Laporan

Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri

Sistem Persamaan Linier, Determinan, dan Aplikasinya

oleh

Feralezer L. G. Tampubolon / 13519062

Moses Ananta / 13519076

Jonathan Richard Sugandhi / 13519128

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2020

BAB I

DESKRIPSI MASALAH

1. Sistem Persamaan Linear

Persoalan Sistem Persamaan Linear ialah sebagai berikut. Diberikan sistem persamaan linier (SPL) Ax = b dengan n peubah (variable) dan m persamaan adalah berbentuk

$$a_{11}\, x_1 + a_{12}\, x_2 + \dots + a_{1n}\, x_n = b_1$$

$$a_{21}\, x_1 + a_{22}\, x_2 + \dots + a_{2n}\, x_n = b_2$$

$$\vdots \qquad \vdots \qquad \qquad \vdots \qquad \vdots$$

$$a_{m1}\, x_1 + a_{m2}\, x_2 + \dots + a_{mn}\, x_n = b_m$$

yang dalam hal ini x_i adalah peubah, a_{ij} dan b_i adalah koefisien ∈ R. 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, atau hanya satu (unik/tunggal).

2. Determinan

Sebuah matriks M berukuran n × n

$$A_{(mxn)} = \begin{bmatrix} a11 & a12 & . & . & . & a1n \\ a21 & a22 & . & . & . & a2n \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ am1 & am2 & . & . & . & amn \end{bmatrix}$$

Determinannya adalah

$$\mathrm{Det(M)} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{vmatrix}$$

Adapun determinan matriks M berukuran n × n dapat dihitung dengan beberapa cara, yaitu dengan reduksi baris dan ekspansi kofaktor

3. Matriks Balikan

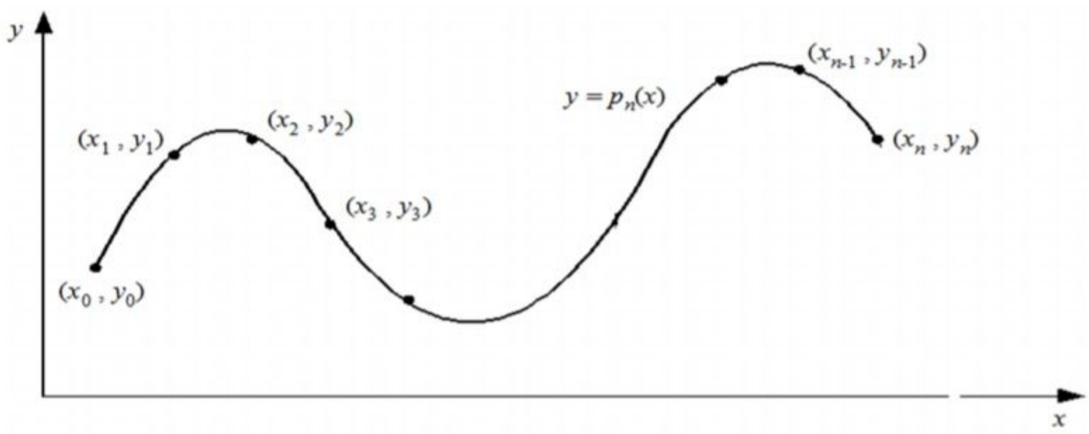
Setiap matriks berukuran n x n yang memiliki determinan tidak sama dengan 0 selalu memiliki matriks balikan. Matriks balikan A' didefinisikan sebagai matriks balikan dari matriks A jika dan hanya jika

AA' = A'A = I dengan I = matriks identitas

Matriks Balikan memiliki banyak kegunaan, salah satunya adalah untuk menyelesaikan sistem persamaan linear. Adapun matriks balikan dapat dihitung dengan memanfaatkan Sistem Eliminasi Gauss-Jordan atau dengan membagi adjoin matriks tersebut dengan determinan matriks.

4. 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 pn(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, …, n.



Setelah polinom interpolasi pn(x) ditemukan, pn(x) dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x0, xn]. Polinom interpolasi derajat n yang menginterpolasi titik-titik (x_0, y_0),(x_1, y_1), ..., (x_n, y_n). adalah berbentuk $p_n(x) = a_0 + a_1x + a2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah p1(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, 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 $p_n(x) = a, + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2, …, a_n,

$$
\begin{aligned}
&a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n = y_0\\
&a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_nx_1^n = y_1\\
&\qquad \ldots \qquad\qquad \ldots\\
&a_0 + a_1x_n + a_2x_n^2 + \ldots + a_nx_n^n = y_n
\end{aligned}
$$

Karena persamaan akhir ini bersifat linear, dapat digunakan sistem eliminasi Gauss Jordan untuk mencari solusi masing-masing koefisiennya.

5. Regresi Linier Berganda

Regresi Linear merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus 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:

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

Karena persamaan akhir ini juga bersifat linear, maka dengan memanfaatkan sistem elmininasi Gauss dapat diperoleh pula masing-masing koefisiennya.

BAB II

TEORI SINGKAT

1. Sistem Persamaan Linear

Berikut adalah cara-cara yang dapat digunakan untuk mendapatkan solusi dari suatu sistem persamaan linear

a. *Metode Eliminasi Gauss*

Metode Eliminasi Gauss adalah metode eliminasi antar-baris dari suatu matriks *augmented* SPL dengan hanya menggunakan 3 cara yaitu, mengganti urutan baris, mengalikan suatu baris dengan suatu konstanta tak nol, dan menambahkan suatu baris dengan baris lainnya. Dari Metode Eliminasi Gauss akan dihasilkan matriks eselon baris. Suatu matriks dikatakan berbentuk eselon baris jika:

- Pada baris yang tak semuanya nol selalu memiliki 1 utama, yaitu nilai tak nol pertama bernilai 1
- Baris yang hanya berisi 0 selalu ada di paling bawah.
- Jika ada 2 atau lebih baris yang memenuhi kriteria pertama, maka baris yang lebih atas adalah baris yang memiliki 1 utama lebih kiri.

b. *Metode Eliminasi Gauss-Jordan*

Metode Eliminasi Gauss Jordan adalah kelanjutan dari Metode Eliminasi Gauss. Setelah didapatkan matriks berbentuk eselon baris, eliminasi dilanjutkan hingga didapatkan matriks eselon baris tereduksi. Adapun matriks eselon baris tereduksi adalah matriks eselon baris dimana setiap kolom yang memuat 1 utama harus memiliki nilai nol di tempat lainnya.

c. *Metode Matriks Balikan*

Sesuai dengan namanya, Metode Matriks Balikan menggunakan sifat matriks balikan untuk mencari solusi persamaan linear.

