TUGAS BESAR I
ALJABAR LINEAR DAN GEOMETRI
APLIKASI MATRIKS

DISUSUN OLEH:
13522029 - Ignatius Jhon Hezkiel Chan
13522043 - Daniel Mulia Putra Manurung
13522054 - Benjamin Sihombing

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

2023

BAB 1
Deskripsi Masalah

Sistem persamaan linear (SPL) banyak ditemukan di dalam penerapan bidang pemrograman dan sains. Dengan banyaknya SPL dan besarnya SPL yang akan sering digunakan, akan sangat sulit untuk menyelesaikan banyak permasalahan tersebut dengan perhitungan manual. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, dan metode matriks balikan. Metode yang terakhir adalah Kaidah Crammer, yang hanya dapat digunakan dengan SPL yang memiliki jumlah persamaan dan jumlah variabel penyusun yang sama. Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

Di dalam Tugas Besar 1 ini, dibuat sebuah library aljabar linear 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).

Library tersebut juga akan digunakan untuk menyelesaikan permasalahan yang melibatkan adanya pemrosesan SPL seperti Interpolasi Polinomial, Regresi Linear Berganda, dan Bicubic Spline Interpolation.

BAB 2
Teori Singkat

2.1 OBE

Operasi Baris Elementer merupakan suatu operasi pada matriks yang dapat digunakan untuk memperoleh invers suatu matriks atau memperoleh penyelesaian dari sebuah Sistem Persamaan Linier (SPL). Untuk mendapatkan solusi dari SPL, OBE dilakukan pada matriks augmented hingga terbentuk matriks eselon baris (matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol) atau matriks eselon baris tereduksi (matriks eselon baris dengan sifat setiap kolom yang memiliki 1 utama memiliki nol di tempat lain). Adapun tiga OBE terhadap matriks augmented yakni:

1. Kalikan sebuah baris dengan konstanta tidak nol.
2. Pertukarkan dua buah baris.
3. Tambahkan sebuah baris dengan kelipatan baris lainnya.

2.2 Eliminasi Gauss

Eliminasi Gauss adalah metode untuk mencari solusi dari suatu sistem persamaan linear. Metode ini akan merubah matriks awal menjadi bentuk matriks eselon baris dengan menggunakan OBE.

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

2.3 Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah metode untuk mencari solusi dari suatu sistem persamaan linear. Metode ini akan merubah matriks awal menjadi bentuk matriks eselon tereduksi baris dengan menggunakan OBE.

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

2.4 Determinan

Untuk setiap matriks persegi, kita dapat mengasosiasikan suatu angka yang merupakan determinan matriks tersebut. Matriks persegi sendiri merupakan suatu matriks yang memiliki jumlah kolom dan baris yang sama. Determinan untuk suatu matriks A yang memiliki n baris dan kolom dilambangkan dengan:

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

Metode utama untuk mencari determinan yaitu metode reduksi baris dan ekspansi kofaktor. Metode reduksi baris memanfaatkan determinan matriks segitiga. Berikut merupakan contoh dari matriks segitiga.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{12} & a_{22} & 0 & 0 \\ a_{13} & a_{23} & a_{33} & 0 \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}$$

Dengan rumus

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

$$\det(A) = a_{11}a_{22} \dots a_{nn}$$

Jika suatu matriks A adalah suatu matriks yang berukuran n x n, terdapat matriks B yang merupakan hasil manipulasi dari matriks A. Hubungan kedua determinan matriks tersebut diatur dengan:

1. Jika matriks A dikali sebuah barisnya dengan konstanta k sehingga menghasilkan matriks B, maka $\det(B) = k \det(A)$.
2. Jika matriks A dilakukan pertukaran antara dua baris sehingga menghasilkan matriks B, maka $\det(B) = -\det(A)$.
3. Jika sebuah matriks A ditambah kelipatan k baris lainnya sehingga menghasilkan matriks B, maka $\det(B) = \det(A)$.

