LAPORAN TUGAS BESAR 1
IF 2123 ALJABAR LINIER DAN GEOMETRI
SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA



Disusun oleh:

Ronggur Mahendra Widya Putra (13519008)

Ruhiyah Faradishi Widiaputri (13519034)

Nabila Hannania (13519097)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2020

DAFTAR ISI

BAB 1

Deskripsi Masalah

1.1. Abstraksi

Sistem persamaan linier (SPL) $Ax = b$ dengan n peubah (*variable*) dan m persamaan adalah berbentuk

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
&: \\
&: \\
&: \\
&: \\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m
\end{aligned}
$$

yang dalam hal ini x_i adalah peubah, a_{ij} dan b_i adalah koefisien $\in$ 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).

Sebuah matriks M berukuran $n \times n$

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

determinannya adalah

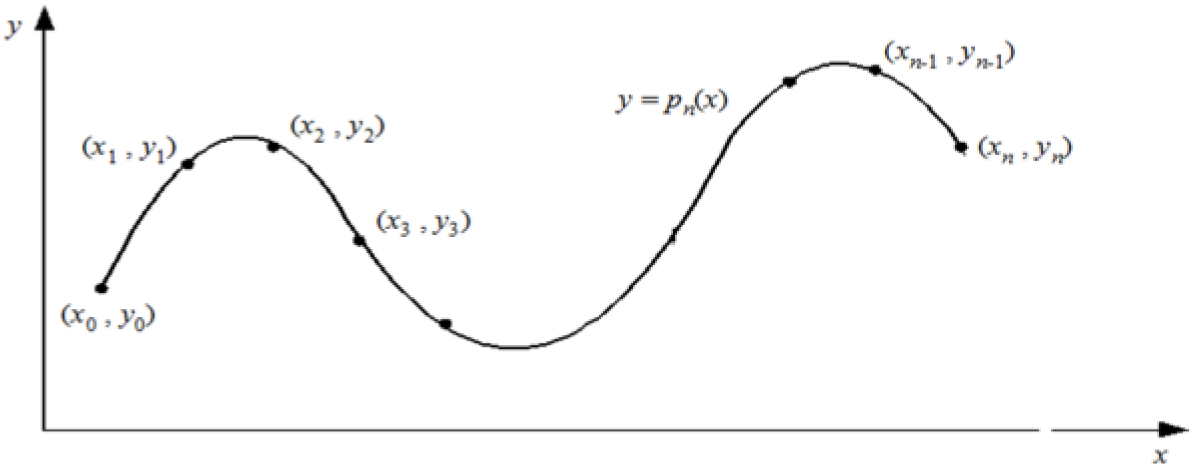
$$
\det(M) = \begin{vmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{vmatrix}
$$

Determinan matriks M berukuran $n \times n$ dapat dihitung dengan beberapa cara: reduksi baris dan ekspansi kofaktor.

SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

1.2. Interpolasi Polinom

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.



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, …, a_n,

$$a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n = y_1$$

$$\ldots \qquad\qquad \ldots$$

$$a_0 + a_1x_n + a_2x_{n2} + ... + a_n x_{nn} = y_n$$

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 sisten persamaan lanjar yang terbentuk adalah

$$a0 + 8.0a1 + 64.00a2 = 2.0794$$
$$a0 + 9.0a1 + 81.00a2 = 2.1972$$
$$a0 + 9.5a1 + 90.25a2 = 2.2513$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan a0 = 0.6762, a1 = 0.2266, dan a2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah p2(x) = 0.6762 + 0.2266x - 0.0064x2. Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: p2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

1.3. 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 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 βi dapat digunakan Normal Estimation Equation for Multiple Linear Regression sebagai berikut:

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB 2
Teori Singkat

2.1. Metode eliminasi Gauss

Metode eliminasi Gauss adalah salah satu metode yang digunakan untuk menyelesaikan persamaan matriks yang berbentuk

Ax = b

Untuk menyelesaikan sistem persamaan linier dengan menggunakan metode eliminasi Gauss, sistem persamaan linier tersebut dinyatakan terlebih dahulu dalam bentuk matriks augmented.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2k} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} & b_k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}. \longrightarrow \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1k} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2k} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} & b_k \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}.$$

Gambar: mengubah persamaan matriks (kiri) menjadi matriks
augmented (kanan)

Matriks augmented ini kemudian diubah menjadi matriks eselon baris. Matriks eselon baris adalah matriks yang memiliki 1 utama pada setiap baris kecuali pada baris yang seluruhnya nol. Untuk mengubah matriks augmented menjadi bentuk matriks eselon baris kita dapat menerapkan operasi baris elementer.

Setelah diperoleh matriks eselon baris langkah selanjutnya adalah dengan menyelesaikan persamaan yang bersesuaian dengan matriks eselon baris dengan menggunakan substitusi mundur.

2.2. Metode eliminasi Gauss-Jordan

Metode eliminasi Gauss adalah metode yang digunakan untuk menyelesaikan persamaan matriks yang berbentuk

Ax = b

dengan menggunakan matriks eselon baris tereduksi. Matriks eselon baris tereduksi merupakan bentuk khusus dari matriks eselon baris di mana setiap kolom yang memiliki 1 utama memiliki nol di tempat lain.

$$
\begin{pmatrix}
1 & * & * & * & * & * & * & * & * \\
0 & 1 & * & * & * & * & * & * & * \\
0 & 0 & 0 & 1 & * & * & * & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Gambar: contoh matriks eselon baris tereduksi

Untuk menyelesaikan sistem persamaan linier dengan metode Gauss-Jordan, pertama-tama sistem persamaan linier tersebut dinyatakan dalam bentuk matriks augmented lalu diubah ke dalam bentuk matriks eselon baris tereduksi. Dari hasil matriks eselon baris tereduksi kita dapat secara langsung mengetahui solusi dari sistem persamaan linier.

2.3. Determinan

Untuk menentukan determinan suatu matrik terdapat 2 metode yang akan dijelaskan di bawah ini

2.3.1. Metode Ekspansi Kofaktor

Metode Ekspansi Kofaktor adalah metode untuk menentukan determinan dengan cara menjumlahkan perkalian semua elemen dari satu baris bebas dengan determinan dari matriks kofaktor dari elemen tersebut.

2.3.2. Metode Operasi Baris Elementer(OBE)

Metode OBE adalah metode untuk menentukan determinan dengan cara merubah matriks menjadi matriks segitiga bawah, kemudian determinan diperoleh dari perkalian elemen diagonal utama.

2.4. Matriks balikan

Matriks balikan adalah matriks yang apabila dikalikan dengan suatu matriks bujursangkar menghasilkan sebuah matriks satuan. Jika A merupakan sebuah matriks bujursangkar, maka balikannya ditulis dengan notasi A^{-1}. Jika suatu matriks bukan merupakan matriks bujursangkar, maka matriks tersebut tidak memiliki matriks balikan (inverse).

Matriks balikan, A^{-1}, banyak dipakai dalam pengolahan matriks. Misalnya dalam pengukuran statistik, pencocokan fungsi pada data hasil pengamatan menggunakan metode kuadrat terkecil (least square). Di sini, nilai A^{-1} memberikan informasi tentang galat mutlak yang dikandung data. Selain itu, matriks balikan juga dapat dipakai untuk menghitung solusi sistem persamaan linier.

Untuk matriks 2 × 2,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

matriks balikannya adalah

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad , \ a_{11}a_{22} - a_{12}a_{21} \neq 0.$$

Nilai $a_{11}a_{22} - a_{12}a_{21}$ ini disebut determinan. Determinan dilambangkan dengan dua buah garis tegak (| |). Bila determinan A = 0, matriks A tidak mempunyai balikan, sehingga dinamakan matriks singular. Sistem persamaan linier yang mempunyai matriks A singular (sistem singular) tidak mempunyai solusi yang unik, yaitu solusinya banyak atau solusinya tidak ada.

Untuk matriks n x n, matriks balikannya dapat diperoleh dengan metode eliminasi GaussJordan, yaitu:

$$[\,A \mid I\,] \xrightarrow{\text{eliminasi G - J}} [\,I \mid A^{-1}]$$

$$\underbrace{\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{array}\right]}_{A \qquad\qquad I} \longrightarrow \underbrace{\left[\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & p_{11} & p_{12} & \dots & p_{1n} \\ 0 & 1 & \dots & 0 & p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & \dots & 1 & p_{n1} & p_{n2} & \dots & p_{nn} \end{array}\right]}_{I \qquad\qquad A^{-1}}$$

- Metode Matriks Balikan

Misalkan *A*-1 adalah matriks balikan dari *A*. Hasil kali *A* dengan *A*-1 menghasilkan matriks identitas *I*,

$$AA^{-1} = A^{-1}A = I$$

Bila matriks *A* dikalikan dengan *I* akan menghasilkan matriks *A* sendiri,

$$AI = IA = A$$

Berdasarkan dua kesamaan di atas, sistem persamaan linier $Ax = b$ dapat diselesaikan sebagai berikut:

$Ax = b$

$A^{-1} Ax = A^{-1} b$ {kalikan kedua ruas dengan A^{-1}}

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

$x = A^{-1} b$

Jadi, penyelesaian sistem persamaan linier $Ax = b$ adalah $x = A^{-1} b$ dengan syarat A^{-1} ada (matriks A mempunyai inverse).