Misalkan dimiliki SPL sebagai berikut:

$$
\begin{array}{l}
a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + ... + a_{1n} x_n = b_1 \\
a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + ... + a_{2n} x_n = b_2 \\
a_{31} x_1 + a_{32} x_2 + a_{33} x_3 + ... + a_{3n} x_n = b_3 \\
\;\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
a_{n1} x_1 + a_{n2} x_2 + a_{n3} x_3 + ... + a_{nn} x_n = b_n
\end{array}
$$

SPL tersebut dapat kita representasikan dalam bentuk matriks sebagai:

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & .... & a_{1n} \\ a_{21} & a_{22} & a_{23} & .... & a_{2n} \\ a_{31} & a_{32} & a_{33} & .... & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & .... & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}
$$

Misal:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \quad \text{dan } \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Dengan mensubstitusikan ketiganya ke dalam persamaan di atas, kita peroleh:

$\mathbf{Ax} = \mathbf{b}$

Misal A^{-1} adalah matriks balikan dari A,

$$\mathbf{A^{-1}} = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & \cdots & a'_{1n} \\ a'_{21} & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ a'_{31} & a'_{32} & a'_{33} & \cdots & a'_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a'_{n1} & a'_{n2} & a'_{n3} & \cdots & a'_{nn} \end{bmatrix}$$

Mengalikan kedua sisi persamaan dengan $\mathbf{A^{-1}}$ menghasilkan bentuk:

$\mathbf{A^{-1}Ax} = \mathbf{A^{-1}b}$

Yang dengan sifat matriks balikan dapat disederhanakan menjadi:

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

Maka diperoleh x_1 sampai x_n, karena:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & \cdots & a'_{1n} \\ a'_{21} & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ a'_{31} & a'_{32} & a'_{33} & \cdots & a'_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a'_{n1} & a'_{n2} & a'_{n3} & \cdots & a'_{nn} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Sehingga:

$$\begin{array}{l} x_1 = a'_{11}\, b_1 + a'_{12}\, b_2 + a'_{13}\, b_3 + \cdots + a'_{1n}\, x_n \\ x_2 = a'_{21}\, b_1 + a'_{22}\, b_2 + a'_{23}\, b_3 + \cdots + a'_{2n}\, x_n \\ x_3 = a'_{31}\, b_1 + a'_{32}\, b_2 + a'_{33}\, b_3 + \cdots + a'_{3n}\, x_n \\ \vdots \\ x_n = a'_{n1}\, b_1 + a'_{n2}\, b_2 + a'_{n3}\, b_3 + \cdots + a'_{nn}\, x_n \end{array}$$

d. Kaidah Cramer

Kaidah Cramer adalah suatu formula mencari solusi persamaan linear dengan memanfaatkan determinan. Solusi akan didapat dengan menggunakan determinan matriks koefisien dan determinan matriks koefisien yang salah satu kolomnya diganti oleh matriks hasil.
Misalkan dimiliki SPL sebagai berikut:

$$\begin{array}{l} a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 + \cdots + a_{1n}\, x_n = b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + a_{23}\, x_3 + \cdots + a_{2n}\, x_n = b_2 \\ a_{31}\, x_1 + a_{32}\, x_2 + a_{33}\, x_3 + \cdots + a_{3n}\, x_n = b_3 \\ \vdots \\ a_{n1}\, x_1 + a_{n2}\, x_2 + a_{n3}\, x_3 + \cdots + a_{nn}\, x_n = b_n \end{array}$$

SPL tersebut dapat kita representasikan dalam bentuk matriks sebagai:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Misal:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \text{ dan } \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Kaidah Cramer menyatakan bahwa nilai x_j, di mana $j \in \{1, 2, 3, \dots, n\}$, dapat dicari dengan:

$x_j = \det(\mathbf{A_j}) / \det(\mathbf{A})$

Di mana $\mathbf{A_j}$ adalah matriks yang entri pada kolom ke-j nya diganti dengan matriks **b**. Misalnya:

$$\mathbf{A_1} = \begin{bmatrix} b_1 & a_{12} & a_{13} & \dots & a_{1n} \\ b_2 & a_{22} & a_{23} & \dots & a_{2n} \\ b_3 & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_n & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}, \ \mathbf{A_2} = \begin{bmatrix} a_{11} & b_1 & a_{13} & \dots & a_{1n} \\ a_{21} & b_2 & a_{23} & \dots & a_{2n} \\ a_{31} & b_3 & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & b_n & a_{n3} & \dots & a_{nn} \end{bmatrix}, \text{ dst.}$$

2. Determinan

Jika **A** adalah suatu matriks 2×2,

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

maka det(**A**) didefinisikan sebagai:

$$\det(\mathbf{A}) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = (a \times d) - (b \times c)$$

Untuk matriks n×n yang lebih besar dari 2×2, determinan matriks tersebut diperoleh dengan menjumlahkan hasil perkalian satu baris/kolom sembarang dengan kofaktor baris/kolom tersebut. Baris/kolom manapun yang dipilih akan menghasilkan determinan yang sama.

Contoh: jika **F** adalah suatu matriks 3x3 dengan

$$\mathbf{F} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Maka, dengan memilih baris 1, diperoleh:

$$\det(\mathbf{F}) = (1 \times (-1)^{1+1} \times \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix}) + (2 \times (-1)^{1+2} \times \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix}) + (3 \times (-1)^{1+3} \times \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix})$$
$$\det(\mathbf{F}) = (1 \times 1 \times ((5 \times 9) - (6 \times 8))) + (2 \times (-1) \times ((4 \times 9) - (6 \times 7))) + (3 \times 1 \times ((4 \times 8) - (5 \times 7)))$$
$$\det(\mathbf{F}) = (1 \times (45 - 48)) + ((-2) \times (36 - 42)) + (3 \times (32 - 35)) = -3$$

$\det(\mathbf{F}) = -3 + ((-2) \times (-6)) + (3 \times (-3))$
$\det(\mathbf{F}) = -3 + 6 + (-3)$
$\det(\mathbf{F}) = 0$

Jadi determinan dari matriks **F** adalah 0.

3. Matriks Balikan

Suatu matriks n x n dikatakan memiliki matriks balikan jika dan hanya jika determinan matriks tersebut tidak bernilai 0. Untuk mencari matriks balikan dari suatu matriks, dapat digunakan beberapa cara, yaitu dengan menggunakan metode eliminasi gauss jordan dan menggunakan determinan dan adjoin.

a. Metode eliminasi gauss jordan
   Pertama-tama, suatu matriks n x n akan ditempelkan matriks identitas berukuran n x n juga
   Berikut contoh untuk suatu matriks M 3 x 3 :

$$M \mid I = \left[\begin{array}{ccc|ccc} a & b & c & 1 & 0 & 0 \\ d & e & f & 0 & 1 & 0 \\ g & h & i & 0 & 0 & 1 \end{array}\right] \rightarrow \begin{bmatrix} a\,b\,c & 1\,0\,0 \\ d\,e\,f & 0\,1\,0 \\ g\,h\,i & 0\,0\,1 \end{bmatrix}$$

   Dengan menggunakan metode eliminasi gauss jordan, akan dibuat sedemikian sehingga bagian kiri (matriks awal 3x3) menjadi matriks identitas. Hasil di sebelah kanan merupakan matriks balikan dari matriks

b. Metode adjoin
   Pertama-tama dicari nilai determinan. Setelah itu perlu dibuat matriks minor dari matriks n x n yang ingin dicari matriks balikannya. Matriks minor dari suatu matriks adalah matriks dimana komponen a_{ij} dengan i adalah baris dan j adalah kolom dari matriks minor adalah determinan dari matriks awal tanpa semua elemen di baris i dan kolom j. Setelah itu dibuat matriks kofaktor dengan mengalikan setiap elemen pada matriks minor dengan $(-1)^{i+j}$. Setelah didapat matriks kofaktor, cari matriks adjoint dengan cara mentranspose matriks kofaktor, Matriks balikan adalah matriks adjoin dibagi dengan determinan matriks.

4. Interpolasi Polinomial

Misal kita mempunyai 2 buah titik (x_1, y_1) dan (x_2, y_2) pada suatu bidang koordinat kartesius. Kita ingin mencari persamaan garis yang melalui kedua titik tersebut. Cara yang biasanya kita lakukan adalah dengan mengubah persamaan $(y - y_1) / (y_2 - y_1) = (x - x_1) / (x_2 - x_1)$ ke dalam bentuk umum $y = mx + c$. Tapi bagaimana bila kita ingin mencari persamaan yang dibentuk oleh *lebih dari dua titik*? Interpolasi polinom adalah metode untuk menentukan suatu persamaan polinom P(x) derajat n yang melewati (n + 1) titik.