Perhitungan determinan melalui reduksi baris dapat memanfaatkan properti determinan matriks segitiga dan aturan terkait determinan hasil manipulasi matriks. Pertama-tama dapat dilakukan Operasi Baris Elementer (OBE) untuk memperoleh matriks segitiga bawah atau atas. Setelah itu, determinan dari matriks dapat diperoleh dengan mempertimbangkan aturan determinan hasil manipulasi matriks. Cara kedua untuk memperoleh determinan yaitu melalui ekspansi kofaktor. Untuk suatu matriks A yang berukuran n x n, didefinisikan Mij yang merupakan minor entri dan Cij yang merupakan kofaktor entri. Minor entri (Mij) didapat dengan menghitung determinan dari submatriks yang elemen-elemennya tidak berada pada baris i dan kolom j. Sedangkan kofaktor entri dapat diperoleh melalui rumus:

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

Kofaktor entri berkoresponden dengan minor entri. Maka dari itu, jika sudah diketahui minor entri, kofaktor entri juga dapat diperoleh melalui pola tanda kofaktor entri.

$$\begin{bmatrix} + & - & + & - & \dots \\ - & + & - & + & \dots \\ + & - & + & - & \dots \\ - & + & - & + & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Dengan ini, determinan matriks bisa didapatkan dengan:

$$\det(A) = a_{11}C_{11} + a_{12}C_{12} + \cdots + a_{1n}C_{1n}$$
$$\det(A) = a_{21}C_{21} + a_{22}C_{22} + \cdots + a_{2n}C_{2n}$$
$$\dots$$
$$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \cdots + a_{nn}C_{nn}$$

2.5 Matriks Balikan

Matriks balikan adalah matriks yang akan menghasilkan matriks identitas jika matriks balikan dikalikan dengan matriks aslinya. $A^{-1}A = I$, $dengan\ A$: $matriks\ asli$, A^{-1}: $matriks\ balikan$, I: $matriks\ identitas$

2.6 Matriks Kofaktor

Misalkan A adalah suatu matriks berukuran n x n dan Cij adalah kofaktor entri dari aij, maka matriks kofaktor dari A adalah

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

2.7 Matriks Adjoin

Matriks adjoin adalah matriks transpose dari matriks kofaktor. Matriks transpose adalah matriks yang diperoleh dengan menukar baris dan kolom suatu matriks.

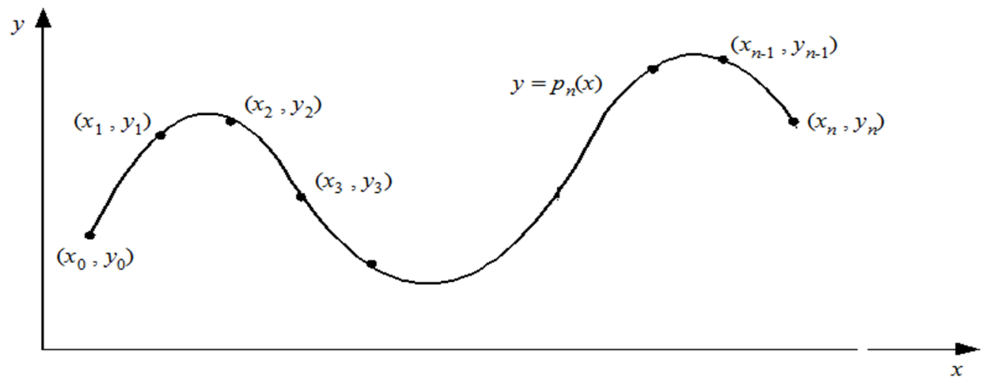
2.8 Kaidah Cramer

Kaidah Cramer adalah salah satu metode yang bisa digunakan untuk mencari solusi dari SPL. Kaidah ini memanfaatkan determinan matriks untuk mencari solusi SPL tersebut.

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

2.9 Interpolasi Polinomial

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



Gambar 2. Ilustrasi beberapa titik yang diinterpolasi secara polinomial.

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

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0, y_0), (x_1, y_1), dan (x_2, y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0, y_0), (x_1, y_1), (x_2, y_2), dan (x_3, y_3), polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \ldots, a_n$,

$$a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n = y_0$$
$$a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n = y_1$$

$$\ldots \qquad \ldots$$
$$a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n = y_n$$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0, a_1, …, a_n, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada $x = 9.2$. Polinom kuadratik berbentuk $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