Penyelesaian dengan SPL metode matriks balikan tidak lebih bagus daripada metode eliminasi Gauss, sebab lebih banyak proses komputasi yang dibutuhkan. Metode matriks balikan biasanya digunakan untuk penyelesaian sejumlah SPL dengan matriks *A* yang sama tetapi dengan matriks kolom *b* yang berbeda - beda:

$Ax = b_I$

$Ax = b_{II}$

$Ax = b_{III}$

...

2.5. Matriks kofaktor

- Minor

Yang dimaksud dengan Minor unsur a_{ij} adalah determinan upa - matriks (*submatrix*) yang elemen-elemennya tidak berada pada baris *i* dan kolom *j*. Minor unsur a_{ij} dinotasikan dengan M_{ij}.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad M_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$$

Minor Matriks A (ordo 3x3)

$$|M_{11}| = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} \quad |M_{21}| = \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} \quad |M_{31}| = \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}$$

$$|M_{12}| = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} \quad |M_{22}| = \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \quad |M_{32}| = \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}$$

$$|M_{13}| = \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \quad |M_{23}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \quad |M_{33}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

- Kofaktor

Kofaktor dari baris ke-i dan kolom ke-j ditulis dengan

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

Cara cepat untuk menentukan penggunaan tanda (+) atau (-) pada C_{ij} adalah dari susunan :

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

2.6. Matriks Adjoin

- Adjoin

Adjoin dari suatu Matriks A adalah transpose dari matriks kofaktornya.

adj(A) = transpose matriks kofaktor

Contoh :

Terdapat Matriks A :

$$A = \begin{bmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{bmatrix}$$

Kemudian dicari matriks kofaktor A :

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

Adjoin dari A adalah transpose matriks kofaktor A:

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

2.7. Kaidah Cramer

Jika **Ax = b** adalah sistem persamaan linear dengan n variabel di mana determinan **A** tidak sama dengan nol, maka sistem persamaan linier tersebut mempunyai solusi yang unik. Solusinya adalah:

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

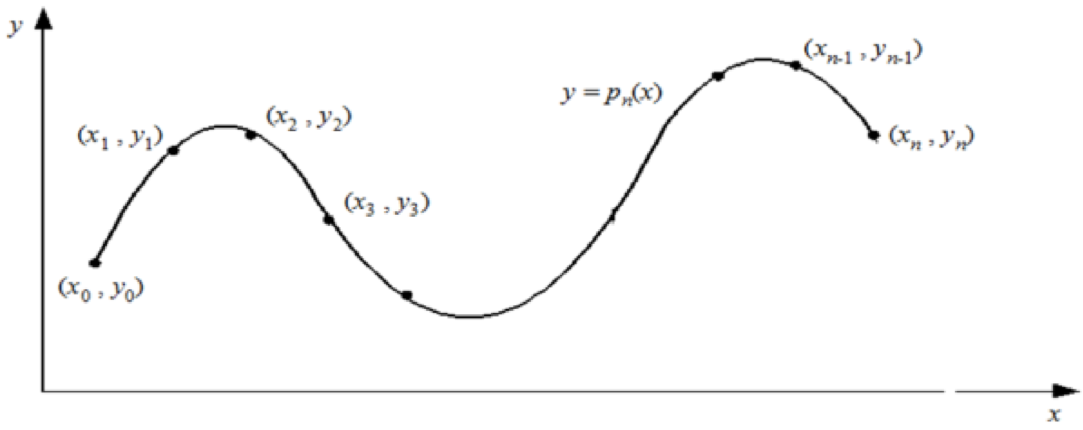
di mana Aj adalah matriks yang diperoleh dari penggantian elemen-elemen di kolom ke-j matriks A dengan elemen-elemen matriks **b**

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

2.8. interpolasi Polinom

Interpolasi dapat diartikan sebagai perkiraan suatu nilai tengah dari suatu set nilai yang diketahui. Interpolasi dengan pengertian yang lebih luas merupakan upaya mendefinisikan suatu fungsi dekatan fungsi analitik yang tidak diketahuin atau pengganti fungsi rumit yang tak mungkin diperoleh persamaan analitiknya. Masalah umum interpolasi adalah menjabarkan data untuk fungsi dekatan, dan salah satu metode penyelesaiannya dinamakan Metoda prinsip Substitusi.

Bila data diketahui mempunyai ketelitian yang sangat tinggi, maka kurva cocokannya dibuat melalui setiap titik. Kita katakan di sini bahwa kita menginterpolasi titik-titik data dengan sebuah fungsi. Bila fungsi cocokan yang digunakan berbentuk polinom, polinom tersebut dinamakan polinom interpolasi. Pekerjaan menginterpolasi titik data dengan sebuah polinom disebut interpolasi (dengan) polinom.



Terdapat 3 metode interpolasi :

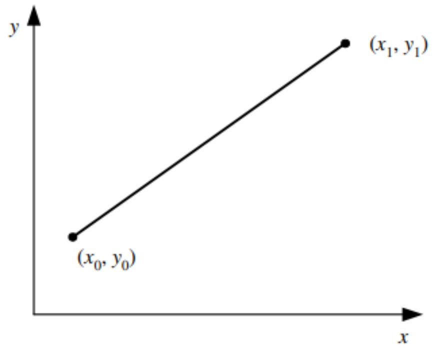
1. Interpolasi Lanjar

Interpolasi lanjar adalah interpolasi dua buah titik dengan sebuah garis lurus. Misal diberikan dua buah titik, (x_0, y_0) dan (x_1, y_1). Polinom yang menginterpolasi kedua titik itu adalah

$$p_1(x) = a_0 + a_1x$$

$$y_0 = a_0 + a_1x_0$$

$$y_1 = a_0 + a_1x_1$$



$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} \qquad a_0 = \frac{x_1y_0 - x_0y_1}{x_1 - x_0}$$

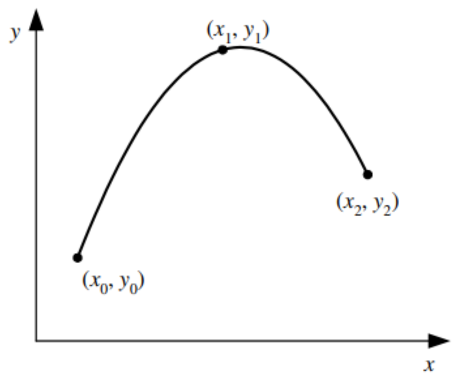
$$p_1(x) = \frac{x_1y_0 - x_0y_1}{x_1 - x_0} + \frac{(y_1 - y_0)x}{(x_1 - x_0)}$$

$$p_1(x) = y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)}(x - x_0)$$

2. Interpolasi Kuadratik

Misal diberikan tiga buah titik data, (x_0, y_0), (x_1, y_1), dan (x_2, y_2). Polinom yang menginterpolasi ketiga buah titik itu adalah polinom kuadrat yang berbentuk:

$$p_2(x) = a_0 + a_1x + a_2x^2$$



$y_0 = a_0 + a_1x_0 + a_2x_0^2$

$y_1 = a_0 + a_1x_1 + a_2x_1^2$

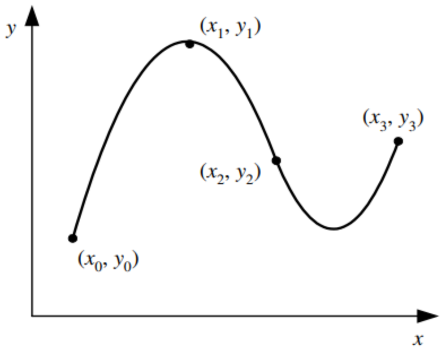
$y_2 = a_0 + a_1x_2 + a_2x_2^2$

Kemudian, nilai a_0, a_1, dan a_2 dicari menggunakan metode eliminasi Gauss

3. Interpolasi Kubik

Misal diberikan empat buah titik data, (x_0, y_0), (x_1, y_1), (x_2, y_2), dan (x_3, y_3). Polinom yang menginterpolasi keempat buah titik itu adalah polinom kubik yang berbentuk:

$$p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$



$y_0 = a_0 + a_1x_0 + a_2x_0^2 + a_3x_0^3$

$y_1 = a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3$

$y_2 = a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3$

$y_3 = a_0 + a_1x_3 + a_2x_3^2 + a_3x_3^3$

Kemudian, nilai a_0, a_1, a_2, dan dicari menggunakan metode eliminasi Gauss

Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi :

$$p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$

$$\begin{aligned}
a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n &= y_0 \\
a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n &= y_1 \\
\ldots \qquad\qquad & \ldots \\
a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n &= y_n
\end{aligned}$$

2.9. Regresi Linier Berganda

Regresi linier Berganda adalah suatu metode analisis hubungan antara dua atau lebih variabel independen (X_1, X_2,….X_n) dengan variabel dependen (Y). Metode ini pertama menggunakan formula *Normal Estimation Equation for Multiple Linier Regression* sebagai berikut :

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

Kemudian menggunakan Metode Eliminasi Gauss diatas untuk mendapatkan nilai (b_1, b_2,….b_n) dan dimasukan ke rumus umum regresi linier berganda yaitu.

