Laporan Tugas Besar 1 IF2123 Aljabar Linier dan Geometri
Sistem Persamaan Linier, Determinan, dan Aplikasinya
Semester I Tahun 2023/2024

Disusun oleh

- 13522066 Nyoman Ganadipa Narayana
- 13522084 Dhafin Fawwaz Ikramullah
- 13522117 Mesach Harmasendro

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

DAFTAR ISI

BAB I DESKRIPSI MASALAH

A. Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Kami 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-1b), 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).

$$\begin{bmatrix} 0 & \mathbf{2} & 1 & -1 \\ 0 & 0 & \mathbf{3} & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad \begin{bmatrix} 0 & \mathbf{1} & 0 & -\frac{2}{3} \\ 0 & 0 & \mathbf{1} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Gambar 1.1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

Di dalam Tugas Besar 1 ini, Kami 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. Penjelasan tentang interpolasi dan regresi adalah seperti di bawah ini.

B. Spesifikasi Tugas

1. Buatlah pustaka (*library* atau *package*) dalam **Bahasa Java** untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah *Cramer* (kaidah *Cramer* khusus untuk SPL dengan *n* peubah dan *n* persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.
2. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi 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
```

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

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. Masukan kemudian dilanjutkan dengan satu buah baris berisi satu buah nilai x yang akan ditaksir menggunakan fungsi interpolasi yang telah didefinisikan. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513) dan akan mencari nilai y saat x = 8.3, maka di dalam *file text* ditulis sebagai berikut:

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

4. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x_{1i}, x_{2i}, …, x_{ni}, nilai y_i, dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari *file*, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$).

6. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

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

   dan untuk regresi adalah

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

7. Untuk persoalan *bicubic spline interpolation*, masukan dari *file text* (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai a dan b untuk mencari nilai $f(a, b)$.

   Misalnya jika nilai dari $f(0, 0)$, $f(1, 0)$, $f(0, 1)$, $f(1, 1)$, $f_x(0, 0)$, $f_x(1, 0)$, $f_x(0, 1)$, $f_x(1, 1)$, $f_y(0, 0)$, $f_y(1, 0)$, $f_y(0, 1)$, $f_y(1, 1)$, $f_{xy}(0, 0)$, $f_{xy}(1, 0)$, $f_{xy}(0, 1)$, $f_{xy}(1, 1)$ berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 serta nilai *a* dan *b* yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi *file text* ditulis sebagai berikut:

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

Luaran yang dihasilkan adalah nilai dari *f*(0.5, 0.5).

8. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam *file***.
9. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20).
10. Program **tidak harus** berbasis GUI, cukup *text-based* saja, namun boleh menggunakan GUI (memakai kakas *Eclipse* misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

```
MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi linier 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 TEORI SINGKAT

A. Sistem Persamaan Linear

Sistem Persamaan Linear dapat direpresentasikan dalam bentuk matriks seperti di bawah ini:

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

SPL dengan pendekatan matriks ini bisa diselesaikan dengan banyak cara. Beberapa diantaranya yaitu metode Eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode Matriks Balikan, dan kaidah Cramer.

1. Metode Eliminasi Gauss

Metode eliminasi gauss dilakukan dengan membuat matriks SPL menjadi sebuah matriks eselon baris dengan cara *OBE.* Salah satu contoh bentuk matriks eselon baris adalah sebagai berikut:

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

Setelah matriks SPL diubah ke dalam bentuk eselon baris maka akan terbentuk sistem persamaan baru yang lebih sederhana. Melalui sistem persamaan yang baru tersebut bisa dicari seluruh nilai dari variabel yang terdapat dalam suatu SPL.

2. Metode Eliminasi Gauss Jordan

Metode eliminasi gauss dilakukan seperti metode eliminasi gauss tetapi dengan membuat matriks SPL menjadi matriks eselon baris tereduksi. Salah satu contoh bentuk matriks eselon baris tereduksi adalah sebagai berikut:

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

Setelah diperoleh matriks SPL dalam bentuk matriks eselon baris tereduksi maka juga akan terbentuk sistem persamaan baru yang lebih sederhana. Melalui sistem persamaan tersebut akan dapat dicari nilai dari semua variabel dalam suatu SPL Dalam beberapa kasus nilai dari variabel dalam suatu SPL bisa didapat secara langsung dengan mudah tanpa perlu repot-repot menyelesaikan sistem persamaan baru yang terbentuk.

3. Metode Matriks Balikan

Misalkan suatu Sistem Persamaan Linear dituliskan sebagai Ax = B, dimana A, x, dan B merupakan sebuah matriks. Matriks x merupakan matriks yang merepresentasikan variabel yang terdapat dalam SPL sedangkan matriks A merepresentasikan nilai koefisien dari x dan matriks B merepresentasikan hasil dari suatu sistem persamaan. Untuk memperoleh nilai x dapat dilakukan dengan cara menginverse matriks A sehingga persamaan dapat diubah menjadi bentuk $x = A^{-1}B$. Dengan melakukan perkalian matriks biasa maka dapat diperoleh nilai dari seluruh variabel x dalam suatu SPL yang diberikan.

4. Kaidah Crammer

Misalkan suatu Sistem Persamaan Linear dituliskan sebagai Ax = B, dimana A, x, dan B merupakan sebuah matriks. Matriks x merupakan matriks yang merepresentasikan variabel yang terdapat dalam SPL sedangkan matriks A merepresentasikan nilai koefisien dari x dan matriks B merepresentasikan hasil dari suatu sistem persamaan. Kaidah Crammer dapat dilakukan dengan cara membuat matriks baru $A_1 \ldots A_n$ dengan ukuran matriks yang sama dengan matriks A. Matriks $A_1 \ldots A_n$ memiliki isi yang sama dengan matriks A hanya saja salah satu kolomnya yaitu kolom ke *n* akan diganti menjadi sama seperti isi dari matriks B. Untuk mencari nilai dari variabel x_n dapat dilakukan dengan cara membagi determinan dari matriks A_n dengan determinan dari matriks A seperti tertulis dalam persamaan di bawah ini:

$$x_n = \frac{\det(A_n)}{\det(A)}$$

B. Determinan Matriks

Determinan matriks dapat kita cari menggunakan metode reduksi baris dan metode ekspansi kofaktor.

1. Metode Reduksi Baris

Metode reduksi baris dilakukan dengan melakukan OBE pada matriks sampai diperoleh matriks segitiga (segitiga bawah atau atas). Misalnya seperti gambar di bawah ini.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \overset{\text{OBE}}{\sim} \begin{bmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \dots & a'_{3n} \\ 0 & 0 & 0 & a'_{nn} \end{bmatrix}$$

Maka misalkan P menyatakan banyaknya operasi OBE, determinannya yaitu

$$det(A) = (-1)^P a'_{11} a'_{22} \dots a'_{nn}$$

Jika selama OBE terdapat perkalian baris-baris matriks dengan k_1, k_2, …, k_m, maka determinannya yaitu

$$det(A) = \frac{(-1)^P a'_{11} a'_{22} \dots a'_{nn}}{k_1 k_2 \dots k_m}$$

2. Metode Ekspansi Kofaktor

Metode ekspansi kofaktor dilakukan dengan memilih satu baris atau kolom, lalu menjumlahkan masing-masing perkalian antara elemen dengan kofaktor pada baris atau kolom yang dipilih tadi. Misalkan matriks A berukuran n×n.

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

Misalkan
M_{ij} = minor entri aij atau determinan submatrix yang elemen-elemennya tidak berada pada baris i dan kolom j
$C_{ij} = (-1)^{i+j} M_{ij}$ = kofaktor entri a_{ij}
Maka determinan dapat dihitung dengan

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

Secara baris

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

Secara kolom

C. Balikan Matriks

Balikan matriks atau inverse matriks dapat dimanfaatkan dalam berbagai macam perhitungan lain. Cara untuk mencari balikan dari sebuah matriks dapat dilakukan dengan dua metode yaitu metode Gauss Jordan dan metode Adjoin.

1. Metode Gauss Jordan

Misalkan A adalah matriks persegi dengan ukuran n x n. Tahap awal untuk mencari balikan dari matriks A dengan metode ini adalah dengan cara membuat matriks augmented dengan bentuk [A | I] dimana matriks I merupakan identitas matriks yang mempunyai ukuran yang sama dengan matriks A. Setelah terbentuk matriks augmented maka bisa dilakukan eliminasi Gauss Jordan hingga terbentuk matriks augmented yang baru. Matriks augmented hasil dari eliminasi Gauss Jordan nantinya akan berbentuk $[I \mid A^{-1}]$ dimana matriks A^{-1} merupakan dari balikan dari matriks A yang sedang dicari.

2. Metode Adjoin

Misalkan A adalah matriks persegi dengan ukuran n x n dan C_{ij} adalah kofaktor entri a_{ij}. Untuk mencari matriks balikan dari Metode adjoin ini dapat dilakukan dengan rumus berikut:

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

Adjoin dari A sendiri merupakan transpose dari matriks kofaktor. Matriks kofaktor dari A memiliki bentuk sebagai berikut:

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

D. Interpolasi

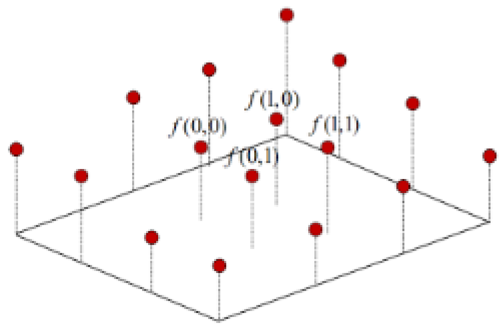
Interpolasi yang akan dibahas di sini adalah interpolasi polinom dan interpolasi bicubic spline.

1. Interpolasi Polinom

Interpolasi Polinom merupakan sebuah teknik matematika yang digunakan untuk mengaproksimasi atau memprediksi nilai-nilai di antara titik-titik data yang diketahui. Interpolasi polinom akan mengasumsikan pola data yang dimiliki mengikuti pola polinomial berderajat satu maupun berderajat tinggi. Jika terdapat $n+1$ buah titik maka persamaan polinom yang akan terbentuk adalah $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Tahap awal penyelesaian interpolasi ini adalah dengan cara membuat n buah sistem persamaan lanjar dari n buah titik yang diketahui. Misalkan diketahui sebuah titik (x_0, y_0) maka akan terbentuk suatu sistem persamaan lanjar $a_0 + a_1x_0 + a_2x_0^2 + \dots + a_n x_0^n = y_0$. Buat semua titik yang diketahui menjadi bentuk sistem persamaan tersebut lalu selesaikan keseluruhan sistem persamaan tersebut sehingga diperoleh nilai dari a_0, a_1, a_2, …, a_n. Untuk menyelesaikan sistem persamaan tersebut bisa dengan cara merepresentasikan sistem persamaan menjadi sebuah matriks kemudian menyelesaikannya dengan menggunakan metode eliminasi Gauss. Jika sudah diperoleh seluruh nilai konstanta a_0, a_1, a_2, …, a_n maka bisa dibentuk persamaan interpolasi $p_n(x)$ yang tepat. Melalui persamaan interpolasi yang telah terbentuk akan dapat diperkirakan nilai y di sembarang titik di dalam selang $[x_0, x_n]$.

2. Interpolasi Bicubic Spline

Interpolasi bicubic spline adalah salah satu metode interpolasi yang digunakan untuk mengaproksimasi fungsi dua dimensi (fungsi yang bergantung pada dua variabel independen) atau menginterpolasi data dalam bentuk grid. Metode ini berfokus pada interpolasi dalam dua dimensi, tetapi konsepnya dapat diperluas ke dimensi yang lebih tinggi. Interpolasi ini melibatkan konsep spline dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diketahui. Pendekatan ini menghasilkan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat dibandingkan jenis interpolasi lainnya. Untuk membuat pemodelan dari interpolasi ini akan diperlukan 16 titik dalam bidang dengan 4 titik di bagian pusat sebagai acuan utama dan 12 titik di sekitarnya sebagai nilai aproksimasi turunan dari keempat titik pusat tersebut.



Persamaan polinom yang diperlukan dalam interpolasi ini ada 4 yaitu sebagai berikut:

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

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

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

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

Dalam tugas ini akan diberikan nilai dari 16 titik yang diketahui kemudian melalui nilai dari 16 titik tersebut akan dibangun 16 bentuk sistem persamaan lanjar dengan bentuk seperti persamaan diatas. Keseluruhan sistem persamaan lanjar tersebut akan dibuat dalam bentuk matriks dengan persamaan penyelesaian matriks sebagai berikut:

$$y = Xa$$

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

Matriks X merupakan sebuah matriks yang bisa dibangun melalui 4 persamaan yang telah ditunjukkan di atas dimana matriks X menandakan nilai koefisien untuk setiap komponen a_{ij}. Sama seperti interpolasi polinom untuk menyelesaikan persoalan interpolasi bicubic spline ini perlu dicari nilai dari seluruh konstanta a_{ij}. Jika seluruh nilai konstanta telah didapat maka dapat terbentuk persamaan $f(x, y)$ yang tepat. Melalui persamaan tersebut dapat diaproksimasi nilai z di setiap titik (x,y) dimana x dan y dinormalisasi dalam rentang *[0, 1]*

E. Regresi Linier Berganda

Regresi linear berganda adalah sebuah metode statistik yang digunakan untuk memodelkan hubungan antara satu variabel dependen (atau variabel target) dan dua atau lebih variabel independen (atau prediktor). Tujuannya adalah untuk memahami dan memprediksi bagaimana variabel dependen berubah seiring dengan perubahan variabel independen. Pada dasarnya regresi linear menggunakan prinsip yang sama dengan interpolasi polinom. Pada regresi linear terdapat persamaan umum yang bisa digunakan untuk regresi linear berganda, yaitu sebagai berikut:

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

Dalam tugas ini akan diberikan sampel sejumlah m dengan jumlah peubah x adalah n, serta semua nilai-nilai x_{1i}, x_{2i}, …, x_{ni}, dan nilai y_i. Data yang diberikan tersebut kemudian akan digunakan untuk mencari nilai dari setiap β_i dengan menggunakan *Normal Estimation Equation for Multiple Linear Regression* yang memiliki bentuk sebagai berikut:

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

Sistem persamaan tersebut kemudian akan diubah kedalam bentuk matriks SPL dan dapat diselesaikan dengan metode eliminasi Gauss. Setelah didapat nilai dari setiap β_i maka akan dapat dibentuk persamaan regresi linear berganda yang tepat. Melalui persamaan tersebut dapat diaproksimasi suatu nilai y dari suatu nilai x_k.