5. Regresi Linier Berganda

Regresi linear berganda merupakan sebuah persamaan dari regresi linear yang melibatkan lebih dari satu variabel yang memiliki persamaan sebagai berikut

$$y = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_k x_{ki} + \varepsilon_i$$

Dimana,

- y merupakan variabel terikat atau *response*

- x merupakan variabel bebas atau *prediktor*
- β merupakan koefisien regresi
- ε merupakan sisa

BAB III

IMPLEMENTASI PROGRAM

3.1. Struktur *Class*

3.1.1. *Class* Matrix

i. Konstruktor

Matrix(int row, int col)

ii. Attribut

double[][] DF
int ROWCOUNT
int COLCOUNT

iii. Method

int GetROWCOUNT() Method untuk mengambil nilai efektif baris matriks
int GetCOLCOUNT() Method untuk mengambil nilai efektif kolom matriks
int GetFirstRowID() Method untuk mengambil indeks baris pertama matriks
int GetLastRowID() Method untuk mengambil indeks baris terakhir matriks
int GetLastColID() Method untuk mengambil indeks kolom pertama matriks
double GetElement(int rowID, int colID) Method untuk mengambil nilai matriks yang berindeks [rowID,colID]
double[] GetRow(int rowID) Method untuk mengambil nilai baris matriks yang berindeks [rowID]
void SetElement(int rowID, int colID, double elementVal) Method untuk merubah nilai matriks yang berindeks [rowID,colID] menjadi elementVal
void PasteDFFrom(Matrix anotherSmallerMatrix)

Proses : Nilai matriks di*copy* ke matriks lainnya I.S. Matriks Terdefinisi F.S. Terbentuk Matriks lain yang merupakan salinan matriks awal
void PrintMatrix() I.S. Matriks Terdefinisi F.S. Nilai setiap indeks matriks efektif ditulis satu per satu ke layar
boolean IsValidRowID(int rowID) Method untuk mengecek apakah rowID termasuk pada indeks efektif baris matriks
boolean IsValidColID(int colID) Method untuk mengecek apakah colIDtermasuk pada indeks efektif kolom matriks
void OBE_SwapRow(int row1, int row2) I.S. Matriks Terdefinisi F.S.Nilai semua kolom matriks pada baris row1 dan baris row2 ditukar
void OBE_SumRow(int row1, int row2, double k) I.S. Matriks Terdefinisi F.S Semua elemen matriks di row1 ditambahkan dengan k kali setiap elemen di row2
void OBE_ScaleRow(int row, double k) I.S. Matriks Terdefinisi F.S Semua elemen matriks di row dikali dengan k
int GetLeadingValue_Row_ColID(int row) I.S. Matriks Terdefinisi F.S : Mencari indeks kolom di baris row yang tak nol pertama kali Bila input tidak valid, return -1 Bila semua elemen bernilai nol return COLCOUNT
double GetLeadingValue_Row_Element(int row) I.S. Matriks Terdefinisi F.S : Mencari nilai baris tak nol pertama bila seluruh baris adalah 0 akan di return 0
int GetLeadingValue_Col_RowID(int col) I.S. Matriks Terdefinisi F.S : mencari nilai leading dari kolom col, me-return indeks baris dengan kolom tak nol pertama dari row bila input tidak valid, return -1 jika semua nilai di kolom bernilai 0 return ROWCOUNT
int GetLeadingValue_Col_RowID_Next(int col) I.S. Matriks terdefinisi F.S. mencari nilai leading dari kolom col, me-return indeks baris dengan kolom tak nol kedua dari row

bila input tidak valid, return -1 jika semua nilai di kolom bernilai 0 return ROWCOUNT
int GetLeadingValue_Col_RowID_FromRow(int col, int row) I.S. Matriks Terdefinisi F.S : mencari nilai leading dari kolom col, me-return indeks baris dengan kolom tak nol pertama dari row bila input col tidak valid, return -1 bila input row tidak valid, return -2 jika semua nilai di kolom bernilai 0 return ROWCOUNT
int GetLeadingValue_Col_RowID_FromRow_Next(int col, int row)
void UpperTriangularSelectionSort() I.S. Matriks Terdefinisi F.S. Matriks terkonversi menjadi matriks segitiga atas
void OBE_ScaleRow_LeadingOne(int row) I.S. Matriks Terdefinisi F.S. Didapat 1 utama pada row, dengan cara membagi semua elemen row dengan elemen tak nol pertama dari row.
void Convert_RowEchelon() I.S. Matriks Terdefinisi F.S. Matriks terkonversi menjadi matriks Eselon baris
void Convert_ReducedRowEchelon() I.S. Matriks Terdefinisi F.S. Matriks terkonversi menjadi matriks Eselon baris tereduksi
void Convert_Minor(int row, int col) I.S. Matriks Terdefinisi F.S. Matriks terkonversi menjadi matriks minor
Matrix Create_FromUserInput() Proses : Membaca masukan row dan col dan nilai setiap index matriksnya dari masukan user I.S. Sembarang F.S. Terdefinisi matrix dengan nilai elemen efektifnya, berukuran row x col
Matrix Create_FromTxt() I.S. File txt sesuai format F.S Menerima input berupa path dari file txt, membaca dan menghasilkan output berupa Matrix
Matrix makeHilbertAugmented(int N) I.S. N Terdefinisi F.S. Terbentuk augmented matriks Hilbert
boolean is_all_zero_row(int row) I.S. Matriks dan row Terdefinisi F.S. mengembalikan true jika semua elemen matriks pada baris row bernilai 0 dan mengembalikan

false jika tidak
boolean is_all_zero_row_except_last(int row) I.S. Matriks dan row Terdefinisi F.S. mengembalikan true jika semua elemen matriks pada baris row bernilai 0 kecuali elemen terakhir pada baris tersebut dan mengembalikan false jika tidak
boolean has_zero_row_except_last() I.S. Matriks terdefinisi F.S. mengembalikan true jika elemen matriks pada baris row ada yang bernilai 0 kecuali elemen terakhir pada baris tersebut dan mengembalikan false jika tidak
boolean only_have_leadingOne_row(int row) I.S. Matriks terdefinisi F.S. mengembalikan true jika hanya terdapat 1 utama pada row, yaitu elemen kecuali 1 utama bernilai 0.
void ParametricOnGauss() I.S. Matriks input terdefinisi dan merupakan matriks eselon baris atau matriks eselon baris tereduksi. Hasil terbaik saat input berupa matriks eselon baris tereduksi. Jumlah kolom maksimal matriks input adalah 27. F.S. Menghasilkan output berupa solusi parametrik atau solusi double dari x_i

3.1.2. *Class* Inverse

i. Konstruktor

Inverse()

ii. Method

void makeAugmentedIdentity(Matrix matrix) I.S. matrix terdefinisi F.S. terbentuk matriks identitas di kanan matrix
void InverseOBE(Matrix matrix) Method yang melakukan OBE pada matrix sehingga terbentuk matriks baris tereduksi
boolean IsInverseOBEValid(Matrix matrix) Method yang memeriksa jika matrix memiliki matriks identitas di sebelah kananya
void PrintInverseOBE(Matrix matrix) Method yang menampilkan inverse dari matrix dengan metode OBE
Matrix MakeCoFactorMatrix(Matrix matrix) I.S. matrix terdefinisi F.S. terbentuk matriks kofaktor
Matrix Transpose(Matrix matrix) I.S. matrix terdefinisi F.S. terbentuk matriks hasil transpose matrix

Matrix InverseAdjoint(Matrix matrix) I.S. matrix terdefinisi F.S. terbentuk matriks adjoint
void PrintInverseAdjoint(Matrix matrix) I.S. matrix terdefinisi F.S. hasil inverse matriks dengan metode adjoint ditampilkan ke layar
void InverseSPL(Matrix matrix) I.S. matrix terdefinisi F.S. didapati solusi persamaan linear dari matrix yang diselesaikan dengan metode inverse yang kemudian ditampilkan ke layar

3.1.3. *Class* Determinan

i. Konstruktor

Determinant()

ii. Attribut

double Det Nilai determinan

iii. Method

void OBE(Matrix matrix) I.S. matrix terdefinisi F.S. didapati nilai determinan Det dari hasil OBE matrix
double Kofaktor_Calculator(Matrix matrix) I.S. matrix terdefinisi F.S. dikembalikan sebuah nilai double hasil operasi kofaktor terhadap matrix
void Kofaktor(Matrix matrix) I.S. matrix terdefinisi F.S. didapati nilai determinan Det dari matrix dengan metode kofaktor
void PrintDeterminant() I.S. Det terdefinisi F.S. menampilkan ke layar nilai Det
double GetDeterminant() I.S. Det terdefinisi F.S. mengembalikan nilai Det

3.1.4. *Class* Cramer

i. Konstruktor

Cramer()

ii. Method

void CramerSPL(Matrix matrix) I.S. matrix terdefinisi F.S. didapati solusi persamaan linear dari matrix yang diselesaikan dengan metode Cramer yang kemudian ditampilkan ke layar

3.1.5. *Class* Interpolation

i. Konstruktor

Interpolation(Matrix listOfCoordinates)

ii. Attribut

Matrix INTERPOLATIONMATRIX

iii. Method

void Create_InterpolationMatrix(Matrix listOfCoordinates) I.S. listOfCoordinates terdefinisi F.S. terbentuk matriks INTERPOLATIONMATRIX yang nilainya sudah di konversi ke matriks eselon baris tereduksi terlebih dahulu
double Interpolate(double x)

3.1.6. *Class* MultiRegression

i. Konstruktor

MultiRegression()

ii. Attribut

double[] y
double[] beta
double[] estimation
Matrix x
Matrix reg
int n

int k

iii. Method

void BacaReg() Proses : Membaca masukan y, beta, estimation, x, n, k dari user I.S. sembarang F.S. y, beta, estimation, x, n, k terdefinisi nilainya
double SumCol(int j) I.S. j dan matrix x terdefinisi F.S. dikembalikan nilai hasil penjumlahan semua elemen matrix x pada kolom j
double Sum2Col(int j1,int j2) I.S. j1, j2, dan matrix x terdefinisi F.S. dikembalikan nilai hasil penjumlahan elemen-elemen matrix x pada kolom j1 yang dikalikan dengan elemen-elemen matrix x pada kolom j2
double SumY() I.S. y terdefinisi F.S. dikembalikan nilai hasil penjumlahan semua elemen y
double Sum2Y(int j) I.S. j, x ,y terdefinisi F.S. dikembalikan nilai hasil penjumlahan setiap baris pada elemen y yang dikalikan dengan setiap baris pada elemen matrix x pada kolom j
void MakeReg() I.S. x,j,y terdefinisi F.S. terbentuk sistem persamaan linear dengan metode Normal Estimation Equation yang direpresentasikan oleh matriks augmented matrix reg
Matrix SolveReg() I.S. reg terdefinisi F.S. dikembalikan matrix hasil konversi reg menjadi matriks eselon baris tereduksi
void GetBeta() I.S. reg terdefinisi F.S. nilai beta terdefinisi
double Estimate() I.S. beta terdefinisi F.S. dikembalikan nilai double yang merupakan hasil estimasi dari masukan estimation
void ProsesReg() I.S.sembarang F.S. menjalankan BacaReg(),MakeReg(),GetBeta(),dan GetBeta() secara berturut-turut

3.2. Garis Besar Program

Program dimulai dengan menjalankan Main.java. Di situ user akan diberikan pilihan. Berdasarkan pilihan user, akan dijalankan salah satu dari antara SistemPersamaanLinier.java, Determinan.java, MatriksBalikan.java, InterpolasiPolinom.java, atau RegresiLinierBerganda .java. Fungsi-fungsi matematis dan definisi objek-objek yang digunakan program semuanya tersimpan di dalam file-file .java pada folder flib. Program akan memanggil file-file tersebut ketika diperlukan.

BAB IV

EKSPERIMEN

4. Eksperimen

a. Sistem Persamaan Linear

1a.

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

Hasil run dengan metode gauss :

```
Matriks Input:
     1.000     1.000    -1.000    -1.000     1.000
     2.000     5.000    -7.000    -5.000    -2.000
     2.000    -1.000     1.000     3.000     4.000
     5.000     2.000    -4.000     2.000     6.000
Matriks eselon:
     1.000     1.000    -1.000    -1.000     1.000
     0.000     1.000    -1.667    -1.000    -1.333
    -0.000    -0.000     1.000    -1.000     1.000
     0.000     0.000     0.000     0.000     1.000
Linear Equation has no real solution

-----------------------------------------------------------------
```

Hasil run dengan metode gauss-jordan:

```
Matriks Input:
     1.000     1.000    -1.000    -1.000     1.000
     2.000     5.000    -7.000    -5.000    -2.000
     2.000    -1.000     1.000     3.000     4.000
     5.000     2.000    -4.000     2.000     6.000
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.667     0.000
     0.000     1.000     0.000    -2.667     0.000
    -0.000    -0.000     1.000    -1.000     0.000
     0.000     0.000     0.000     0.000     1.000
Linear Equation has no real solution

-----------------------------------------------------------------
```

Hasil run dengan metode matriks balikan :

```
Path lengkap dari file .txt matriks: ./test/1a.txt
x1: NaN
x2: NaN
x3: NaN
x4: NaN
```

Hasil run dengan kaidah cramer :

```
Path lengkap dari file .txt matriks: ./test/1a.txt
x1: Infinity
x2: -Infinity
x3: -Infinity
x4: -Infinity
```

1b.

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

Hasil run dengan metode gauss:

```
Matriks Input:
     1.000    -1.000     0.000     0.000     1.000     3.000
     1.000     1.000     0.000    -3.000     0.000     6.000
     2.000    -1.000     0.000     1.000    -1.000     5.000
    -1.000     2.000     0.000    -2.000    -1.000    -1.000
Matriks eselon:
     1.000    -1.000     0.000     0.000     1.000     3.000
     0.000     1.000     0.000    -1.500    -0.500     1.500
     0.000     0.000     0.000     1.000    -1.000    -1.000
     0.000     0.000     0.000     0.000     0.000     0.000
x1= 3.0 + 1.0(1.5 + 1.5d + 0.5e) - 1.0e
x2= 1.5 + 1.5(-1.0 + 1.0e) + 0.5e
x3= c
x4= -1.0 + 1.0e
x5= e

----------------------------------------------------------------
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     1.000    -1.000     0.000     0.000     1.000     3.000
     1.000     1.000     0.000    -3.000     0.000     6.000
     2.000    -1.000     0.000     1.000    -1.000     5.000
    -1.000     2.000     0.000    -2.000    -1.000    -1.000
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.000    -1.000     3.000
     0.000     1.000     0.000     0.000    -2.000     0.000
     0.000     0.000     0.000     1.000    -1.000    -1.000
     0.000     0.000     0.000     0.000     0.000     0.000
x1= 3.0 + 1.0e
x2= 0.0 + 2.0e
x3= c
x4= -1.0 + 1.0e
x5= e
```

Hasil run dengan metode matriks balikan:

```
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
```

```
Path lengkap dari file .txt matriks: ./test/1b.txt
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
x5: NaN
```

Hasil run dengan kaidah Cramer:

1c.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

Hasil run dengan metode gauss:

```
Matriks Input:
     0.000     1.000     0.000     0.000     1.000     0.000     2.000
     0.000     0.000     0.000     1.000     1.000     0.000    -1.000
     0.000     1.000     0.000     0.000     0.000     1.000     1.000
Matriks eselon:
     0.000     1.000     0.000     0.000     1.000     0.000     2.000
     0.000     0.000     0.000     1.000     1.000     0.000    -1.000
    -0.000    -0.000    -0.000    -0.000     1.000    -1.000     1.000
x1= a
x2= 2.0 - 1.0(1.0 + 1.0f)
x3= c
x4= -1.0 - 1.0(1.0 + 1.0f)
x5= 1.0 + 1.0f
x6= f

----------------------------------------------------------------
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     0.000     1.000     0.000     0.000     1.000     0.000     2.000
     0.000     0.000     0.000     1.000     1.000     0.000    -1.000
     0.000     1.000     0.000     0.000     0.000     1.000     1.000
Matriks eselon tereduksi:
     0.000     1.000     0.000     0.000     0.000     1.000     1.000
     0.000     0.000     0.000     1.000     0.000     1.000    -2.000
    -0.000    -0.000    -0.000    -0.000     1.000    -1.000     1.000
x1= a
x2= 1.0 - 1.0f
x3= c
x4= -2.0 - 1.0f
x5= 1.0 + 1.0f
x6= f

----------------------------------------------------------------
```

Hasil run dengan metode matriks balikan:

```
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
```

Hasil run dengan kaidah cramer:

```
Path lengkap dari file .txt matriks: ./test/1c.txt
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
x5: NaN
x6: NaN
```

1d6. Matriks Hilbert dengan n = 6

Hasil run dengan metode gauss:

```
Matriks Input:
     1.000     0.500     0.333     0.250     0.200     0.167     1.000
     0.500     0.333     0.250     0.200     0.167     0.143     0.000
     0.333     0.250     0.200     0.167     0.143     0.125     0.000
     0.250     0.200     0.167     0.143     0.125     0.111     0.000
     0.200     0.167     0.143     0.125     0.111     0.100     0.000
     0.167     0.143     0.125     0.111     0.100     0.091     0.000
Matriks eselon:
     1.000     0.500     0.333     0.250     0.200     0.167     1.000
     0.000     1.000     1.000     0.900     0.800     0.714    -6.000
     0.000     0.000     1.000     1.500     1.714     1.786    30.003
     0.000     0.000     0.000     1.000     2.006     2.785  -140.579
     0.000     0.000     0.000     0.000     1.000     2.682   744.411
    -0.000    -0.000    -0.000    -0.000    -0.000     1.000  2158.664
x1= -358.77804526779386 - 0.5(-1547.9009981942017 - 1.0000060000240003c - 0.9000036000144003d - 0.8000072000288003e) - 0.333333(-3825.022375878429 - 1.500114009636836d - 1.71427
71424833925e) - 0.25(-6152.280429035973 - 2.0062233080620944e) - 0.2(-5045.250152924339)
x2= -1547.9009981942017 - 1.0000060000240003(-3825.022375878429 - 1.500114009636836d - 1.7142771424833925e) - 0.9000036000144003(-6152.280429035973 - 2.0062233080620944e) - 0.80
00072000288003(-5045.250152924339)
x3= -3825.022375878429 - 1.500114009636836(-6152.280429035973 - 2.0062233080620944e) - 1.7142771424833925(-5045.250152924339)
x4= -6152.280429035973 - 2.0062233080620944(-5045.250152924339)
x5= -5045.250152924339
x6= 2158.6639542788544
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     1.000     0.500     0.333     0.250     0.200     0.167     1.000
     0.500     0.333     0.250     0.200     0.167     0.143     0.000
     0.333     0.250     0.200     0.167     0.143     0.125     0.000
     0.250     0.200     0.167     0.143     0.125     0.111     0.000
     0.200     0.167     0.143     0.125     0.111     0.100     0.000
     0.167     0.143     0.125     0.111     0.100     0.091     0.000
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.000     0.000     0.000    11.540
     0.000     1.000     0.000     0.000     0.000     0.000    46.617
     0.000     0.000     1.000     0.000     0.000     0.000 -1130.945
     0.000     0.000     0.000     1.000     0.000     0.000  3969.618
     0.000     0.000     0.000     0.000     1.000     0.000 -5045.250
    -0.000    -0.000    -0.000    -0.000    -0.000     1.000  2158.664
x1= 11.540412090797318
x2= 46.61669408394607
x3= -1130.9449694655714
x4= 3969.618022764683
x5= -5045.250152924339
x6= 2158.6639542788544

-----------------------------------------------------------------
```

Hasil run dengan metode matriks balikan:

```
Path lengkap dari file .txt matriks: ./test/1d6.txt
x1: 11.540412090771728
x2: 46.61669408059518
x3: -1130.944969452068
x4: 3969.618022733211
x5: -5045.250152890351
x6: 2158.663954265922
```

Hasil run dengan kaidah cramer:

```
Path lengkap dari file .txt matriks: ./test/1d6.txt
x1: 11.540405162368982
x2: 46.61663728218961
x3: -1130.9437001254032
x4: 3969.6134915657444
x5: -5045.244423027063
x6: 2158.661548503386
```

