_{Cout} = 150 m^3/s dan m_{Ain} = 1300 dan m_{Cin} = 200 mg/s.

Solusi:

```
Pilih metode penyelesaian SPL:
1. Metode Gauss
2. Metode Gauss-Jordan
3. Metode Cramer
4. Metode Matriks Balikan
2
1. Input dari file
2. Input secara manual
2
Jumlah baris (m):
3
Jumlah kolom (n):
4
Masukkan elemen-elemen matriks:
Elemen [0][0]: 60
Elemen [0][1]: -40
Elemen [0][2]: -80
Elemen [0][3]: -1300
Elemen [1][0]: 40
Elemen [1][1]: -60
Elemen [1][2]: -20
Elemen [1][3]: 0
Elemen [2][0]: 80
Elemen [2][1]: 20
Elemen [2][2]: -150
Elemen [2][3]: -200
Solusi SPL:
x1 = 190.33333333333334
x2 = 88.66666666666667
x3 = 114.66666666666667
```

5. 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	0.1	0.3	0.5	0.7	0.9	1.1	1.3
f(*x*)	0.003	0.067	0.148	0.248	0.370	0.518	0.697

Lakukan pengujian pada nilai-nilai berikut:

x = 0.2 *f*(*x*) = ?

x = 0.55 *f*(*x*) = ?

x = 0.85 $f(x)$ = ?

x = 1.28 $f(x)$ = ?

Solusi:

```
Masukkan elemen-elemen matriks:
Elemen [0][0]: 0.1
Elemen [0][1]: 0.003
Elemen [1][0]: 0.3
Elemen [1][1]: 0.067
Elemen [2][0]: 0.5
Elemen [2][1]: 0.148
Elemen [3][0]: 0.7
Elemen [3][1]: 0.248
Elemen [4][0]: 0.9
Elemen [4][1]: 0.370
Elemen [5][0]: 1.1
Elemen [5][1]: 0.518
Elemen [6][0]: 1.3
Elemen [6][1]: 0.697
Masukkan nilai x yang ingin diprediksi: 0.2
Persamaan polinomial: y =  -0.022976562500000127 + 0.24000000000000202x + 0.19739583333332383x^2 + 1.943974495266508E-14x^3 + 0.026041666666648422x^4 + 7.10801814904417E-15x^5 -7.2350
73091134577E-16x^6
Hasil prediksi Y pada x = 0.2 adalah 0.032960937500000016
```

b. Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6,567	12.624
30/06/2022	7	21.807
08/07/2022	7,258	38.391
14/07/2022	7,451	54.517
17/07/2022	7,548	51.952
26/07/2022	7,839	28.228
05/08/2022	8,161	35.764
15/08/2022	8,484	20.813
22/08/2022	8,709	12.408
31/08/2022	9	10.534

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

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

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

Tanggal (desimal) = 6 + (17/30) = 6,567

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

a. 16/07/2022

b. 10/08/2022

c. 05/09/2022

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

Solusi:

a.

```
Masukkan elemen-elemen matriks:
Elemen [0][0]: 0.1
Elemen [0][1]: 0.003
Elemen [1][0]: 0.3
Elemen [1][1]: 0.067
Elemen [2][0]: 0.5
Elemen [2][1]: 0.148
Elemen [3][0]: 0.7
Elemen [3][1]: 0.248
Elemen [4][0]: 0.9
Elemen [4][1]: 0.370
Elemen [5][0]: 1.1
Elemen [5][1]: 0.518
Elemen [6][0]: 1.3
Elemen [6][1]: 0.697
Masukkan nilai x yang ingin diprediksi: 0.55
Persamaan polinomial: y =  -0.022976562500000127 + 0.240000
73091134577E-16x^6
Hasil prediksi Y pada x = 0.55 adalah 0.17111865234375004
```

b.

```
Masukkan nilai x yang ingin diprediksi: 0.85
Persamaan polinomial: y =  -0.022976562500000127 + 0.24000
73091134577E-16x^6
Hasil prediksi Y pada x = 0.85 adalah 0.33723583984375005
```