BAB III IMPLEMENTASI PUSTAKA dan PROGRAM dalam JAVA

A. Implementasi Pustaka

Pustaka yang dibuat berisi Tipe Data Abstrak (Abstract Data Type) Matrix beserta fungsi-fungsi primitifnya dan Tipe Data Abstrak lainnya yang menggunakan ADT Matriks sebagai salah satu pondasinya. 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), serta Utils (untuk utility) yang memuat fungsi-fungsi yang dipakai berulang kali untuk memudahkan pembuatan pustaka dan program.

1. Matrix.java

Berisi ADT Matrix yang terdiri dari attribute dan juga fungsi/ prosedur primitifnya.

Properties

// Banyak baris
public int row;
// Banyak kolom matriks
public int col;
// Matriks yang disimpan
public double matrix[][];

Constructor
// konstruktor matriks public Matrix() public Matrix(int row, int col)
Class Methods

// Metode prosedur untuk menginisialisasi matriks dengan jumlah baris dan kolom yang diberikan.
public void initMatrix(int row, int col)
 	
// Metode untuk memeriksa apakah matriks tersebut adalah matriks persegi (jumlah baris = jumlah kolom).
public boolean isSquare()
 	
// Metode untuk mendapatkan representasi String dari matriks.
public String getMatrixString()
 	
// Metode procedure untuk membaca matriks dari input pengguna.
public void readMatrix()

```
// Metode procedure untuk membaca matriks persegi dari input pengguna.
public void readSquareMatrix()

// Metode procedure untuk membaca matriks dari file.
public void readMatrixFromFile()

// Metode procedure untuk membaca matriks dari input pengguna baris per baris.
public void readMatrixFromUserInput()

// Metode untuk memilih metode pembacaan matriks dari pengguna (keyboard atau file).
public void chooseReadMatrixMethodFromUserInput()

// Metode untuk menampilkan matriks.
public void displayMatrix()

// Metode untuk menampilkan matriks augmented dengan garis vertikal pada kolom tertentu.
public void displayAugmentedMatrix(int afterCol)

// Metode untuk memeriksa apakah matriks tersebut merupakan matriks eselon baris.
public boolean isEchelon()

// Metode untuk memeriksa apakah matriks tersebut dalam bentuk matriks eselon baris
tereduksi. Mengembalikan true atau false
public boolean isReducedEchelon()

// Metode procedure untuk menukar dua baris matriks.
public void swapRow(int baris1, int baris2)

// Metode procedure untuk mengubah matriks menjadi matriks segitiga.
public int toMatrixSegitiga()

// Metode untuk menghitung determinan matriks dengan metode yang dipilih (RowReduction
atau CofactorExpansion).
public double getDeterminant(DeterminantMethod method)

// Metode untuk menghitung determinan matriks dengan metode default (CofactorExpansion).
public double getDeterminant()

// Metode untuk mendapatkan indeks kolom yang tidak seluruhnya berisi angka nol dalam
rentang baris tertentu.
public int getColomnNotEntirelyZero(int startRow, int endRow)

// Metode untuk membuat salinan matriks.
public Matrix getCopyMatrix()
```

```
// Metode untuk mengubah matriks menjadi matriks eselon baris.
public void toRowEchelon()

// Metode untuk mengalikan matriks dengan matriks lain.
public void multiply(Matrix m)

// Metode untuk mengalikan matriks dengan matriks dalam bentuk array 2D.
public void multiply(double[][] m)

// Metode untuk mengubah matriks menjadi matriks eselon baris tereduksi.
public void toReducedRowEchelon()

// Metode untuk menghapus -0 dan menggantinya dengan 0 dalam matriks.
public void normalizeMatrix()

// Metode untuk mendapatkan matriks invers dengan metode yang dipilih (GaussJordan atau
Adjoin).
public Matrix getInverse(InverseMethod method)

// Metode untuk mendapatkan matriks invers dengan metode default (Adjoin).
public Matrix getInverse()

// Metode untuk mendapatkan matriks kofaktor berdasarkan baris dan kolom tertentu.
public Matrix getCofactor(int row, int col)

// Metode untuk mendapatkan matriks adjoin.
public Matrix getAdjoin()

// Metode untuk mentransposisi matriks.
public Matrix transpose()

// Metode untuk mengalikan matriks dengan konstanta.
public Matrix MultiplyByConst(double x)

// Metode untuk mengalikan matriks dengan matriks lain.
public Matrix multiplyBy(Matrix m)
```

2. 1. Class SPL

Berisi ADT SPL

Properties
// Kesepakatan: Ax = B; // Matriks A

```
public Matrix A;
// Variabel x
public Parametric x[];
// Matriks B
public Matrix B;
// Matriks Augmented
public Matrix augmentedMatrix;
private Map<Integer, Integer> parametricLinking = new HashMap<Integer, Integer>();
// Metode SPL
private SPLMethod method = SPLMethod.Gauss;
// Tampilkan proses atau tidak
private boolean showProcess = false;

// Metode-metode penyelesaian SPL
public enum SPLMethod {
   Gauss, GaussJordan, Inverse, Cramer
}
```

Constructor

```
// Konstruktor
public SPL(int countEquations, int countVariables)

public SPL()
```

Class Methods

```
// Membaca variabel dari input pengguna
public void readVariablesFromUserInput()

// Membaca variabel dari file teks
public void readVariablesFromTextFile()

// Menyelesaikan SPL
public void solve()

// Menampilkan solusi ke file
public void displaySolutionToFile()

// Menampilkan solusi ke terminal
public void displaySolutionToTerminal()

// Menampilkan solusi
public void displaySolution()
```

```
// Mengatur metode penyelesaian SPL
public SPL setMethod(SPLMethod method

// Mengatur tampilan proses
public SPL setShowProcess(boolean showProcess)

// Mengatur matriks
public SPL setMatrix(Matrix matrix)


// Mengisi SPL dari matriks
public void fromMatrix(Matrix m)

// Mendapatkan salinan SPL
public SPL getCopySPL()

// Menyelesaikan SPL tanpa merubah atribut apapun
public void solve(boolean stay)


// Menampilkan semua persamaan
public void showEquations()


// Mengatur solusi SPL
public void setSolution()

// Mengatur solusi SPL dengan status khusus
public void setSolution(boolean isHasSolution, String error)

// Memeriksa apakah SPL memiliki solusi
public boolean hasSolution(boolean stay)


// Memeriksa apakah SPL memiliki solusi
public boolean hasSolution(boolean stay)

// Menampilkan solusi SPL
public void showSolution(boolean stay)

// Menampilkan matriks augmented
public void displayAugmentedMatrix()

// Mengambil string yang berisi solusi SPL
public String getSolutionString()
```

2. 2 Parametric.java

Berisi ADT Parametric

Properties

// konstanta c
private double c
 	
// array nilai-nilai variabel parametrik sementara
private double temporaryParametricVariables[]
 	
// status apakah nilai variabel parametrik telah ditugaskan atau tidak
private boolean isAssigned

Constructor
// Inisialisasi variabel parametrik dengan jumlah variabel yang digunakan dalam SPL public Parametric(int countVariable)
Class Methods

// Mengambil elemen parametrik sementara dengan indeks tertentu
public double getTmpParamElmt(int idx)
 	
// Mengambil seluruh elemen parametrik sementara
public double[] getTmpParamElmt()
 	
// Mengatur elemen parametrik sementara dengan indeks tertentu ke nilai tertentu
public void setTmpParamElmt(int idx, double val)
 	
// Mendapatkan konstanta yang terkait dengan variabel parametrik
public double getConstant()
 	
// Mengatur nilai konstanta variabel parametrik
public void setConstant(double val)
 	
// Mendapatkan status apakah nilai variabel parametrik telah ditugaskan atau tidak
public boolean getIsAssigned()

3. Interpolasi.java

Berisi ADT Interpolasi

Properties
// data masukan titik akan diubah ke dalam bentuk matriks spl dan disimpan dalam

```
// variabel ini
private Matrix matrix

// matriks representasi dari masukan GUI
private Matrix inputMatrix

// nilai x yang mau diaproksimasi hasilnya
private double x

// Objek spl untuk membantu menyelesaikan persoalan interpolasi
private SPL spl

// menyimpan bentuk persamaan interpolasi yang didapatkan
private String persamaan

// hasil dari nilai p(x) untuk nilai x yang telah dimasukan pengguna (nilai x
// yang ingin diaproksimasi hasilnya)
private double result

// boolean untuk penanda apakah nilai x yang mau diaproksimasi nilai f(x) nya berada di dalam range interpolasi yang sesuai atau tidak
private boolean isValidX

//range untuk nilai x yang valid
private double[] batas
```

Constructor

```
// Inisialisasi interpolasi
public interpolasi()
```

Class Methods

```
// prosedur untuk mengambil data dari masukan user melalui terminal, data
// masukan kemudian akan diolah dan disesuaikan menjadi bentuk matriks spl
public void readVariablesFromUserInput()

// prosedur untuk mengambil data dari masukan user melalui file, data masukan
// kemudian akan diolah dan disesuaikan menjadi bentuk matriks spl
public void readVariablesFromTextFile()

// prosedur untuk menyelesaikan persoalan interpolasi polinom, hasil dari
// prosedur ini adalah persamaan interpolasi dan hasil aproksimasi untuk nilai x
// yang dimasukan
public void solve()
```

```
// prosedur untuk menampilkan solusi dari interpolasi polinom kepada pengguna
public void displaySolution()

// prosedur untuk menampilkan solusi dari interpolasi polinom kepada pengguna
// melalui file (solusi akan dituliskan di file)
public void displaySolutionToFile()

// prosedur untuk menampilkan solusi dari interpolasi polinom kepada pengguna
// melalui terminal
public void displaySolutionToTerminal()

// fungsi untuk menetapkan nilai matrix spl dari inputMatrix GUI
public Interpolasi setMatrix(Matrix inputMatrix)

// setter untuk atribut x
public Interpolasi setX(double newX)

// getter untuk atribut x
public double getX()

// getter untuk atribut inputMatrix
public Matrix getInputMatrix()

// getter untuk atribut solution (atribut dari parent class)
public String getSolutionString()

// prosedur untuk menetapkan nilai dari inputMatrix dari masukan file (GUI)
public void setVariablesFromFile(File file) throws Exception
```

4. BicubicSplineInterpolation.java

Berisi ADT BicubicSplineInterpolation

Properties

```
//masukan dari GUI akan direpresentasikan sebagai sebuah matriks
private Matrix inputMatrix = new Matrix(4, 4);

//matrix X yang merupakan koefisien dari a
private Matrix matrixX;

//matriks yang berisi nilai dari f, fx, fy, dan fxy yang diberikan oleh pengguna
private Matrix matrixF;

// matriks a yang akan dicari nilainya
private Matrix matrixA;
```

//balikan dari matrix x
private Matrix inverseX;
 	
//masukan sebuah nilai a dan b dari pengguna untuk nantinya akan diaproksimasi nilai dari
//f(a,b)
private double a;
private double b;
 	
//hasil dari aproksimasi f(a,b)
private double result = 0;

Constructor
// Inisialisasi BicubicSplineInterpolation public BicubicSplineInterpolation()
Class Methods

//prosedur untuk mengambil data dari masukan user melalui terminal, data masukan kemudian
//akan diolah dan disesuaikan menjadi bentuk matriks spl
public void readVariablesFromUserInput()
 	
// prosedur untuk mengambil data dari masukan user melalui file, data masukan kemudian
// akan diolah dan disesuaikan menjadi bentuk matriks spl
public void readVariablesFromTextFile()
 	
//prosedur untuk menyelesaikan persoalan interpolasi bicubic spline, hasil dari prosedur ini
// adalah nilai dari f(a,b)
public void solve()
 	
//prosedur untuk menampilkan solusi dari interpolasi bicubic spline kepada pengguna melalui
//file (solusi akan dituliskan di file)
public void displaySolutionToFile()
 	
//prosedur untuk menampilkan solusi dari interpolasi bicubic spline kepada pengguna melalui
//terminal
public void displaySolutionToTerminal()
 	
//prosedur untuk menampilkan solusi dari interpolasi bicubic spline kepada pengguna
public void displaySolution()
 	
//fungsi untuk meng-genarate matrix x
public void setMatrix(Matrix m)
 	
//setter untuk atribut getInputMatrix

public Matrix getInputMatrix()
 	
//setter untuk atribut inputMatrix sekaligus untuk menguabah inputMatrix menjadi matrixF
public void setInputMatrix(Matrix m)
 	
//setter untuk atribut a dan b
public void setX(double[] x)
 	
//getter untuk atribut a dan b
public double[] getX()
 	
//prosedur untuk menetapkan nilai dari inputMatrix dari masukan file (GUI)
public void setVariablesFromFile(File file) throws Exception

5. MultipleLinearRegression.java

Berisi ADT Parametric

Properties

// Matriks input yang mewakili data pengguna
public Matrix matrix;
 	
// Matriks untuk regresi
private Matrix mlrMatrix;
 	
// Metode Simplex untuk penyelesaian
private SPL spl;
 	
// Nilai yang akan diestimasi
private double x[];
 	