1d10. Matriks Hilbert dengan n = 10

Hasil run dengan metode gauss:

```
Matriks Input:
     1.000     0.500     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     1.000
     0.500     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.000
     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.000
     0.250     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.000
     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.000
     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.000
     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.000
     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.059     0.000
     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.059     0.056     0.000
     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.059     0.056     0.053     0.000
Matriks eselon:
Ini jalan
     1.000     0.500     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     1.000
     0.000     1.000     1.000     0.900     0.800     0.714     0.643     0.583     0.533     0.491    -6.000
     0.000     0.000     1.000     1.500     1.714     1.786     1.786     1.750     1.697   136.647    30.003
     0.000     0.000     0.000     1.000     2.006     2.785     3.344     3.723  2111.720 -3157.737  -140.579
     0.000     0.000     0.000     0.000     1.000     2.682     4.430 38140.004-76497.560 49403.122   744.411
    -0.000    -0.000    -0.000    -0.000    -0.000     1.000-227071.051609023.130-589453.183245734.990  2158.664
     0.000     0.000     0.000     0.000     0.000     0.000     1.000    -2.682     2.596    -1.082    -0.010
    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000     1.000    -2.005     1.295     0.020
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     1.000-221537749505940.000328975186896146.00014558087332220.750
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     1.000    -1.634    -0.096
```

```
x1= 1.0 - 0.5(-6.000024000096001 - 1.0000060000240003c - 0.9000036000144003d - 0.8000072000288003e - 0.7142848571394288f - 0.6428605714422858g - 0.5833343333373334h - 0.5333361333445336i - 0.4909099636398546j) - 0.333333(30.00300025898251 - 1.500114009636836d - 1.7142771424833925e - 1.7858385824695262f - 1.7857835776548248g - 1.7501150100908758h - 1.6970444309128194i - 136.6472136717198j) - 0.25(-140.5789633376317 - 2.0062233080620944e - 2.7849177051305665f - 3.3435548092036322g - 3.72282325404823h - 2111.7196913841876i + 3157.7374285171127j) - 0.2(744.4113310512869 - 2.682057794359095f - 4.429607700365184g - 38140.00389338817h + 76497.55986022178i - 49403.12206609378j) - 0.166667(2158.6639542788544 + 227071.0512592636g - 609023.1298008951h + 589453.1832498119i - 245734.98969355068j) - 0.142857(-0.00950665512476004 + 2.6821012897557064h - 2.595921951553248i + 1.0822053098442217j) - 0.125(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) - 0.111111(-0.09596486848744372 + 1.6338727745287314j) - 0.1j
x2= -6.000024000096001 - 1.0000060000240003(30.00300025898251 - 1.500114009636836d - 1.7142771424833925e - 1.7858385824695262f - 1.7857835776548248g - 1.7501150100908758h - 1.6970444309128194i - 136.6472136717198j) - 0.9000036000144003(-140.5789633376317 - 2.0062233080620944e - 2.7849177051305665f - 3.3435548092036322g - 3.72282325404823h - 2111.7196913841876i + 3157.7374285171127j) - 0.8000072000288003(744.4113310512869 - 2.682057794359095f - 4.429607700365184g - 38140.00389338817h + 76497.55986022178i - 49403.12206609378j) - 0.7142848571394288(2158.6639542788544 + 227071.0512592636g - 609023.1298008951h + 589453.1832498119i - 245734.98969355068j) - 0.6428605714422858(-0.00950665512476004 + 2.6821012897557064h - 2.595921951553248i + 1.0822053098442217j) - 0.5833343333373334(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) - 0.5333361333445336(-0.09596486848744372 + 1.6338727745287314j) - 0.4909099636398546j
x3= 30.00300025898251 - 1.500114009636836(-140.5789633376317 - 2.0062233080620944e - 2.7849177051305665f - 3.3435548092036322g - 3.72282325404823h - 2111.7196913841876i + 3157.7374285171127j) - 1.7142771424833925(744.4113310512869 - 2.682057794359095f - 4.429607700365184g - 38140.00389338817h + 76497.55986022178i - 49403.12206609378j) - 1.7858385824695262(2158.6639542788544 + 227071.0512592636g - 609023.1298008951h + 589453.1832498119i - 245734.98969355068j) - 1.7857835776548248(-0.00950665512476004 + 2.6821012897557064h - 2.595921951553248i + 1.0822053098442217j) - 1.7501150100908758(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) - 1.6970444309128194(-0.09596486848744372 + 1.6338727745287314j) - 136.6472136717198j
x4= -140.5789633376317 - 2.0062233080620944(744.4113310512869 - 2.682057794359095f - 4.429607700365184g - 38140.00389338817h + 76497.55986022178i - 49403.12206609378j) - 2.7849177051305665(2158.6639542788544 + 227071.0512592636g - 609023.1298008951h + 589453.1832498119i - 245734.98969355068j) - 3.3435548092036322(-0.00950665512476004 + 2.6821012897557064h - 2.595921951553248i + 1.0822053098442217j) - 3.72282325404823(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) - 2111.7196913841876(-0.09596486848744372 + 1.6338727745287314j) + 3157.7374285171127j
x5= 744.4113310512869 - 2.682057794359095(2158.6639542788544 + 227071.0512592636g - 609023.1298008951h + 589453.1832498119i - 245734.98969355068j) - 4.429607700365184(-0.00950665512476004 + 2.6821012897557064h - 2.595921951553248i + 1.0822053098442217j) - 38140.00389338817(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) + 76497.55986022178(-0.09596486848744372 + 1.6338727745287314j) - 49403.12206609378j
x6= 2158.6639542788544 + 227071.0512592636(-0.00950665512476004 + 2.6821012897557064h - 2.595921951553248i + 1.0822053098442217j) - 609023.1298008951(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) + 589453.1832498119(-0.09596486848744372 + 1.6338727745287314j) - 245734.98969355068j
x7= -0.00950665512476004 + 2.6821012897557064(1.455808733222075E13 + 2.0050487753940223i - 1.2947048169287498j + 2.2153774950594E14i - 3.28975186896146E14j) - 2.595921951553248(-0.09596486848744372 + 1.6338727745287314j) + 1.0822053098442217j
x8= 1.455808733222075E13 + 2.0050487753940223(-0.09596486848744372 + 1.6338727745287314j) - 1.2947048169287498j + 2.2153774950594E14(-0.09596486848744372 + 1.6338727745287314j) - 3.28975186896146E14j
x9= -0.09596486848744372 + 1.6338727745287314j
x10= j
```

Hasil run dengan metode gauss jordan

```
Matriks Input:
     1.000     0.500     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     1.000
     0.500     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.000
     0.333     0.250     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.000
     0.250     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.000
     0.200     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.000
     0.167     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.000
     0.143     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.000
     0.125     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.059     0.000
     0.111     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.059     0.056     0.000
     0.100     0.091     0.833     0.077     0.071     0.067     0.063     0.059     0.056     0.053     0.000
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000    11.625     6.636
     0.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000   -16.544   -10.443
     0.000     0.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000   -82.186   -16.543
     0.000     0.000     0.000     1.000     0.000     0.000     0.000     0.000     0.000  -162.237   -32.793
     0.000     0.000     0.000     0.000     1.000     0.000     0.000     0.000     0.000   237.789    48.487
     0.000     0.000     0.000     0.000     0.000     1.000     0.000     0.000     0.000    -2.256    -0.262
     0.000     0.000     0.000     0.000     0.000     0.000     1.000     0.000     0.000    -2.169    -0.227
    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000     0.000     0.00032989310551973.3126701753664120.857
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     1.000     0.000-32989310551975.250-6701753664121.031
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     1.000    -1.634    -0.096
x1= 6.636110211930081 - 11.624814346207614j
x2= -10.443278849343429 + 16.54431482623113j
x3= -16.54348323677334 + 82.18576937786747j
x4= -32.792984156779596 + 162.23711917207106j
x5= 48.48661382306642 - 237.7889864030999j
x6= -0.26213297175437905 + 2.2563990377552035j
x7= -0.22712230082195206 + 2.169469203149112j
x8= -6.701753664121031E12 + 3.298931055197525E13j
x9= -0.09596486848744372 + 1.6338727745287314j
x10= 6.701753664120857E12

---------------------------------------------------------------
```

Hasil run dengan metode matriks balikan:

```
Path lengkap dari file .txt matriks: ./test/1d10.txt
x1: 4.274537907572639
x2: -7.08231371714405
x3: 0.1524940919831831
x4: 0.1653647651885352
x5: 0.17995981151224216
x6: 0.1962527982392904
x7: 0.21360373855506198
x8: 0.22958952965621904
x9: 0.23595515864083202
x10: 0.20314924901270232
```

Hasil run dengan kaidah cramer:

2a. SPL berbentuk matriks augmented :

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

Hasil run dengan metode gauss:

```
Matriks Input:
     1.000    -1.000     2.000    -1.000    -1.000
     2.000     1.000    -2.000    -2.000    -2.000
    -1.000     2.000    -4.000     1.000     1.000
     3.000     0.000     0.000    -3.000    -3.000
Matriks eselon:
     1.000    -1.000     2.000    -1.000    -1.000
     0.000     1.000    -2.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000
x1= -1.0 + 1.0(0.0 + 2.0c) - 2.0c + 1.0d
x2= 0.0 + 2.0c
x3= c
x4= d
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     1.000    -1.000     2.000    -1.000    -1.000
     2.000     1.000    -2.000    -2.000    -2.000
    -1.000     2.000    -4.000     1.000     1.000
     3.000     0.000     0.000    -3.000    -3.000
Matriks eselon tereduksi:
     1.000     0.000     0.000    -1.000    -1.000
     0.000     1.000    -2.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000
x1= -1.0 + 1.0d
x2= 0.0 + 2.0c
x3= c
x4= d
```

Hasil run dengan metode matriks balikan:

```
Path lengkap dari file .txt matriks: ./test/2a.txt
x1: NaN
x2: NaN
x3: NaN
x4: NaN
```

Hasil run dengan kaidah cramer:

```
Path lengkap dari file .txt matriks: ./test/2a.txt
x1: NaN
x2: NaN
x3: NaN
x4: NaN
```

2b. SPL berbentuk matriks augmented:

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

Hasil run dengan metode gauss:

```
Matriks Input:
     2.000     0.000     8.000     0.000     8.000
     0.000     1.000     0.000     4.000     6.000
    -4.000     0.000     6.000     0.000     6.000
     0.000    -2.000     0.000     3.000    -1.000
     2.000     0.000    -4.000     0.000    -4.000
     0.000     1.000     0.000    -2.000     0.000
Matriks eselon:
     1.000     0.000     4.000     0.000     4.000
    -0.000     1.000    -0.000    -1.500     0.500
    -0.000    -0.000     1.000    -0.000     1.000
     0.000     0.000     0.000     1.000     1.000
     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000
x1= 0.0
x2= 2.0
x3= 1.0
x4= 1.0
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     2.000     0.000     8.000     0.000     8.000
     0.000     1.000     0.000     4.000     6.000
    -4.000     0.000     6.000     0.000     6.000
     0.000    -2.000     0.000     3.000    -1.000
     2.000     0.000    -4.000     0.000    -4.000
     0.000     1.000     0.000    -2.000     0.000
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.000     0.000
     0.000     1.000     0.000     0.000     2.000
     0.000     0.000     1.000     0.000     1.000
     0.000     0.000     0.000     1.000     1.000
     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000
x1= 0.0
x2= 2.0
x3= 1.0
x4= 1.0
```

Hasil run dengan metode matriks balikan:

```
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
x5: NaN
x6: NaN
```

Hasil run dengan kaidah cramer:

```
Path lengkap dari file .txt matriks: ./test/2b.txt
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
```

3a. SPL :

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

Setelah dikonversi menjadi matriks *augmented*,
Hasil run dengan metode gauss:

```
Matriks Input:
     8.000     1.000     3.000     2.000     0.000
     2.000     9.000    -1.000    -2.000     1.000
     1.000     3.000     2.000    -1.000     2.000
     1.000     0.000     6.000     4.000     3.000
Matriks eselon:
     1.000     0.125     0.375     0.250     0.000
     0.000     1.000    -0.200    -0.286     0.114
     0.000     0.000     1.000    -0.195     0.760
     0.000     0.000     0.000     1.000    -0.258
x1= 0.064527027027027 - 0.125(0.04054054054054057 + 0.19999999999999998c) - 0.375(0.7094594594594594)
x2= 0.04054054054054057 + 0.19999999999999998(0.7094594594594594)
x3= 0.7094594594594594
x4= -0.258108108108108
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     8.000     1.000     3.000     2.000     0.000
     2.000     9.000    -1.000    -2.000     1.000
     1.000     3.000     2.000    -1.000     2.000
     1.000     0.000     6.000     4.000     3.000
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.000    -0.224
     0.000     1.000     0.000     0.000     0.182
     0.000     0.000     1.000     0.000     0.709
     0.000     0.000     0.000     1.000    -0.258
x1= -0.22432432432432436
x2= 0.18243243243243246
x3= 0.7094594594594594
x4= -0.258108108108108
```

Hasil dengan metode matriks balikan:

```
Path lengkap dari file .txt matriks: ./test/3a.txt
x1: -0.22432432432432434
x2: 0.1824324324324324
x3: 0.7094594594594594
x4: -0.25810810810810814
```

Hasil dengan kaidah cramer:

```
Path lengkap dari file .txt matriks: ./test/3a.txt
x1: -0.22432432432432434
x2: 0.18243243243243243
x3: 0.7094594594594594
x4: -0.2581081081081081
```

3b. SPL

b.

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

Setelah melakukan konversi ke matriks *augmented*,
Hasil run dengan metode gauss:

```
Matriks Input:
     0.000     0.000     0.000     0.000     0.000     0.000     1.000     1.000     1.000    13.000
     0.000     0.000     0.000     1.000     1.000     1.000     0.000     0.000     0.000    15.000
     1.000     1.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000     8.000
     0.000     0.000     0.043     0.000     0.043     0.750     0.043     0.750     0.614    14.790
     0.000     0.250     0.914     0.250     0.914     0.250     0.914     0.250     0.000    14.310
     0.614     0.750     0.043     0.750     0.043     0.000     0.043     0.000     0.000     3.810
     0.000     0.000     1.000     0.000     0.000     1.000     0.000     0.000     1.000    18.000
     0.000     1.000     0.000     0.000     1.000     0.000     0.000     1.000     0.000    12.000
     1.000     0.000     0.000     1.000     0.000     0.000     1.000     0.000     0.000     6.000
     0.043     0.750     0.614     0.000     0.043     0.750     0.000     0.000     0.043    10.510
     0.914     0.250     0.000     0.250     0.914     0.250     0.000     0.250     0.914    16.130
     0.043     0.000     0.000     0.750     0.043     0.000     0.614     0.750     0.043     7.040
Matriks eselon:
     1.000     1.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000     8.000
     0.000     1.000    -4.198     5.513     0.315     0.000     0.315     0.000     0.000    -8.098
    -0.000    -0.000     1.000    -1.253    -0.061    -0.000    -0.253    -0.000    -0.000     1.943
     0.000     0.000     0.000     1.000     0.065     1.398     1.254     0.000     0.080    16.802
     0.000     0.000     0.000     0.000     1.000     1.339     0.195     0.264     1.028    24.158
    -0.000    -0.000    -0.000    -0.000    -0.000     1.000     0.320    -0.763     0.011     5.111
     0.000     0.000     0.000     0.000     0.000     0.000     1.000     0.035    -0.811    -1.840
    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000     1.000     0.313     7.571
    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000     1.0002993171299363.555
    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000     1.000
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
Linear Equation has no real solution
```

Hasil run dengan metode gauss jordan:

```
Matriks Input:
     0.000     0.000     0.000     0.000     0.000     0.000     1.000     1.000     1.000    13.000
     0.000     0.000     0.000     1.000     1.000     1.000     0.000     0.000     0.000    15.000
     1.000     1.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000     8.000
     0.000     0.000     0.043     0.000     0.043     0.750     0.043     0.750     0.614    14.790
     0.000     0.250     0.914     0.250     0.914     0.250     0.914     0.250     0.000    14.310
     0.614     0.750     0.043     0.750     0.043     0.000     0.043     0.000     0.000     3.810
     0.000     0.000     1.000     0.000     0.000     1.000     0.000     0.000     1.000    18.000
     0.000     1.000     0.000     0.000     1.000     0.000     0.000     1.000     0.000    12.000
     1.000     0.000     0.000     1.000     0.000     0.000     1.000     0.000     0.000     6.000
     0.043     0.750     0.614     0.000     0.043     0.750     0.000     0.000     0.043    10.510
     0.914     0.250     0.000     0.250     0.914     0.250     0.000     0.250     0.914    16.130
     0.043     0.000     0.000     0.750     0.043     0.000     0.614     0.750     0.043     7.040
Matriks eselon tereduksi:
     1.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     1.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     1.000     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000     1.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000     0.000     1.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     1.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     1.000     0.000
    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000    -0.000     1.000
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
Linear Equation has no real solution
```

Hasil dengan metode matriks balikan:

```
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
x5: NaN
x6: NaN
x7: NaN
x8: NaN
x9: NaN
x10: NaN
x11: NaN
x12: NaN
```

Hasil dengan kaidah cramer:

```
ERROR: Undefined determinant. Matrix must be n x n in dimension.
ERROR: Undefined determinant. Matrix must be n x n in dimension.
x1: NaN
x2: NaN
x3: NaN
x4: NaN
x5: NaN
x6: NaN
x7: NaN
x8: NaN
x9: NaN
```

5. Interpolasi

Diketahui data seperti pada soal, berikut hasil run :

```
x = 0.2
P(x) = -2.1990232555541397E10
```

```
x = 0.55
P(x) = -1.6630113370110 89E11
```

```
x = 0.85
P(x) = -3.9719857553397 02E11
```

```
Path lengkap dari file .txt matriks: ./test/5.txt
Polinom interpolasi P(x) berhasil dibuat!
Sekarang input x, dan P(x) akan ditampilkan.
x = 1.28
P(x) = -9.00719925473619E11
```

6. Interpolasi

Diketahui data seperti pada soal, setelah mengonversi tanggal ke dalam tanggal desimal dan menggunakannya sebagai input, berikut adalah hasilnya:

```
x = 5.806
P(x) = 22794.69125020504
x = 8.9677
P(x) = 175757.657289505
x = 9.5
P(x) = 68216.42804718018
```

7. Interpolasi

Dengan menggunakan n = 5, berikut adalah beberapa contoh titik hasil interpolasi:

```
Path lengkap dari file .txt matriks: ./test/7.txt
Polinom interpolasi P(x) berhasil dibuat!
Sekarang input x, dan P(x) akan ditampilkan.
x = 0.3
P(x) = 0.36751318359375O2
```

```
Path lengkap dari file .txt matriks: ./test/7.txt
Polinom interpolasi P(x) berhasil dibuat!
Sekarang input x, dan P(x) akan ditampilkan.
x = 1
P(x) = 0.5345937500000005
```

8. Regresi

Diketahui diberikan sebuah data seperti di soal. Data tersebut akan dimasukan terlebih dahulu oleh program

```
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
----------------------------------------------------------------
Pilihan anda: 5
----------------------------------------------------------------
Regresi Linier Berganda
Masukan Data
Masukan nilai n = 20
Masukan nilai k = 3

Masukan Data X:
X[1][1] = 72.4
X[2][1] = 41.6
X[3][1] = 34.3
X[4][1] = 35.1
X[5][1] = 10.7
X[6][1] = 12.9
X[7][1] = 8.3
X[8][1] = 20.1
X[9][1] = 72.2
X[10][1] = 24
X[11][1] = 23.2
X[12][1] = 47.4
X[13][1] = 31.5
X[14][1] = 10.6
X[15][1] = 11.2
X[16][1] = 73.3
```

```
X[18][2] = 78.7
X[19][2] = 86.8
X[20][2] = 70.9

X[1][3] = 29.18
X[2][3] = 29.35
X[3][3] = 29.24
X[4][3] = 29.27
X[5][3] = 29.78
X[6][3] = 29.39
X[7][3] = 29.69
X[8][3] = 29.48
X[9][3] = 29.09
X[10][3] = 29.6
X[11][3] = 29.38
X[12][3] = 29.35
X[13][3] = 29.63
X[14][3] = 29.56
X[15][3] = 29.48
X[16][3] = 29.4
X[17][3] = 29.28
X[18][3] = 29.29
X[19][3] = 29.03
X[20][3] = 29.37

Masukan Data  Y:
Y[1] = 0.9
Y[2] = 0.91
Y[3] = 0.96
Y[4] = 0.89
Y[5] = 1
```

Kemudian setelah dimasukan nilai yang akan ditaksir(Xk), akan proses data tersebut dengan Normal Estimation Equation sehingga didapatkan sistem persamaan linearnya.

```
Y[12] = 0.94
Y[13] = 1.1
Y[14] = 1.1
Y[15] = 1.1
Y[16] = 0.91
Y[17] = 0.87
Y[18] = 0.78
Y[19] = 0.82
Y[20] = 0.95

Masukan Nilai Yang Akan Ditaksir Xk:
Xk[1] = 50
Xk[2] = 76
Xk[3] = 29.3

Persamaan Linear Hasil Normal Estimation Equation
dalam bentuk matriks augmented :
    20.000   863.100  1530.400   587.840    19.420
   863.100 54876.890 67000.090 25283.395   779.477
  1530.400 67000.090117912.320 44976.867  1483.437
   587.840 25283.395 44976.867 17278.509   571.122

Didapati Beta:
Beta[0] : -3.5077781408831465
Beta[1] : -0.0026249907458783875
Beta[2] : 7.989410472218425E-4
Beta[3] : 0.1541550301982891

Estimasi nilai y : 0.938434226221665
```

Didapati persamaan linearnya dan perhatikan bahwa persamaan tersebut ekuivalen dengan persamaan yang ada pada studi kasus. Kemudian dicari nilai betanya dengan gauss-jordan, supaya dapat didapati estimasi dari masukan taksiran sebelumnya dengan nilai hasil estimasi adalah 0.938434226221665

BAB V

KESIMPULAN, SARAN DAN REFLEKSI

5.1. Kesimpulan

Dari hasil eksperimen dan pembuatan program, penulis menyimpulkan beberapa hal seperti berikut:
Meninjau dari proses pengerjaan dan eksperimen yang dilakukan, penulis dapat menarik beberapa kesimpulan :

1. Sistem Persamaan Linier, Interpolasi, dan regresi linear merupakan metode yang sangat berguna yang dapat diterapkan dalam berbagai bidang keilmuan baik ilmu murni maupun terapan.
2. Penggunaan teknologi github sangat membantu dalam menjaga kelancaran alur pengerjaan proyek dan membantu mengontrol versi kode yang ada.
3. Bahasa Pemrograman java dapat dijalankan pada sistem operasi apapun yang pemrogramannya berbasis objek.

5.2. Saran

Dari tugas yang sudah dibuat, tim penulis menyarankan supaya program ini dapat dipublikasikan, setidaknya kepada khalayak ITB, agar program ini memiliki nilai kebermanfaatan.

5.3. Refleksi

Dari proses pembuatan hingga penyelesaian tugas ini, penulis mendapatkan beberapa pengalaman berharga seperti berikut:

- mendapatkan pengalaman dalam mengimplementasikan bahasa java untuk membuat program
- Dapat membuat algoritma untuk menentukan Sistem Persamaan Linier ,Determinan, Invers Gauss, Gauss-Jordan, Interpolasi, dan regresi linear
- Dapat mengenal anggota tim dengan lebih dekat dan mampu saling aktif dalam bekerja sama , berpendapat, dan tolong menolong.
- Mendapat pengalaman tentang bagaimana memanajemen waktu dengan lebih baik dan tidak menunda suatu pekerjaan.
- Belajar mengetahui dengan jelas perintah yang diberikan supaya tidak terjadi *miss-communication*

DAFTAR PUSTAKA

Profematika. (2019). Eliminasi Gauss dan Contoh Penerapannya. [online] https://www.profematika.com/eliminasi-gauss-dan-contoh-penerapannya/ [Diakses pada September 2020].

Wikipedia. (2018). Eliminasi Gauss. [online] https://id.wikipedia.org/wiki/Eliminasi_Gauss [Diakses pada September 2020].

Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Sistem Persamaan Linier, Determinan, dan Aplikasinya . (2020, September), Bandung: Program Studi Informatika Institut Teknologi Bandung.

Bremer, M. (n.d.). *Math 261A -Spring 2012 Multiple Linear Regression*. [online] Available at: http://mezeylab.cb.bscb.cornell.edu/labmembers/documents/supplement%205%20-%20multiple%20regression.pdf.

Machine Learning From Scratch. (2020). *Multiple Linear Regression: Explained, Coded & Special Cases*. [online]: https://mlfromscratch.com/linear-regression-from-scratch [Diakses pada 1 Oktober 2020].