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

BAB 3
Implementasi program dalam Java

3.1. Metode eliminasi Gauss,

- Nama file : MatSPL.java
- Deskripsi:

  Class ini mengandung atribut-atribut dan method-method yang diperlukan dalam menghitung solusi-solusi dari Sistem Persamaan Linier dengan menggunakan metode, Gauss, metode Gauss-Jordan, dan metode Cramer.
- Atribut:
  - Mat[][] adalah matriks yang bertipe double dengan ukuran baris x kolom.
  - baris adalah atribut bertipe integer yang berisi ukuran baris matriks Mat
  - kolom adalah atribut bertipe integer yang berisi ukuran kolom matriks Mat.
- Method:
  - isiMatriks(double[][] Mat, int baris, int kolom)

    Method ini merupakan prosedur yang menerima masukan elemen-elemen matriks Mat[][] yang berukuran baris x kolom. Method ini digunakan dalam menu penyelesaian SPL dengan menggunakan metode Gauss, metode Gauss-Jordan, dan metode Crammer.
  - tulisMatriks(double[][] Mat, int baris, int kolom)

    Method ini digunakan untuk mencetak matriks Mt[][] yang berukuran baris x kolom ke layar. Method ini memang tidak digunakan dalam menu penyelesaian SPL tapi sangat dipakai saat mengetes method lainnya.
  - matriksEselonBaris(double[][] Mat, int baris, int kolom)

Method ini digunakan untuk mengubah matriks masukan Mat[][] yang berukuran baris x kolom menjadi bentuk matriks eselon baris. Method ini digunakan dalam menu penyelesaian SPL dengan menggunakan metode Gauss dan metode Gauss-Jordan.

- barisNol(double[][] Mat, int kolom, int brs)

  Method ini akan mengembalikan nilai boolean apakah baris ke-brs di matriks Mat[][] berisi elemen nol semua. Method ini digunakan di dalam method matriksEselonBaris.

- barisNolKebawah(double[][] Mat, int brs, int baris, int kolom)

  Method ini berguna untuk memindahkan baris dalam matriks Mat[][] yang elemennya nol semua (yaitu di baris ke-brs) ke bagian bawah matriks. Method ini akan dijalankan saat method barisNol di dalam method matriksEselonBaris menghasikan true.

- barisPatokan(double[][] Mat, int brs, int kol, int baris)

  Method ini digunakan pada matriks Mat[][] yang berukuran baris x kolom untuk mencari pada baris ke berapa (brs) di suatu kolom ke-kol terdapat 1 utama. Method ini digunakan dalam method matriks EselonBaris.

- tukarBaris(double[][] Mat, int brs, int kol, int baris)

  Method ini digunakan untuk mengganti elemen-elemen antara baris ke-brs1 dan brs-2. Method ini digunakan dalam method matriksEselonBaris.

- SPLGauss (double[][] Mat, int baris, int kolom)

  Method ini digunakan untuk mengembalikan array yang isinya hasil dari SPL dengan metode Gauss apabila solusinya pasti unik. Method ini akan digunakan oleh class interpolasi dan class regresi linier.

- solusiSPLGauss(double[][] Mat, int baris, int kolom)

  Method ini digunakan untuk mencetak solusi SPL dengan menggunakan metode Gauss baik di layar maupun pada file txt. Solusi dari SPL ini bisa jadi unik, memiliki parameter (solusi tidak unik), atau tidak memiliki solusi sama sekali.

- barisMana$(double[][] Mat, int $, int baris, int kolom)

Method ini berguna untuk mencari di baris mana $ menjadi angka bukan nol pertama di matriks Mat yang berukuran baris x kolom. Method ini digunakan dalam method solusiSPLGauss.

- pertamaBrsBukanNol(double[][] Mat, int brs, int kolom)

  Method ini digunakan untuk mengeluarkan pada kolom keberapa dalam suatu baris (baris ke-brs) pertama kali dijumpai elemen tidak nol di matriks Mat.

- banyakBukanNolBaris(double[][] Mat, int brs, int kolom)

  Method ini berfungsi untuk menghitung berapa banyak elemen pada satu baris (baris ke-brs) selain di bagian kolom terakhir yang nilainya bukan nol di matriks Mat.

- banyak$SesudahElArr(String[] arr, int kolel, int kolom)

  Method ini digunakan untuk menghitung berapa banyak elemen yang bernilai “$” di array arr mulai dari arr[kolel] sampai arr[kolom-2] (selain kolom terakhir)

- sblVarTakNol_(double[][] Mat, String[] arr, int brs, int kolel)

  Method ini mengembalikan berapa banyak variabel yang nilainya tidak nol di baris mulai dari kolom ke-0 sampai di kolom ke-(kolel-1)

- matriksEselonReduksi(double[][] Mat, int baris, int kolom)

  Method ini berguna untuk mengubah matriks Mat yang berukuran baris x kolom menjadi ke bentuk matriks eselon tereduksi

- kofaktor(double[][] MatriksInput)

  Method ini digunakan untuk mengembalikan nilai determinan MatriksInput dengan menggunakan metode kofaktor

- solusiSPLCramer(double[][] Mat, int baris, int kolom)

  Method ini digunakan untuk mengembalikan solusi SPL dengan menggunakan metode cramer di mana banyak baris harus sama dengan banyak kolom - 1. Hasil penghitungan kemudian akan ditampilkan ke layar dan juga disimpan dalam bentuk file txt.

- solusiSPLGaussJordan(double[][] Mat, int baris, int kolom)

Method ini digunakan untuk mengembalikan solusi SPL dengan menggunakan metode Gauss-Jordan. Hasil penghitungan kemudian akan ditampilkan ke layar dan juga disimpan dalam bentuk file txt.

- solusiSPLGaussJordan1(double[][] Mat, int baris, int kolom)

  Method ini digunakan untuk mengembalikan solusi SPL dengan menggunakan metode Gauss-Jordan. Merupakan perbaikan dari method sebelumnya dan banyak mengambil inspirasi dari method solusiSPLGauss

- banyakTidakNolSesudahElmt(double[][] Mat, int kolapa, int bar, int kolom)

  Method ini berfungsi mengembalikan banyaknya elemen tidak nol di Mat[bar][kolapa] sampai Mat[bar][kolom-2] pada matriks Mat.

3.2 Metode Eliminasi Gauss-Jordan

File class yang digunakan dalam menyelesaikan SPL menggunakan metode Gauss-Jordan sama dengan file class untuk menyelesaikan SPL dengan menggunakan metode eliminasi Gauss yaitu class MatSPL.java

3.3. Determinan

Nama file : Determinan.java

Deskripsi : Program kelasin ini menerima suatu matriks dari user atau file lalu menggunakan metode Operasi Baris ELementer (OBE) atau ekspansi Kofaktor menghitung Determinan dari matriks tersebut lalu memberikan ya ke layar atau menuliskanya ke suatu file

Atribut :

- Double[][] matriksInput : matriks input dari user
- Integer i,j,k, :indeks dari matriksInput
- Integer a,b : indeks dari matriks minor
- Double[][] minor : matriks minor dari matriks input
- Double det,ret : hasil determinan dari matriks
- Integer cntswap : jumlah matriks ditukar
- Double tmp : variabel sementara untuk swap

- Boolean found :boolean untuk ditemukanya elemen yang tidak 0
-

Method :

- kofaktor(double[][] MatriksInput) -> Double

  {fungsi ini menerima input berupa matriks persegi lalu menggunakan metode ekspansi kofaktor untuk mendapatkan determinannya}
- OBE(double[][] MatriksInput) -> Double

  {fungsi ini menerima input berupa matriks persegi lalu menggunakan metode Operasi Baris ELementer (OBE) untuk mendapatkan determinannya}
- bacaFile() ->double[][]

  { fungsi membaca suatu file dengan directory dari user dengan format:

  <kolom>

  <Baris>

  <elemen 1,1> <elemen 1,2> .. <elemen 1,kolom>

  : :

  : :

  <elemen baris,1> <elemen baris,2> .. <elemen baris,kolom>

  Lalu mengembalikan suatu matriks sesuai dengan format diatas

  }
- bacaMatriks() -> double[][]

  { fungsi membaca inpput dari user dengan format:

  <kolom>

  <Baris>

  <elemen 1,1> <elemen 1,2> .. <elemen 1,kolom>

  : :

  : :

  <elemen baris,1> <elemen baris,2> .. <elemen baris,kolom>

  Lalu mengembalikan suatu matriks sesuai dengan format diatas

  }

- tulisMatriks(double[][] Matriks)

  {tulis matriks ini ada untuk keperluan debugging dimana prosedur ini memberikan matriks kelayar}

- tulisFile(double hasil, string directory)

  {prosedur mendapatkan suatu double yang merupakan hasil perhitungan Operasi Bilangan ELementer atau ekspansi kofaktor dari suatu matriks lalu menuliskan ke suatu file pada directory yang didapatkan}

3.4. Metode Matriks Balikan

❖ Nama class file : SPLMatrixBalikan.java

❖ Deskripsi :

Program class ini menerima input matriks yang berasal dari user (keyboard) atau dari file. Sebelum matriks di-input, program akan meminta jumlah persamaan SPL (m) yang ingin dicari (diasumsikan jumlah persamaannya sama dengan jumlah variabel agar matriks dapat dieksekusi). Untuk pilihan input dari user, kita harus memasukkan elemen matriks A (orde mxm) dan b (orde mx1) secara terpisah. Untuk input dari file, akan diminta nama (alamat) file yang akan di-input-kan. Input dari file itu berupa matriks augmented, maka harus dipisahkan dulu matriks A dan b terlebih dahulu. Kemudian, dicari inverse dari matriks A dengan memanggil method yang ada di file class MatrixBalikan.java. Setelah didapat inverse A, matriks balikan tersebut dikalikan dengan matriks b (A-1.b) untuk memperoleh solusi SPL. Jika matriks A memiliki determinan yang tidak sama dengan nol, maka solusi yang diperoleh adalah solusi tunggal dan program akan menampilkan nilai dari solusi SPL. Namun, jika determinan matriks A sama dengan nol, maka solusinya itu bukan merupakan solusi yang tunggal (unik), maka akan ditampilkan pesan "SPL tidak memiliki solusi tunggal (unik)". Setelah itu, hasil eksekusi akan disimpan ke dalam file dengan format txt dengan penamaan sesuai dengan keinginan kita.

❖ Atribut :

- m (integer) : jumlah persamaan
- input (integer) : jenis input
- i,j,k (integer) : indeks
- det (double) : determinan matriks
- matrixA (matriks of double)
- matrixB (matriks of double)

- Alamat (string) : nama file input
- n (integer) : panjang matriks A
- MInv (matriks of double) : matriks balikan A
- X (matriks of double) : solusi SPL
- alamat2 (string) : nama file output

❖ Method :

- SPLInverse () : menjalankan semua perintah yang ada pada deskripsi file class SPLMatrixBalikan.java

3.5. Matriks Balikan

❖ Nama class file : MatrixBalikan.java

❖ Deskripsi :

Program ini menerima 2 jenis input, yaitu dari user (keyboard) dan dari file. Kemudian, akan meminta input jumlah baris dan kolom(n), jumlah baris dan kolom harus sama agar dapat dieksekusi. Untuk input dari user, kita akan menginputkan setiap elemen matriks dalam betuk matriks itu sendiri. Untuk input dari file, akan diminta nama (alamat) file yang akan di-input-kan, lalu program akan mengubah data dari file kebentuk matriks (nxn). Kemudian, akan dicari determinan dari matriks tersebut. Jika determinannya sama dengan nol, maka akan keluar pesan "Matriks Balikan Tidak Ada". Jika determinan tidak nol, pertama akan dicari inverse matriks dengan memanggil method inverse pada file tersebut. Pada method inverse ini, akan dicari matriks kofaktor dari matriks input, lalu di-tranpose membentuk matriks adjoin. Untuk mencari inverse dari matriks input, natriks adjoinnya dikalikan dengan 1/determinan matriks input. Setelah didapat matriks balikannya, akan ditampilkan ke layar dan disimpan ke dalam file. sebelumnya , program akan meminta input nama file output dari matriks yang telah dieksekusi.

❖ Atribut :

- n (integer) : jumlah baris dan kolom matrisk
- i,j,k,h (integer) : indeks
- input (integer) : jenis input
- det (double) : determinan matriks
- min (matriks of double) : matriks minor
- newone (matriks of double) : matriks penyimpanan
- brs, kol (integer) : jumlah baris dan kolom matriks newone

- co (matriks of double) : matriks kofaktor
- M1(matriks of double) : penyimpanan matriks minor
- Trans (matriks of double) : matriks transpose
- MInv (matriks of double) : matriks inverse
- alamat (string) : nama file input
- alamat2 (string) : nama file output

  Method :
- determinan (double[][] MatriksInput) : menerima input matriks dan mengembalikan determinan (double) matriks tersebut. Pada method ini, memanggil method yang ada pada file class Determinan.java (method OBE)
- minor (double a[][]) : menerima input matriks dan mengembalikan matriks minor-nya (matriks of double).
- cofactor (double a[][]) : menerima input matriks dan mengembalikan matriks kofaktornya (matriks of double). Method ini memanggil method minor.
- Transpose (double a[][]) : menerima input matriks dan mengembalikan tranpose dari matriks tersebut (matriks of double).
- inverse (double a[][]) : menerima input matriks dan mengembalikan matriks balikannya (matriks of double). Method ini memanggil method determinan, cofactor, dan Transpose.
- MainInverse() : meminta input untuk jenis input matriks, jumlah baris dan kolom matriks input, elemen matriks input, alamat/nama file input (untuk yang input dari file), mencari determinan, mencari inverse matriks, menampilkan matriks ke layar, dan menyalin matriks solusi ke file, serta meminta input nama file output-nya.

3.6. Kaidah Cramer

File class yang digunakan dalam menyelesaikan SPL menggunakan metode Cramer sama dengan file class untuk menyelesaikan SPL dengan menggunakan metode eliminasi Gauss yaitu class MatSPL.java

3.7. Interpolasi Polinom

❖ Nama file : interpolasi.java

❖ Deskripsi :

Program class ini dapat menerima input dari user (keyboard) atau dari file. Setelah dipilih akan menggunakan input dari mana, program akan meminta input jumlah titik (n). Untuk input dari user, elemennya akan diinput dalam bentuk matriks (nx2). Untuk input dari file, akan diminta input nama file, kemudian data dalam file akan disimpan dalam matriks (nx2). Lalu titik - titik tersebut akan diubah kebentuk persamaan polinom, kemudian dibentuk matriks augmented dari persamaan yang didapat. Matriks augmented tersebut dicari solusinya dengan menggunakan method SPLGauss pada class MatSPL.java. Setelah didapat, akan ditampilan persamaan interpolasi polinomnya. Kemudian, program meminta input nilai X yang ingin dicari nilai fungsinya. Kemudian, akan dicari dan ditampilkan nilai fungsi dari X dengan memasukkan nilai X ke persamaan polinom yang didapatkan tadi. Terakhir, hasil eksekusi akan disimpan ke dalam file, program akan meminta input nama file output hasil eksekusi.

❖ Atribut :

- input (integer) : jenis input
- n (integer) : jumlah baris dan kolom matriks
- i,j,k (integer) : indeks
- ListPoint (matriks of double) : matriks yang menyimpan titik koordinat
- alamat (string) : nama file input
- M (matriks of double) : matriks augmented
- pangkat (double) : hasil pangkat elemen ListPoint
- SPL (array of double) : solusi matriks augmented
- X (double) : nilai X yang ingin dicari
- hasil (double) : nilai fungsi dari X
- hasilpanglat (double)
- alamat2 (string) : nama file output

Method :

- inter() : menjalankan semua perintah yang ada pada deskripsi file class interpolasi.java

3.8. Regresi Linier Berganda

❖ Nama file : Determinan.java

❖ Deskripsi :

Program ini menerima Koefisien dari dari semua nilai X (X_1, X_2,….X_n) dan variabel dependen Y dari n jumlah bersamaan dan memproses input tersebut menjadi suatu matriks lalu memprosesnya lagi menggunakan formula *Normal Estimation Equation for Multiple Linier Regression* lalu setelah didapatkan persamaan dari b didapatkan semua nilai b (b_1, b_2,….b_n) dan memaanfaatkan fungsi eliminasi gauss yang sudah dibuat sebelumnya didapatkan semua nilai B lalu memasukanya ke persamaan umum regresi linier berganda.

❖ Atribut :

- Double[][] matriksInput : matriks input dari user
- Double [][] persBeta : matriks hasil dari *Normal Estimation Equation for Multiple Linier Regression*
- Integer n,k,i,j,a : index untuk matriks
- Double temp : nilai sementara untuk perhitungan *Normal Estimation Equation for Multiple Linier Regression*
- Double[] hasil : array 1 dimensi hasil dari eliminasi gauss dari persBeta
- MatSPL Gauss : object yang memiliki method untuk eliminasi gauss
- Sttring alamat : alamat untuk tulis dan bacaFile
- Scanner in, inFile, myObj : object yang memiliki method untuk input dari user
- Double [][] matriks : matris yang akan di tulis ke layar atau di tulis ke file

❖ Method :

- RegresiLinier(double[][] matriksInput) -> double[]

  {menerima suatu matriks input dengan format (X_1, X_2, .. X_n Y) lalu memproses matriks tersebut dengan rumus *Normal Estimation Equation for Multiple Linier Regression* dan juga mencari solusi dari matriks hasil *Normal Estimation Equation for Multiple Linier Regression* dan mengembalikan suatu array 1 dimensi yang merupakan hasil dari b (b_1, b_2,….b_n)}
- bacaFile() -> double[][]

  { fungsi membaca suatu file dengan directory dari user dengan format:

  <kolom>

  <Baris>

  <elemen 1,1> <elemen 1,2> .. <elemen 1,kolom>

  : :

  : :

<elemen baris,1> <elemen baris,2> .. <elemen baris,kolom>

Lalu mengembalikan suatu matriks sesuai dengan format diatas

}

- bacaSPL() ->double[][]

  { fungsi ini membaca input dari user dengan format:

  <kolom>

  <Baris>

  <elemen 1,1> <elemen 1,2> .. <elemen 1,kolom>

  : :

  : :

  <elemen baris,1> <elemen baris,2> .. <elemen baris,kolom>

  Lalu mengembalikan suatu matriks sesuai degan format diatas

  }

- tulisMatriks(double[])

  {prosedur mendapatkan suatu array 1 dimensi yang merupakan hasil perhitungan RegresiLinier lalu menuliskan ke layar sesuai dengan persamaan umum regresi linier berganda}

- tulisFile(double[] hasil, String directory)

  {prosedur mendapatkan suatu array 1 dimensi yang merupakan hasil perhitungan RegresiLinier lalu menuliskan ke suatu file sesuai dengan format persamaan umum regresi linier berganda}

3.9. MENU (main program)

- ❖ Nama class file : Main.java
- ❖ Deskripsi :

Program ini mengeksekusi semua file class yang telah dibuat. Pada program ini akan ditampilkan menu, apa saja operasi yang dapat dijalankan. Kemudian, program akan meminta input menu yang dipilih. Jika memilih Sistem Persamaan Linear, maka akan keluar lagi menu metode apa yang akan digunakan, program kembali meminta input metode yang dipilih.

❖ Atribut :

- input (integer) : pilihan menu awal
- inputSPL : pilihan menu SPL
- opsi (integer) : pilihan input
- baris1, kolom1, baris2, kolom2, baris3, kolom3 (integer) : jumlah baris dan kolom matriks
- M1, M2, M3 : matriks augmented
- i,j,k (integer) : indeks
- alamat1, alamat2, alamat3 (string) : alamat file input
- input : jenis input
- inputregrasilinier : jenis input

❖ Method :

- Menu() : menjalankan program sesuai dengan deskripsi class Main.java

BAB 4
Eksperimen

4.1. Metode eliminasi Gauss

1. Test case: Studi kasus nomor 1 bagian a

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

- Masukan dari keyboard
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 4
isi matriks:
Masukkan persamaan ke-1 :
x1 = 1
x2 = 1
x3 = -1
x4 = -1
Hasil = 1
Masukkan persamaan ke-2 :
x1 = 2
x2 = 5
x3 = -7
x4 = -5
Hasil = -2
Masukkan persamaan ke-3 :
```

```
x1 = 2
x2 = -1
x3 = 1
x4 = 3
Hasil = 4
Masukkan persamaan ke-4 :
x1 = 5
x2 = 2
x3 = -4
x4 = 2
Hasil = 6
isi nama file:
OutputSPL1(Gauss).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss:
SPL tidak memiliki solusi
```

- Dari eksekusi program diperoleh bahwa SPL tersebut tidak memiliki solusi
- File output: OutputSPL1(Gauss).txt

2. Test case: Studi kasus no.1 bagian b

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

- Masukan dari keyboard
- Eksekusi program :

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 5
isi matriks:
Masukkan persamaan ke-1 :
x1 = 1
x2 = -1
x3 = 0
x4 = 0
x5 = 1
Hasil = 3
Masukkan persamaan ke-2 :
x1 = 1
x2 = 1
x3 = 0
x4 = -3
x5 = 0
Hasil = 6
Masukkan persamaan ke-3 :
```

```
x1 = 2
x2 = -1
x3 = 0
x4 = 1
x5 = -1
Hasil = 5
Masukkan persamaan ke-4 :
x1 = -1
x2 = 2
x3 = 0
x4 = -2
x5 = -1
Hasil = -1
isi nama file:
OutputSPL2(Gauss).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss:
Solusi SPL:
x1 = 3.0
x2 = 0
x3 = a
x4 = -1.0
x5 = b
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi berupa parameter (solusinya tidak unik)
- File output: OutputSPL2(Gauss).txt

3. Test case : studi kasus no.1 bagian c

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

- Masukan dari file txt (InputSPL3.txt)
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 3
Masukkan banyak variabel: 6
Masukkan alamat file : C://Users//farad//Documents//InputSPL3.txt
isi nama file:
OutputSPL3(Gauss).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss:
Solusi SPL:
x1 = a
x2 = 1.0
x3 = b
x4 = -2.0
x5 = 1.0
x6 = c
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi berupa parameter (tidak unik)
- File output: OutputSPL3(Gauss).txt

4. Test case : studi kasus no. 1 bagian d

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

- Masukan dari keyboard
- Eksekusi program :

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan: 6
Masukkan banyak variabel: 6
isi matriks:
Masukkan persamaan ke-1 :
x1 = 1
x2 = 0.5
x3 = 0.33
```

```
x4 = 0.25
x5 = 0.2
x6 = 0.17
Hasil = 1
Masukkan persamaan ke-2 :
x1 = 0.5
x2 = 0.33
x3 = 0.25
x4 = 0.2
x5 = 0.17
x6 = 0.14
Hasil = 0
Masukkan persamaan ke-3 :
x1 = 0.33
x2 = 0.25
```

```
x3 = 0.25
x4 = 0.2
x5 = 0.17
x6 = 0.14
Hasil = 0
Masukkan persamaan ke-3 :
x1 = 0.33
x2 = 0.25
x3 = 0.2
x4 = 0.17
x5 = 0.14
x6 = 0.125
Hasil = 0
Masukkan persamaan ke-4 :
x1 = 0.25
x2 = 0.2
x3 = 0.17
x4 = 0.14
x5 = 0.125
x6 = 0.11
Hasil = 0
Masukkan persamaan ke-5 :
```

```
x1 = 0.2
x2 = 0.17
x3 = 0.14
x4 = 0.125
x5 = 0.11
x6 = 0.1
Hasil = 0
Masukkan persamaan ke-6 :
x1 = 0.17
x2 = 0.14
x3 = 0.125
x4 = 0.11
x5 = 0.1
x6 = 0.09
Hasil = 0
isi nama file:
OutputSPL4(Gauss).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss:
Solusi SPL:
x1 = 2.582590456213958
x2 = -0.8127949659151135
x3 = 2.123754588358736
x4 = -13.633980073413642
x5 = -16.70162558993192
x6 = 28.657577346617707
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi unik
- File output: OutputSPL4(Gauss).txt

5. Test case : studi kasus no. 2 bagian a

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

- Masukan dari file txt (InputSPL5.txt)
- Eksekusi program :

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 4
Masukkan alamat file : C://Users//farad//Documents//Smt3//InputSPL5.txt
isi nama file:
OutputSPL5(Gauss).java
Hasil perhitungan SPL dengan menggunakan metode Gauss:
Solusi SPL:
x1 = -1.0
x2 = 2.0a
x3 = a
x4 = b
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi tidak unik (berupa parameter)
- File output : OutputSPL5(Gauss).txt

6. Test case: studi kasus no.2 bagian b

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

- Masukan dari keyboard
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan: 6
Masukkan banyak variabel: 4
isi matriks:
Masukkan persamaan ke-1 :
x1 = 2
x2 = 0
x3 = 8
x4 = 0
Hasil = 8
Masukkan persamaan ke-2 :
x1 = 0
x2 = 1
x3 = 0
x4 = 4
Hasil = 6
```

```
Masukkan persamaan ke-3 :
x1 = -4
x2 = 0
x3 = 6
x4 = 0
Hasil = 6
Masukkan persamaan ke-4 :
x1 = 0
x2 = -2
x3 = 0
x4 = 3
Hasil = -1
Masukkan persamaan ke-5 :
x1 = 2
x2 = 0
x3 = -4
x4 = 0
Hasil = -4
Masukkan persamaan ke-6 :
x1 = 0
x2 = 1
x3 = 0
x4 = -2
Hasil = 0
isi nama file:
OutputSPL6(Gauss).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss:
Solusi SPL:
x1 = 0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi unik
- File output : OutputSPL6(Gauss).txt

7. Test case : studi kasus no 3 bagian a
    - Masukan dari file txt (InputSPL7.txt
    - Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 1
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 4
Masukkan alamat file : C://Users//farad//Documents//InputSPL7.txt
isi nama file:
OutputSPL7(Gauss).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss:
Solusi SPL:
x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi unik
- File output : OutputSPL7(Gauss).txt

4.2. Metode eliminasi Gauss-Jordan

1. Test case : studi kasus nomor 1 bagian a

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

- Masukan dari file txt (InputSPL1.txt)
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 4
Masukkan alamat file : C://Users//farad//Documents//InputSPL1.txt
isi nama file:
OutputSPL1(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
SPL tidak memiliki solusi
```

- Dari eksekusi program diperoleh bahwa SPL tersebut tidak memiliki solusi
- File output: OutputSPL1(GaussJordan).txt

2. Test case : studi kasus nomor 1 bagian b

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

- Masukan dari file txt (InputSPL2.txt)
- Eksekusi program

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 5
Masukkan alamat file : C://Users/farad/Documents/InputSPL2.txt
isi nama file:
OutputSPL2(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
Solusi SPL:
x1 = 3.0
x2 = 0
x3 = a
x4 = -1.0
x5 = b
```

- Dari eksekusi program diperoleh bahwa SPL tersebut solusinya tidak unik
- File output: OutputSPL2(GaussJordan).txt

3. Test case : studi kasus no.1 bagian c

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

- Masukan dari file txt (InputSPL3.txt)
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 3
Masukkan banyak variabel: 6
Masukkan alamat file : C://Users/farad/Documents/InputSPL3.txt
isi nama file:
OutputSPL3(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
Solusi SPL:
x1 = a
x2 = 1.0
x3 = b
x4 = -2.0
x5 = 1.0
x6 = c
```

- Dari eksekusi program diperoleh bahwa SPL tersebut memiliki solusi unik
- File output: OutputSPL3(GaussJordan).txt

4. Test case : studi kasus nomor 1 bagian d

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

- Masukan dari keyboard
- Eksekusi program (untuk n=6)

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan: 6
Masukkan banyak variabel: 6
isi matriks:
Masukkan persamaan ke-1 :
x1 = 1
x2 = 0.5
x3 = 0.33
x4 = 0.25
x5 = 0.2
x6 = 0.17
Hasil = 1
Masukkan persamaan ke-2 :
x1 = 0.5
x2 = 0.33
x3 = 0.25
x4 = 0.2
x5 = 0.17
```

```
x6 = 0.14
Hasil = 0
Masukkan persamaan ke-3 :
x1 = 0.33
x2 = 0.25
x3 = 0.2
x4 = 0.17
x5 = 0.14
x6 = 0.125
Hasil = 0
Masukkan persamaan ke-4 :
x1 = 0.25
x2 = 0.2
x3 = 0.17
x4 = 0.14
x5 = 0.125
x6 = 0.11
Hasil = 0
Masukkan persamaan ke-5 :
x1 = 0.2
x2 = 0.17
x3 = 0.14
x4 = 0.125
x5 = 0.11
x6 = 0.1
Hasil = 0
Masukkan persamaan ke-6 :
x1 = 0.17
x2 = 0.14
x3 = 0.125
x4 = 0.11
x5 = 0.1
x6 = 0.09
Hasil = 0
```

```
isi nama file:
OutoutSPL4(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
Solusi SPL:
x1 = 2.58
x2 = -0.81
x3 = 2.12
x4 = -13.63
x5 = -16.70
x6 = 28.66
```

- Dari eksekusi program diperoleh bahwa SPL tersebut memiliki solusi unik
- File output: OutputSPL4(GaussJordan).txt

5. Test case: studi kasus nomor 2 bagian a

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

- Masukan dari file (InputSPL5.txt)
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 4
Masukkan alamat file : C://Users//farad//Documents//Smt3//InputSPL5.txt
isi nama file:
OutputSPL5(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
Solusi SPL:
x1 = -1.0
x2 = 2.0a
x3 = a
x4 = b
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi unik
- File output: OutputSPL5(GaussJordan).txt

6. Test case : studi kasus no. 2 bagian b

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

- Masukan dari file txt : InputSPL6.txt
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 6
Masukkan banyak variabel: 4
Masukkan alamat file : C://Users//farad//Documents//InputSPL6.txt
isi nama file:
OutputSPL6(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
Solusi SPL
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi unik
- File output: OutputSPL6(GaussJordan).txt

7. Test case : studi kasus no. 3 bagian a

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

- Masukan dari file txt (InputSPL7.txt)
- Solusi yang diperoleh unik
- Eksekusi:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 2
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan: 4
Masukkan banyak variabel: 4
Masukkan alamat file : C://Users//farad//Documents//InputSPL7.txt
isi nama file:
OutputSPL7(GaussJordan).txt
Hasil perhitungan SPL dengan menggunakan metode Gauss-Jordan:
Solusi SPL:
x1 = -0.22
x2 = 0.18
x3 = 0.71
x4 = -0.26
```

- Dari eksekusi program diperoleh bahwa SPL memiliki solusi unik
- File output: OutputSPL7(GaussJordan).txt

4.3. Determinan

4.3.1. Metode Ekspansi Kofaktor

TEST CASE 1 :

Perhitungan Manual

$$M = \begin{matrix} -2 & 4 & 5 \\ 1 & 3 & -7 \\ -1 & 4 & -8 \end{matrix}$$

$$Det(M) = -2[(3 \times -8) - (-7 \times 4)] \times -4[(1 \times -8) - (-7 \times -1)] - 5[(1 \times 4) - (3 \times -1)]$$

$$Det(M) = 87$$

Perhitungan Program

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 2
1.Metode kofaktor
2. Metode OBE
1
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukan Jumlah Kolom : 3
Masukan Jumlah Baris : 3
-2 4 -5
1 3 -7
-1 4 -8
Determinan Matriks dengan kofaktor = 17.0

Process finished with exit code 0
```

4.3.2. Metode OBE

TEST CASE 1 :

Perhitungan Manual

$$M = \begin{array}{ccc} -2 & 4 & 5 \\ 1 & 3 & -7 \\ -1 & 4 & -8 \end{array} \qquad \sim R2 + ½ R1 ; R3 - ½ R1$$

$$M = \begin{matrix} -2 & 4 & 5 \\ 0 & 5 & -4.5 \\ 0 & 2 & -11.5 \end{matrix} \quad \sim R3 - 5/2\ R2$$

$$M = \begin{matrix} -2 & 4 & 5 \\ 0 & 5 & -9/2 \\ 0 & 0 & -87/10 \end{matrix}$$

$$Det(M) = -2 \times 5 \times -8.7$$
$$Det(M) = 87$$

Perhitungan Program

```
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 2
1.Metode kofaktor
2. Metode OBE
2
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukan Jumlah Kolom : 3
Masukan Jumlah Baris : 3
-2 4 -5
1 3 -7
-1 4  -8
Determinan Matriks dengan OBE = 17.0

Process finished with exit code 0
```

4.4. SPL Metode Matriks Balikan

1. Studi Kasus 1 (a)
   - Masukkan dari user (keyboard)
   - Solusi yang diperoleh tidak tunggal (unik) sehinggal tidak dapat menghasilkan nilai.
   - Eksekusi :

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 3
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan jumlah persamaan : 4
Masukkan Elemen Matriks A :
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2
Masukkan Elemen Matriks B :
1
-2
4
6
SPL tidak memiliki solusi tunggal (unik)
```

2. Studi Kasus 1 (d)
    - Masukkan dari user (keyboard)
    - Solusi yang diperoleh adalah solusi tunggal (unik) artinta matriks A memiliki matriks balikan
    - Eksekusi :

      Nilai n = 6

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 3
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan jumlah persamaan : 6
Masukkan Elemen Matriks A :
1 0.5 0.333 0.25 0.2 0.167
0.5 0.333 0.25 0.2 0.167 0.143
0.333 0.25 0.2 0.167 0.143 0.125
0.25 0.2 0.167 0.143 0.125 0.111
0.2 0.167 0.143 0.125 0.111 0.1
0.167 0.143 0.125 0.111 0.1 0.091
Masukkan Elemen Matriks B :
1
0
0
0
0
0
```

```
Hasil penyelesaian SPL dengan matriks balikan :
X1 = 8.080836
X2 = -23.558848
X3 = -31.735398
X4 = 108.017925
X5 = -43.699406
X6 = -17.952854
```

3. Studi Kasus 2 (a)
    - Masukkan dari user (keyboard)
    - Solusi yang diperoleh tidak tunggal (unik) sehinggal tidak dapat menghasilkan nilai.
    - Eksekusi :

- ```
  MENU
  1. Sistem Persamaan Linier
  2. Determinan
  3. Matriks balikan
  4. Interpolasi Polinom
  5. Regresi linier berganda
  6. Keluar
  Pilih menu : 1
  1. Metode eliminasi Gauss
  2. Metode eliminasi Gauss-Jordan
  3. Metode matriks balikan
  4. Kaidah Cramer
  Pilih metode : 3
  Pilih jenis input :
  1. Input Keyboard
  2. Input File
  Jenis input : 1
  Masukkan jumlah persamaan : 4
  Masukkan Elemen Matriks A :
  1 -1 2 -1
  2 1 -2 -2
  -1 2 -4 1
  3 0 0 -3
  Masukkan Elemen Matriks B :
  -1
  -2
  1
  -3
  SPL tidak memiliki solusi tunggal (unik)
  ```

4. Studi Kasus 3 (a)
    - Masukkan dari user (keyboard)
    - Solusi yang diperoleh adalah solusi tunggal (unik) artinta matriks A memiliki matriks balikan
    - Eksekusi :
  ```

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 3
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan jumlah persamaan : 4
Masukkan Elemen Matriks A :
8 1 3 2
2 9 -1 -2
1 3 2 -1
1 0 6 4
Masukkan Elemen Matriks B :
0
1
2
3
Hasil penyelesaian SPL dengan matriks balikan :
X1 = -0.224324
X2 = 0.182432
X3 = 0.709459
X4 = -0.258108
```

5. Studi Kasus 1(b),1(c), 2(b), dan 3(b) : Tidak dapat dieksekusi di dalam program SPL Metode Matriks Balikan karena jumlah baris dan kolom pada matriks A tidak sama sehingga tidak dapat dicari matriks balikannya.

4.5. Matriks Balikan

1. Test Case 1

   Diketahui suatu matriks A

$$A = \begin{bmatrix} 1 & 5 & 5 \\ -1 & -1 & 0 \\ 2 & 4 & 3 \end{bmatrix}$$

   Perhitungan manual :

Untuk mencari inversenya, pertama dicari dulu matriks kofaktor A

$$\begin{bmatrix} -3 & 3 & -2 \\ 5 & -7 & 6 \\ 5 & -5 & 4 \end{bmatrix}$$

Kemudian, dicari determinan dan matriks adjoin A. matriks adjoin A adalah transpose dari matriks kofaktor A.

$$det(A) = 2$$

$$A^{-1} = \frac{1}{det(A)} adj(A) = \frac{1}{2} \begin{bmatrix} -3 & 5 & 5 \\ 3 & -7 & -5 \\ -2 & 6 & 4 \end{bmatrix} = \begin{bmatrix} -\frac{3}{2} & \frac{5}{2} & \frac{5}{2} \\ \frac{3}{2} & -\frac{7}{2} & -\frac{5}{2} \\ -1 & 3 & 2 \end{bmatrix}$$

Perhitungan program :

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 3
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan jumlah baris dan kolom : 3
Masukkan Elemen Matriks :
1 5 5
-1 -1 0
2 4 3
Matriks Balikan :
-1.5 2.5 2.5
1.5 -3.5 -2.5
-1.0 3.0 2.0
```

2. Test Case 2

Diketahui suatu matriks D

$$D = \begin{bmatrix} 4 & -10 \\ -2 & 5 \end{bmatrix}.$$

Perhitungan manual :

Dicari determinan dari matriks D,

$$\det(D) = \begin{vmatrix} 4 & -10 \\ -2 & 5 \end{vmatrix} = (4)(5) - (-2)(-10) = 0.$$

Karena determinannya 0, maka matriks tersebut tidak memiliki matriks balikan.