// Persamaan sebagai string
private String persamaan = "";
 	
// Hasil dari estimasi
private double result = 0;
 	
// Flag untuk menentukan apakah akan mencetak ke file
private boolean isPrintFile = false;

Constructor
// Inisialisasi MultipleLinearRegression public MultipleLinearRegression()

Class Methods

// Metode untuk membaca variabel dari input pengguna
public void readVariablesFromUserInput()
 	
// Metode untuk membaca variabel dari file teks
public void readVariablesFromTextFile()
 	
// Metode untuk menyelesaikan masalah regresi
public void solve()
 	
// Metode untuk menampilkan solusi ke dalam file
public void displaySolutionToFile()
 	
// Metode untuk menampilkan solusi ke terminal
public void displaySolutionToTerminal()
 	
// Metode tambahan untuk menginisialisasi objek dengan matriks
public void init(Matrix matrix)
 	
// Metode tambahan untuk menampilkan solusi
public void display()
 	
// Metode setter untuk nilai 'x'
public void setX(double[] x)
 	
// Metode getter untuk nilai 'x'
public double[] getX()

6. SimpleImage.java

Berisi ADT SimpleImage

Properties

// Matriks yang menyimpan nilai RGB dari suatu foto (index 0 RED, index 1 GREEN, index 2
// BLUE)
private Matrix RGBMatrix[];
 	
// Matriks yang menyimpan nilai grayscale dari suatu foto
private Matrix grayScaleMatrix;
 	
// Matriks hasil perkalian matrix I dan matrix X inverse
private Matrix matrixXI;
 	
// Matriks x yang merupakan koefisien dari a (bicubic spline interpolation)
private Matrix matrixX;

```
// Matriks dengan ukuran panjang dan lebar ditambah 2 untuk mempermudah saat perhitungan
// interpolasi
private Matrix RGBMatrixHandleSides[];
private Matrix grayScaleMatrixHandleSides;

// Lebar foto
private int width;

// Tinggi foto
private int height;

// Balikan dari matrix x
private Matrix matrixXInv;

// Matrix D yang merupakan koefisien dalam persamaan I, Ix, Iy, dan Ixy
private Matrix matrixD;

// Pilihan warna untuk foto yang akan diperbesar
private ColorOptions color;

// Berapa kali perbesaran foto yang diinginkan
private double factor;

// Nama file dari foto yang akan diperbesar
private String fileName;

// Enum untuk pilihan warna yang tersedia
public enum ColorOptions {
  RED, GREEN, BLUE, GRAYSCALE, NORMAL
}
```

Constructor

```
// Inisialisasi SimpleImage
public SimpleImage(String fileName) throws IOException
```

Class Methods

```
// Setter factor
public void setFactor(double factor)

// Setter colorOptions
public void setOptions(ColorOptions color)
```

```
// Getter RGBMatrix untuk warna merah
public Matrix getRedMatrix()

// Getter RGBMatrix untuk warna hijau
public Matrix getGreenMatrix()

// Getter RGBMatrix untuk warna biru
public Matrix getBlueMatrix()

// Getter grayScaleMatrix
public Matrix getGrayscaleMatrix()

// Prosedur untuk memulai proses perbesaran foto
public void EnlargeImage() throws IOException

// Fungsi yang digunakan untuk memperbesar suatu matrix yang merepresentasikan sebuah
foto
private Matrix EnlargedMatrix(Matrix m, double factor)

// Prosedur untuk membantu proses inisialisasi objek SimpleImage
private void init()

// Prosedur untuk meng-generate matrix X
public void setMatrixX()

// Fungsi untuk meng-generate matrix D
private void setMatrixD()

// Getter matrixD
public Matrix getMatrixD()

// Fungsi untuk mendapatkan nilai a pada sebuah sistem persamaan interpolasi bicubic spline
// Nilai a akan disimpan dalam bentuk matrix
private Matrix getMatrixA(int tlx, int tly, ColorOptions color)

// Fungsi untuk menghitung aproksimasi nilai f(x,y) dengan nilai a berasal dari matrix a yang
//dikirimkan lewat parameter
private double interpolate(double x, double y, Matrix a)
```

7. Solvable.java

Berisi ADT Solvable

Properties

// Data solusi dari suatu permasalahan
protected String solution = "";
 	
// Setter solution
public String getSolutionString()
 	
// Status dari suatu persoalan
protected State state;
 	
// Boolean untuk menyimpan data apakah solusi perlu dituliskan di file atau tidak
private boolean isPrintFile = false;

Constructor
// Inisiasi Solvable Public Solvable()
Class Methods

// Prosedur untuk menanyakan kepada pengguna mau memilih masukan dari mana
public void chooseReadVariablesMethodFromUserInput() {}
 	
// Metode untuk membaca variabel dari input pengguna
public void readVariablesFromUserInput(){};
 	
// Metode untuk membaca variabel dari file teks
public void readVariablesFromTextFile(){};
 	
// Enumerator untuk menunjukkan status Solvable
protected enum State {
    UninitializedVariable, Unsolved, Solved
}
 	
// Getter untuk state
protected State getState()
 	
// Setter untuk state
protected void setState(State newState) {}
 	
// Prosedur untuk menyelesaikan permasalahan
public void solve(){};
 	
// Prosedur untuk menampilkan solusi
public void displaySolution(){};

8. Input.java

Berisi ADT Input

Properties

// Scanner untuk input dari pengguna
private static Scanner userInput = new Scanner(System.in);
 	
// Indikator kesuksesan input
private static boolean success = false;

Constructor
// Inisialisasi Input public Input();
Class Methods

// Menutup scanner input.
public static void closeScanner();
 	
// Mendapatkan objek Scanner untuk input.
public static Scanner getScanner();
 	
// Input dengan validasi integer bersamaan dengan validasi tambahan.
public static int getInt(String errorMessage, Function<Integer, Boolean> validator);
 	
// Input dengan validasi double bersamaan dengan validasi tambahan.
public static double getDouble(String errorMessage, Function<Double, Boolean> validator);
 	
// Input tanpa validasi integer dengan pesan kesalahan default.
public static int getInt();
 	
// Input tanpa validasi double dengan pesan kesalahan default.
public static double getDouble();

9. Utils.java

Berisi ADT Utils

Properties

// Nilai toleransi
private static double tolerance = 1e-6;
 	
// Scanner untuk input dari pengguna
private static Scanner userInput = new Scanner(System.in);

Constructor
// Inisialisasi Utils public Utils()
Class Methods

// Mendapatkan input integer dari pengguna dengan pesan khusus dan pesan kesalahan jika
// tipe data tidak sesuai.
public static int getIntInput(String message, String mismatchMessage)
 	
// Mendapatkan nilai toleransi yang digunakan untuk perbandingan nilai double.
public static double getTolerance()
 	
// Memeriksa apakah dua nilai double hampir sama dengan toleransi yang diberikan.
public static boolean isEqual(double val1, double val2)
 	
// Memeriksa apakah dua nilai double tidak hampir sama dengan toleransi yang diberikan.
public static boolean isNotEqual(double val1, double val2)
 	
// Menambah atau mengurangkan elemen-elemen dalam dua larik double.
public static void plusMinusList(double[] l1, double[] l2, boolean plus)
 	
// Mengubah l1 menjadi l1+l2*multiplier atau l1-l2*multiplier.
public static void plusMinusList(double[] l1, double[] l2, boolean plus, double multiplier)
 	
// Mengalikan semua elemen dalam larik double dengan suatu nilai.
public static void multiplyListBy(double[] l, double multiplier)
 	
// Menggabungkan dua larik integer dengan operasi OR atau AND.
public static void andOrList(Integer[] l1, double[] l2, boolean and)
 	
// Menulis teks ke dalam file dengan nama file tertentu.
public static void printFile(String s, String fileName)

10. Main.java

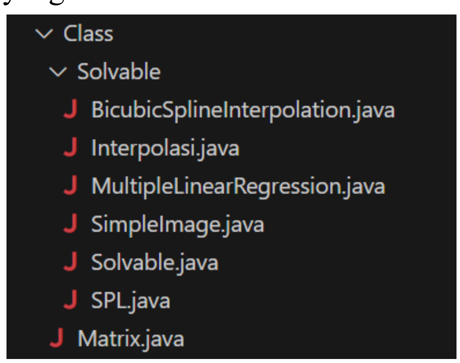
Berisi ADT Main

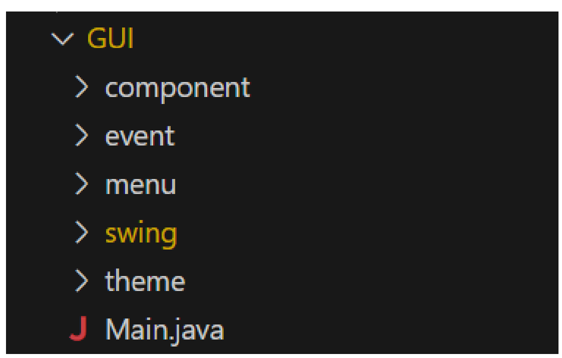
Properties
Constructor
Class Methods

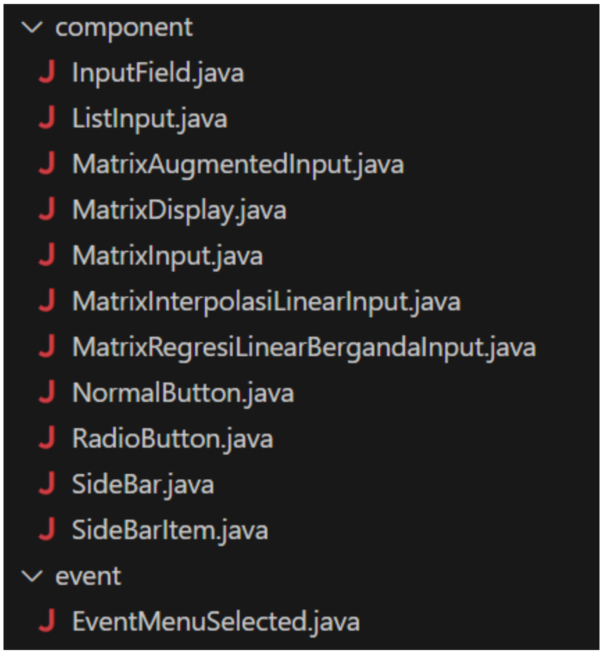
```
// Metode utama yang menjalankan aplikasi Library Matrix.
public static void main(String[] args)

// Menghandle menu "Coba lagi?" dan menjalankan instruksi yang diberikan.
static void handleCobaLagi(Runnable currentInstruction)

// Menghandle opsi menu "Interpolasi Bicubic Spline."
static void handleBicubic()

// Menghandle opsi menu "Interpolasi polinom."
static void handleInterpolasi()

// Menghandle opsi menu "Determinan."
static void handleDeterminan()

// Menghandle opsi menu "Matrix Balikan."
static void handleMatrixBalikan()

// Menghandle opsi menu "Regresi Linier Berganda."
static void RegresiLinierBerganda()

// Menghandle opsi menu "Sistem Persamaaan Linier."
static void handleSPL()
```

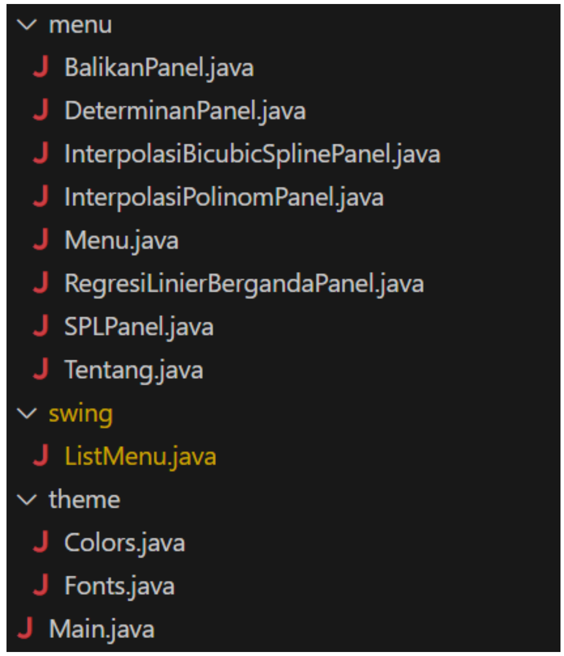
B. Struktur Class

Struktur dari class yang dibuat terbagi-bagi sesuai dengan tugasnya. Secara garis besar, class dibagi ke dalam 2 folder yaitu class untuk pustaka dari matrix dan GUI untuk simulasi dari yang ada di dalam Class.









Untuk folder Class terdapat pustaka dari matriks untuk implementasi dari operasi matrix pada umumnya misalnya untuk determinan dan balikan matriks. Di dalam Solvable terdapat masing-masing class untuk menyelesaikan masalah-masalah yang dapat dicari solusinya dengan matriks. SPL merupakan class untuk menyelesaikan Sistem Persamaan Linier. Class ini banyak dimanfaatkan oleh class class lain untuk menemukan solusi masing-masing. Interpolasi digunakan untuk mencari solusi dari persoalan interpolasi. BicubicSplineInterpolasion digunakan untuk mencari solusi dari persoalan Bicubic Spline Interpolation. MultipleLinearRegression digunakan untuk menyelesaikan masalah Multiple Linear Regression. Sedangkan SimpleImage untuk permasalahan gambar. Semua class ini diwariskan dari class Solvable.

Untuk folder GUI dibagi lagi menjadi berbagai hal untuk mengimplementasikan semua yang ada pada folder Class. Component berisi komponen-komponen yang digunakan untuk menampilkan user interface. Event untuk menerapkan event yang digunakan oleh class lain. Menu merupakan landasan dari tampilan yang akan berisi class pada component. Swing berisi alat bantu untuk membuat menu list. Theme merupakan pallete yang bisa digunakan untuk menghias GUI. Sedangkan Main merupakan entry point dari GUI.

BAB IV EKSPERIMEN

Jika masukan dalam bentuk sebuah file maka file masukan tersebut harus diletakan pada folder test/input. Untuk keluaran hasil dalam bentuk file, file tersebut akan ada di folder test/output.

A. Studi Kasus Sistem Persamaan Linear

Untuk pengujian program dalam penyelesaian SPL, diberikan beberapa tes:

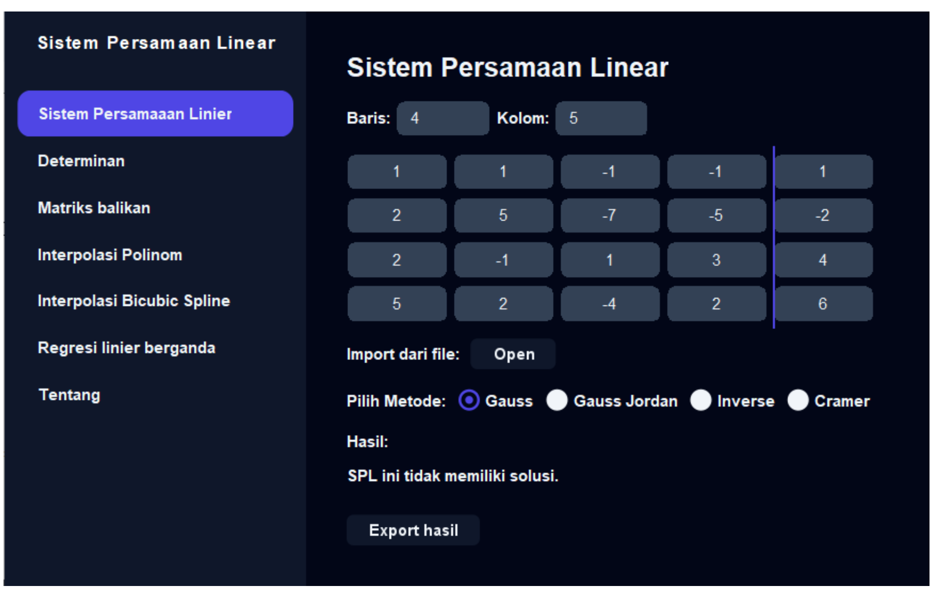
1. Menentukan solusi SPL dari:

   a.

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

   Bagian akhir penyelesaian program:

```
--> Setelah itu, ubah ke bentuk sistem persamaan linear.

(1) . . .   1.0000x1 +1.0000x2 -1.0000x3 -1.0000x4 = 1.0000
(2) . . .   1.0000x2 -1.6667x3 -1.0000x4 = -1.3333
(3) . . .   1.0000x3 -1.0000x4 = 1.0000
(4) . . .   0.0000 = 1.0000

--> Cek apakah ada suatu persamaan yang tidak konsisten.
***) Dapat dilihat bahwa persamaan ke-4 tak konsisten, karena
 tidak akan ada
***) nilai variabel yang memenuhi persamaan tsb.


--> Kesimpulannya, SPL tersebut tidak memiliki solusi.
Program selesai.
```



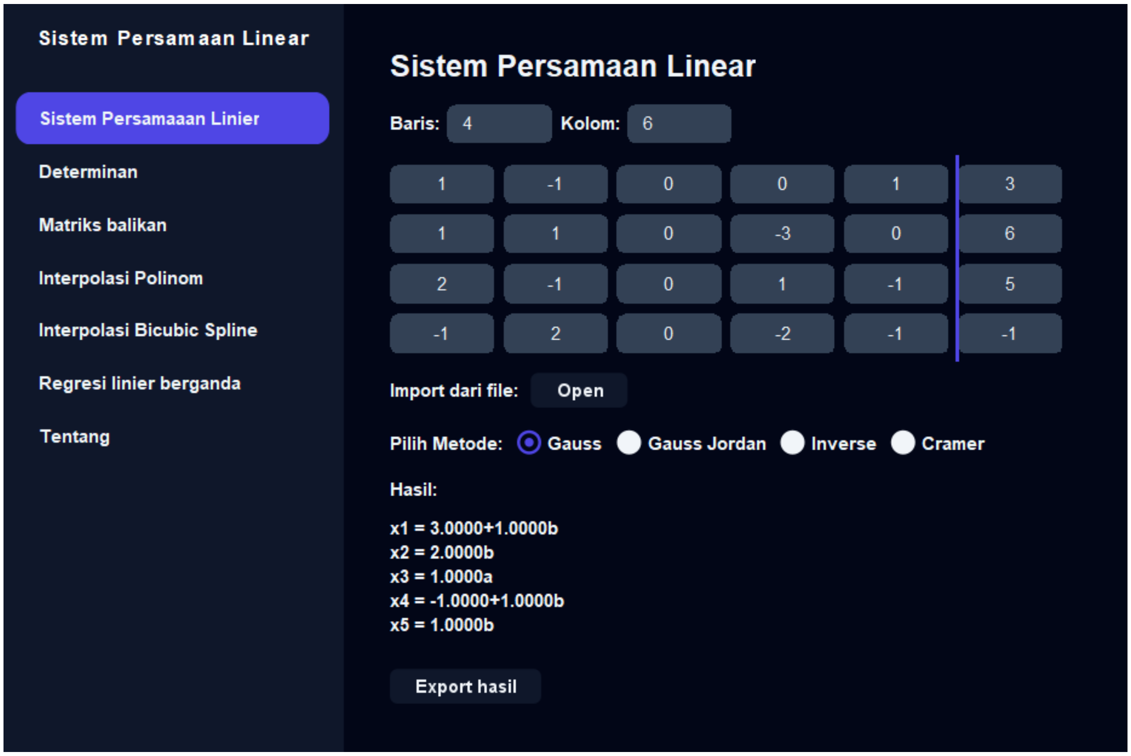
Analisis: prosedur penyelesaian SPL menggunakan gauss bisa menangani suatu persamaan yang tidak konsisten, alhasil dapat menangani apabila SPL tidak memiliki solusi.

b.

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

Potongan akhir penyelesaian SPL program:

```
--> Setelah itu, ubah ke bentuk sistem persamaan linear.

(1) . . .   1.0000x1 -1.0000x2 +1.0000x5 = 3.0000
(2) . . .   1.0000x2 -1.5000x4 -0.5000x5 = 1.5000
(3) . . .   1.0000x4 -1.0000x5 = -1.0000
(4) . . .   0.0000 = 0.0000

--> Cek apakah ada suatu persamaan yang tidak konsisten.
  | Semua persamaan konsisten, lanjut ke step berikutnya.

--> Sekarang, selesaikan persamaan satu persatu mulai dari
persamaan yang paling bawah.

 -> dari persamaan :  (4) . . .   0.0000 = 0.0000
 -> dari persamaan :  (3) . . .   1.0000x4 -1.0000x5 = -1.0000

 -> dari persamaan :  (2) . . .   1.0000x2 -1.5000x4 -0.5000x5 = 1.5000
  | Kita dapati:
 | Jadikan x3 sebagai base parametrik, katakanlah c
 | x2 = +2.0000e

 -> dari persamaan :  (1) . . .   1.0000x1 -1.0000x2 +1.0000x5 = 3.0000
  | Kita dapati:
 | x1 = 3.0000 +1.0000e

--> Wrapping up, sekaligus merapihkan penamaan variabel, solusi dari SPL tersebut adalah

x1 = 3.0000+1.0000b
x2 = 2.0000b
x3 = 1.0000a
x4 = -1.0000+1.0000b
x5 = 1.0000b
```



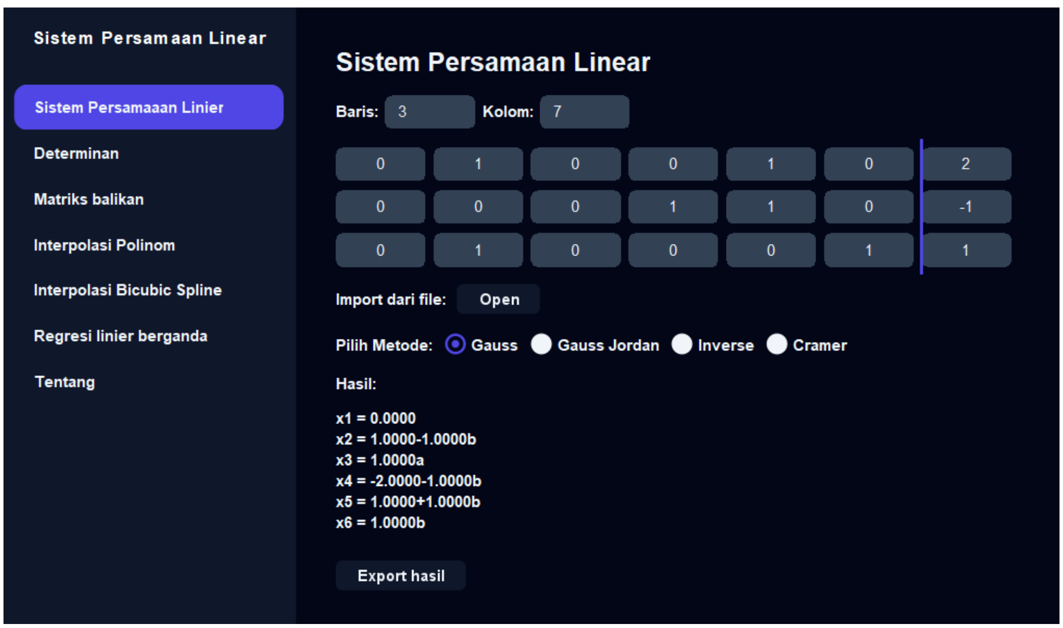
Analisis: prosedur penyelesaiaan SPL menggunakan metode gauss dapat menangani SPL yang memiliki solusi banyak.

c.

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

Potongan akhir penyelesaian SPL program:

```
--> Setelah itu, ubah ke bentuk sistem persamaan linear.

(1) . . .   1.0000x2 +1.0000x5 = 2.0000
(2) . . .   1.0000x4 +1.0000x5 = -1.0000
(3) . . .   1.0000x5 -1.0000x6 = 1.0000

--> Cek apakah ada suatu persamaan yang tidak konsisten.
  | Semua persamaan konsisten, lanjut ke step berikutnya.

--> Sekarang, selesaikan persamaan satu persatu mulai dari
persamaan yang paling bawah.

 -> dari persamaan :  (3) . . .   1.0000x5 -1.0000x6 = 1.0000

  | Kita dapati:
 | Jadikan x6 sebagai base parametrik, katakanlah f
 | x5 = 1.0000 +1.0000f

 -> dari persamaan :  (2) . . .   1.0000x4 +1.0000x5 = -1.0000
  | Kita dapati:
 | x4 = -2.0000 -1.0000f

 -> dari persamaan :  (1) . . .   1.0000x2 +1.0000x5 = 2.0000

  | Kita dapati:
 | x2 = 1.0000 -1.0000f

--> Wrapping up, sekaligus merapihkan penamaan variabel, solusi dari SPL tersebut adalah

x1 = 0.0000
x2 = 1.0000-1.0000b
x3 = 1.0000a
x4 = -2.0000-1.0000b
x5 = 1.0000+1.0000b
x6 = 1.0000b
```



Analisis: program penyelesaian SPL menggunakan Gauss yang telah dibuat dapat menangani studi kasus apabila solusi dari SPL yang diberikan memiliki lebih dari 1 bentuk x berupa parametrik.

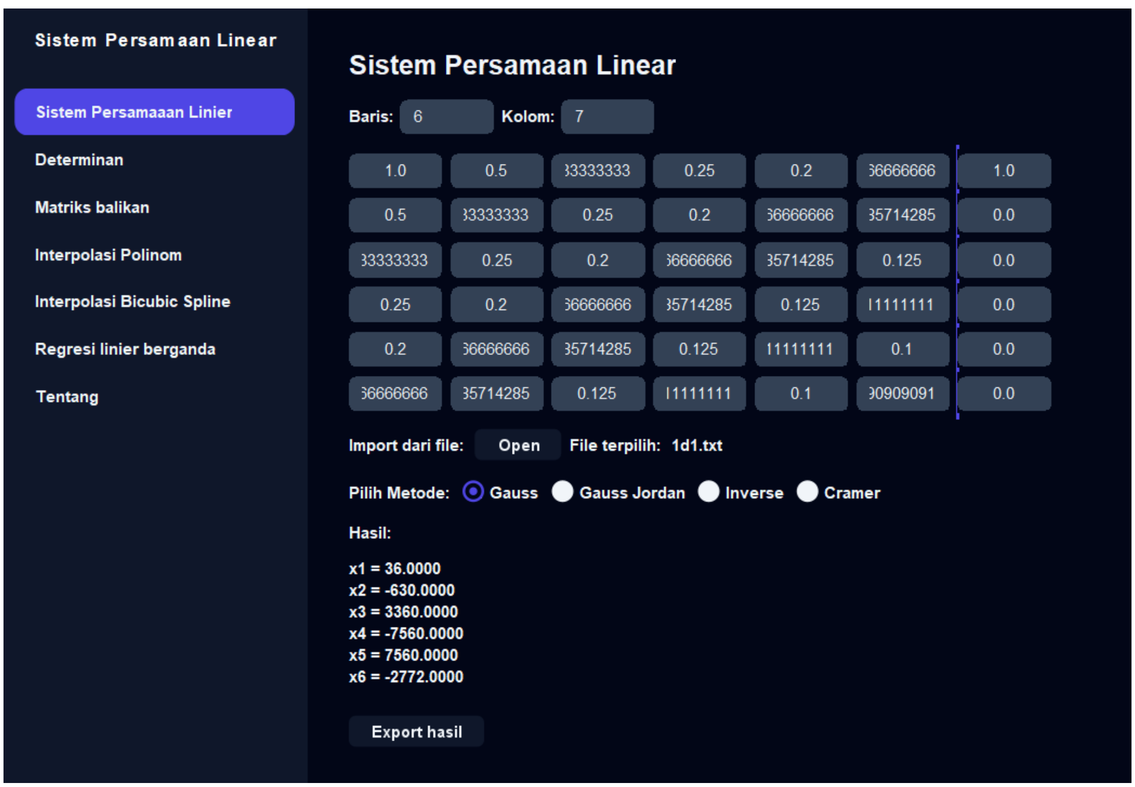
d.

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

Potongan akhir penyelesaian program:

1. Untuk n=6,

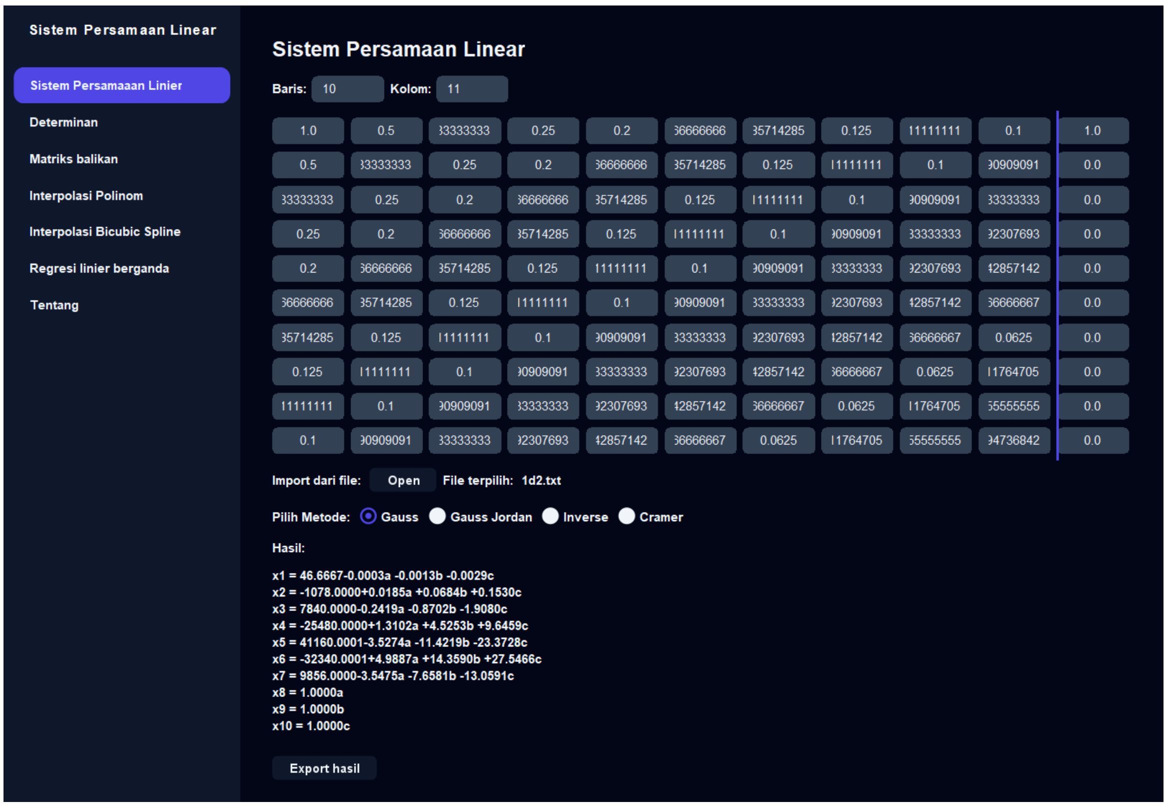
```
--> Wrapping up, sekaligus merapihkan penamaan variabel, solusi dari SPL tersebut adalah

x1 = 36.0000
x2 = -630.0000
x3 = 3360.0000
x4 = -7560.0000
x5 = 7560.0000
x6 = -2772.0000
```



Analisis: Program penyelesaian SPL menggunakan metode gauss dapat menyelesaikan matriks hilbert khususnya untuk. Namun, apabila digunakan metode matriks balikan atau metode Crammer, akan menghasilkan solusi yang berbeda. Kemungkinan hal ini disebabkan karena determinan dari matriks Hilbert untuk n = 6.

2. Untuk n = 10,

```
--> Wrapping up, sekaligus merapihkan penamaan variabel, solusi dari SPL tersebut adalah

x1 = 46.6667-0.0003a -0.0013b -0.0029c
x2 = -1078.0000+0.0185a +0.0684b +0.1530c
x3 = 7840.0000-0.2419a -0.8702b -1.9080c
x4 = -25480.0000+1.3102a +4.5253b +9.6459c
x5 = 41160.0001-3.5274a -11.4219b -23.3728c
x6 = -32340.0001+4.9887a +14.3590b +27.5466c
x7 = 9856.0000-3.5475a -7.6581b -13.0591c
x8 = 1.0000a
x9 = 1.0000b
x10 = 1.0000c
```



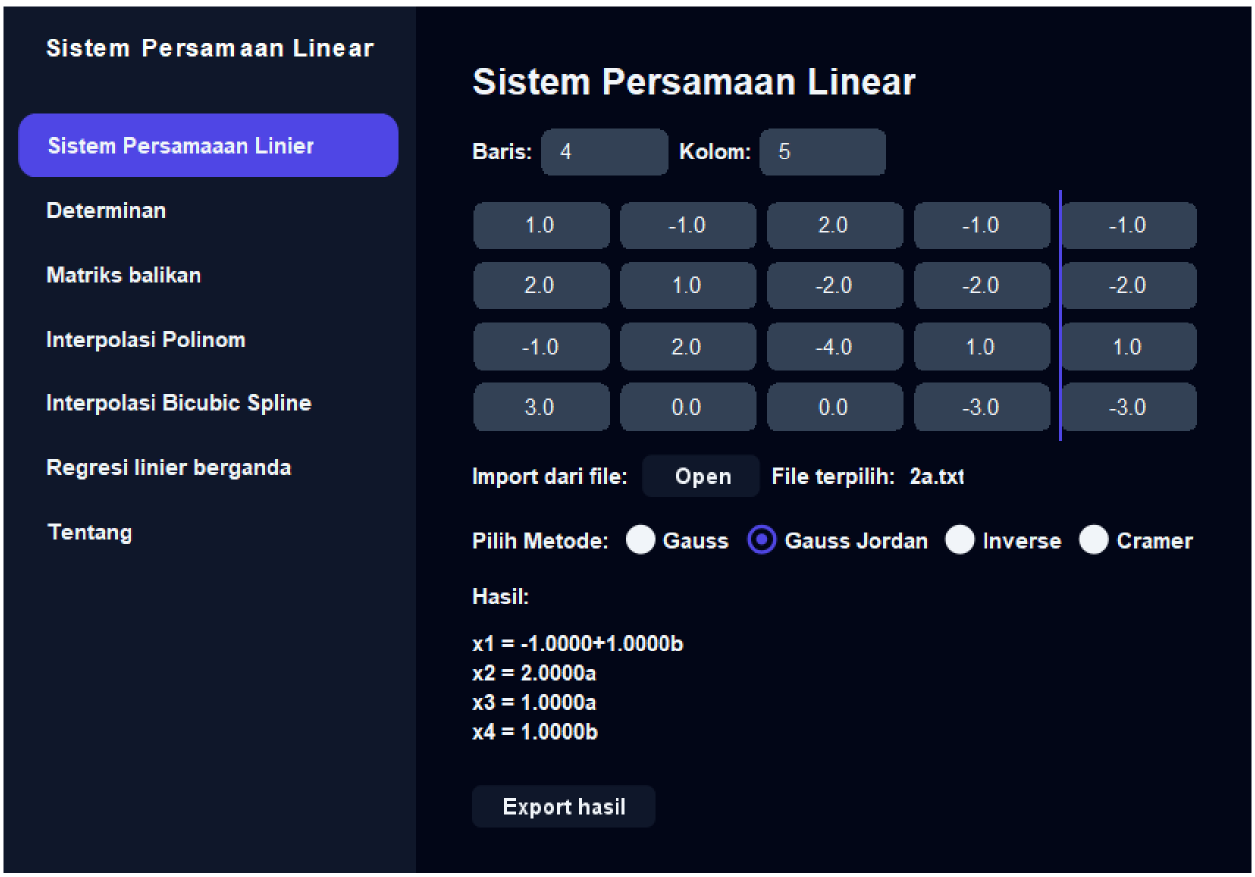
Analisis: Penyelesaian SPL tersebut menggunakan metode Gauss menghasilkan jawaban di atas. Seharusnya, solusi dari SPL tersebut berupa solusi tunggal. Kemungkinan, hal ini disebabkan karena saat melakukan eliminasi gauss ke baris eselon, ada suatu nilai yang sangat kecil padahal tak nol yang dianggap nol karena kami menggunakan toleransi sebesar 10^(-8) untuk menghindari floating point error.

2. Akan dicari solusi dari 2 SPL berbentuk augmented matriks sebagai berikut

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

Solusi:

```
--> Wrapping up, sekaligus merapihkan penamaan variabel, solusi dari SPL tersebut adalah

x1 = -1.0000+1.0000b
x2 = 2.0000a
x3 = 1.0000a
x4 = 1.0000b
```



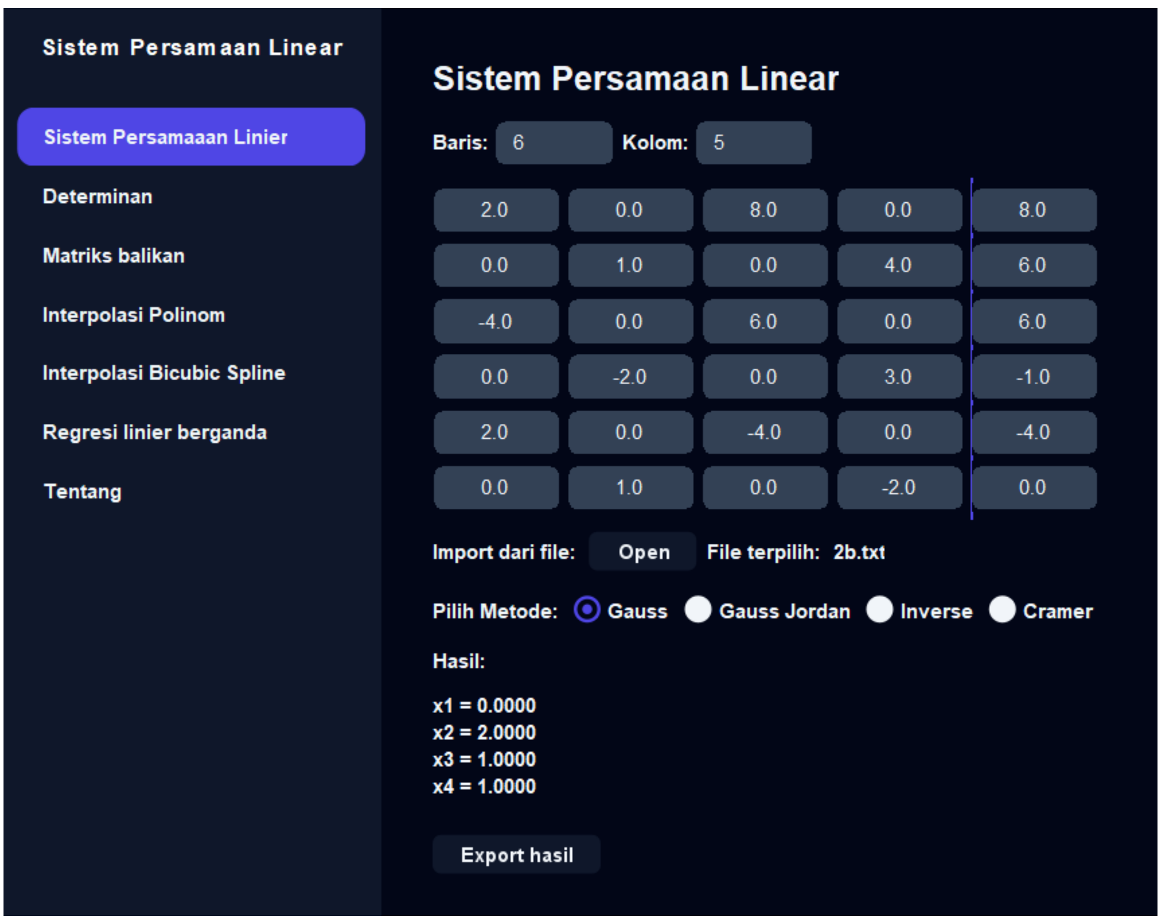
Analisis:

SPL ini diselesaikan dengan metode eliminasi Gauss-Jordan dan ternyata didapat solusi berupa persamaan parametrik. SPL tersebut memiliki solusi parametrik karena dalam bentuk matriks eselon baris tereduksi terdapat baris yang berisi angka 0 semua. SPL ini juga tidak akan bisa diselesaikan dengan metode matriks balikan ataupun metode Cramer karena matriks masukan ini memiliki determinan 0.

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

Solusi:

```
--> Wrapping up, sekaligus merapihkan penamaan variabel, solusi dari SPL tersebut adalah

x1 = 0.0000
x2 = 2.0000
x3 = 1.0000
x4 = 1.0000
```



Analisis:

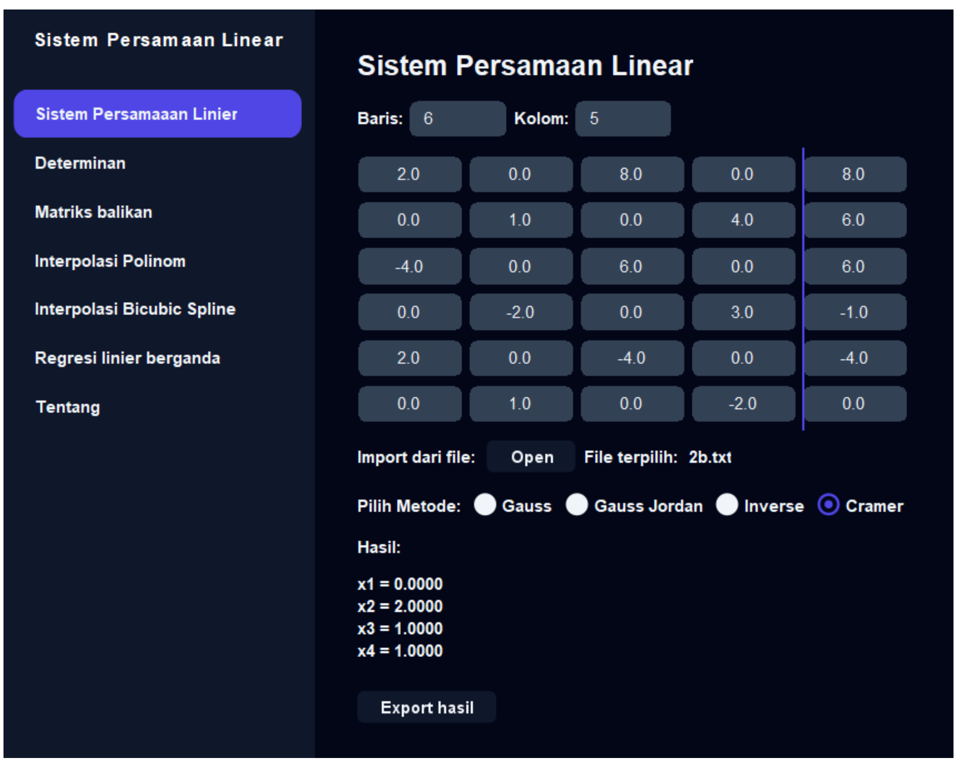
SPL ini diselesaikan dengan metode eliminasi Gauss dan didapat nilai dari x seperti yang ditampilkan pada gambar diatas. SPL ini tidak akan bisa diselesaikan dengan metode inverse dan metode Cramer karena memiliki panjang baris dan kolom yang tidak sama.

3. Akan diselesaikan SPL dengan bentuk sebagai berikut

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

Solusi:

```
--> Wrapping up, solusi SPL ini adalah
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
```

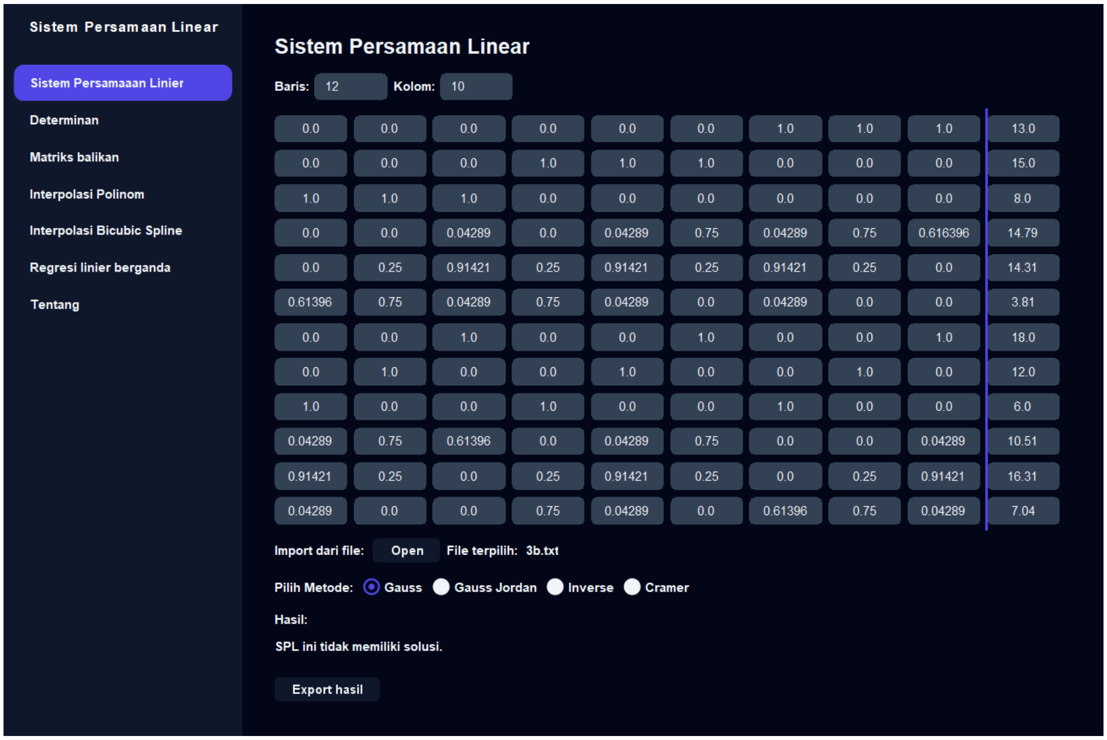


Analisis: SPL diatas akan diubah terlebih dahulu menjadi bentuk augmented matrix kemudian akan dimasukan ke dalam program kami. SPL diatas diselesaikan dengan metode Cramer dan didapat data nilai x seperti yang tertampil pada gambar. SPL sebenarnya bisa diselesaikan dengan semua metode yang tersedia dalam program kami.

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

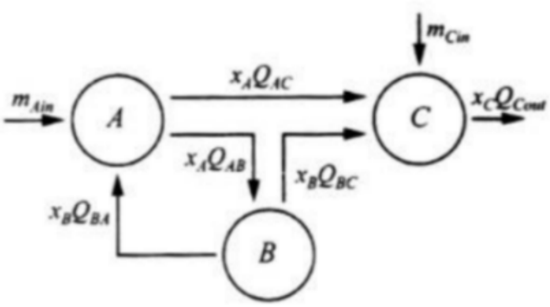
Solusi SPL menggunakan program penyelesaian:

```
(1) . . .   1.0000x1 +1.0000x2 +1.0000x3 = 8.0000
(2) . . .   1.0000x2 +3.6568x3 +1.0000x4 +3.6568x5 +1.0000x6 +3.6568x7 +1.0000x8 = 57.2400
(3) . . .   1.0000x3 +1.0000x5 +17.4866x6 +1.0000x7 +17.4866x8 +14.3716x9 = 344.8356
(4) . . .   1.0000x4 +1.0000x5 +1.0000x6 = 15.0000
(5) . . .   1.0000x5 +16.4866x6 +1.0000x7 +17.4866x8 +13.3716x9 = 326.8356
(6) . . .   1.0000x6 +0.0342x7 +1.0342x8 +0.8562x9 = 19.5356
(7) . . .   1.0000x7 +1.0000x8 +1.0000x9 = 13.0000
(8) . . .   1.0000x8 -0.2301x9 = 4.8054
(9) . . .   1.0000x9 = 5.0236
(10) . . .   0.0000 = 1.0000
(11) . . .   0.0000 = 0.0000
(12) . . .   0.0000 = 0.0000

--> Cek apakah ada suatu persamaan yang tidak konsisten.
***) Dapat dilihat bahwa persamaan ke-10 tak konsisten, karena tidak akan ada
***) nilai variabel yang memenuhi persamaan tsb.


--> Kesimpulannya, SPL tersebut tidak memiliki solusi.
Program selesai.
```



Analisis: SPL di atas diselesaikan menggunakan metode Gauss, yaitu matriks A (LHS) dijadikan matriks eselon baris terlebih dahulu, sehabis itu program kami dapat menangani bahwa SPL tidak memiliki solusi dengan mengecek persamaan ke 10 inkonsisten.

4. Akan diselesaikan persoalan sistem reaktor sebagai berikut:



Bentuk persamaan dari sistem reaktor tersebut adalah sebagai berikut:

$$\text{A:} \quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$$

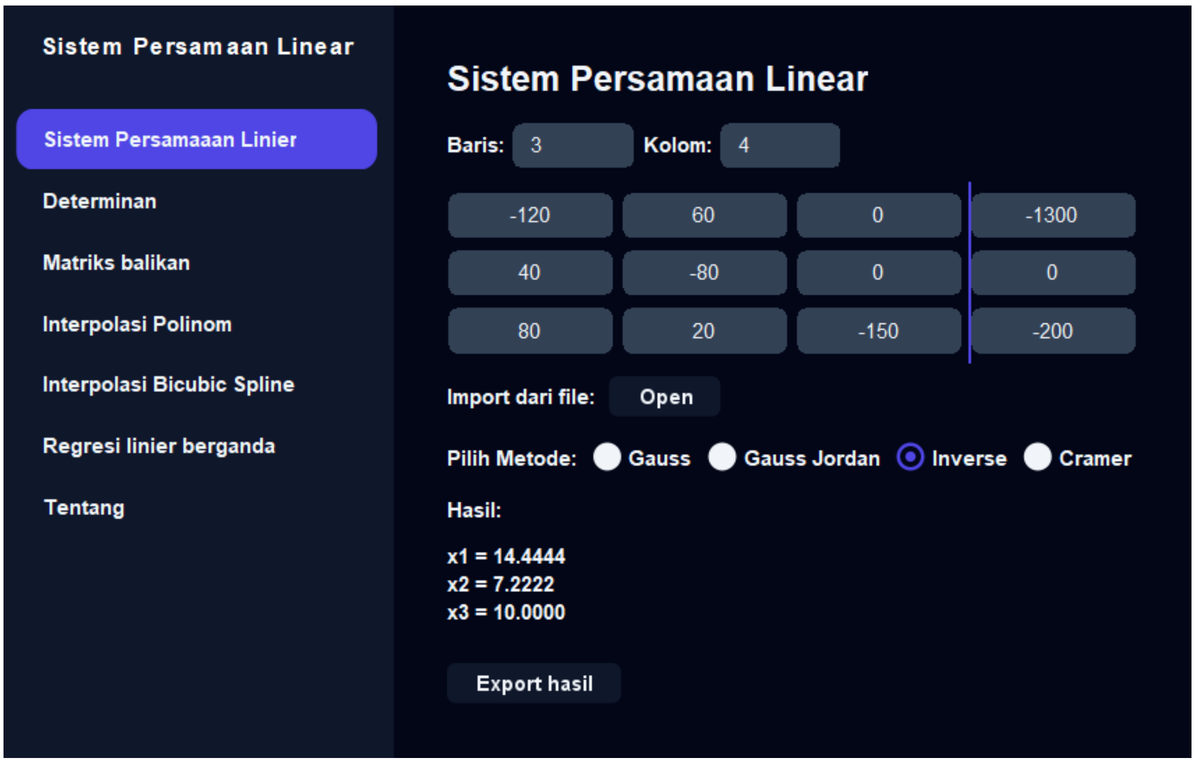
$$\text{B:} \quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0$$

$$\text{C:} \quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0$$

Nilai dari parameter Q yang akan digunakan dalam persamaan ini adalah $Q_{AB} = 40$, $Q_{AC} = 80$, $Q_{BA} = 60$, $Q_{BC} = 20$ dan $Q_{Cout} = 150\ m^3/s$ dan $m_{Ain} = 1300$ dan $m_{Cin} = 200\ mg/s$.

Solusi:

```
--> Simpulkan.
x1 = 14.4444
x2 = 7.2222
x3 = 10.0000
```



Analisis:
Pertama-tama akan dibentuk SPL dengan variabel xa, xb dan xc yang tepat. Bentuk dari matriks augmented untuk SPL yang terbentuk pada soal ini adalah sebagai berikut:

```
[-120, 60, 0, -1300]
[40, -80, 0, 0]
[80, 20, -150, -200]
```

Kolom pertama adalah koefisien dari xa, kolom kedua adalah koefisien dari xb, kolom ketiga adalah koefisien untuk xc, dan kolom keempat adalah nilai m. Matriks SPL tersebut kemudian diselesaikan dengan metode inverse dan diperoleh nilai xa = 14.44, xb = 7.222, dan x3 = 10.

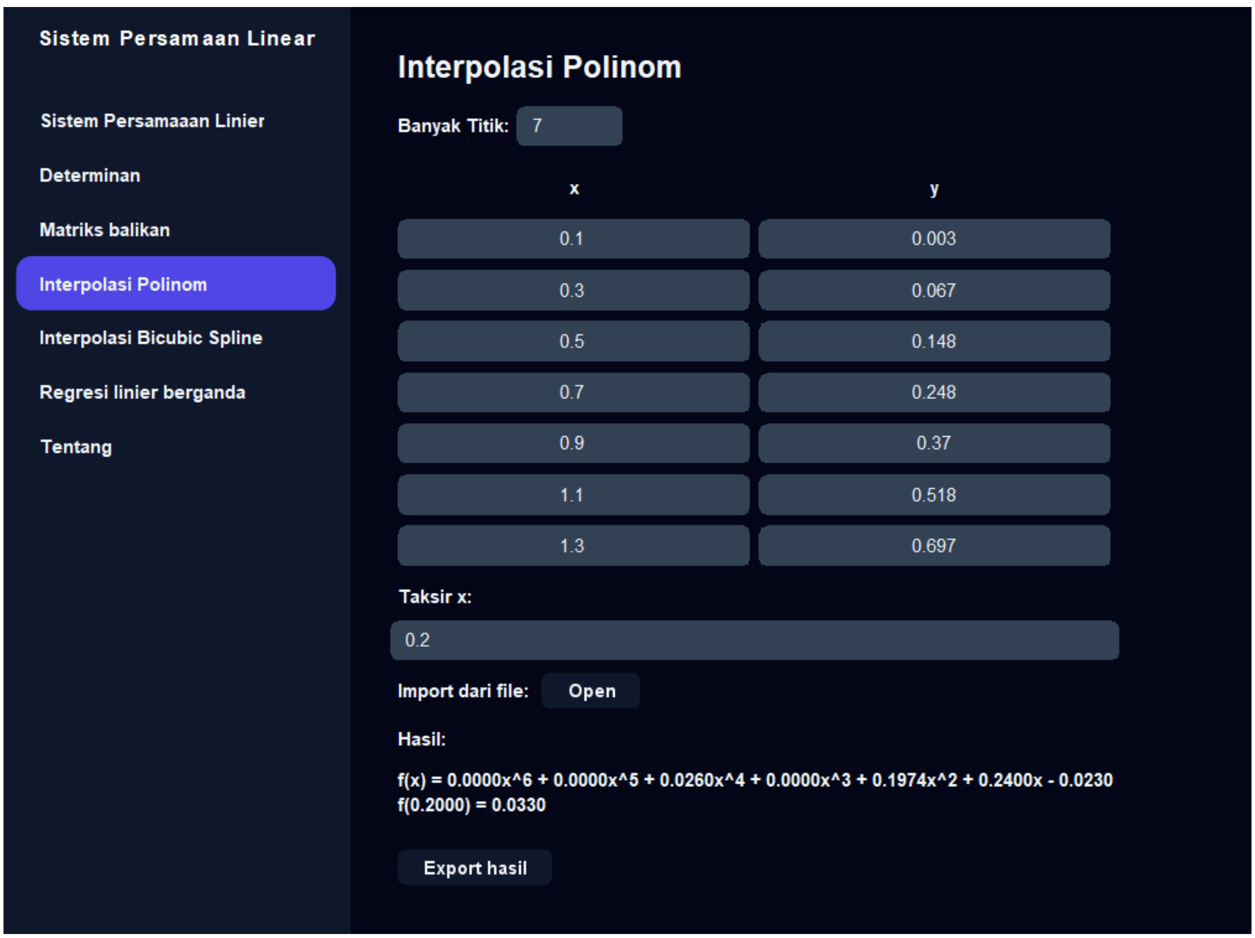
B. Studi Kasus Interpolasi

1. Berikut akan ditampilkan hasil dari studi kasus interpolasi no 5a dengan data masukan sebagai berikut

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

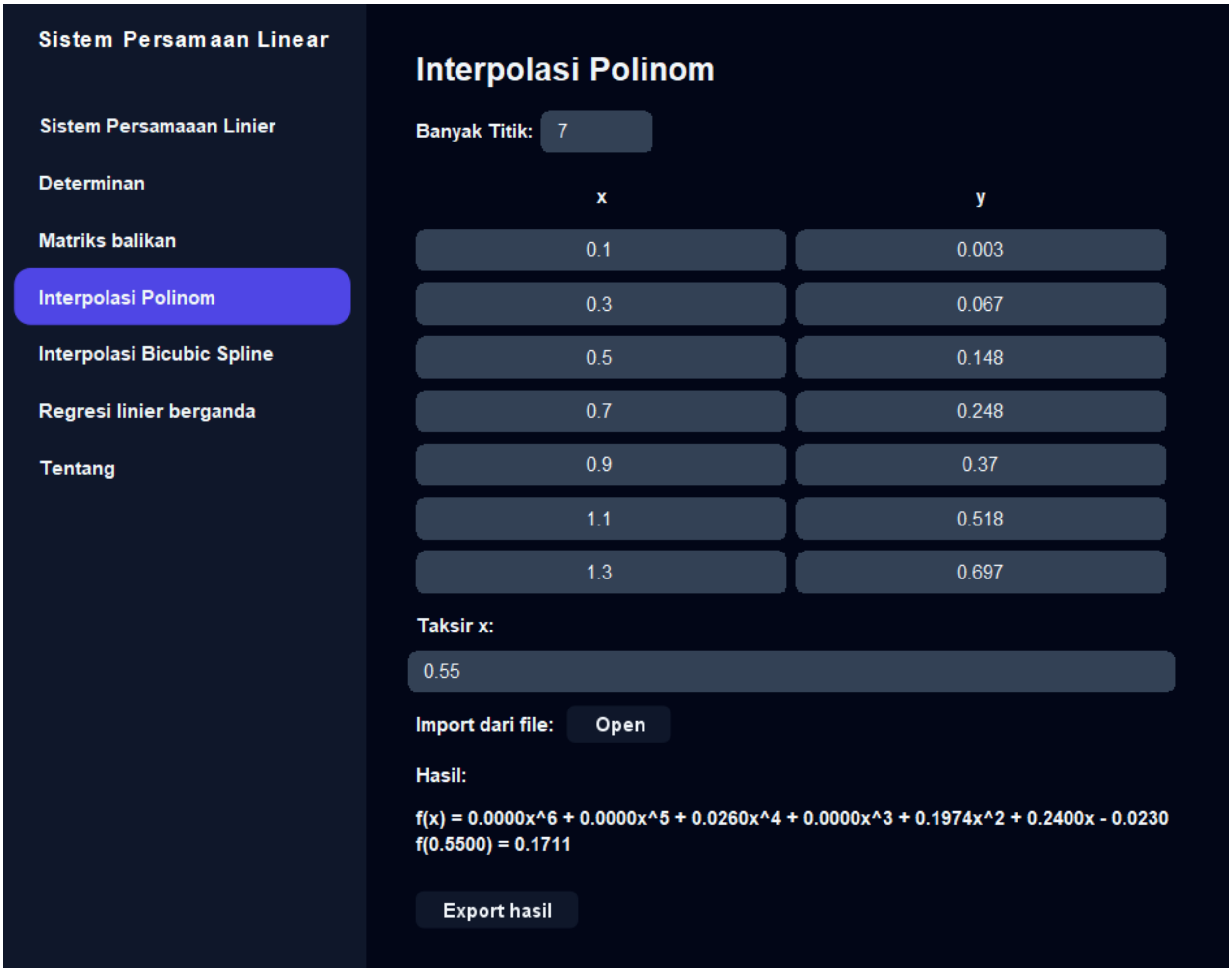
Solusi untuk x = 0.2

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = 0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 0.2000

f(0.2000) = 0.0000(0.2000^6) + 0.0000(0.2000^5) + 0.0260(0.2000^4) + 0.0000(0.2000^3) + 0.1974(0.2000^2) + 0.2400(0.2000) - 0.0230
f(0.2000) = 0.0000(0.0001) + 0.0000(0.0003) + 0.0260(0.0016) + 0.0000(0.0080) + 0.1974(0.0400) + 0.2400(0.2000) - 0.0230
f(0.2000) = 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0079 + 0.0480 - 0.0230
f(0.2000) = 0.0330
```



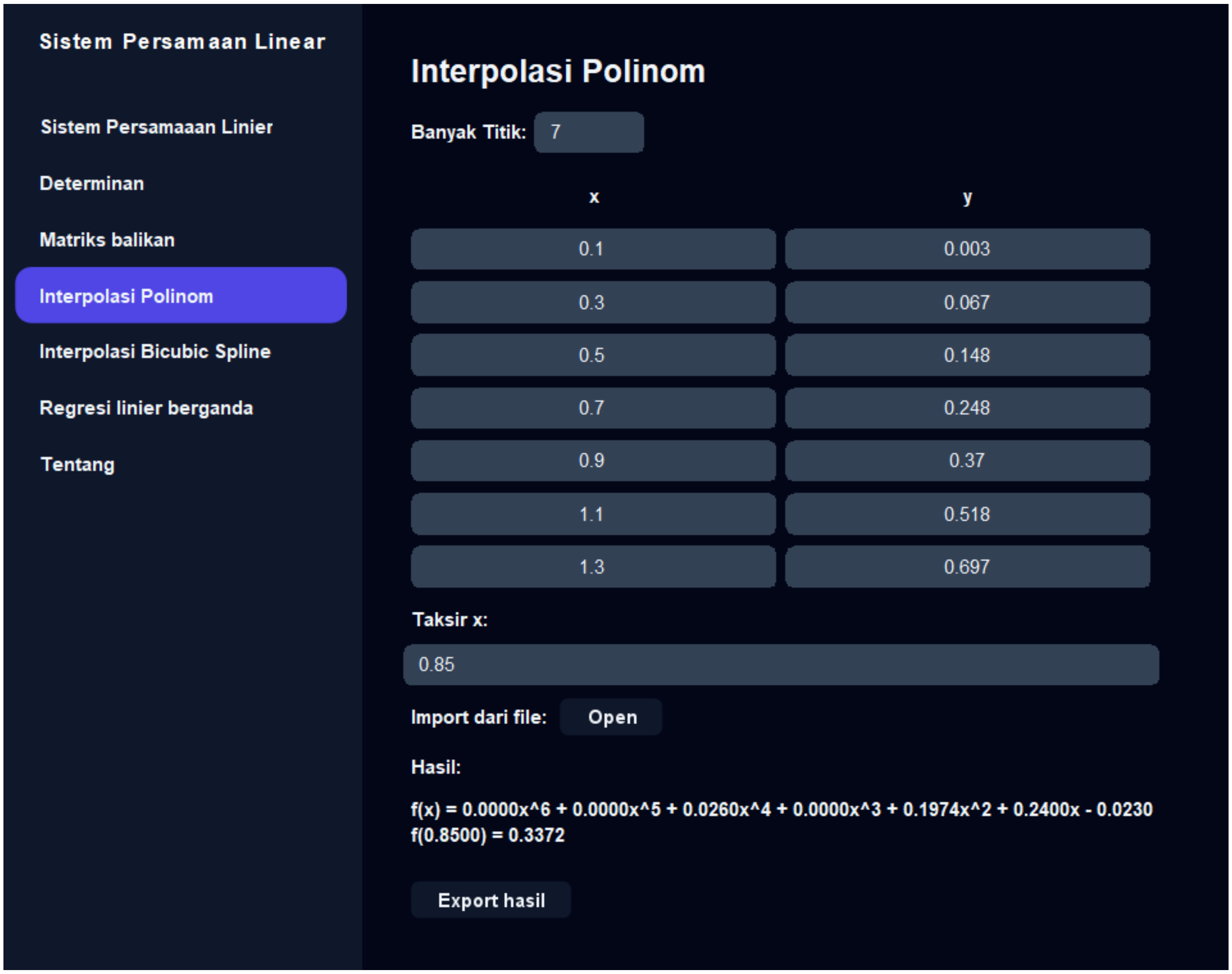
Solusi untuk x = 0.55

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = 0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 0.5500

f(0.5500) = 0.0000(0.5500^6) + 0.0000(0.5500^5) + 0.0260(0.5500^4) + 0.0000(0.5500^3) + 0.1974(0.5500^2) + 0.2400(0.5500) - 0.0230
f(0.5500) = 0.0000(0.0277) + 0.0000(0.0503) + 0.0260(0.0915) + 0.0000(0.1664) + 0.1974(0.3025) + 0.2400(0.5500) - 0.0230
f(0.5500) = 0.0000 + 0.0000 + 0.0024 + 0.0000 + 0.0597 + 0.1320 - 0.0230
f(0.5500) = 0.1711
```



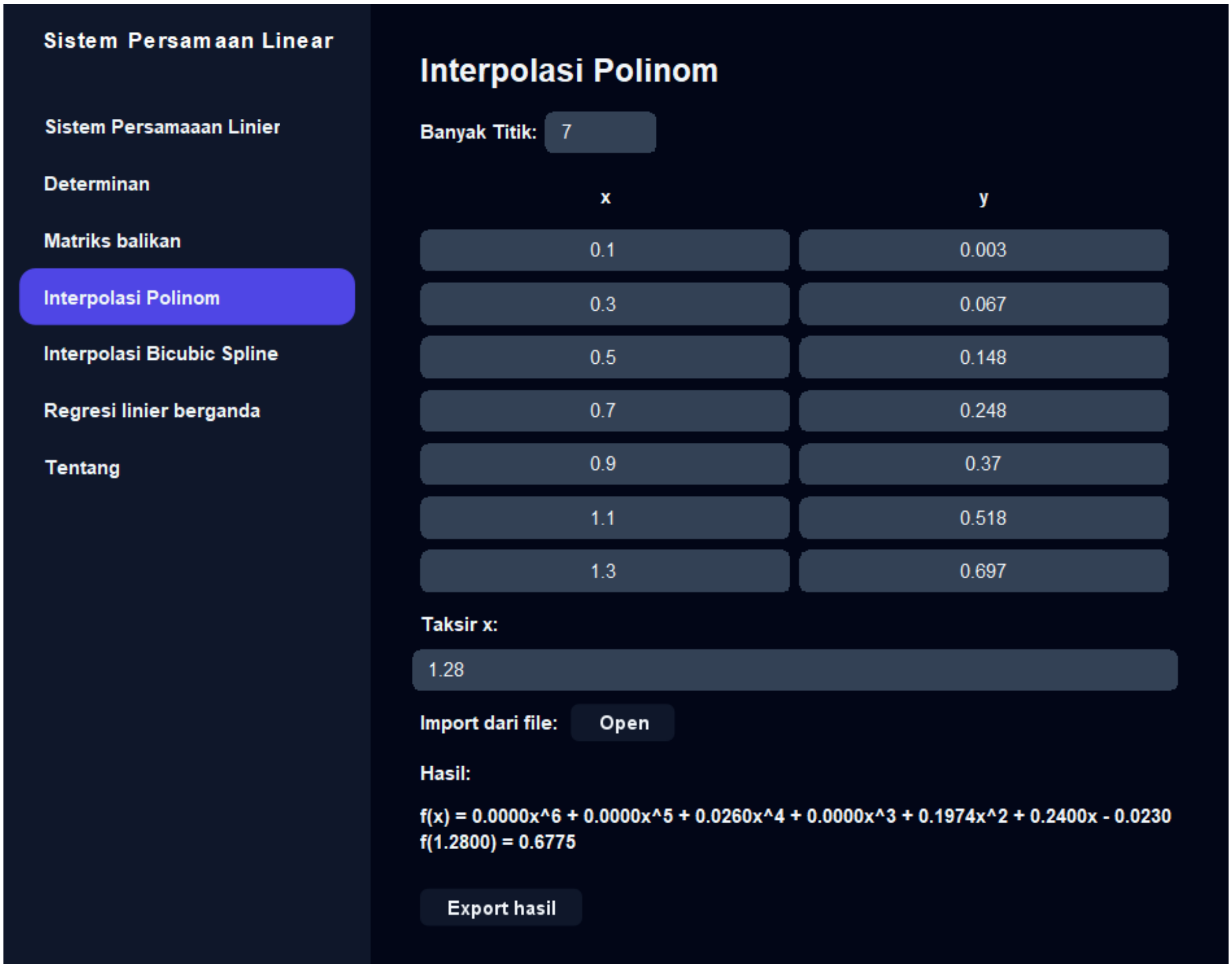
Solusi untuk x = 0.85

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = 0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 0.8500