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

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

2.10 Regresi Linier Berganda

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

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

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

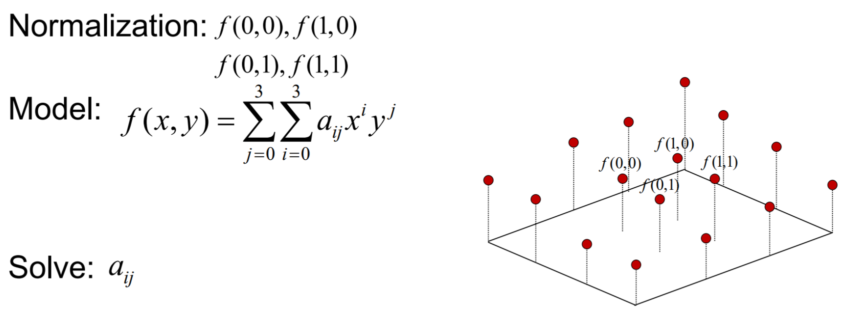
$$\begin{array}{lllll}
nb_0 + b_1\sum_{i=1}^{n} x_{1i} & + b_2\sum_{i=1}^{n} x_{2i} & + \cdots & + b_k\sum_{i=1}^{n} x_{ki} & = \sum_{i=1}^{n} y_i \\
b_0\sum_{i=1}^{n} x_{1i} + b_1\sum_{i=1}^{n} x_{1i}^2 & + b_2\sum_{i=1}^{n} x_{1i}x_{2i} & + \cdots & + b_k\sum_{i=1}^{n} x_{1i}x_{ki} & = \sum_{i=1}^{n}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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

2.11 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 spline*.

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.

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

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

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

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

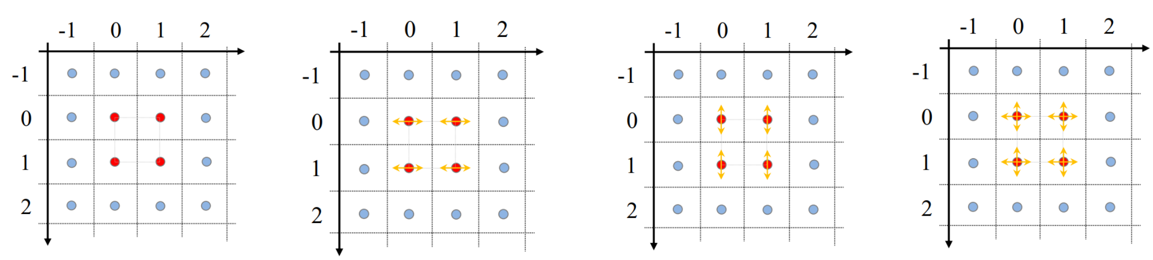
Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi *X* yang membentuk persamaan penyelesaian sebagai berikut.

$$y = Xa$$

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

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 x 1^{1-1} x 1^0 = 1, sesuai dengan isi matriks X.

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



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

BAB 3
Implementasi

3.1 Folder Matriks

a. Matrix.java

i. Atribut

- double [][] mem
- int rowEff
- int colEff

ii. Metode

Fungsi/Prosedur	Deskripsi
public Matrix(int row, int col)	Konstruktor Matrix dengan row sebagai rowEff dan col sebagai colEff
public void setRow (int n)	Setter rowEff senbagi n
public void setCol (int n)	Setter colEff senbagi n
public void readMatrix()	Membaca elemen matrix dari keyboard
public void matrixIdentity()	Menjadikan matrix menjadi matrix identitas
public int getRow()	Getter mengembalikan nilai rowEff
public int getCol()	Getter mengembalikan nilai colEff
public void bagiRow(int row, double div)	Menghasilkan matrix dengan row ke - row yang telah dibagi div
public void tambahRow(int rowDest, int rowX, double rasio)	Menghasilkan matrix dengan row ke - rowDest yang telah ditambahkan dangan rowX yang telah dikali rasio.
public int findColLead1(int row)	Mengembalikan letak kolom dimana leading 1 berada dalam suatu baris row.
public Matrix kali(Matrix a, Matrix b)	Mengembalikan matrix hasil kali matrix a dan b

public void displayMatrix()	Menampilkan matix ke layar
public boolean isSingular()	Mengembalikan true jika matrix singular dan false jika tidak
public boolean isSquare()	Mengembalikan true jika matrix adalah matrix kotak dan false jika tidak
public Matrix transpose()	Melakukan transpose terhadap matrix
public double sigmaSampel(Matrix m, int bykSampel, int roww, int coll)	Menghitung hasil penjumlahan variabel pada tiap sample

b. OBE.java

i. Metode

Fungsi/Prosedur	Deskripsi
public static void Gauss(Matrix m)	Melakukan OBE pada matrix m hingga matrix eselon
public static void GaussJordan(Matrix m)	Melakukan OBE pada matrix m hingga matrix eselon tereduksi
public static double kofaktor(Matrix m, int p, int q)	Mengembalikan nilai kofaktor dari matrix m kolom q dan p
public static double determinanKofaktor(Matrix m)	Mengembalikan nilai determinan dari matrix m dengan cara kofaktor
public static double determinanRB(Matrix mdet)	Mengembalikan nilai determinan dari matrix m dengan cara reduksi baris
public static void mainmenu()	Menampilkan “Main menu” dari operasi pencarian Determinan

c. Invers.java

i. Metode

Fungsi/Prosedur	Deskripsi

public static void menuInvers()	Menampilkan “Main menu” dari operasi Invers terhadap Matriks
public static Matrix invers(Matrix m)	Mengembalikan hasil balikan dari matriks m
public static Matrix setUpInvers(Matrix m)	Mengembalikan matriks yaitu matriks m yang telah ditambahkan matriks identitas
public static Matrix takeInvers(Matrix m)	Mengembalikan matrix hasil invers yang masil terkonkat dengan matriks identitas
public static Matrix matrixKofak(Matrix m)	Mengembalikan matrix kofaktor dari m.
public static Matrix inversAdj(Matrix m)	Mengembalikan matrix balikan dari m, dengan metode menggunakan adjoin

3.2 Folder Aplikasi

a. SPL.java

i. Metode

Fungsi/Prosedur	Deskripsi
public static boolean cekAllArray(String[] arr, String val)	Mengembalikan boolean apakah semua elemen array arr adalah val.
public static boolean cekRangeArray(double[] arr, int val, int awal, int akhir)	Mengembalikan boolean apakah semua elemen dari array arr dari int awal hingga int akhir, adalah val
public static boolean cekAdaSolusi(Matrix m)	Mengembalikan boolean apakah matrix m memiliki solusi
public static String[] SPLGauss(Matrix m)	Mengembalikan string yang menyatakan hasil dari persamaan dalam matrix m yang telah dioperasikan dengan SPL Gauss
public static String[] SPLGaussJordan(Matrix m)	Mengembalikan string yang menyatakan hasil dari persamaan dalam matrix m yang telah

	dioperasikan dengan SPL Gauss Jordan
public static String kaidahCrammer(Matrix m)	Mengembalikan string yang menyatakan hasil dari persamaan dalam matrix m yang telah dioperasikan dengan Kaidah Crammer
public static String[] SPLInvers(Matrix M)	Mengembalikan string yang menyatakan hasil dari persamaan dalam matrix m yang telah dioperasikan dengan metode Invers
public static void menuSPL()	Menampilkan “Main menu” dari operasi Sistem Persamaan Linear

b. RLB.java

i. Metode

public static double[] solveRLB(Matrix m)	Membuat matrix augmented yang berisi normal estimation equation for multiple linear regression. Kemudian menyelesaikannya dengan gauss, lalu mengembalikan solusi dari SPL tersebut yakni nilai b0 b1 .. bk dalam bentuk array double.
public static void menuRLB()	Menampilkan “Main menu” dari operasi Regresi Linear Berganda

c. Interpolasi.java

i. Metode

Fungsi/Prosedur	Deskripsi
public static Matrix inToMatrix(int n)	Melakukan penginputan matrix dari keyboard, lalu mengubahnya menjadi matrix persamaan polinomial

	dan mengembalikan matrixnya
public static Matrix inToMatrixv2(Matrix m)	Melakukan penginputan matrix dari matrix lain, lalu mengubahnya menjadi matrix persamaan polinomial dan mengembalikan matrixnya
public static Matrix MatrixInterpolasi(Matrix m)	Mengembalikan matrix hasil matrix m yang telah dilakukan OBE.
public static String InterpolasiPrint(Matrix m)	Mengembalikan string yang menyatakan persamaan hasil Interpolasi matrix m
public static String HasilYPrint(Matrix m, double d)	Mengembalikan string yang menyatakan hasil dari nilai yang ditaksir
public static void mainMenu()	Menampilkan “Main menu” dari operasi Interpolasi Polinomial

d. Bicubic.java

i. Metode

Fungsi/Prosedur	**Deskripsi**
public static int f(int x, int y, int i, int j)	Mengembalikan koefisien aij dari f(x,y).
public static int fx(int x, int y, int i, int j)	Mengembalikan koefisien aij dari f_x(x,y).
public static int fy(int x, int y, int i, int j)	Mengembalikan koefisien aij dari f_y(x,y).
public static int fxy(int x, int y, int i, int j)	Mengembalikan koefisien aij dari f_{xy}(x,y).
public static Matrix setBicubic(Matrix m)	Mempersiapkan matriks dengan menambah matriks identitas di sebelah kanan matriks awal.

	Mengembalikan matriks yang sudah ditambahkan.
public static void menubicubic()	Melakukan proses penginputan matriks, memproses interpolasi dengan memanggil fungsi-fungsi lain, dan melakukan proses pengoutputan hasil.
public static Matrix buatX()	Membuat matriks X dan mengembalikan matriks X tersebut.
public static double bicubicInterpolation(double x, double y, Matrix mIn, Matrix mX)	Menghitung hasil interpolasi dan mengembalikan hasil interpolasi *bicubic spline*.

3.3 Folder IO

a. IOput.java

i. Metode

Fungsi/Prosedur	**Deskripsi**
public static int inputMode()	Menerima pilihan input mode dan mengembalikan nilai 0 atau 1.
public static int outputMode()	Menerima pilihan output mode dan mengembalikan nilai 0 atau 1.
public static void readFileToMatrix(Matrix mm)	Menerima matriks dari file dan menyimpannya pada program.
public static void readFileToMatrixInterpolasi(Matrix mm, Matrix x)	Menerima matriks dan x dari file dan menyimpannya pada program.
public static void readFileToMatrixBicubic(Matrix mm, Matrix xy)	Menerima matriks dan x,y dari file dan menyimpannya pada program.

public static void readFileToMatrixRegresi(Matrix mm, Matrix x)	Menerima matriks dan x_0-x_n dari file dan menyimpannya pada program.
public static void readKeyboardToMatrix(Matrix m)	Menerima matriks dari keyboard dan menyimpannya pada program.
public static void writeMatrixAndStringToFile(Matrix m, String str)	Menuliskan matriks dan string dari keyboard dan menyimpannya pada file.
public static void writeMatrixToFile(Matrix m)	Menuliskan matriks dari keyboard dan menyimpannya pada file.
public static void writeStringToFile(String str)	Menuliskan string dari keyboard dan menyimpannya pada file.
public static void writeArrayStringToFile(String[] listStr)	Menuliskan sejumlah string dari keyboard dan menyimpannya pada file.

3.4 Garis Besar Program

Program akan menampilkan menu pilihan-pilihan yang ingin diselesaikan oleh user:

1. Sistem Persamaan Linear
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi *Bicubic Spline*
6. Regresi Linear Berganda
7. Keluar Kalkulator

Setiap pilihan di atas akan memanggil fungsi-fungsi yang dituliskan sesuai dengan pilihan yang user panggil.

BAB 4
Eksperimen

4.1 Studi Kasus 1

<table>
<tr><td>a</td><td>$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$

SPl tersebut tidak memiliki solusi.
Kembali ke Menu Utama.......</td></tr>
<tr><td>b</td><td>$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$

Solusi dari SPL tersebut adalah:
x1 = a+3.00
x2 = 2.00a
x3 = b
x4 = a-1.00
x5 = a
Kembali ke Menu Utama.......</td></tr>
<tr><td>c</td><td>$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$

SPL3.txt
Matriks bukan persegi.
Matriks tidak memiliki balikan sehingga solusi SPL tidak bisa diselesaikan.
Tentukan metode output!
1. Penulisan ke File
2. Penulisan ke Terminal
2
SPl tersebut tidak memiliki solusi.</td></tr>
</table>

d

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

```
x1 = -3.21
x2 = 46.50
x3 = -134.72
x4 = 204.14
x5 = -223.30
x6 = 114.25
Kembali ke Menu Utama.......
```

```
Solusi dari SPL tersebut adalah:
x1 = 3.03
x2 = 12.07
x3 = -68.12
x4 = 34.61
x5 = 34.49
x6 = 8.25
x7 = -8.52
x8 = 29.79
x9 = -5.72
x10 = -43.88
Kembali ke Menu Utama.......
```

4.2 Studi Kasus 2

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 dari SPL tersebut adalah:
x1 = b-1.00
x2 = 2.00a
x3 = a
x4 = b
Kembali ke Menu Utama.......
```

b

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

```
Solusi dari SPL tersebut adalah:
x1 = 0.00
x2 = 2.00
x3 = 1.00
x4 = 1.00
Kembali ke Menu Utama.......
```

4.3 Studi Kasus 3

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 dari SPL tersebut adalah
x1 = -0.22
x2 = 0.18
x3 = 0.71
x4 = -0.26
Kembali ke Menu Utama.......
```

b

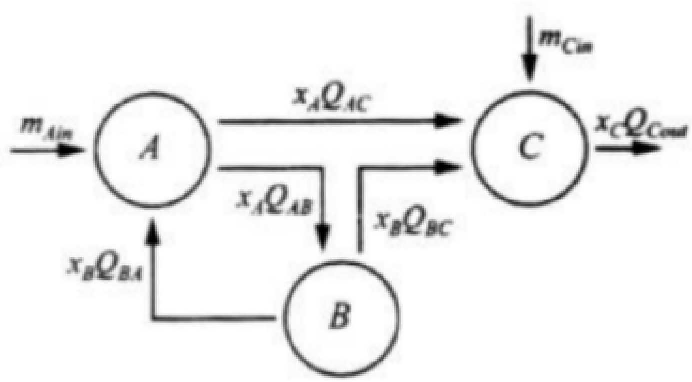
$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\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

```
SPL tidak bisa diselesaikan menggunakan Kaidah Cramer.
Kembali ke Menu Utama.......
```

```
Tentukan metode output!
1. Penulisan ke File
2. Penulisan ke Terminal
2
SPl tersebut tidak memiliki solusi.
Kembali ke Menu Utama.......
```

4.4 Studi Kasus 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:

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

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.

Hasil	`Solusi dari SPL tersebut adalah:` `x1 = 190.33` `x2 = 88.67` `x3 = 114.67` `Kembali ke Menu Utama.......`

4.5 Studi Kasus 5

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:

$x = 0.2$ $f(x) = ?$
$x = 0.55$ $f(x) = ?$
$x = 0.85$ $f(x) = ?$
$x = 1.28$ $f(x) = ?$

a	`Polinom yang melalui titik-titik tersebut adalah:` `f(x) = -0.02298 + 0.24000x + 0.19740x^2 + 0.00000x^3 + 0.02604x^4 + 0.00000x^5 - 0.00000x^6` `Nilai y dari nilai x yang diinginkan adalah:` `f(0.2) = 0.0330`
b	`Polinom yang melalui titik-titik tersebut adalah:` `f(x) = -0.02298 + 0.24000x + 0.19740x^2 + 0.00000x^3 + 0.02604x^4 + 0.00000x^5 - 0.00000x^6` `Nilai y dari nilai x yang diinginkan adalah:` `f(0.55) = 0.1711`
c	`Polinom yang melalui titik-titik tersebut adalah:` `f(x) = -0.02298 + 0.24000x + 0.19740x^2 + 0.00000x^3 + 0.02604x^4 + 0.00000x^5 - 0.00000x^6` `Nilai y dari nilai x yang diinginkan adalah:` `f(0.85) = 0.3372`
d	`Polinom yang melalui titik-titik tersebut adalah:` `f(x) = -0.02298 + 0.24000x + 0.19740x^2 + 0.00000x^3 + 0.02604x^4 + 0.00000x^5 - 0.00000x^6` `Nilai y dari nilai x yang diinginkan adalah:` `f(1.28) = 0.6775`

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.

a	`Polinom yang melalui titik-titik tersebut adalah:` `f(x) = 7187066071657.87 - 9346993079172.33x + 5334203055240.20x^2 - 1756810186361.38x^3`

	+ 368550807175.53x^4 - 51131876760.13x^5 + 4695806315.43x^6 - 275474539.42x^7 + 9372849.24x^8 - 140993.71x^9 Nilai y dari nilai x yang diinginkan adalah: f(7.516) = 53537.9961
b	Polinom yang melalui titik-titik tersebut adalah: f(x) = 7187066071657.87 - 9346993079172.33x + 5334203055240.20x^2 - 1756810186361.38x^3 + 368550807175.53x^4 - 51131876760.13x^5 + 4695806315.43x^6 - 275474539.42x^7 + 9372849.24x^8 - 140993.71x^9 Nilai y dari nilai x yang diinginkan adalah: f(8.322) = 36343.7695
c	Polinom yang melalui titik-titik tersebut adalah: f(x) = 7187066071657.87 - 9346993079172.33x + 5334203055240.20x^2 - 1756810186361.38x^3 + 368550807175.53x^4 - 51131876760.13x^5 + 4695806315.43x^6 - 275474539.42x^7 + 9372849.24x^8 - 140993.71x^9 Nilai y dari nilai x yang diinginkan adalah: f(9.166) = -659015.5156
d	05/10/2022 f(x) = 7187066071657.8670000000 - 9346993079172.3280000000x + 5334203055240.1950000000x^2 - 1756810186361.3809000000x^3 + 368550807175.5339400000x^4 - - 51131876760.1327500000x^5 + 4695806315.428793 0000x^6 - 275474539.4206693000x^7 + 9372849.2391013200x^8 - 140993.7122486359x^9 f(10.161) = -435606190.4688

c. Sederhanakan fungsi $f(x)$ yang memenuhi kondisi

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

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

Hasil	```0 0 0.4 0.419 0.8 0.507 1.2 0.561 1.6 0.584 2 0.577``` ```Polinom yang melalui titik-titik tersebut adalah: f(x) = + 2.03850x - 3.56510x^2 + 3.25260x^3 - 1.42904x^4 + 0.23763x^5```

4.6 Studi Kasus 6

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.

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

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

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

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

Hasil	`Persamaan regresinya adalah f(x) = -3.51-0.0x1 0.0x2 + 0.15x3` `Hasil dari taksiran nya adalah` `y = 0.8849999999999998` `Kembali ke Menu Utama.....`

4.7 Studi Kasus 7

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

Tentukan nilai:

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

a	f(x,y) = ΣiΣj(a_ij * x^i * y^j) = 21.0 Kembali ke Menu Utama.....
b	f(x,y) = ΣiΣj(a_ij * x^i * y^j) = 87.796875 Kembali ke Menu Utama.....
c	f(x,y) = ΣiΣj(a_ij * x^i * y^j) = 117.732177734375 Kembali ke Menu Utama.....
d	f(x,y) = ΣiΣj(a_ij * x^i * y^j) = 128.57518699999997 Kembali ke Menu Utama.....

BAB 5
Kesimpulan, Saran, Komentar, dan Refleksi

5.1 Kesimpulan

Dari pengerjaan tugas ini, kami berhasil menerapkan pelajaran Aljabar Linear pada suatu program komputer. Kami berhasil mengimplementasikan baik struktur data matriks, kaidah-kaidah & properti-properti sebuah matriks, dan menyelesaikan beberapa permasalahan riil/nyata dengan menggunakan penerapan matriks. Permasalah yang dapat diselesaikan pada program ini adalah solusi SPL, determinan matriks, matriks balikan, interpolasi polinom, interpolasi *bicubic spline*, dan regresi linear berganda. Setiap fungsi program dapat berjalan dengan baik dan ketelitian yang cukup baik.

5.2 Saran dan Komentar

Saran dan komentar untuk tugas besar kedepannya adalah berikan 1 jawaban benar untuk salah satu studi kasus/*testcase* karena setiap studi cenderung memiliki banyak percobaan/jawaban. Hal ini dilakukan agar setidaknya bisa memastikan kode yang dibuat benar untuk 1 contoh.

5.3 Refleksi

Dari tugas besar ini, kami belajar untuk komunikasi dan bekerja sama satu sama lain dengan baik. Kami juga belajar bagaimana cara membagi waktu dengan baik (walaupun akhirnya tetap *keos*). Selain itu, kami juga belajar untuk memahami sesuatu yang baru dengan cepat (pada tugas ini mempelajari bahasa pemrograman Java).

DAFTAR REFERENSI

Munir,Rinaldi. 2022. “Sistem persamaan linier (Bagian 1: Metode eliminasi Gauss)”. Diakses 5 Oktober 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm

Munir,Rinaldi. 2022. “Sistem persamaan linier (Bagian 3: Metode eliminasi Gauss-Jordan)”. Diakses 5 Oktober 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm

Munir,Rinaldi. 2022. “Determinan (Bagian 1: menghitung determinan dengan reduksi baris)”. Diakses 5 Oktober 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm

Munir,Rinaldi. 2022. “Determinan (Bagian 2: menghitung determinan dengan ekspansi kofaktor” Diakses 5 Oktober 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm

REPOSITORY

Link repository dari Tugas Besar 1 IF 2123 Aljabar Linear dan Geometri kelompok Beaulo adalah https://github.com/Gryphuss/Algeo01-22029