c.

```
Masukkan nama file:
doc/contoh.txt
Masukkan nilai x yang ingin diprediksi: 1.28
Persamaan polinomial: y =  -0.022976562500000127 + 0.24000000
73091134577E-16x^6
Hasil prediksi Y pada x = 1.28 adalah 0.6775418374999999
```

d.

```
doc/contoh.txt
Masukkan nilai x yang ingin diprediksi: 7516
Persamaan polinomial: y =  -567473.9108886719 + 154628.294183
E-10x^6 + 8.56589591621408E-15x^7 -3.7584392830393265E-19x^8
Hasil prediksi Y pada x = 7516.0 adalah 53470.890563964844
```

```
Masukkan nama file:
doc/contoh.txt
Masukkan nilai x yang ingin diprediksi: 8323
Persamaan polinomial: y =  -567473.9108886719 + 154628.29418
E-10x^6 + 8.56589591621408E-15x^7 -3.7584392830393265E-19x^8
Hasil prediksi Y pada x = 8323.0 adalah 40390.56591796875
```

6. Studi Kasus Regresi Linear dan Kuadratik Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

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

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

$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$

$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$

$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$

$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$

Silahkan terapkan model-model ini pada *Multiple Quadratic Equation* juga dan bandingkan hasilnya. Sistem persamaan linear tidak akan diberikan untuk kasus ini.

Solusi:

```
------------MAIN-----------
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi
0. KELUAR
5
1. Input dari file
2. Input secara manual
1
Masukkan nama file:
doc/contoh.txt
f(50.0,76.0) = -9.48777429699000
```

7. Studi Kasus Interpolasi *Bicubic Spline*

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

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

Tentukan nilai:

$f(0, 0)$ = ?

```
0. KELUAR
5
1. Input dari file
2. Input secara manua
1
Masukkan nama file:
doc/conto.txt
f(0.0,0.0) = 21.0
```

$f(0.5, 0.5)$ = ?

```
Masukkan nama file:
doc/conto.txt
f(0.5.0.5) = 71.97656250000058
```

$f(0.25, 0.75)$ = ?

```
1
Masukkan nama file:
doc/conto.txt
f(0.25,0.75) = 57.867065429687926
```

$f(0.1, 0.9)$ = ?

```
Masukkan nama file:
doc/conto.txt
f(0.1,0.9) = 53.26848650000025
```

Bab 5

Kesimpulan

Kesimpulan

Kami Cepalta kelompok 47 Tugas Besar 1 Aljabar Linier dan Geometri telah berhasil membuat pustaka dalam Bahasa Java. Pustaka tersebut mampu 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. Selain itu juga, kami telah berhasil menggunakan pustaka tersebut untuk menyelesaikan berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Saran

Masih terdapat kasus-kasus yang tidak ditangani, seperti validasi input/parameter, dan juga terdapat beberapa kasus spesial yang belum dapat diatasi. Oleh karena itu, sebaiknya dikaji lebih lanjut mengenai kasus-kasus spesial ataupun kasus tertentu meskipun tidak dijabarkan pada spesifikasi.

Refleksi

Masih belum diterapkannya best practice pada github yang terkadang membuat kerancuan informasi terkait progress pengerjaan tugas besar ini. Penambahan fitur fungsi sebaiknya dilakukan di local terlebih dahulu dan dipastikan berfungsi baru dipush agar tidak terjadi konflik. Selain itu, kami menyadari bahwa perlu adanya analisis lebih lanjut terhadap contoh-contoh kasus yang diberikan dari spesifikasi. Dengan selesainya kami membuat pustaka, masih perlu dikaji lebih lanjut terkait kasus-kasus yang mungkin saja tidak ditangani oleh kode yang sudah kami buat.

Lampiran

Tautan *repository* :

https://github.com/bevindav/Algeo01-2304

Tautan video:

https://bit.ly/VideoAlgeoCepalta

Referensi:

https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/037-Elementary-Linear-Algebra-Applications-Version-Howard-Anton-Chris-Rorres-Edisi-1-2013.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm

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

Asistensi 1 – 16 Oktober 2024: