TUGAS BESAR 1 IF2123
ALJABAR LINIER DAN GEOMETRI
2022/2023

Oleh

Ulung Adi Putra	13521122
Satria Octavianus Nababan	13521168
Nathan Tenka	13521172

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

BAB I
DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}b$), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

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

SPESIFIKASI TUGAS

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

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

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

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

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

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

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x_0 , y_0), (x_1 , y_1), ..., (x_n , y_n), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

```
8.0 2.0794
9.0 2.1972
```

9.5 2.2513

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x1i , x2i , ..., xni , nilai yi , dan nilai-nilai xk yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran (output) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya x4 = -2, x3 = 2s – t, x2 = s, dan x1 = t.)

6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing

-3 7 8.3 11 -4
3 4.5 2.8 10
12 0.5 -10 -9
12 0 0.1 0.2

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah $f(x)$= -0.0064x + 0.2266 + 0.6762 , 2 x f(5) =… dan untuk regresi adalah f(x) = -9.5872 + 1.0732x 1 , $f(xk$)= …

8. Untuk persoalan interpolasi bicubic, masukan dari file text (.txt) yang berisi matriks berukuran 4x4 yang berisi nilai f(i,j) dengan i dan j adalah indeks matriks diikuti dengan nilai a dan b untuk mencari nilai f(a,b). Misalnya jika nilai dari f(-1,-1), f(-1,0), f(-1,1), f(-1,2),f(0,-1), f(0,0), f(0,1), f(0,2), f(1,-1), f(1,0), f(1,1), f(1,2), f(2,-1), f(2,0), f(2,1), f(2,2) berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9 ,10, 11, 12, 13, 14, 15 ,16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi file text ditulis sebagai berikut:

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

luaran yang dihasilkan adalah nilai dari f(0.5,0.5).
masukannya adalah matriks 4 x 4, diikuti oleh nilai a dan b, maka luarannya adalah nilai f(a,b).

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

10. Bahasa program yang digunakan adalah Java.

11. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).

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

MENU
1. Sistem Persamaaan Linier
2. Determinan

3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic
6. Regresi linier berganda
7. Keluar

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

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

Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB II
LANDASAN TEORI

2.1 Eliminasi Gauss

Pada eliminasi Gauss, SPL direpresentasikan dalam bentuk matriks augmented dengan tiap baris berisi koefisien masing-masing persamaan dan kolom terakhir merupakan hasil dari persamaan. Matriks tersebut akan diubah menjadi matriks eselon baris, yaitu matriks yang setiap barisnya memiliki satu utama. Ciri-ciri matriks eselon baris adalah sebagai berikut :

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

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

Gambar 2.1.1 : Contoh matriks eselon baris

Reduksi menjadi matriks eselon baris dilakukan dengan 3 operas baris elementer (OBE), yaitu :

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

Setelah reduksi, dilakukan *backwards substitution*, yaitu substitusi persamaan dari baris terakhir matriks sampai ke persamaan di baris pertama. Contoh penyelesaian SPL dengan eliminasi Gauss adalah sebagai berikut :

Contoh 1: Selesaikan SPL berikut dengan eliminasi Gauss

$$2x_1 + 3x_2 - x_3 = 5$$
$$4x_1 + 4x_2 - 3x_3 = 3$$
$$-2x_1 + 3x_2 - x_3 = 1$$

Penyelesaian:

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \overset{R1/2}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \underset{R3+2R1}{\overset{R2-4R1}{\sim}} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{bmatrix}$$

$$\overset{R2/(-2)}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 6 & -2 & 6 \end{bmatrix} \overset{R3-6R2}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & -5 & -15 \end{bmatrix} \overset{R3/(-5)}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Keterangan: R1 = baris ke-1, Rn = baris ke-n

Matriks eselon baris

Dari matriks *augmented* terakhir:

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

diperoleh persamaan-persamaan linier sbb:

$x_1 + 3/2x_2 - 1/2x_3 = 5/2$ (i)

$x_2 + 1/2x_3 = 7/2$ (ii)

$x_3 = 3$ (iii)

Selesaikan dengan teknik penyulihan mundur sbb:

(iii) $x_3 = 3$

(ii) $x_2 + 1/2x_3 = 7/2 \rightarrow x_2 = 7/2 - 1/2(3) = 2$

(i) $x_1 + 3/2x_2 - 1/2x_3 = 5/2 \rightarrow x_1 = 5/2 - 3/2\,(2) - 1/2\,(3) = 1$

Solusi: $x_1 = 1, x_2 = 2, x_3 = 3$

Gambar 2.1.2 dan 2.1.3 : Contoh penyelesaian SPL dengan eliminasi Gauss

Seperti yang telah disebutkan pada deskripsi masalah, SPL memiliki 3 kemungkinan solusi, yaitu tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal). Pada metode eliminasi Gauss, ciri-ciri dari ketiga solusi tersebut adalah :

1. Tidak ada solusi : terjadi saat ada baris yang seluruh koefisien variabelnya nol tapi hasilnya tidak 0.

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 7 \end{array}\right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Gambar 2.1.4 : Contoh SPL yang tidak memiliki solusi

2. Banyak solusi : terjadi saat ada baris yang hasilnya seluruhnya berisi 0.

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 6 \end{array}\right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Gambar 2.1.5 : Contoh SPL yang memiliki banyak solusi

3. Solusi unik : terjadi saat tidak ada baris yang seluruhnya maupun koefisien variabelnya saja yang bernilai 0.

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 2 & 1 \end{array}\right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

Solusi: $x_1 = 1, x_2 = 0, x_3 = -1$

Gambar 2.1.6 : Contoh SPL yang memiliki solusi unik

2.2 Eliminasi Gauss Jordan

Eliminasi Gauss-Jordan mirip dengan eliminasi Gauss, tetapi matriks augmented diubah menjadi matriks eselon baris tereduksi. Matriks tersebut memiliki sifat yang sama dengan matriks eselon baris dengan tambahan kolom yang berisi 1 utama seluruhnya berisi 0 kecuali satu utamanya sendiri.

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

Gambar 2.2.1 : Contoh mariks eselon baris tereduksi

Pada eliminasi Gauss-Jordan terdapat 2 tahap eliminasi, yaitu fase maju (fase eliminasi Gauss, membuat matriks eselon baris) dan fase mundur (membuat matriks eselon baris tereduksi).

1. Fase maju (*forward phase*) atau fase eliminasi Gauss

   - Menghasilkan nilai-nilai 0 di bawah 1 utama

$$\begin{bmatrix}2&3&-1&5\\4&4&-3&3\\-2&3&-1&1\end{bmatrix} \sim \overset{\text{OBE}}{\dots} \sim \begin{bmatrix}1&3/2&-1/2&5/2\\0&1&1/2&7/2\\0&0&1&3\end{bmatrix}$$

2. Fase mundur (*backward phase*)

   - Menghasilkan nilai-nilai 0 di atas satu utama

$$\begin{bmatrix}1&3/2&-1/2&5/2\\0&1&1/2&7/2\\0&0&1&3\end{bmatrix} \overset{\text{R1 −(3/2)R2}}{\sim} \begin{bmatrix}1&0&-5/4&-11/4\\0&1&1/2&7/2\\0&0&1&3\end{bmatrix} \underset{\text{R2 - (1/2)R3}}{\overset{\text{R1 +(5/4)R3}}{\sim}} \begin{bmatrix}1&0&0&1\\0&1&0&2\\0&0&1&3\end{bmatrix}$$

Matriks eselon baris tereduksi

Gambar 2.2.2 : Fase maju dan mundur pada eliminasi Gauss-Jordan

Karena sudah berupa matriks eselon baris tereduksi, pada eliminasi Gauss-Jordan tidak perlu dilakukan *backwards substitution.* Hasil bisa langsung didapat dari nilai di kolom terakhir matriks augmented untuk tiap variabel. Ciri-ciri kemungkinan solusi Gauss-Jordan sama dengan metode eliminasi Gauss.

2.3 Determinan

Determinan adalah sebuah angka khusus yang bisa dicari pada matriks. Secara geometris, determinan adalah angka yang menunjukkan dengan faktor berapa kali ukuran sebuah daerah diubah oleh sebuah matriks (Jika determinan negatif, berarti orientasi ruangnya dibalik). Determinan bisa digunakan untuk berbagai hal, misalnya mencari matriks balikan (*inverse*) dan menyelesaikan SPL. Determinan hanya bisa dicari pada matriks persegi (matriks yang berukuran n x n). Untuk matriks berukuran 2x2 dan 3x3, determinan bisa dicari dengan cara berikut :

Untuk matriks A berukuran 2 x 2:

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

maka $\det(A) = a_{11}a_{22} - a_{12}a_{21}$

Gambar 2.3.1 : Perhitungan determinan matriks 2x2

Untuk matriks A berukuran 3 x3:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \left[\begin{array}{ccc|cc} a_{11} & a_{12} & a_{13} & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} & a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} & a_{31} & a_{32} \end{array}\right]$$

maka $\det(A) = (a_{11}a_{22}a_{33} + a_{12}a_{21}a_{31} + a_{13}a_{21}a_{32}) - (a_{13}a_{22}a_{31} + a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33})$

Gambar 2.3.2 : Perhitungan determinan matriks 3x3

Determinan matriks persegi dengan dimensi yang lebih besar bisa dilakukan dengan 2 cara, yaitu reduksi baris dan metode ekspansi kofaktor. Pada metode reduksi baris, matriks diubah menjadi matriks segitiga atas/bawah memakai OBE sehingga determinan bisa ditemukan dari hasil perkalian seluruh elemen diagonal matriks. Rumusnya adalah sebagai berikut :

$$\det(A) = \frac{(-1)^p \, a'_{11} \; a'_{22} \ldots a'_{mm}}{k_1 k_2 \ldots k_m}$$

Gambar 2.3.3 : Perhitungan determinan matriks dengan metode reduksi baris

Dengan p adalah banyak pertukaran baris dan k_i adalah konstanta tidak nol yang dipakai untuk mengalikan baris pada matriks.

Metode ekspansi kofaktor menggunakan jumlah dari hasil perkalian setiap elemen salah satu baris/kolom dengan kofaktornya (kofaktor dijelaskan lebih lanjut di subbab 2.5). Perhitungannya seperti berikut :

- Dengan menggunakan kofaktor, maka determinan matriks

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

dapat dihitung dengan <u>salah satu</u> dari persamaan berikut:

Secara baris	Secara kolom
$\det(A) = a_{11}C_{11} + a_{12}C_{12} + \ldots + a_{1n}C_{1n}$	$\det(A) = a_{11}C_{11} + a_{21}C_{21} + \ldots + a_{n1}C_{n1}$
$\det(A) = a_{21}C_{21} + a_{22}C_{22} + \ldots + a_{2n}C_{2n}$	$\det(A) = a_{12}C_{12} + a_{22}C_{22} + \ldots + a_{n2}C_{n2}$
$\vdots$	$\vdots$
$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \ldots + a_{nn}C_{nn}$	$\det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \ldots + a_{nn}C_{nn}$

Gambar 2.3.4 : Perhitungan determinan matriks dengan metode ekspansi kofaktor

2.4 Matriks Balikan

Matriks balikan atau invers matriks adalah suatu matriks yang jika dikalikan dengan matriks asalnya akan menghasilkan matriks identitas sama seperti pada aljabar biasa di mana bilangan X memiliki invers $X^{-1} = \frac{1}{X}$ sehingga $XX^{-1} = 1$. Dalam laporan ini dibahas dua cara menemukan matriks invers yaitu dengan rumus $M^{-1} = \frac{1}{determinan(M)}\ adjoin(M)$ dengan adjoin merupakan transpose dari matriks Kofaktor M atau dengan eliminasi Gauss-Jordan. Dapat dilihat dari rumusnya yang memakai pembagian dengan determinan maka matriks hanya memiliki matriks balikan jika determinan dari matriks tersebut tidak sama dengan nol dan merupakan matriks persegi. Metode eliminasi Gauss-Jordan digunakan untuk menemukan matriks balikan dengan mengubah matriks tersebut menjadi matriks identitas dan melakukan perubahan yang sama terhadap matriks identitas yang berdimensi sama sehingga matriks identitas tersebut berubah menjadi matriks balikan.

Matriks balikan ini dapat digunakan untuk menyelesaikan sistem persamaan linier. Namun, metode matriks balikan ini hanya dapat digunakan jika SPL dapat diubah menjadi matriks persegi dan memiliki determinan yang tidak 0. Misalkan dimiliki SPL yang diubah menjadi bentuk matriks $AX = B$. jika kedua ruas dikalikan dengan matriks balikan A didapat $A^{-1}AX = A^{-1}B$ yang menjadi $X = A^{-1}B$ karena matriks A dikali dengan matriks balikannya akan menjadi matriks identitas yang jika dikali dengan matriks X hasilnya akan sama dengan matriks X. Dari sini didapat solusi SPL dengan menggunakan matriks balikan.

2.5 Matriks Kofaktor

Matriks kofaktor adalah suatu matriks yang terdiri dari kofaktor matriks. Nilai kofaktor sebuah elemen adalah nilai determinan dari elemen-elemen matriks yang tidak sebaris dan tidak sekolom dengan elemen tersebut. Nilai kofaktor ini digabungkan dengan $+$ atau $-$ tergantung letak elemennya pada matriks yang mengikuti gambar berikut :

$$\begin{vmatrix} + & - & + & \cdots \\ - & + & - & \cdots \\ + & - & + & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{vmatrix}$$

Gambar 2.5.1 : Posisi tanda +/- untuk kofaktor

Matriks kofaktor ini dipakai dalam perhitungan determinan matriks melalui ekspansi Laplace atau ekspansi kofaktor. Ekspansi kofaktor ini adalah metode perhitungan determinan matriks dengan memilih suatu baris atau kolom dalam matriks, mengalikan tiap elemen dalam baris atau kolom tersebut dengan determinan kofaktornya, dan menjumlahkan setiap nilai yang didapat dari perkalian elemen dengan determinan kofaktornya.

2.6 Matriks Adjoin

Matriks adjoin adalah transpose dari sebuah matriks kofaktor. Adjoin M dapat dinotasikan dengan Adj (M). Transpose merupakan operasi pertukaran elemen pada baris menjadi kolom dan sebaliknya. Adjoin matriks digunakan dalam menentukan invers matriks. Berikut merupakan matriks kofaktor M yang diubah menjadi Adj (M).

$$\begin{bmatrix} -12 & 6 & -8 \\ -4 & 2 & -8 \\ 12 & -10 & 8 \end{bmatrix} \begin{bmatrix} -12 & -4 & 12 \\ 6 & 2 & -10 \\ -8 & -8 & 8 \end{bmatrix}$$

Gambar 2.6.1 : Perubahan matriks kofaktor menjadi adjoin

2.7 Kaidah Cramer

Kaidah Cramer adalah rumus yang dapat digunakan untuk menyelesaikan sistem persamaan linier. Metode ini menggunakan determinan suatu matriks dan determinan matriks lain yang diperoleh dengan mengganti salah satu kolom dengan vektor yang terdiri dari hasil tiap persamaan. Jika Ax = b adalah sebuah sistem linier *n* yang tidak diketahui dan det(A) tidak bernilai 0 maka persamaan tersebut mempunyai penyelesaian yang unik

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

Gambar 2.7.1 : Rumus Kaidah Cramer

2.8 Interpolasi Polinom

Interpolasi polinomial digunakan untuk mencari titik-titik antara dari *n* buah titik $P_1(X1, Y1)$, $P_2(X_2, Y_2)$, $P_3(X_3, Y_3)$, ..., $P_n(X_n, Y_n)$ dengan menggunakan pendekatan fungsi polinomial berpangkat n - 1:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

Gambar 2.8.1 : Persamaan umum interpolasi polinom

Dengan memasukkan nilai dari setiap titik ke dalam persamaan polinomial di atas, akan diperoleh persamaan simultan dengan *n* persamaan dan *n* variabel bebas:

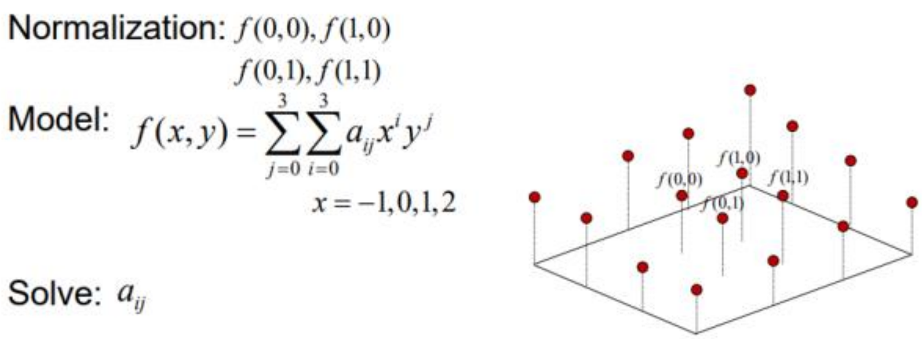
$$\begin{aligned}
y_1 &= a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 + \ldots + a_{n-1} x_1^{n-1} \\
y_2 &= a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 + \ldots + a_{n-1} x_2^{n-1} \\
y_3 &= a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 + \ldots + a_{n-1} x_3^{n-1} \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
y_n &= a_0 + a_1 x_n + a_2 x_n^2 + a_3 x_n^3 + \ldots + a_{n-1} x_n^{n-1}
\end{aligned}$$

Gambar 2.8.2 : SPL persamaan polinom

Persamaan simultan tersebut bisa diselesaikan dengan eliminasi Gauss.

2.9 Interpolasi Bicubic

Interpolasi bicubic merupakan pengembangan lebih lanjut dari interpolasi cubic. Interpolasi bicubic digunakan untuk menginterpolasi data 2 dimensi dan bisa digunakan untuk manipulasi citra (misal untuk perbesaran gambar). Persamaan interpolasi bicubic dimodelkan sebagai berikut :



Gambar 2.9.1 : Model interpolasi bicubic

Dengan mensubstitusi nilai f(x,y) yang diketahui akan didapat persamaan sebagai berikut

$$y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Gambar 2.9.2 : Persamaan interpolasi bicubic

Nilai vektor **a** bisa didapat dengan mengalikan invers matriks X dengan vektor nilai **y**. Dalam kasus dengan titik-titik seperti di atas, nilai yang bisa diinterpolasi hanyalah pada rentang ([0..1],[0..1]) karena interpolasi bicubic memerlukan 16 nilai yang ada di sekitar titik yang ingin diinterpolasi.

2.10 Regresi Linier Berganda

Regresi linier berganda adalah cara lain untuk memprediksi nilai selain interpolasi polinom. Regresi linier memiliki rumus umum :

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

Gambar 2.10.1 : Rumus umum regresi linier

Nilai semua βi bisa didapatkan dengan menyelesaikan SPL yang didapat dari *Normal Estimation Equation for Multiple Linear Regression* seperti berikut :

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

Gambar 2.7.1 : Rumus *Normal Estimation Equation for Multiple Linear Regression*

Selanjutnya SPL tersebut bisa diselesaikan dengan eliminasi Gauss untuk mendapat nilai βi.

BAB III
IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

3.1 Implementasi Matriks (Matrix.java)

Struktur matriks diimplementasikan dengan c*lass* Matrix, yang memiliki atribut dan metode sebagai berikut :

- Atribut :
  - public ArrayList<ArrayList<Double>> mtrx : bagian matriks dari ADT Matrix, diimplementasikan memakai ArrayList.
  - public int row, public int col : jumlah baris dan kolom matriks.
- Metode :
  - Getter :
    1. public int getRow()
       Mengembalikan jumlah baris Matrix.
    2. public int getCol()
       Mengembalikan jumlah kolom Matrix.
    3. public double getElmt(int i, int j)
       Mengembalikan elemen di indeks (i,j).
  - Setter :
    1. public void setSize(int nRow, int nCol)
       “Mengosongkan” matriks (menjadi hanya berisi 0) dan mengubah ukurannya menjadi nRow x nCol.
    2. public void setElmt(int i, int j, double val)
       Mengubah nilai elemen pada indeks i,j menjadi val.
    3. public void addRow()
       menambah baris baru pada Matrix.
    4. public void addElmt(int i, double val)
       Menambah elemen sekaligus kolom baru pada baris i.
  - Operasi input :
    1. public void readMatrix(int nRow, int nCol, String fileName, Scanner input)
       Jika nRow dan nCol diberi dan fileName null, input matriks dibaca dari keyboard secara per baris. Jika fileName diberi, membaca isi Matrix dari file. Matrix terisi sesuai input.
  - Operasi output :
    1. public void displayMatrix(String fileName)
       Mencetak isi Matrix ke layar dan/atau file (jika fileName tidak null)
  - Operasi-operasi lain :
    1. public static Matrix multiplyMatrix(Matrix m1, Matrix m2)
       Mengembalikan Matrix hasil perkalian Matrix m1 dan m2.
    2. public static Matrix transpose(Matrix m)
       Mengembalikan transpose Matrix m
    3. public static Matrix copyMatrix(Matrix m)
       Mengembalikan salinan Matrix m.
    4. public static void swap(Matrix m, int row1, int row2)
       Menukarkan baris row1 dengan row2 pada Matrix m.

3.2 Implementasi Eliminasi Gauss dan Gauss Jordan (SPL.java fungsi gaussElim dan gaussjordanElim)

Fungsi gaussElim dan gaussjordanElim berada di dalam *class* SPL yang memiliki atribut dan metode :

- Atribut :
  - static String newline : karakter newline (untuk pencetakan hasil)

- Metode :
  - static Matrix inputSPL(Scanner input)
    Menerima input SPL dalam bentuk Matrix augmented (per baris) dari keyboard maupun file. Jika dari keyboard akan diminta banyak persamaan dan variabel diikuti Matrix augmentednya (per baris). Jika dari file akan diminta *path* file. Mengembalikan Matrix augmented berisi SPL tersebut.

  - static void forwardElimination(Matrix m) (overloader, boolean inv bernilai false) dan static void forwardElimination(Matrix m, boolean inv)
    Melakukan eliminasi tahap maju pada Matrix m. Boolean inv digunakan untuk menentukan batas jumlah kolom yang menjadi faktor eliminasi (jika diminta inverse matriks, yang menjadi faktor eliminasi hanya sampai saat nomor kolom = nomor baris. Jika matriks *augmented*, kolom terakhir tidak menjadi faktor eliminasi). Algoritmanya sebagai berikut :
    1. Iterasi tiap kolom (r) dan tentukan elemen mutlak terbesar dari kolom r dengan k (baris yang sedang diiterasi) = r (untuk mencegah pembagian dengan 0).
    2. Jika kolom seluruhnya nol (elemen mutlak terbesar adalah nol), cek kolom berikutnya tanpa melanjutkan iterasi baris. Jika tidak, tukar baris yang berisi elemen “maksimum” dengan baris yang sedang diiterasi.
    3. Bagi baris “maksimum” dengan elemen maksimum tadi supaya terbentuk satu utama.
    4. Kurangi setiap baris di bawah baris maksimum dengan n kali baris maksimum. n adalah elemen baris yang akan dikurang pada kolom r.
    5. Ulangi sampai baris “maksimum” adalah baris terakhir.

  - static void backwardElimination(Matrix m)
    Menerima Matrix m dan melakukan eliminasi tahap mundur pada Matrix m (syarat : Matrix m sudah merupakan matriks eselon baris) dengan algoritma sebagai berikut :
    1. Mulai iterasi dari baris terakhir (i).
    2. Kurangi setiap baris di atas baris i dengan n kali baris i. n adalah elemen pada kolom j=i di baris yang bersangkutan.
    3. Ulangi sampai i adalah baris kedua dari atas.

  - static boolean zeroRow(Matrix m, int rowIdx, boolean SPL), static boolean zeroRow(Matrix m, int rowIdx, boolean SPL, boolean inv)
    Memeriksa apakah baris rowIdx pada Matrix m merupakan baris yang hanya terdiri dari 0. Boolean SPL dan inv digunakan untuk menentukan batas kolom yang dicek. Jika SPL bernilai true, maka kolom yang dicek hanya sampai kolom kedua terakhir. Jika inv bernilai true, maka kolom yang dicek hanya sampai saat indeks kolom tidak melebihi indeks baris. Jika keduanya false, maka seluruh kolom akan dicek.

- public static double[] gaussElim (Scanner input), public static double[] gaussElim (Matrix m, boolean SPL), public static double[] gaussElim (Matrix m, boolean SPL, String outputFile)
  Melakukan eliminasi Gauss pada Matrix augmented m dan menuliskan hasilnya ke layar dan/atau file. Parameter Scanner input berarti Matrix m belum ada dan akan dilakukan input Matrix m lebih dulu oleh pengguna. Boolean SPL dipakai untuk menentukan apakah hasil perlu dicetak ke layar dan/atau file (karena fungsi ini dipakai di fungsi lain juga). String outputFile merupakan nama file untuk output hasil. Jika tidak disimpan ke dalam file, outputFile bernilai null.

- public static double[] gaussjordanElim (Scanner input), public static double[] gaussjordanElim (Matrix m, String outputFile) :
  Melakukan eliminasi Gauss-Jordan pada Matrix m dan menuliskan hasilnya ke layar dan/atau file. Parameter Scanner input berarti Matrix m belum ada dan akan dilakukan input Matrix m oleh pengguna lebih dulu. String outputFile merupakan nama file untuk output hasil. Jika outputFile bernilai null, hasil tidak disimpan ke file.

• Alur kerja program eliminasi Gauss dan Gauss-Jordan (gaussElim dan gaussjordanElim) :
  1. Menerima Matrix augmented m (jika belum ada) dan nama file *output* (opsional).
  2. Melakukan forwardElimination dilanjut backwardElimination untuk gaussjordanElim.
  3. Untuk gaussElim dilakukan *backward substitution*.
  4. Cek apakah SPL memiliki solusi berdasarkan ciri-ciri matriks seperti pada Bab 2 dan output hasil sesuai dengan jenis solusinya.

3.3 Implementasi Determinan (Determinant.java)

• Atribut : Tidak ada

• Metode
  - static Matrix inputDeterminant(Scanner input)
    Menerima input berupa Matrix m yang akan dicari determinannya baik dari keyboard maupun dari file. Jika dari keyboard, akan diminta dimensi Matrix dan isi Matrixnya. Jika dari file, akan diminta *path* file.
  - static void minor(Matrix m,Matrix temp, int a, int b)
    Membuat matriks minor dari matriks m.
  - public static double determinanCofactor (Scanner input):
    Menerima input Matrix lalu mengembalikan hasil determinannya dengan metode kofaktor.
  - public static double determinanCofactor(Matrix m):
    Menghasilkan determinan Matrix m dengan metode cofactor. Diasumsikan m matriks persegi dan relatif kecil (ukuran kurang dari 10x10).
  - static int detRowReduction(Matrix m) :
    Mengubah Matrix m menjadi matriks segitiga atas.
  - public static double determinanReduction(Scanner input) :
    Menerima input Matrix lalu menghasilkan determinannya dengan metode reduksi baris. Algoritmanya mirip dengan forwardElimination tetapi mengembalikan jumlah pertukaran baris yang dilakukan.