Perhitungan Program :

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 3
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan jumlah baris dan kolom : 2
Masukkan Elemen Matriks :
4 -10
-2 5
Matriks Balikan Tidak Ada
```

4.6. Kaidah Cramer

1. Test case : studi kasus nomor 1 bagian a

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

- Masukan berupa input dari keyboard.
- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 4
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan dan variabel: 4
isi matriks:
Masukkan persamaan ke-1 :
x1 = 1
x2 = 1
x3 = -1
x4 = -1
Hasil = 1
Masukkan persamaan ke-2 :
x1 = 2
x2 = 5
x3 = -7
x4 = -5
Hasil = -2
Masukkan persamaan ke-3 :
```

```
x1 = 2
x2 = -1
x3 = 1
x4 = 3
Hasil = 4
Masukkan persamaan ke-4 :
x1 = 5
x2 = 2
x3 = -4
x4 = 2
Hasil = 6
isi nama file:
OutputSPL1(Cramer).txt
Hasil perhitungan SPL dengan menggunakan metode Crammer:
SPL tidak memiliki solusi tunggal (unik)
```

- Solusi yang diperoleh tidak unik
- File output: OutputSPL1(Cramer).txt

2. Test case : studi kasus nomor 1 bagian d

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

- Masukan dari keyboard
- Eksekusi program (untuk n=6)

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 4
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 1
Masukkan banyak persamaan dan variabel: 6
isi matriks:
Masukkan persamaan ke-1 :
x1 = 1
x2 = 0.5
x3 = 0.33
x4 = 0.25
x5 = 0.2
x6 = 0.17
Hasil = 1
Masukkan persamaan ke-2 :
x1 = 0.5
x2 = 0.33
x3 = 0.25
x4 = 0.2
x5 = 0.17
x6 = 0.14
```

```
Hasil = 0
Masukkan persamaan ke-3 :
x1 = 0.33
x2 = 0.25
x3 = 0.2
x4 = 0.17
x5 = 0.14
x6 = 0.125
Hasil = 0
Masukkan persamaan ke-4 :
x1 = 0.25
x2 = 0.2
x3 = 0.17
x4 = 0.14
x5 = 0.125
x6 = 0.11
Hasil = 0
Masukkan persamaan ke-5 :
```

```
x1 = 0.2
x2 = 0.17
x3 = 0.14
x4 = 0.125
x5 = 0.11
x6 = 0.1
Hasil = 0
Masukkan persamaan ke-6 :
x1 = 0.17
x2 = 0.14
x3 = 0.125
x4 = 0.11
x5 = 0.1
x6 = 0.09
Hasil = 0
isi nama file:
OutputSPL4(Cramer).txt
Hasil perhitungan SPL dengan menggunakan metode Crammer:
Solusi SPL:
x1 = 2.58
x2 = -0.81
x3 = 2.12
x4 = -13.63
x5 = -16.70
x6 = 28.66
```

- Dari eksekusi program diperoleh solusi yang diperoleh unik
- File output: OutputSPL4(Cramer).txt

3. Test case : dari studi kasus nomor 2 bagian a

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

- Masukan berupa file txt (InputSPL5.txt)

- Eksekusi program:

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 4
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan dan variabel: 4
Masukkan alamat file : C://Users//farad//Documents//Smt3//InputSPL5.txt
isi nama file:
OutputSPL5(Cramer).txt
Hasil perhitungan SPL dengan menggunakan metode Crammer:
SPL tidak memiliki solusi tunggal (unik)
```

- Dari eksekusi program diperoleh bahwa SPL tersebut tidak memiliki solusi yang unik.
- File output : OutputSPL5(Cramer).txt

4. Test case : dari studi kasus nomor 3 bagian a

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

- Masukan dari file txt (InputSPL7.txt)
- Eksekusi program

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
Pilih metode : 4
Pilih metode masukan:
1. Masukan dari keyboard
2. Masukan dari file txt
-------------------------
Metode masukan: 2
Masukkan banyak persamaan dan variabel: 4
Masukkan alamat file : C://Users//farad//Documents//InputSPL7.txt
isi nama file:
OutputSPL7(Crammer).txt
Hasil perhitungan SPL dengan menggunakan metode Crammer:
Solusi SPL:
x1 = -0.22
x2 = 0.18
x3 = 0.71
x4 = -0.26
```

- Dari eksekusi program diperoleh bahwa SPL tersebut memiliki solusi yang unik.
- File output : OutputSPL7(Cramer).txt

4.7. Interpolasi Polinom

1. Studi Kasus 5
    - Input dari user (keyboard)
    - Setelah koordinat yang diketahui pada soal di-input-kan ke program Interpolas Polinom, akan diminta nilai X yang ingin dicari. Setelah itu didapatkan “hampiran” nilai hasil dari X yang kita masukkan
    - Eksekusi :

- ```
  MENU
  1. Sistem Persamaan Linier
  2. Determinan
  3. Matriks balikan
  4. Interpolasi Polinom
  5. Regresi linier berganda
  6. Keluar
  Pilih menu : 4
  Pilih jenis input :
  1. Input Keyboard
  2. Input File
  Jenis input : 1
  Masukkan banyak titik : 7
  Masukkan titik :
  0.1 0.003
  0.3 0.067
  0.5 0.148
  0.7 0.248
  0.9 0.370
  1.1 0.518
  1.3 0.697
  Nilai X yang ingin di cari : 0.2
  Solusi atau nilai Y yang dihasilkan adalah 0.032961
  ```

  ```
  Nilai X yang ingin di cari : 0.55
  Solusi atau nilai Y yang dihasilkan adalah 0.171119
  ```

  ```
  Nilai X yang ingin di cari : 0.85
  Solusi atau nilai Y yang dihasilkan adalah 0.337236
  ```

  ```
  Nilai X yang ingin di cari : 1.28
  Solusi atau nilai Y yang dihasilkan adalah 0.677542
  ```

2. Studi Kasus 6
   - Input dari user (keyboard)
   - Setelah koordinat yang diketahui pada soal di-input-kan ke program Interpolas Polinom, akan diminta nilai X yang ingin dicari. Setelah itu didapatkan “hampiran” nilai hasil dari X yang kita masukkan.
   - Eksekusi :

     Untuk tanggal 25/05/20 diperoleh tanggal (desimal) : 5,833

     Untuk tanggal 30/08/20 diperoleh tanggal (desimal) : 9

     Untuk tanggal 15/09/20 : diperoleh tanggal (desimal) : 9.5
  ```

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 4
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan banyak titik : 10
Masukkan titik :
4.8 8211
5 10118
5.516 17025
5.710 20796
6.5 39294
7.194 64958
8.097 113134
8.258 123503
9.033 177571
9.333 145510
Nilai X yang ingin di cari : 5.833
Solusi atau nilai Y yang dihasilkan adalah 23368.095723
```

```
Nilai X yang ingin di cari : 9
Solusi atau nilai Y yang dihasilkan adalah 176872.858099
```

```
Nilai X yang ingin di cari : 9.5
Solusi atau nilai Y yang dihasilkan adalah 68216.427094
```

3. Studi Kasus 7

Titik - titik yang diketahui berdasarkan selang dan fungsi yang diberikan :

x = 0	f(x) = 0
x = 0,4	f(x) = 0,419
x = 0,8	f(x) = 0,507
x = 1,2	f(x) = 0,56
x = 1,6	f(x) = 0,584
x = 2	f(x) = 0,577

Kemudian nilai tersebut dimasukkan ke dalam program dan didapatkan persamaan polinom dari interpolasi yang dilakukan.

```
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 4
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukkan banyak titik : 6
Masukkan titik :
0 0
0.4 0.419
0.8 0.507
1.2 0.56
1.6 0.584
2 0.577
Persamaan interpolasi polinomnya :
p(x) = 0.000000 + (2.030167) * x^1 + (-3.524479) * x^2 + (3.188802) * x^3 + (-1.389974) * x^4 + (0.229492) * x^5
```

4.8. Regresi Linier Berganda

TEST CASE 1 :

Perhitungan Manual

(i). $12Y_1 = 1\ X_1 + 10\ X_2$

(ii). $18Y_2 = 2\ X_1 + 1\ X_2$

(iii). $24Y_3 = 3\ X_1 + 2\ X_2$

(iv). $30Y_4 = 4\ X_1 + 3\ X_2$

X_1	1	2	3	4
X_2	10	1	2	3
Y	12	18	24	30

Dimasukan ke rumus *Normal Estimation Equation for Multiple Linier Regression* didapatkan:

$$4\beta_0 + 10\beta_1 + 16\beta_2 = 84$$
$$10\beta_0 + 30\beta_1 + 30\beta_2 = 240$$
$$16\beta_0 + 30\beta_1 + 114\beta_2 = 276$$

Lalu menggunakan fungsi Sistem Persamaan Linier Gauss akan didapatkan nilai β

$\beta_0 = 6$, $\beta_1 = 6$, $\beta_2 = 0$

Dan Dimasukan ke persamaan umum regresi linier berganda maka akan didapatkan

$Y = 6 + 6 X_1 + 0 X_2$

Perhitungan Program

```
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
MENU
1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
Pilih menu : 5
Pilih jenis input :
1. Input Keyboard
2. Input File
Jenis input : 1
Masukan Jumlah persamaan (baris) : 4
Masukan Jumlah x (Kolom+1) : 2
Masukan X11 : 1
Masukan X12 : 10
Masukan Y1 : 12
Masukan X21 : 2
Masukan X22 : 1
Masukan Y2 : 18
Masukan X31 : 3
Masukan X32 : 2
Masukan Y3 : 24
Masukan X41 : 4
Masukan X42 : 3
Masukan Y4 : 30
y = 6.00  + 6.00  X1 + 0.00  X2
Process finished with exit code 0
```

TEST CASE 2 (studi kasus no. 8) :

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.

Dimasukan ke rumus *Normal Estimation Equation for Multiple Linier Regression* didapatkan:

$$20\,\beta_0 + 863.1\,\beta_1 + 1530.4\,\beta_2 + 587.84\,\beta_3 = 19.42$$

$$863.1\,\beta_0 + 54876.89\,\beta_1 + 67000.09\,\beta_2 + 25283.395\,\beta_3 = 779.477$$

$$1530.4\,\beta_0 + 67000.09\,\beta_1 + 117912.32\,\beta_2 + 44976.867\,\beta_3 = 1483.437$$

$$587.84\,\beta_0 + 25283.395\,\beta_1 + 44976.867\,\beta_2 + 17278.5086\,\beta_3 = 571.1219$$

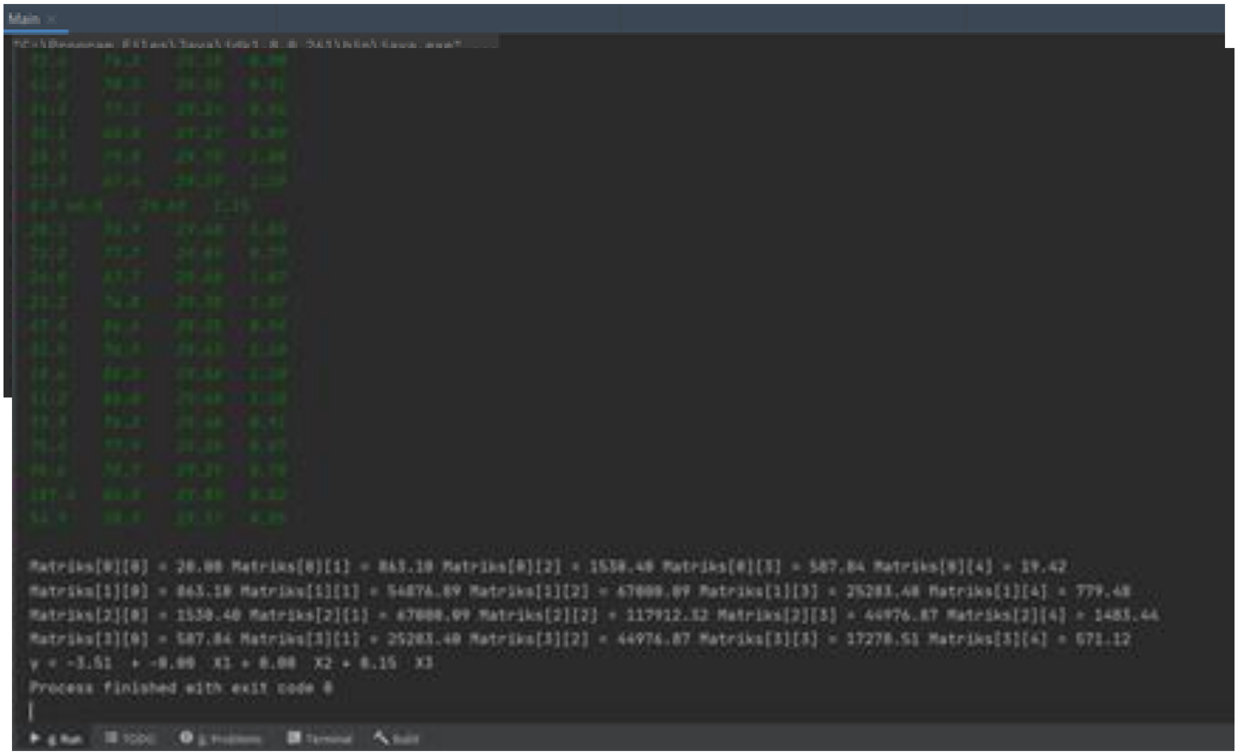
Lalu menggunakan fungsi Sistem Persamaan Linier Gauss akan didapatkan nilai β

$\beta_0 = -3.51$, $\beta_1 = 0.00$, $\beta_2 = 0.00$,$\beta_3 = 0.15$

Dan Dimasukan ke persamaan umum regresi linier berganda maka akan didapatkan

$$Y = -3.51 + 0\,X_1 + 0\,X_2 + 0.15\,X_3$$

Perhitungan Program



Main
Matriks[0][0] = 20.00 Matriks[0][1] = 863.10 Matriks[0][2] = 1530.40 Matriks[0][3] = 587.84 Matriks[0][4] = 19.42
Matriks[1][0] = 863.10 Matriks[1][1] = 54876.89 Matriks[1][2] = 67000.09 Matriks[1][3] = 25283.40 Matriks[1][4] = 779.40
Matriks[2][0] = 1530.40 Matriks[2][1] = 67000.09 Matriks[2][2] = 117912.32 Matriks[2][3] = 44976.87 Matriks[2][4] = 1483.44
Matriks[3][0] = 587.84 Matriks[3][1] = 25283.40 Matriks[3][2] = 44976.87 Matriks[3][3] = 17278.51 Matriks[3][4] = 571.12
y = -3.51 + -0.00 X1 + 0.00 X2 + 0.15 X3
Process finished with exit code 0

BAB 5

Kesimpulan, Saran, dan Refleksi

Kesimpulan :

1. Program berjalan sesuai ekspektasi tanpa error fatal
2. Semua spesifikasi program terpenuhi

Saran :

1. Dibuat GUI agar program lebih user friendly
2. Minimalisir variabel - variabel yang di hardcoded seperti directory untuk tulis dan isi file dan dibuat menjaid input dari user
3. Perbanyak test case agar dapat lebih terverifikasi program berjalan sesuai spesifikasi
4. Untuk masukan dari file tidak perlu lagi memasukka jumlah baris dan kolom matriks input agar lebih mudah

Refleksi diri :

1. Lebih sering tekun membaca spesifikasi program agar tidak ada spek yang salah.
2. Saat mengedit program, jangan lupa pull terlebih dahulu agar tidak conflict
3. Perbanyak diskusi antar anggota
4. Permatang perencanaan pengerjaan tugas besar
5. Ada 2 method solusiGaussJordan di dalam class MatSPL.java. hal ini karena awalnya sudah dibuat method tersendiri untuk membuat solusi Gauss-Jordan yang berbeda dari solusi Gauss (method solusiSPLGaussJordan(double[][] Mat, int baris, int kolom) namun di solusi berupa parameter terjadi kesalahan output. Oleh sebab itu diganti methodnya menjadi sama seperti method untuk mencari solusi Gauss yaitu method solusiSPLGaussJordan1(double[][] Mat, int baris, int kolom) untuk menjalankan program main. Jadi sebaiknya dalam membuat program lebih teliti dan siap untuk kondisi terburuk.

Referensi

Anton, Howard, Chris Rosses, Elementary Linear Algebra 11th edition, Wiley
https://www.geeksforgeeks.org/java-tutorial/?ref=lbp
https://www.investopedia.com/terms/m/mlr.asp
https://www.statistikian.com/2018/01/penjelasan-tutorial-regresi-linear-berganda.html#:~:text=Regresi%20Linear%20Berganda%20adalah%20model,disebut%20dengan%20multiple%20linear%20regression.&text=Skala%20data%20yang%20dimaksud%20diatas%20adalah%20pada%20semua%20variabel%20terutama%20variable%20terikat.
https://mathworld.wolfram.com/GaussianElimination.html
https://qph.fs.quoracdn.net/main-qimg-cbac2ebd54114d88cfff996a90e202a5