– public static double determinanReduction(Matrix m)
  Menghasilkan determinan matrix m dengan metode reduksi baris, diasumsikan m matriks persegi.

- Alur

  A. Metode reduksi baris
     1. Menerima input dimensi Matrix m dan isinya (per baris) atau *path* file.
     2. Melakukan reduksi baris pada Matrix m.
     3. Mengalikan setiap elemen diagonal dari Matrix m yang sudah direduksi untuk mendapat determinan.
     4. Jika banyak pertukaran baris ganjil, kalikan hasil tersebut dengan –1 dan kembalikan nilai determinan. Jika tidak, langsung kembalikan nilai determinan.

  B. Metode ekspansi kofaktor
     1. Menerima input dimensi Matrix m dan isinya (per baris) atau *path* file.
     2. Membuat matriks minor dari tiap elemen baris pertama.
     3. Mengalikan tiap elemen baris pertama dengan kofaktornya dan jumlahkan hasilnya.

3.4 Implementasi Inverse Matriks (Inverse.java)

- Atribut (Tidak ada)

- Metode
  - static Matrix inputInverse(Scanner input)
    Menerima input Matrix m dari keyboard maupun dari file. Jika dari keyboard, diminta dimensi Matrix dan isinya. Jika dari file, diminta path file. Mengembaikan Matrix hasil bacaan.
  - public static Matrix matrixCofactor(Matrix m)
    Mengembalikan matriks kofaktor dari matriks m. Algoritma sesuai yang diajarkan di kelas.
  - public static Matrix adjoint(Matrix m)
    Mengembalikan Matrix adjoint dari Matrix m dengan mentranspose matriks kofaktor dari m.
  - public static Matrix inverseAdjoint(Scanner input)
    Menerima input Matrix lalu menghasilkan inversenya dengan metode adjoin.
  - public static Matrix inverseAdjoint(Matrix m)
    Menghasilkan inverse matriks dengan metode adjoint, diasumsikan m matriks persegi.
  - public static Matrix inverseGaussJordan(Scanner input)
    Menerima input matriks lalu menghasilkan inverse matriks dengan metode Gauss-Jordan.
  - public static Matrix inverseGaussJordan(Matrix m)
    Mengasilkan invers matriks m dengan eliminasi Gauss Jordan.

- Alur

  A. Metode adjoin
     1. Menerima input Matrix m (jika belum ada).

2. Menghitung determinan Matrix m. Jika 0, kembalikan matriks tidak memiliki inverse. Jika tidak, lanjut ke langkah berikutnya.
3. Membuat Matrix adjoin dari Matrix m.
4. Bagi tiap elemen Matrix adjoin dengan determinan Matrix m.
5. Kembalikan Matrix inverse tersebut.

B. Metode Gauss-Jordan
1. Menerima input Matrix m (jika belum ada).
2. Meng-*augment* Matrix m dengan matriks identitas berdimensi sama.
3. Melakukan eliminasi tahap maju dan mundur pada Matrix m.
4. Mengecek apakah ada baris yang berisi 0 semua (di bagian kiri matriks identitas). Jika ada, berarti matriks tidak memiliki inverse. Jika tidak, lanjut ke langkah berikutnya.
5. Isi Matrix inverseM dengan elemen Matrix m bagian augmented.
6. Kembalikan Matrix inverseM.

3.5 Implementasi Kaidah Cramer dan Metode Inverse (SPL.java fungsi cramer dan solveSPLInverse)

Fungsi Crammer dan metode inverse diimplementasikan dalam *class* SPL dengan :

- Atribut :
  - static String newline : karakter newline (untuk pencetakan hasil)

- Metode :
  - static Matrix inputSPL(Scanner input)
    Menerima input SPL dalam bentuk Matrix augmented dari keyboard maupun file.

  - Public static double[] cramer(Scanner input), Public static double[] cramer(Matrix m, String outputFile)
    Melakukan pencarian solusi dari sistem persamaan linier dengan kaidah cramer kemudian menampilkan ke layar dan/atau menyimpan ke dalam file.

  - public static Matrix solveSPLInverse(Scanner input), public static Matrix solveSPLInverse(Matrix arr, String outputFile)
    Menghitung solusi SPL dari Matrix arr dengan metode inverse dan mengembalikan Matrix yang berisi solusi serta mencetak hasil ke layar dan/atau file.

- Alur kerja program Kaidah Cramer:
  1. Menerima Matrix augmented m yang berukuran n x n+1 (jika belum ada).
  2. Membuat Matrix temp yang berukuran n x n. Kemudian set elemen Matrix temp dengan elemen Matrix m.
  3. Cari determinan Matrix temp.
  4. Jika determinan Matrix temp tidak 0, maka dibuat array y yang berisi kolom ke n+1 dari Matrix m.
  5. Setelah itu dibuat array result yang berukuran n.
  6. Kemudian dilakukan looping sebanyak n kali. Setiap loop dilakukan copy Matrix temp ke Matrix cramer, setelah itu tiap nilai pada kolom ke-j akan diganti dengan elemen

pada array y. Kemudian set elemen ke-j dari array result dengan nilai determinan matriks cramer dibagi dengan determinan Matrix temp.
7. Lakukan looping hingga loop terakhir. Array result merupakan solusi dari SPL.

- Alur kerja program metode inverse :
  1. Menerima Matrix augmented SPL (jika belum ada) dan nama file output (opsional).
  2. Memisahkan kolom hasil (menjadi Matrix B) dan kolom koefisien variabel (menjadi Matrix A).
  3. Menginverse Matrix A.
  4. Jika Matrix A memiliki inverse, kalikan Matrix A dengan Matrix B dan cetak hasilnya ke layar dan/atau file. Jika tidak, tuliskan Matrix tidak memiliki inverse di terminal.

3.6 Implementasi Interpolasi Polinom

Interpolasi Polinom diimplementasikan dalam class Interpolasi dengan :

- Atribut :
  - static String newline : karakter newline (untuk pencetakan hasil)

- Metode :
  - static Matrix inputTitikInterpolasi(Scanner input) :
    Menerima input pasangan nilai x dan y yang ingin diinterpolasi baik dari file maupun masukan dari keyboard. Jika dari keyboard, diminta jumlah titik dan setiap titiknya. Jika dari file, diminta path file.
  - Static double inputX(Scanner input) :
    Menerima input nilai x yang ingin dicari estimasi nilai fungsinya dari terminal.
  - Public static void interpolasiPolinom(Scanner input), public static void interpolasiPolinom (Matrix titik, double x, String outputFile) :
    Melakukan interpolasi polinom pada pasangan titik (x,y) yang sudah diterima dan menuliskan fungsi hasil interpolasi dan hasil taksiran dari x yang diinginkan ke layar dan/atau file.

- Alur kerja program interpolasi polinom:
  1. Menerima n pasangan titik(x_i,y_i) yang akan diinterpolasi baik dari file maupun keyboard.
  2. Menerima nilai x yang ingin ditaksir nilainya dengan interpolasi.
  3. Membuat Matrix augmented m yang berukuran n x (n+1).
  4. Mengisi Matrix m dengan kolom pertama diisi 1, kolom kedua dengan nilai x_i, kolom kedua diisi dengan nilai x_i^2, dan seterusnya hingga kolom ke-n diisi dengan x_i^{n-1}. Kemudian kolom ke-(n+1) diisi dengan nilai y_i.
  5. Setelah dibuat Matrix augmented m, dilakukan eliminasi gauss dan hasilnya akan disimpan ke array a.
  6. Kemudian hasil taksirannya dihitung dengan rumus $y = a_0 + a_1x + a_2x^2 + …. + a_{n-1}x^{n-1}$.

3.7 Implementasi Interpolasi Bicubic (Bicubic.java)

Interpolasi Bicubic diimplementasikan dalam class Bicubic dengan :

- Atribut : Tidak ada

- Metode :
  - static Matrix inputFBicubic(Scanner input) :
    Menerima input Matrix 4x4 (disertai 2 nilai a dan b) untuk persoalan interpolasi bicubic. Bisa dari keyboard maupun file. Jika dari file, nilai a dan b berada di baris terakhir file. Mengembalikan Matrix persoalan interpolasi bicubic.
  - public static void bicubic(Scanner input), public static void bicubic(Matrix inputMtrx, String outputFile) :
    Melakukan interpolasi bicubic berdasarkan matriks 4x4 (Matrix inputMtrx). Jika parameter berupa Scanner, berarti inputMtrx belum ada dan akan diminta input matriksnya lebih dulu. String outputFile merupakan nama file *output* untuk menyimpan hasil. Mencetak hasil interpolasi bicubic ke layar dan/atau file.

- Alur kerja program bicubic :
  1. Menerima Matrix inputMtrx berukuran 4x4 (jika belum ada) yang juga berisi koordinat target (a,b) di baris terakhir dan nama file output (opsional).
  2. Mengubah inputMtrx menjadi Matrix berukuran 16x1 supaya sesuai dengan persamaan bicubic.
  3. Membuat matriks koefisien X berdasarkan rumus.
  4. Menginvers matriks koefisien X lalu mengalikannya dengan matriks 16x1 tadi untuk mendapat vektor **a**.
  5. Menghitung hasil interpolasi dari nilai a b yang diinput di awal berdasarkan nilai-nilai di vektor **a**.
  6. Mencetak hasil ke layar dan/atau file.

3.8 Implementasi Regresi Linier Berganda

- Atribut :
  - static String newline : karakter newline.
- Metode :
  - static Matrix inputDataRegresi(Scanner input)
    Menerima input data regresi dan mengembalikan data dalam bentuk Matrix augmented dari file maupun keyboard. Format per barisnya adalah xi1 xi2 .. xin yi. Jika dari keyboard diminta jumlah peubah dan banyak persamaan. Jika dari file diminta path file.
  - static double[] inputTargetRegresi(Matrix data, Scanner input)
    Menerima input nilai peubah yang ingin dicari hasilnya. Hanya diterima dari keyboard.
  - public static void multiRegression(Scanner input), public static void multiRegression(Matrix data, double[] target, String outputFile)
    Melakukan regresi linier berganda dari data dan target yang sudah diinput. Jika parameter berupa Scanner, akan diminta input Matrix data dan target terlebih dahulu.
- Alur kerja program regresi linier berganda:
  1. Menerima data, target (jika belum ada), dan nama file output (opsional).
  2. Membuat SPL regresi berdasarkan rumus Normal Estimation Equation for Multiple Linear Regression.
  3. Mengecek apakah SPL memiliki solusi unik.

4. Jika SPL memiliki solusi unik, cetak persamaan regresi dan hasil interpolasi target ke layar dan/atau file. Jika tidak, tuliskan regresi tidak bisa dilakukan.

3.9 Implementasi Output File (FileOutput.java)

- Atribut : Tidak ada
- Metode :
  - static void printFile(String fileName, String output) :
    Membuat file baru dengan nama sesuai String fileName (jika belum ada) dan menuliskan String output ke dalam file tersebut.
- Alur kerja program output file : sama persis dengan fungsi printFile.

3.10 Implementasi Program Utama (LinearAlgebraMenu.java)

- Atribut : Tidak ada
- Metode :
  - public static void menu() :
    Menampilkan menu operasi-operasi yang bisa dilakukan.
- Alur :
  1. Tampilkan menu.
  2. Terima input jenis operasi yang ingin dilakukan.
  3. Lakukan operasi tersebut memakai fungsi yang sudah dibuat.
  4. Ulangi sampai pengguna memutuskan untuk keluar dari program.

Catatan : Setiap fungsi/prosedur yang meminta input meminta parameter Scanner untuk mencegah konflik dengan scanner yang digunakan di program utama atau program yang dibuat user *library*.

BAB IV
EKSPERIMEN

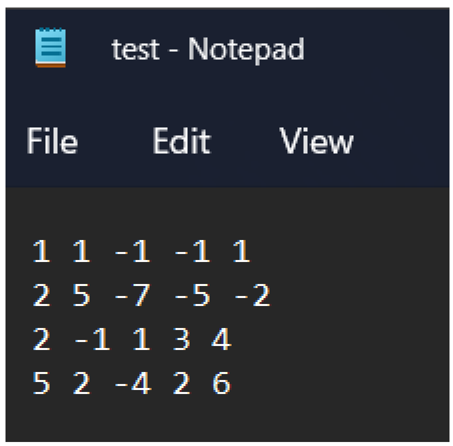
4.1 Solusi SPL Ax = b

a.

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

Gambar 4.1.1.0 : Soal studi kasus 1a

Isi file test.txt



Gambar 4.1.1.1 : Isi file uji coba studi kasus 1a (test.txt)

Hasil eliminasi Gauss (file resultGaussElim.txt tidak dibuat karena tidak ada solusi)

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
N
Masukkan banyak persamaan
4
Masukkan banyak variabel
4
Masukan semua persamaan per baris. Format : ai1 ai2 .. ain bi.
Baris/Persamaan 1
1 1 -1 -1 1
Baris/Persamaan 2
2 5 -7 -5 -2
Baris/Persamaan 3
2 -1 1 3 4
Baris/Persamaan 4
5 2 -4 2 6
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussElim.txt
SPL tidak memiliki solusi
Continue ? Y/N
```

Gambar 4.1.1.2 : Hasil studi kasus 1a eliminasi Gauss

Hasil Eliminasi Gauss-Jordan

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
2
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
N
SPL tidak memiliki solusi
Continue ? Y/N
```

Gambar 4.1.1.3 : Hasil studi kasus 1a eliminasi Gauss-Jordan

Hasil Metode Inverse

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
3
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
N
Matriks tidak memiliki inverse
Continue ? Y/N
```

Gambar 4.1.1.4 : Hasil studi kasus 1a metode inverse

Hasil Kaidah Cramer

```
Continue ? Y/N
Y
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
4
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
N
Matriks koefisien memiliki determinan 0, SPL tidak bisa diselesaikan
Continue ? Y/N
```

Gambar 4.1.1.5 : Hasil studi kasus 1a Kaidah Cramer

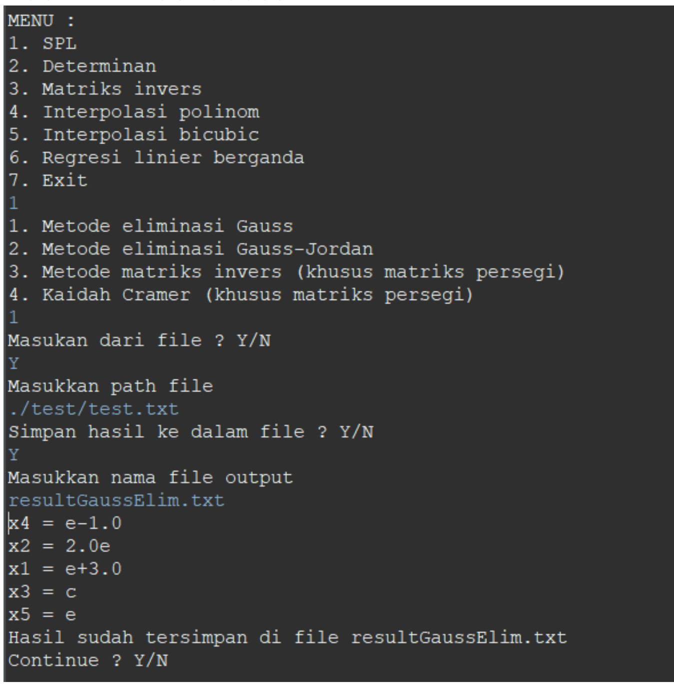
Analisis : SPL di atas tidak memiliki solusi, sehingga ditulis SPL tidak memiliki solusi.

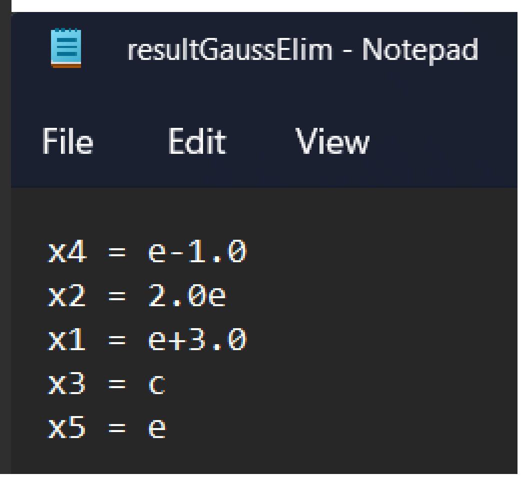
b.

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

Gambar 4.1.2.0 : Soal studi kasus 1b

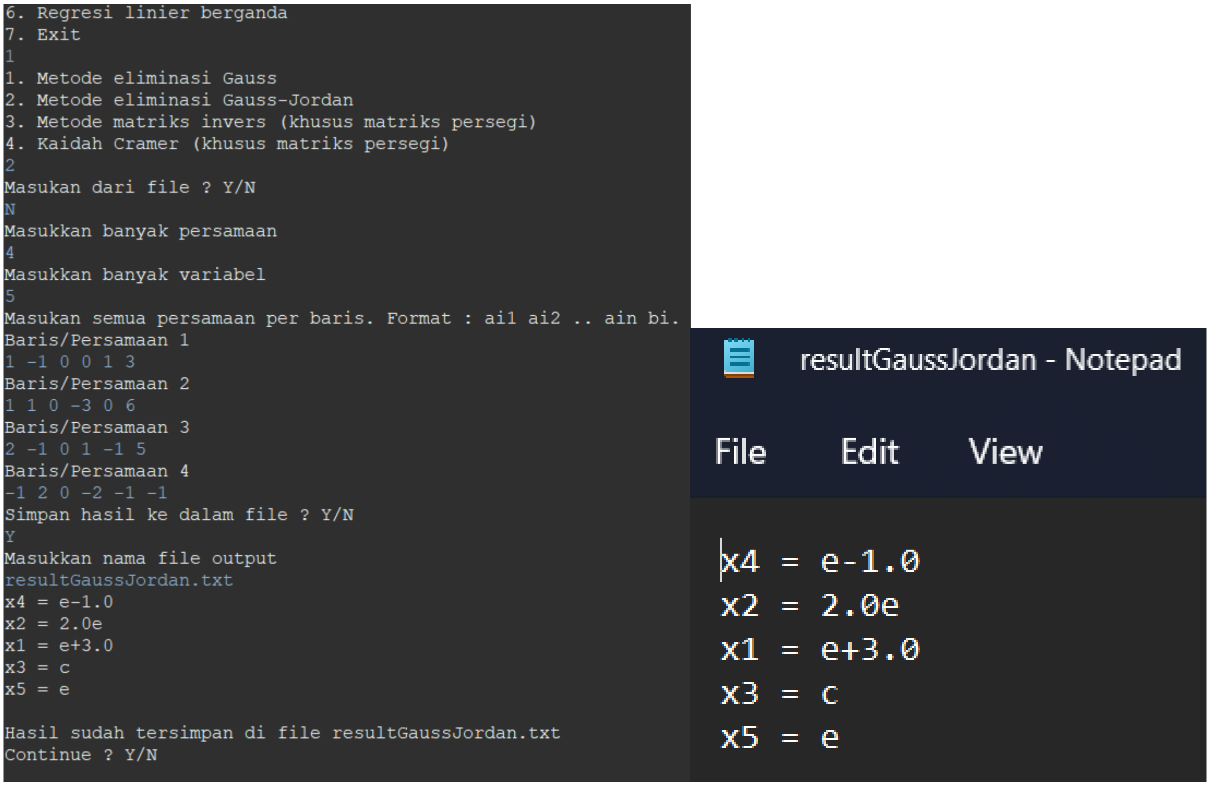
Hasil Eliminasi Gauss





Gambar 4.1.2.1 dan Gambar 4.1.2.2 : Hasil studi kasus 1b eliminasi Gauss

Hasil Eliminasi Gauss Jordan



Gambar 4.1.2.3 dan 4.1.2.4 : Hasil studi kasus 1b eliminasi Gauss-Jordan

Metode inverse matriks dan Kaidah Cramer tidak bisa dilakukan karena matriks A bukan matriks persegi.

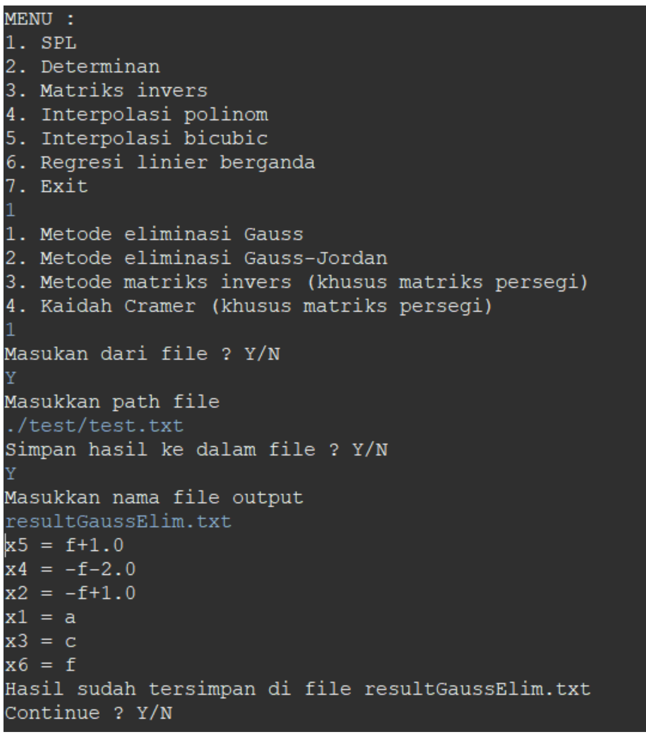
Analisis : SPL tersebut memiliki banyak solusi (terlihat dari jumlah persamaan yang lebih sedikit dari jumlah variabel), sehingga hasilnya merupakan persamaan parametrik.

c.

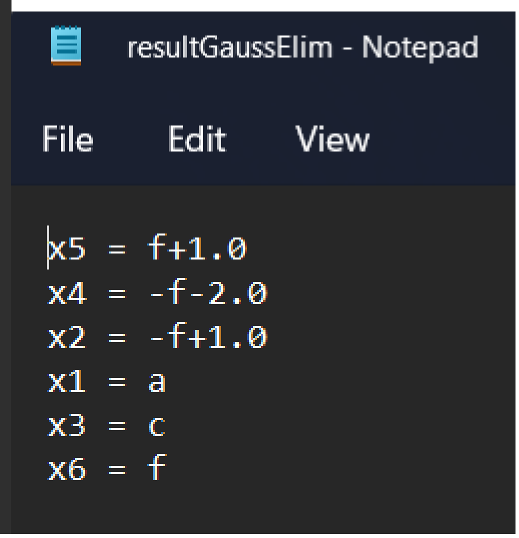
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

Gambar 4.1.3.0 : Soal studi kasus 1c

Hasil Eliminasi Gauss



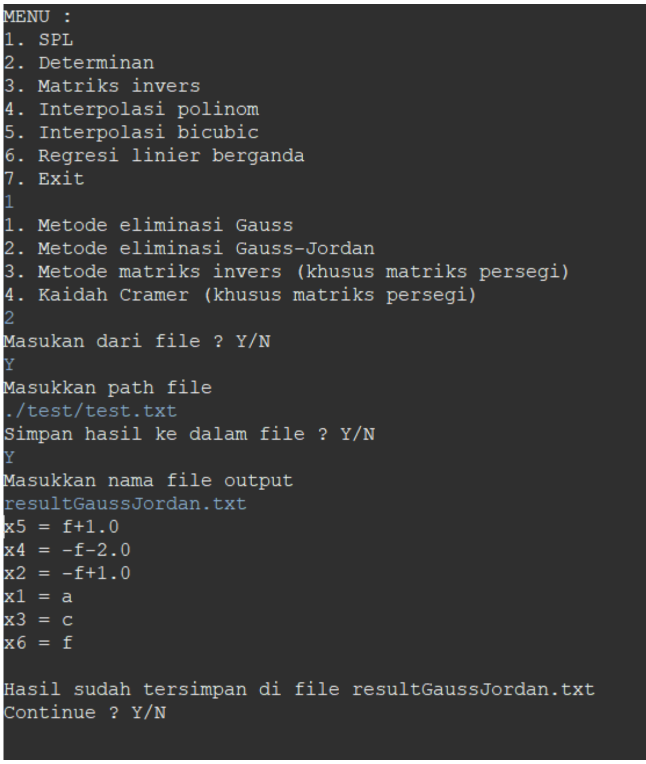
```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussElim.txt
x5 = f+1.0
x4 = -f-2.0
x2 = -f+1.0
x1 = a
x3 = c
x6 = f
Hasil sudah tersimpan di file resultGaussElim.txt
Continue ? Y/N
```

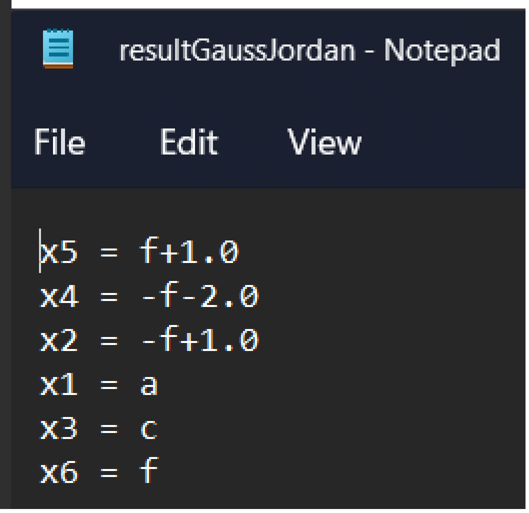



```
x5 = f+1.0
x4 = -f-2.0
x2 = -f+1.0
x1 = a
x3 = c
x6 = f
```

Gambar 4.1.3.1 dan 4.1.3.2 : Hasil studi kasus 1c eliminasi Gauss

Hasil Eliminasi Gauss-Jordan





Gambar 4.1.3.3 dan 4.1.3.4 : Hasil studi kasus 1c eliminasi Gauss-Jordan

Metode inverse matriks dan Kaidah Cramer tidak bisa dilakukan karena matriks A bukan matriks persegi.

Analisis : SPL tersebut memiliki banyak solusi (terlihat dari jumlah persamaan yang lebih sedikit dari jumlah variabel), sehingga hasilnya merupakan persamaan parametrik.

d.

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

H adalah matriks Hilbert. Cobakan untuk n = 6 dan n = 10.

Gambar 4.1.4.0 : Soal studi kasus 1d

n = 6

Hasil Eliminasi Gauss

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussElim.txt
x1 = 36.00000000101886
x2 = -630.0000000310961
x3 = 3360.0000002190764
x4 = -7560.000000585684
x5 = 7560.000000659453
x6 = -2772.0000002637953

Hasil sudah tersimpan di file resultGaussElim.txt
Continue ? Y/N
```

```
resultGaussElim - Notepad
File   Edit   View

x1 = 36.00000000101886
x2 = -630.0000000310961
x3 = 3360.0000002190764
x4 = -7560.000000585684
x5 = 7560.000000659453
x6 = -2772.0000002637953
```

Gambar 4.1.4.1 dan 4.1.4.2 : Hasil studi kasus 1d eliminasi Gauss (N=6)

Hasil Eliminasi Gauss-Jordan

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
2
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussJordan.txt
x1 = 36.000000001018634
x2 = -630.0000000310952
x3 = 3360.0000002190754
x4 = -7560.000000585684
x5 = 7560.000000659453
x6 = -2772.0000002637953

Hasil sudah tersimpan di file resultGaussJordan.txt
Continue ? Y/N
```

```
resultGaussJordan - Notepad
File    Edit    View

x1 = 36.000000001018634
x2 = -630.0000000310952
x3 = 3360.0000002190754
x4 = -7560.000000585684
x5 = 7560.000000659453
x6 = -2772.0000002637953
```

Gambar 4.1.4.3 dan 4.1.4.4 : Hasil studi kasus 1d eliminasi Gauss-Jordan (N=6)

Hasil Metode Inverse

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
3
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultInverseSPL.txt
x1 = 36.000000001018634
x2 = -630.0000000310952
x3 = 3360.0000002190754
x4 = -7560.000000585684
x5 = 7560.000000659453
x6 = -2772.0000002637953

Hasil sudah tersimpan di file resultInverseSPL.txt
Continue ? Y/N
```

```
resultInverseSPL - Notepad
File    Edit    View

x1 = 36.000000001018634
x2 = -630.0000000310952
x3 = 3360.0000002190754
x4 = -7560.000000585684
x5 = 7560.000000659453
x6 = -2772.0000002637953
```

Gambar 4.1.4.5 dan 4.1.4.6 : Hasil studi kasus 1d metode inverse (N=6)

Hasil Kaidah Cramer

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
4
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultCramer.txt
x1 = 36.0000000013453
x2 = -630.0000000277283
x3 = 3360.00000020911
x4 = -7560.000000547803
x5 = 7560.000000613385
x6 = -2772.0000002445704

Hasil sudah tersimpan di file resultCramer.txt
Continue ? Y/N
```

```
resultCramer - Notepad
File   Edit   View

x1 = 36.0000000013453
x2 = -630.0000000277283
x3 = 3360.00000020911
x4 = -7560.000000547803
x5 = 7560.000000613385
x6 = -2772.0000002445704
```

Gambar 4.1.4.7 dan 4.1.4.8 : Hasil studi kasus 1d kaidah Cramer (N=6)

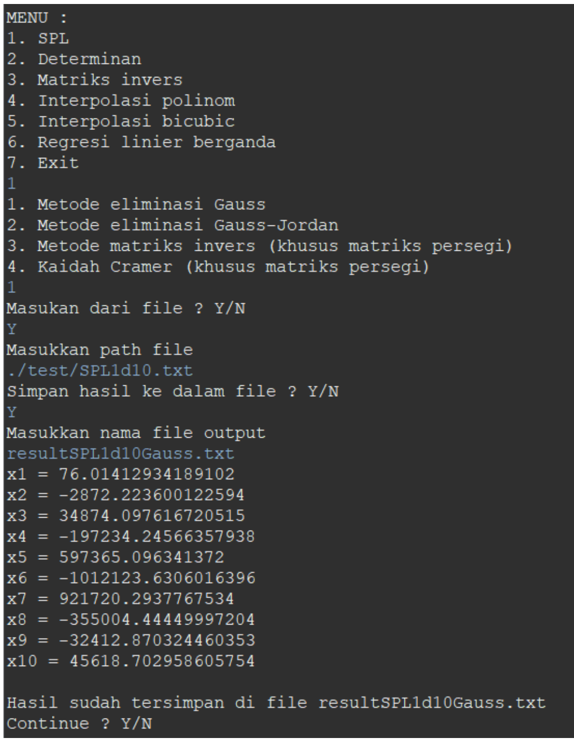
n = 10

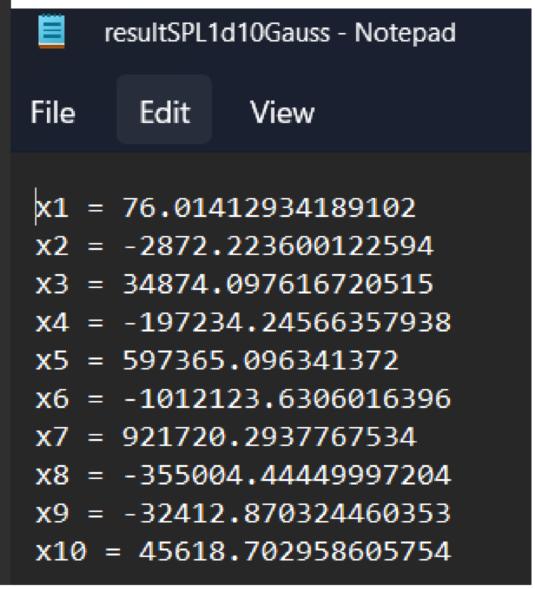
Isi file SPL1d10.txt

```
SPL1d10 - Notepad
File   Edit   View

1.0 0.5 0.3333333333333333 0.25 0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 1
0.5 0.3333333333333333 0.25 0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909
0.3333333333333333 0.25 0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.090909090909
0.25 0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333
0.2 0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.0833333333333333
0.1666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.
0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.
0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.07142857142857142 0.
0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.07142857142857142 0.066666
0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.07142857142857142 0.06666666666666667 0.0625
```

Gambar 4.1.4.9 : Isi file SPLd10.txt

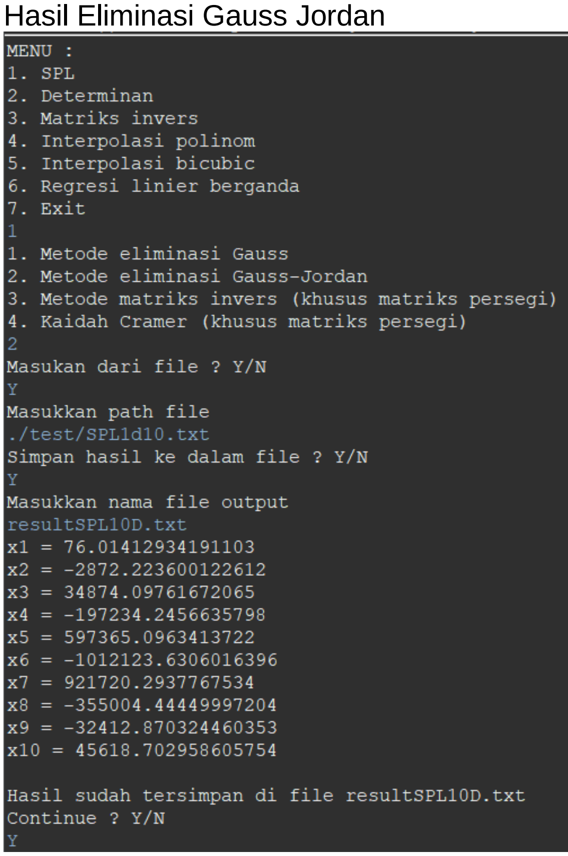
Hasil Eliminasi Gauss

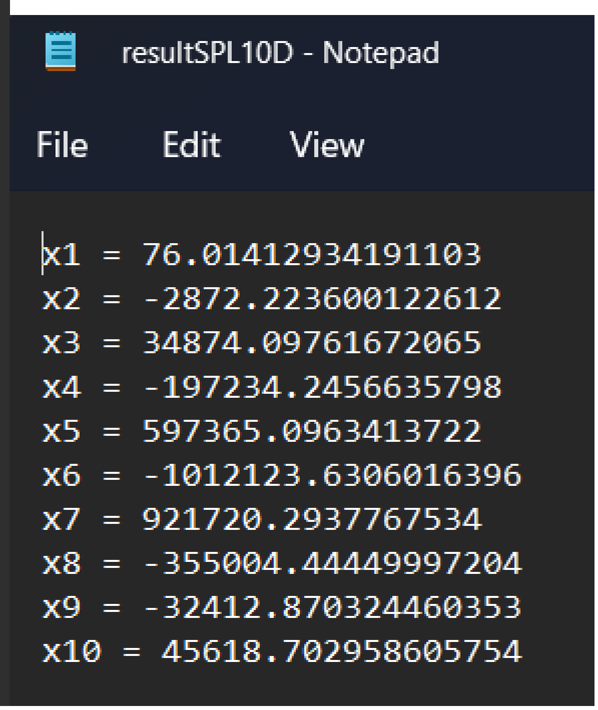




Gambar 4.1.4.10 dan 4.1.4.11 : Hasil studi kasus 1d eliminasi Gauss (N=10)

Hasil Eliminasi Gauss Jordan





Gambar 4.1.4.12 dan 4.1.4.13 : Hasil studi kasus 1d eliminasi Gauss-Jordan (N=10)

Hasil Metode Inverse

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
3
Masukan dari file ? Y/N
Y
Masukkan path file
./test/SPL1d10.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
result10DInverse.txt
x1 = 76.01412934191103
x2 = -2872.223600122612
x3 = 34874.09761672065
x4 = -197234.2456635798
x5 = 597365.0963413722
x6 = -1012123.6306016396
x7 = 921720.2937767534
x8 = -355004.44449997204
x9 = -32412.870324460353
x10 = 45618.702958605754

Hasil sudah tersimpan di file result10DInverse.txt
Continue ? Y/N
Y
```

```
result10DInverse - Notepad
File   Edit   View

x1 = 76.01412934191103
x2 = -2872.223600122612
x3 = 34874.09761672065
x4 = -197234.2456635798
x5 = 597365.0963413722
x6 = -1012123.6306016396
x7 = 921720.2937767534
x8 = -355004.44449997204
x9 = -32412.870324460353
x10 = 45618.702958605754
```

Gambar 4.1.4.14 dan 4.1.4.15 : Hasil studi kasus 1d metode inverse (N=10)

Hasil Kaidah Cramer

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
4
Masukan dari file ? Y/N
Y
Masukkan path file
./test/SPL1d10.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultSPL10DCramer.txt
Matriks koefisien memiliki determinan 0, SPL tidak bisa diselesaikan
Continue ? Y/N
```

Gambar 4.1.4.16 : Hasil studi kasus 1d kaidah Cramer (N=10)

Analisis : Untuk n=6, hasil relatif akurat. Tetapi untuk n=10, hasil kurang akurat karena nilai yang dimasukkan di file bukanlah nilai eksak setiap pecahan.

4.2 SPL Berbentuk Matriks *Augmented*

a.

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

Gambar 4.2.1.0 : Soal studi kasus 2a

Hasil Eliminasi Gauss

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussElim.txt
x2 = 2.0c
x1 = d-1.0
x3 = c
x4 = d
Hasil sudah tersimpan di file resultGaussElim.txt
Continue ? Y/N
```

```
resultGaussElim - Notepad
File   Edit   View

x2 = 2.0c
x1 = d-1.0
x3 = c
x4 = d
```

Gambar 4.2.1.1 dan 4.2.1.2 : Hasil studi kasus 2a eliminasi Gauss

Hasil Eliminasi Gauss-Jordan

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
2
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussJordan.txt
x2 = 2.0c
x1 = d-1.0
x3 = c
x4 = d

Hasil sudah tersimpan di file resultGaussJordan.txt
Continue ? Y/N
```

```
resultGaussJordan - Notepad
File    Edit    View

x2 = 2.0c
x1 = d-1.0
x3 = c
x4 = d
```

Gambar 4.2.1.3 dan 4.2.1.4 : Hasil studi kasus 2a eliminasi Gauss-Jordan

Hasil Metode Inverse

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
3
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultInverseSPL.txt
Matriks tidak memiliki inverse
Continue ? Y/N
```

Gambar 4.2.1.5 : Hasil studi kasus 2a metode inverse

Hasil Kaidah Cramer

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
4
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultCramer.txt
Matriks koefisien memiliki determinan 0, SPL tidak bisa diselesaikan
Continue ? Y/N
```

Gambar 4.2.1.5 : Hasil studi kasus 2a Kaidah Cramer

Analisis **:** SPL memiliki solusi parametrik, sehingga metode inverse dan kaidah cramer tidak bisa dipakai karena determinan matriks koefisien adalah 0.

b.

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

Gambar 4.2.2.0 : Soal studi kasus 2b

Hasil Eliminasi Gauss

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussElim.txt
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0

Hasil sudah tersimpan di file resultGaussElim.txt
Continue ? Y/N
```

```
resultGaussElim - Notepad
File    Edit    View

x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

Gambar 4.2.2.1 dan 4.2.2.2 : Hasil studi kasus 2b eliminasi Gauss

Hasil Eliminasi Gauss-Jordan

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
2
Masukan dari file ? Y/N
Y
Masukkan path file
./test/test.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultGaussJordan.txt
x4 = 1.0
x3 = 1.0
x2 = 2.0
x1 = 0.0

Hasil sudah tersimpan di file resultGaussJordan.txt
Continue ? Y/N
```

```
resultGaussJordan - Notepad
File    Edit    View

x4 = 1.0
x3 = 1.0
x2 = 2.0
x1 = 0.0
```

Gambar 4.2.2.3 dan 4.2.2.4 : Hasil studi kasus 2b eliminasi Gauss-Jordan

Metode inverse matriks dan Kaidah Cramer tidak bisa dilakukan karena matriks A bukan matriks persegi.

Analisis : Jumlah persamaan lebih besar dari jumlah variabel dan SPL memiliki solusi unik.

4.3 SPL

3. SPL berbentuk

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

Gambar 4.3.1.0 : Soal studi kasus 3a

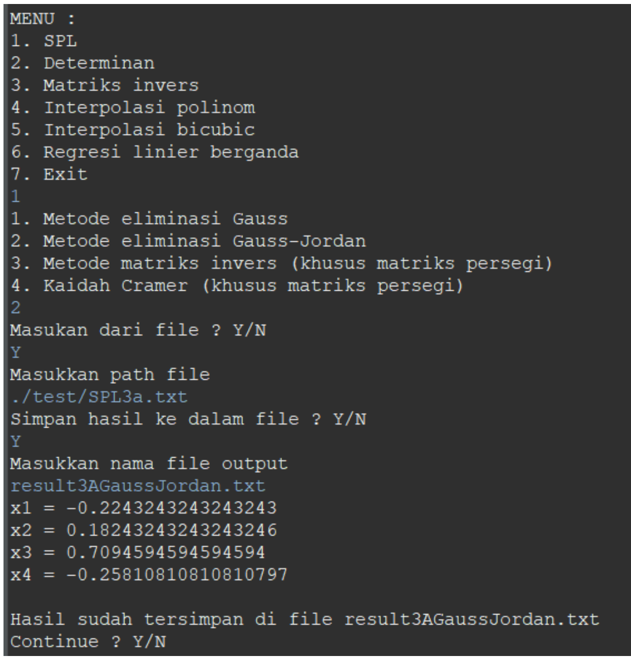
Hasil Eliminasi Gauss

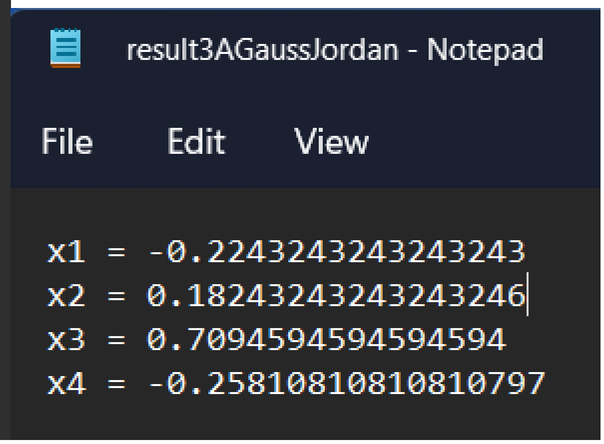
```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
Y
Masukkan path file
./test/SPL3a.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
result3AGauss.txt
x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797

Hasil sudah tersimpan di file result3AGauss.txt
Continue ? Y/N
```

```
result3AGauss - Notepad
File   Edit   View

x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

Gambar 4.3.1.1 dan 4.3.1.2 : Hasil studi kasus 3a eliminasi Gauss

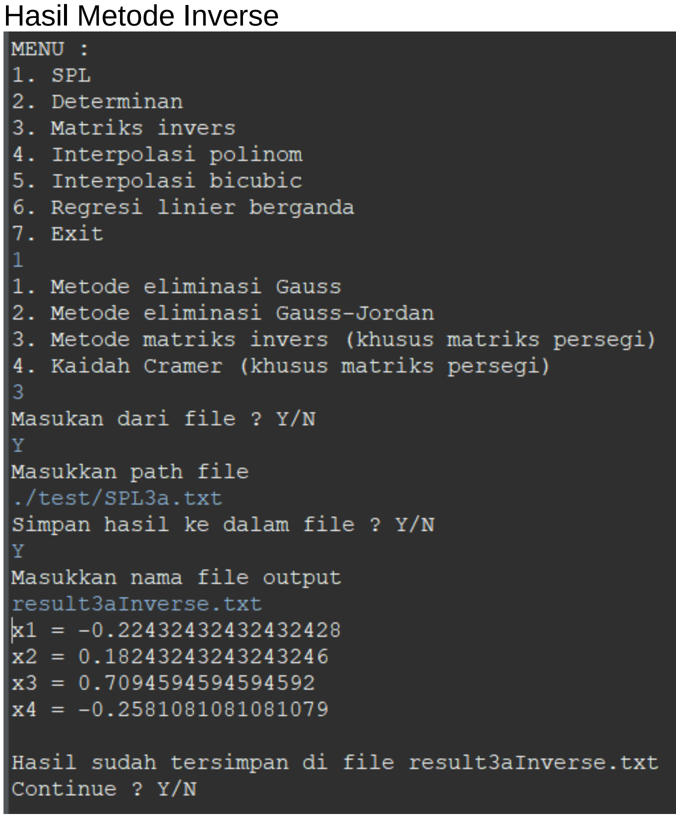
Hasil Eliminasi Gauss-Jordan

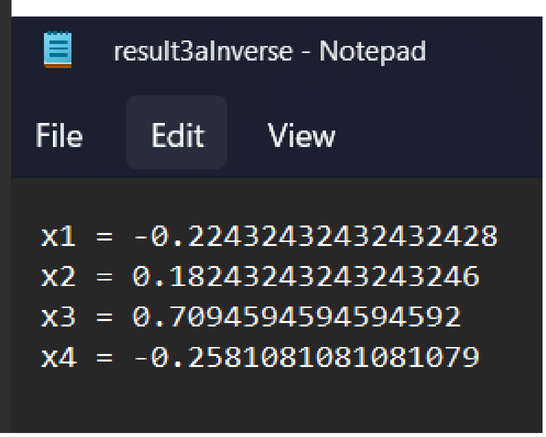




Gambar 4.3.1.3 dan 4.3.1.4 : Hasil studi kasus 3a eliminasi Gauss-Jordan

Hasil Metode Inverse





Gambar 4.3.1.5 dan 4.3.1.6 : Hasil studi kasus 3a metode inverse

Hasil Kaidah Cramer

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
4
Masukan dari file ? Y/N
Y
Masukkan path file
./test/SPL3a.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
result3ACramer.txt
x1 = -0.22432432432432414
x2 = 0.1824324324324324
x3 = 0.7094594594594594
x4 = -0.25810810810810797

Hasil sudah tersimpan di file result3ACramer.txt
Continue ? Y/N
```

```
result3ACramer - Notepad
File    Edit    View

x1 = -0.22432432432432414
x2 = 0.1824324324324324
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

Gambar 4.3.1.7 dan 4.3.1.8 : Hasil studi kasus 3a Kaidah Cramer

Analisis : SPL berhasil diselesaikan dengan benar

b.

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

Gambar 4.3.2.0 : Soal studi kasus 3b

Hasil Eliminasi Gauss

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
1
Masukan dari file ? Y/N
Y
Masukkan path file
./test/SPL3b.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultSPL3bGauss.txt
SPL tidak memiliki solusi
Continue ? Y/N
```

Gambar 4.3.2.1 : Hasil studi kasus 3b eliminasi Gauss

Hasil Eliminasi Gauss-Jordan

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
1
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks invers (khusus matriks persegi)
4. Kaidah Cramer (khusus matriks persegi)
2
Masukan dari file ? Y/N
Y
Masukkan path file
./test/SPL3b.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultSPL3bJordan.txt
SPL tidak memiliki solusi
```

Gambar 4.3.2.2 : Hasil studi kasus 3b eliminasi Gauss-Jordan

Metode Inverse dan Kaidah Cramer tidak bisa digunakan karena matriks bukan matriks persegi

Analisis : Hasil sudah benar karena SPL tidak memiliki solusi

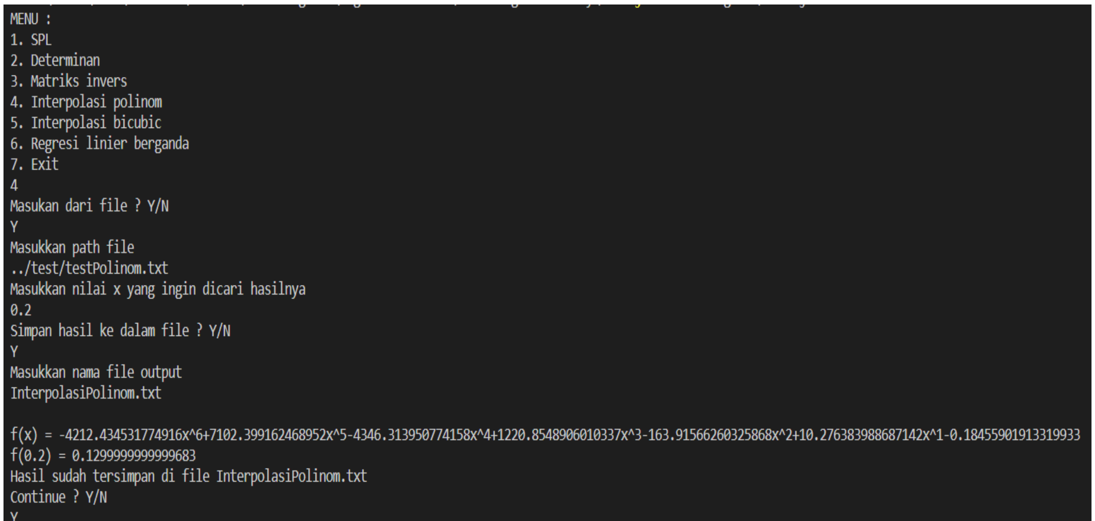
4.4. Interpolasi Pada Fungsi

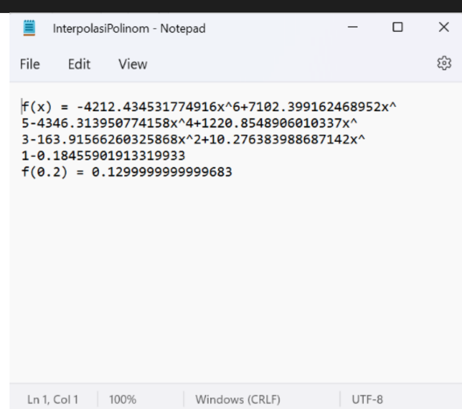
Diberikan tabel pasangan x dan f(x) berikut :

x	0.4	0.7	0.11	0.14	0.17	0.2	0.23
f(x)	0.043	0.005	0.058	0.072	0.1	0.13	0.147

Tabel 4.4.0 : Data interpolasi 4a

1. Untuk x = 0.2





Gambar 4.4.1 dan 4.4.2 : Hasil studi kasus 4a (x = 0.2)

2. Untuk x = 0.55

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
4
Masukan dari file ? Y/N
Y
Masukkan path file
../test/testPolinom.txt
Masukkan nilai x yang ingin dicari hasilnya
0.55
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
InterpolasiPolinom.txt

f(x) = -4212.434531774916x^6+7102.399162468952x^5-4346.313950774158x^4+1220.8548906010337x^3-163.91566260325868x^2+10.276383988687142x^1-0.18455901913319933
f(0.55) = 2.137571620847609
Hasil sudah tersimpan di file InterpolasiPolinom.txt
Continue ? Y/N
```

```
InterpolasiPolinom - Notepad
File   Edit   View

f(x) = -4212.434531774916x^6+7102.399162468952x^
5-4346.313950774158x^4+1220.8548906010337x^
3-163.91566260325868x^2+10.276383988687142x^
1-0.18455901913319933
f(0.55) = 2.137571620847609

Ln 1, Col 1   100%   Windows (CRLF)   UTF-8
```

Gambar 4.4.3 dan 4.4.4 : Hasil studi kasus 4a (x = 0.55)

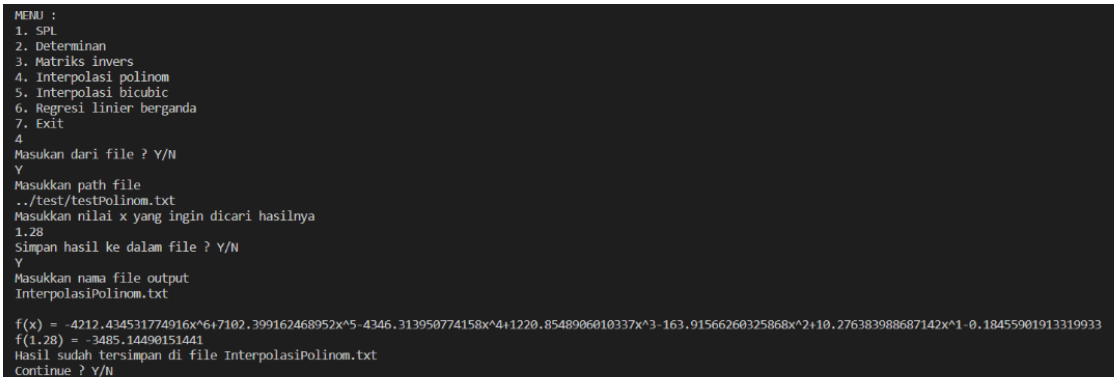
3. Untuk x = 0.85

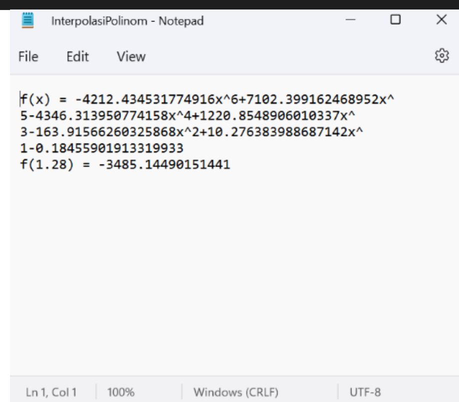
```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
4
Masukan dari file ? Y/N
Y
Masukkan path file
../test/testPolinom.txt
Masukkan nilai x yang ingin dicari hasilnya
0.85
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
InterpolasiPolinom.txt

f(x) = -4212.434531774916x^6+7102.399162468952x^5-4346.313950774158x^4+1220.8548906010337x^3-163.91566260325868x^2+10.276383988687142x^1-0.18455901913319933
f(0.85) = -66.26963931345449
Hasil sudah tersimpan di file InterpolasiPolinom.txt
Continue ? Y/N
```

```
InterpolasiPolinom - Notepad
File   Edit   View

f(x) = -4212.434531774916x^6+7102.399162468952x^
5-4346.313950774158x^4+1220.8548906010337x^
3-163.91566260325868x^2+10.276383988687142x^
1-0.18455901913319933
f(0.85) = -66.26963931345449

Ln 1, Col 1   100%   Windows (CRLF)   UTF-8
```

Gambar 4.4.5 dan 4.4.6 : Hasil studi kasus 4a (x = 0.85)

4. Untuk x = 1.28





Gambar 4.4.7 dan 4.4.8 : Hasil studi kasus 4a (x = 1.28)

Analisis : Fungsi sudah berhasil mengembalikan persamaan dan nilai taksiran.

4.5. Polinom Interpolasi Pada Data Covid-19

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

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

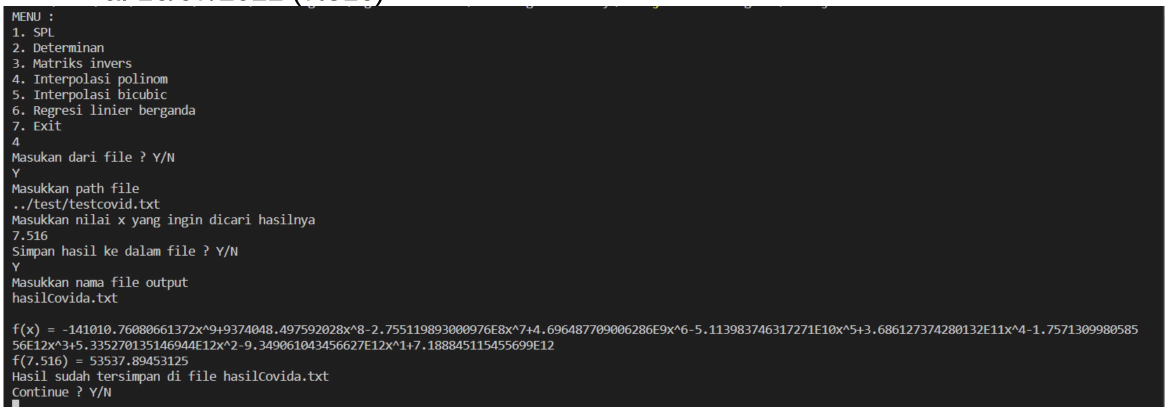
Gambar 4.5.0 : Data studi kasus 4b

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

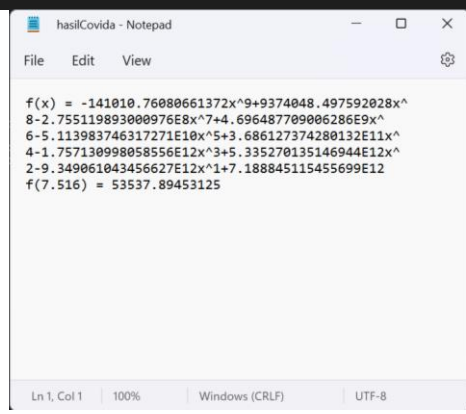
tanggal(desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut)

Digunakan data di atas dengan memanfaatkan polinom interpolasi untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

a. 16/07/2022 (7.516)



```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
4
Masukan dari file ? Y/N
Y
Masukkan path file
../test/testcovid.txt
Masukkan nilai x yang ingin dicari hasilnya
7.516
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
hasilCovida.txt

f(x) = -141010.76080661372x^9+9374048.497592028x^8-2.755119893000976E8x^7+4.696487709006286E9x^6-5.113983746317271E10x^5+3.686127374280132E11x^4-1.7571309980585
56E12x^3+5.335270135146944E12x^2-9.349061043456627E12x^1+7.188845115455699E12
f(7.516) = 53537.89453125
Hasil sudah tersimpan di file hasilCovida.txt
Continue ? Y/N
```

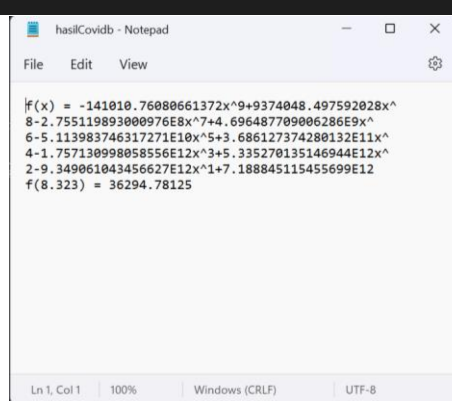


```
hasilCovida - Notepad
File Edit View

f(x) = -141010.76080661372x^9+9374048.497592028x^
8-2.755119893000976E8x^7+4.696487709006286E9x^
6-5.113983746317271E10x^5+3.686127374280132E11x^
4-1.757130998058556E12x^3+5.335270135146944E12x^
2-9.349061043456627E12x^1+7.188845115455699E12
f(7.516) = 53537.89453125

Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

Gambar 4.5.1 dan 4.5.2 : Hasil studi kasus 4b (Tanggal = 7.516)

b. 10/08/2022 (8.323)

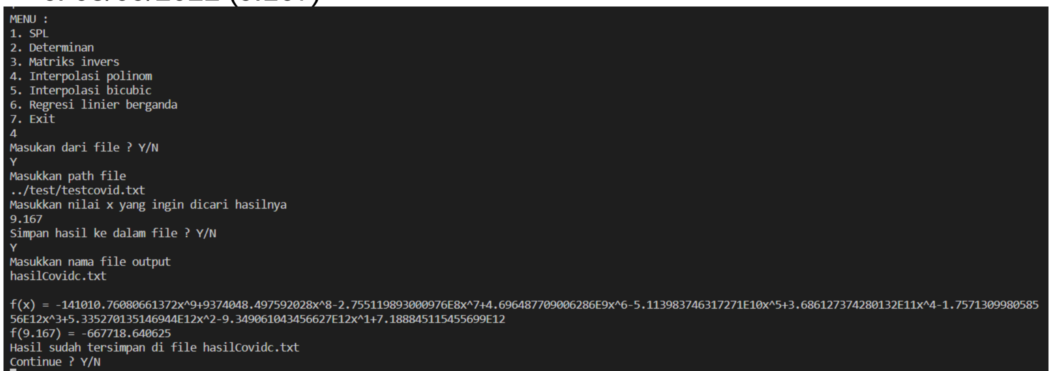
```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
4
Masukan dari file ? Y/N
Y
Masukkan path file
../test/testcovid.txt
Masukkan nilai x yang ingin dicari hasilnya
8.323
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
hasilCovidb.txt

f(x) = -141010.76080661372x^9+9374048.497592028x^8-2.755119893000976E8x^7+4.696487709006286E9x^6-5.113983746317271E10x^5+3.686127374280132E11x^4-1.7571309980585
56E12x^3+5.335270135146944E12x^2-9.349061043456627E12x^1+7.188845115455699E12
f(8.323) = 36294.78125
Hasil sudah tersimpan di file hasilCovidb.txt
Continue ? Y/N
```

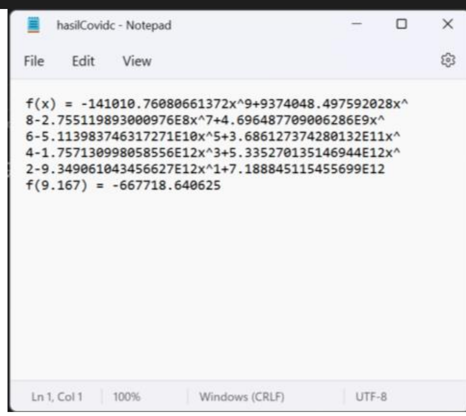


```
hasilCovidb - Notepad
File Edit View

f(x) = -141010.76080661372x^9+9374048.497592028x^
8-2.755119893000976E8x^7+4.696487709006286E9x^
6-5.113983746317271E10x^5+3.686127374280132E11x^
4-1.757130998058556E12x^3+5.335270135146944E12x^
2-9.349061043456627E12x^1+7.188845115455699E12
f(8.323) = 36294.78125

Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

Gambar 4.5.3 dan 4.5.4 : Hasil studi kasus 4b (Tanggal = 8.323)

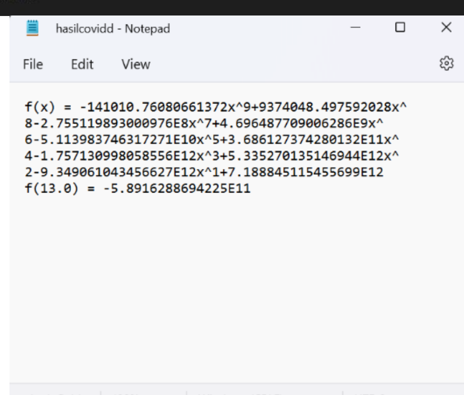
c. 05/09/2022 (9.167)





Gambar 4.5.5 dan 4.5.6 : Hasil studi kasus 4b (Tanggal = 9.167)

d. 31/12/2022 (13)



Gambar 4.5.7 dan 4.5.8 : Hasil studi kasus 4b (Tanggal = 13)

Analisi : Hasil taksiran sudah benar, akan tetapi hasil tersebut tidak bisa dijadikan taksiran yang sebenarnya karena terdapat hasil fungsi yang bernilai negatif.

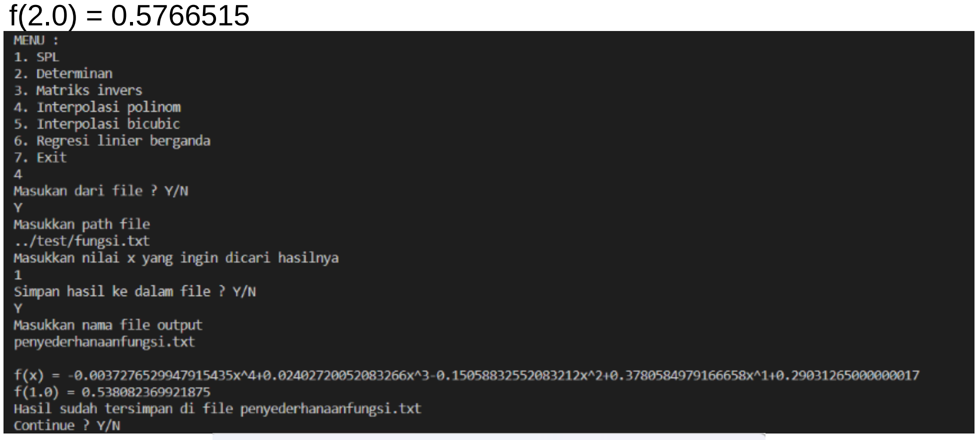
4.6. Penyederhanaan Fungsi

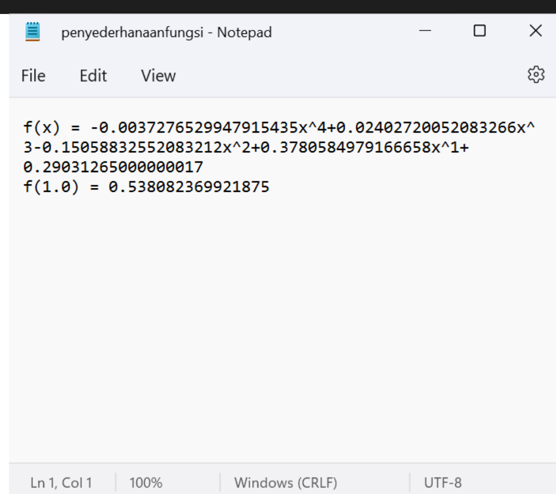
$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

Gambar 4.6.0 : Soal studi kasus 4c

dengan polinom interpolasi derajat n di dalam selang [0, 2].

- N = 5
  Didapat :
  f(0.4) = 0.41888423
  f(0.8) = 0.507158
  f(1.2) = 0.560925
  f(1.6) = 0.583686
  f(2.0) = 0.5766515





Gambar 4.6.1 dan 4.6.2 : Hasil studi kasus 4c

Analisis : didapat fungsi hasil penyederhanaan :
$f(x) = -0.0037276529947915435x^4 + 0.02402720052083266x^3 - 0.15058832552083212x^2 + 0.3780584979166658x + 0.29031265000000017$

4.7. Interpolasi Bicubic

Diberikan matriks input:

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

Tentukan nilai:

f(0,0) = ?

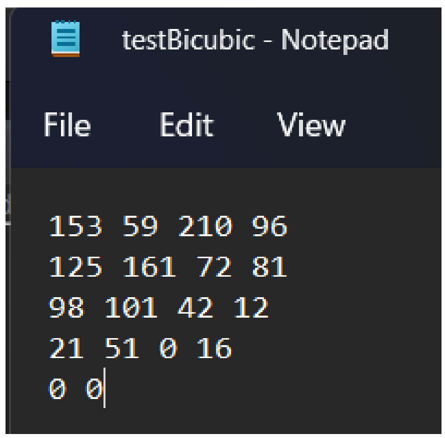
f(0.5, 0.5) = ?

f(0.25, 0.75) = ?

f(0.1, 0.9) = ?

Gambar 4.7.0 : Soal studi kasus interpolasi bicubic

TestBicubic.txt



Gambar 4.7.1 : Isi file TestBicubic.txt

Hasil interpolasi bicubic

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
5
Masukan dari file ? Y/N
Y
Masukkan path file
./test/testBicubic.txt
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultBicubic.txt
f(0.0,0.0) = 161.0
Hasil sudah tersimpan di file resultBicubic.txt
Continue ? Y/N
```

```
resultBicubic - Notepad
File    Edit    View
f(0.0,0.0) = 161.0
```

```
f(0.5,0.5) = 97.72656249999999
```

```
f(0.5,0.5) = 97.72656249999999
```

```
f(0.25,0.75) = 82.50207519531251
```

```
f(0.25,0.75) = 82.50207519531251
```

```
f(0.1,0.9) = 74.69611850000001
```

```
f(0.1,0.9) = 74.69611850000001
```

Gambar 4.7.2 - 4.7.9 : Hasil studi kasus interpolasi bicubic

Analisis : Hasil yang diberikan sudah benar

4.8. Regresi Linier Berganda Pada Tabel Keadaan Cuaca

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian

13

estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Gambar 4.8.0 : Soal studi kasus regresi linier berganda

Hasil regresi linier

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
6
Masukan dari file ? Y/N
Y
Masukkan path file
./test/testRegresi.txt
Masukkan nilai-nilai peubah yang ingin dicari hasilnya. Format : x1 x2 .. xn
50 76 29.30
Simpan hasil ke dalam file ? Y/N
Y
Masukkan nama file output
resultRegresi.txt

f(x) = -3.507778140863008 - 0.002624990745881574x1 + 7.989410472203326E-4x2 + 0.15415503019761256x3
f(50.0,76.0,29.3) = 0.9384342262217071
Hasil sudah tersimpan di file resultRegresi.txt
Continue ? Y/N
```

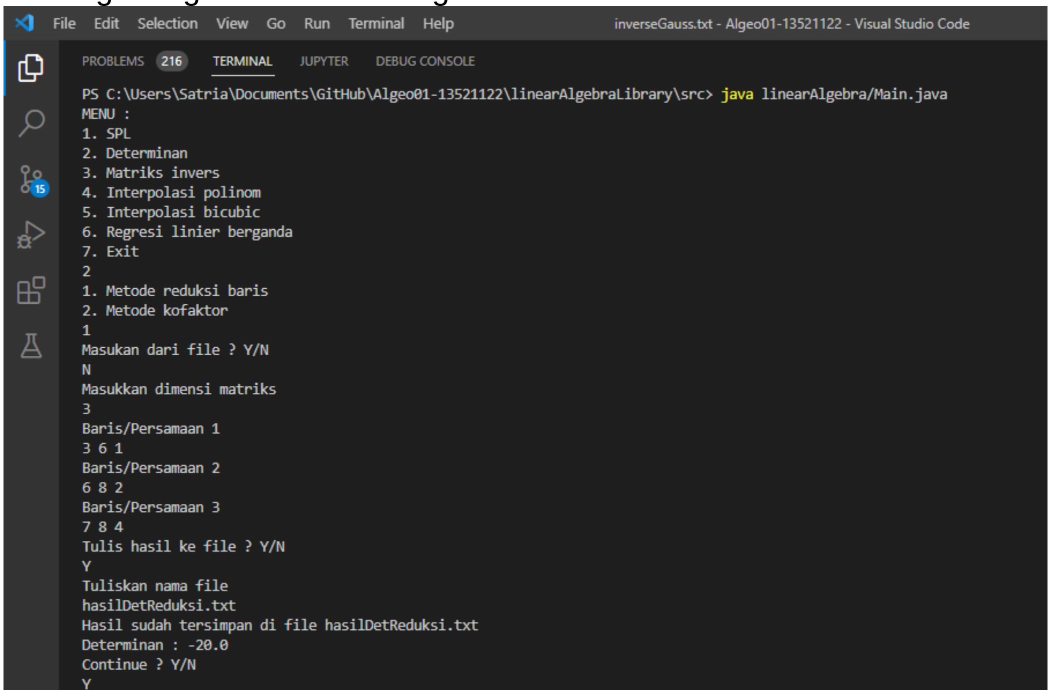
```
resultRegresi - Notepad
File   Edit   View

f(x) = -3.507778140863008 - 0.002624990745881574x1 + 7.989410472203326E-4x2 + 0.15415503019761256x3
f(50.0,76.0,29.3) = 0.9384342262217071
```

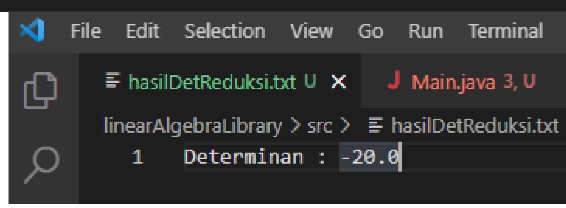
Gambar 4.8.1 dan 4.8.2 : Hasil studi kasus regresi linier berganda

Analisis : Fungsi sudah berhasil mengembalikan persamaan dan hasil regresi untuk nilai yang diminta

4.9. Determinan (Determinant.java)

1. Menghitung determinan dengan metode reduksi baris





Gambar 4.9.1 Hasil Determinan Matriks dengan Metode Reduksi baris

2. Menghitung determinan matriks dengan metode kofaktor

```
PS C:\Users\Satria\Documents\GitHub\Algeo01-13521122\linearAlgebraLibrary\src> java linearAlgebra/Main.java
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
2
1. Metode reduksi baris
2. Metode kofaktor
2
Masukan dari file ? Y/N
Y
Masukkan path file
../test/testDet.txt
Tulis hasil ke file ? Y/N
Y
Tuliskan nama file
hasilDetKofaktor.txt
Hasil sudah tersimpan di file hasilDetKofaktor.txt
Determinan : 9.0
Continue ? Y/N
Y
```

```
hasilDetKofaktor.txt U ×    hasilDetReduksi.txt
linearAlgebraLibrary > src > hasilDetKofaktor.txt
  1    Determinan : 9.0
```

Gambar 4.9.2. Hasil Determinan Matriks dengan Metode Kofaktor

4.10. Inverse (Inverse.java)

1. Menghitung inverse matiks dengan metode eliminasi Gauss-Jordan

```
PS C:\Users\Satria\Documents\GitHub\Algeo01-13521122\linearAlgebraLibrary\src> java linearAlgebra/Main.java
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
3
1. Metode Gauss-Jordan
2. Metode adjoint
1
Masukan dari file ? Y/N
N
Masukkan dimensi matriks
4
Baris/Persamaan 1
8 3 6 9
Baris/Persamaan 2
2 1 7 4
Baris/Persamaan 3
9 7 0 3
Baris/Persamaan 4
1 9 5 8
Tulis hasil ke file ? Y/N
Y
Tuliskan nama file
hasilInversGaussJordan.txt
0.027837259100642428 0.03426124197002138 0.08779443254817984 -0.08137044967880086
-0.11884368308351181 0.046038543897216316 0.0867237687366167 0.07815845824411134
-0.10171306209850112 0.24660956459671668 0.03818700927908637 -0.023197715917202003
0.19379014989293364 -0.2102069950035689 -0.13240542469664524 0.0617416131334761
Hasil sudah tersimpan di file hasilInversGaussJordan.txt
Continue ? Y/N
Y
```

```
linearAlgebraLibrary > src > hasilInversGaussJordan.txt
0.027837259100642428 0.03426124197002138 0.08779443254817984 -0.08137044967880086
-0.11884368308351181 0.046038543897216316 0.0867237687366167 0.07815845824411134
-0.10171306209850112 0.24660956459671668 0.03818700927908637 -0.023197715917202003
0.19379014989293364 -0.2102069950035689 -0.13240542469664524 0.0617416131334761
```

Gambar 4.10.1 Hasil Inverse Matriks dengan metode Eliminasi Gauss-Jordan

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
3
1. Metode Gauss-Jordan
2. Metode adjoint
1
Masukan dari file ? Y/N
N
Masukkan dimensi matriks
3
Baris/Persamaan 1
1 2 3
Baris/Persamaan 2
4 5 6
Baris/Persamaan 3
7 8 9
Matriks tidak memiliki inverse
```

Gambar 4.10.2 Kasus Matiks Tidak Memiliki Inverse Karena Determinan Matriksnya 0

2. Menghitung inverse matriks dengan metode Adjoint

```
PS C:\Users\Satria\Documents\GitHub\Algeo01-13521122\linearAlgebraLibrary\src> java linearAlgebra/Main.java
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
3
1. Metode Gauss-Jordan
2. Metode adjoint
2
Masukan dari file ? Y/N
Y
Masukkan path file
../test/testInverseKecil.txt
Tulis hasil ke file ? Y/N
Y
Tuliskan nama file
hasilInversAdjoint.txt
0.24609375 0.4375 0.0625 -0.1015625
-0.1875 0.0 -0.0 0.125
0.265625 0.25 -0.25 -0.09375
0.05859375 -0.5625 0.0625 0.0234375
Hasil sudah tersimpan di file hasilInversAdjoint.txt
Continue ? Y/N
Y
```

```
hasilInversAdjoint.txt U ×    Main.java 3, U
linearAlgebraLibrary > src > hasilInversAdjoint.txt
1    0.24609375 0.4375 0.0625 -0.1015625
2    -0.1875 0.0 -0.0 0.125
3    0.265625 0.25 -0.25 -0.09375
4    0.05859375 -0.5625 0.0625 0.0234375
```

Gambar 4.10.3 Hasil Inverse Matriks dengan Metode Adjoint

```
MENU :
1. SPL
2. Determinan
3. Matriks invers
4. Interpolasi polinom
5. Interpolasi bicubic
6. Regresi linier berganda
7. Exit
3
1. Metode Gauss-Jordan
2. Metode adjoint
2
Masukan dari file ? Y/N
N
Masukkan dimensi matriks
3
Baris/Persamaan 1
4 8 9
Baris/Persamaan 2
0 0 0
Baris/Persamaan 3
2 7 4
Matriks tidak memiliki inverse
```

Gambar 4.10.4 Kasus Matriks Tidak Memiliki Inverse karena terdapat baris bernilai 0

BAB V
KESIMPULAN, SARAN, REFLEKSI

5.1 Kesimpulan

Dengan mengikuti perkuliahan IF2123 Aljabar Linear dan Geometri selama setengah semester, penulis telah berhasil mengimplementasikan teori teori matriks ke dalam sebuah program dengan bahasa pemrograman Java. Penulis telah membuat program yang mampu untuk menyelesaikan persoalan sistem persamaan linear yang menghasilkan solusi unik, solusi banyak, dan tidak memiliki solusi dengan metode Eliminasi Gauss, Eliminasi Gauss Jordan, metode Balikan, dan metode Cramer. Penulis juga telah membuat program untuk mencari determinan dan balikan dari suatu matriks persegi dengan menggunakan metode Cofactor dan metode matriks reduksi. Selain itu, penulis juga telah membuat program yang dapat melakukan Interpolasi Polinom, Interpolasi Bicubic, dan Regresi Linear Berganda. Semua program yang dibuat dapat dijalankan dengan menerima masukan langsung dari keyboard maupun dari file.

5.2 Saran

Dari program yang sudah dibuat, beberapa saran yang bisa diberi penulis untuk pengembangan lebih lanjut :

- Pembuatan GUI untuk program.
- Pembuatan fitur yang diminta pada soal bonus, yaitu perbesaran gambar menggunakan interpolasi bicubic.
- Penggunaan algoritma yang lebih efisien dan penambahan operasi-operasi aljabar linier lain.
- Masih ada operasi yang bisa dibuat menjadi fungsi sehingga tidak perlu diketik berulang-ulang, seperti input matriks dari file dan output persamaan parametrik pada fungsi eliminasi Gauss dan Gauss-Jordan.
- Penambahan penanganan input yang tidak sesuai.

5.3 Refleksi

Dari tugas besar ini, penulis mendapat ilmu dan pengalaman memrogram dalam bahasa Java, belajar membagi waktu di tengah-tengah semua kesibukan yang lain, membagi tugas, dan berkerja dalam tim sebagai bekal ilmu dan pengalaman bagi kami kedepannya.

DAFTAR REFERENSI

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2022-2023/algeo22-23.htm

https://towardsdatascience.com/what-really-is-a-matrix-determinant-89c09884164c

https://www.geeksforgeeks.org/gaussian-elimination/

https://www.geeksforgeeks.org/filewriter-class-in-java/

https://www.w3schools.com/java/java_files_create.asp

Lampiran

Link Repository : https://github.com/Nat10k/Algeo01-21122