f(0.8500) = 0.0000(0.8500^6) + 0.0000(0.8500^5) + 0.0260(0.8500^4) + 0.0000(0.8500^3) + 0.1974(0.8500^2) + 0.2400(0.8500) - 0.0230
f(0.8500) = 0.0000(0.3771) + 0.0000(0.4437) + 0.0260(0.5220) + 0.0000(0.6141) + 0.1974(0.7225) + 0.2400(0.8500) - 0.0230
f(0.8500) = 0.0000 + 0.0000 + 0.0136 + 0.0000 + 0.1426 + 0.2040 - 0.0230
f(0.8500) = 0.3372
```



Solusi untuk x = 1.28

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = 0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x - 0.0230

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 1.2800

f(1.2800) = 0.0000(1.2800^6) + 0.0000(1.2800^5) + 0.0260(1.2800^4) + 0.0000(1.2800^3) + 0.1974(1.2800^2) + 0.2400(1.2800) - 0.0230
f(1.2800) = 0.0000(4.3980) + 0.0000(3.4360) + 0.0260(2.6844) + 0.0000(2.0972) + 0.1974(1.6384) + 0.2400(1.2800) - 0.0230
f(1.2800) = 0.0000 + 0.0000 + 0.0699 + 0.0000 + 0.3234 + 0.3072 - 0.0230
f(1.2800) = 0.6775
```



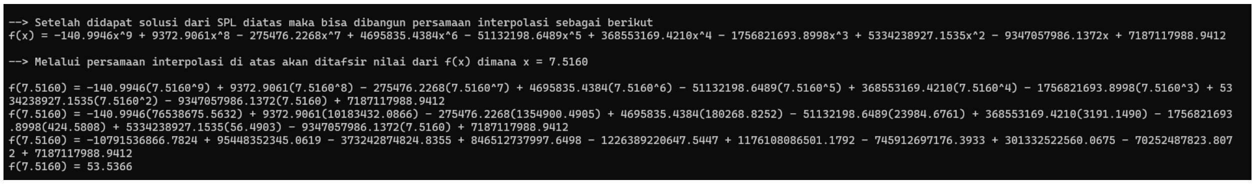
Analis:
Pada studi kasus no 5a ini telah disediakan tabel yang merepresentasikan titik (x, f(x)). Titik-titik tersebut kemudian dimasukan ke dalam program kami dan melalui titik-titik tersebut akan dapat ditemukan persamaan interpolasi polinomnya. Berdasarkan program kami telah didapat persamaan interpolasi polinom $f(x) = 0.026x^4 + 0.1974x^2 + 0.24x - 0.023$. Kemudian akan dilakukan aproksimasi nilai f(x) untuk nilai x yang diberikan dan akan berdasarkan program kami akan diperoleh data sebagai berikut:

Di saat x = 0.2, f(0.2) = 0.033
Di saat x = 0.55, f(0.55) = 0.1711
Di saat x = 0.85, f(0.85) = 0.3372
Di saat x = 1.28, f(1.28) = 0.6775

2. Berikut akan ditampilkan hasil dari studi kasus interpolasi no 5a dengan data masukan sebagai berikut

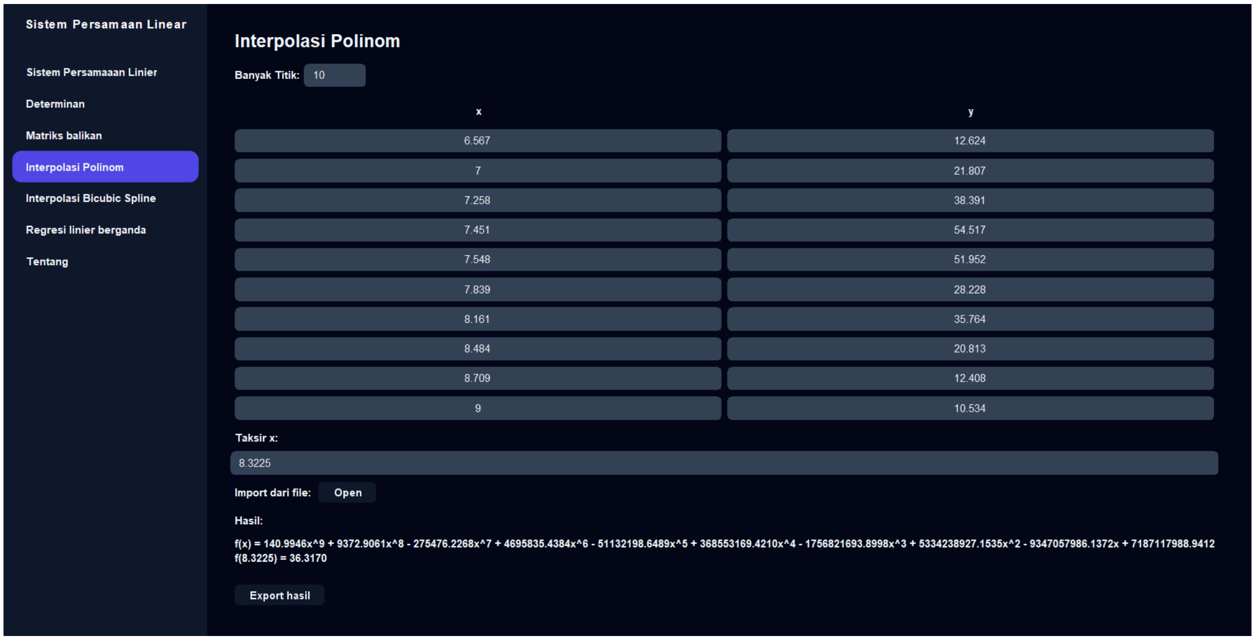
Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6,567	12.624
30/06/2022	7	21.807
08/07/2022	7,258	38.391
14/07/2022	7,451	54.517
17/07/2022	7,548	51.952
26/07/2022	7,839	28.228
05/08/2022	8,161	35.764
15/08/2022	8,484	20.813
22/08/2022	8,709	12.408
31/08/2022	9	10.534

Solusi untuk tanggal 16/07/2022 dimana bentuk desimalnya adalah 7.516



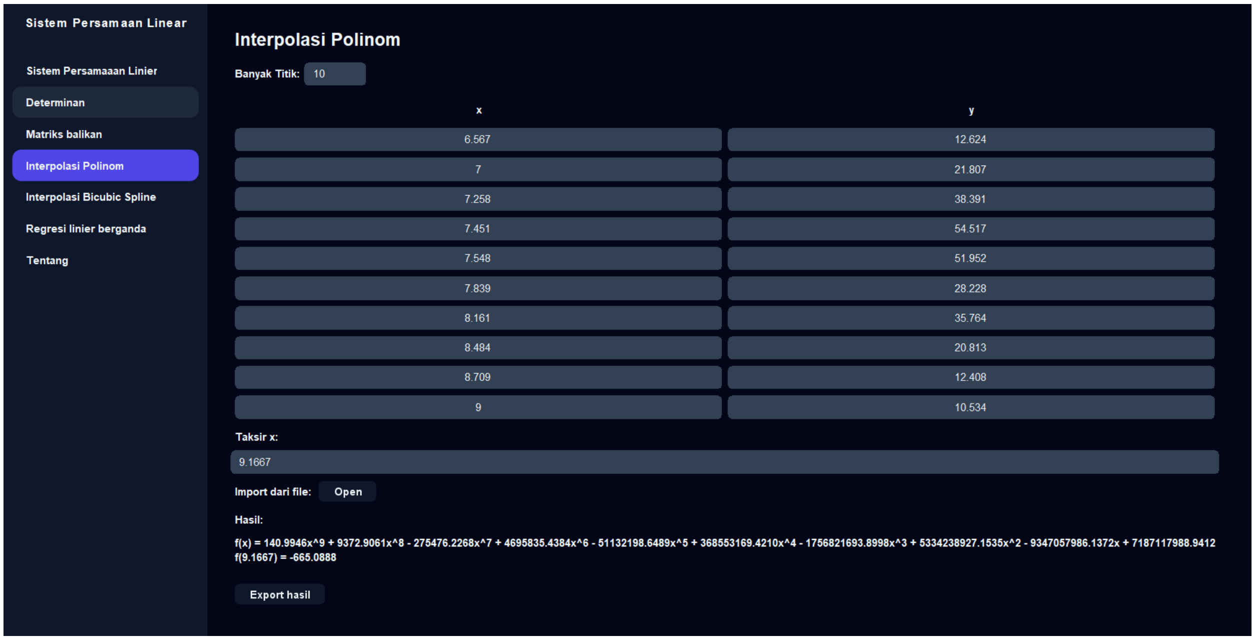
Solusi untuk tanggal 10/08/2022 dimana bentuk desimalnya adalah 8.3225

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = -140.9946x^9 + 9372.9061x^8 - 275476.2268x^7 + 4695835.4384x^6 - 51132198.6489x^5 + 368553169.4210x^4 - 1756821693.8998x^3 + 5334238927.1535x^2 - 9347057986.1372x + 7187117988.9412

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 8.3225

f(8.3225) = -140.9946(8.3225^9) + 9372.9061(8.3225^8) - 275476.2268(8.3225^7) + 4695835.4384(8.3225^6) - 51132198.6489(8.3225^5) + 368553169.4210(8.3225^4) - 1756821693.8998(8.3225^3) + 53
34238927.1535(8.3225^2) - 9347057986.1372(8.3225) + 7187117988.9412
f(8.3225) = -140.9946(191550916.5867) + 9372.9061(23016030.8305) - 275476.2268(2765518.8742) + 4695835.4384(332294.2474) - 51132198.6489(39927.2151) + 368553169.4210(4797.5026) - 175682169
3.8998(576.4497) + 5334238927.1535(69.2640) - 9347057986.1372(8.3225) + 7187117988.9412
f(8.3225) = -27007637158.5897 + 215727094912.6308 - 761834704628.8469 + 1560399103025.6992 - 2041566292484.9368 + 1768134774456.5823 - 1012719324374.8907 + 369470758389.3544 - 77790890089.
6269 + 7187117988.9412
f(8.3225) = 36.3170
```



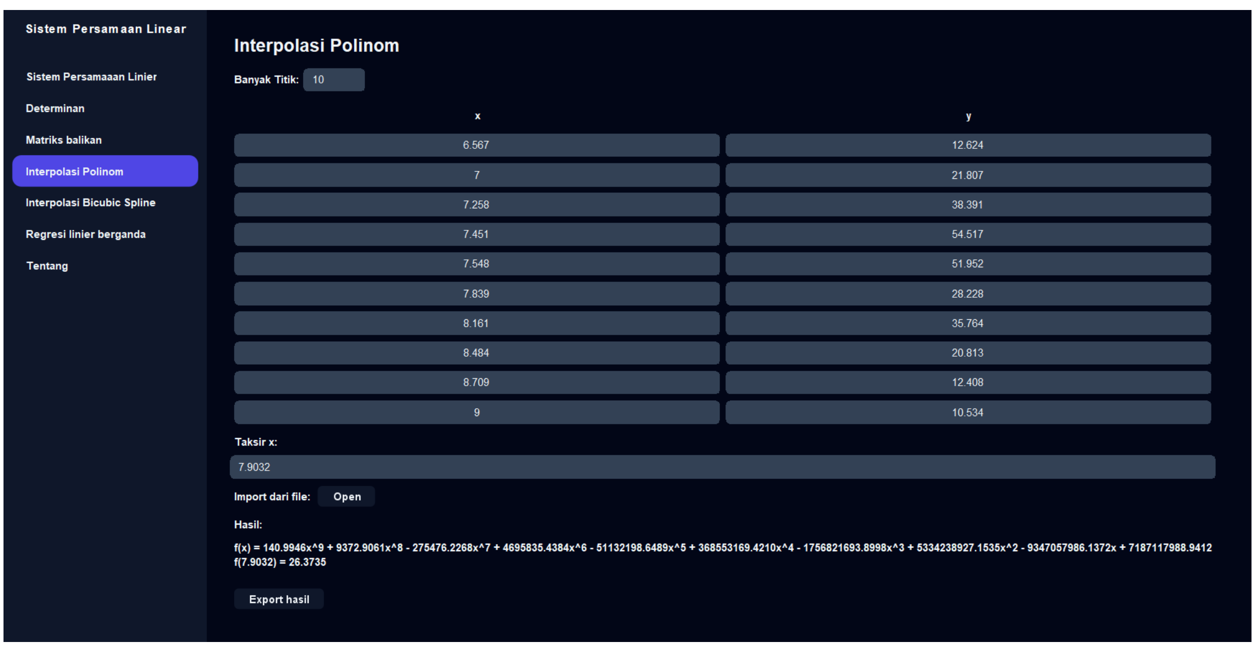
Solusi untuk tanggal 05/09/2022 dimana bentuk desimalnya adalah 9.1667

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = -140.9946x^9 + 9372.9061x^8 - 275476.2268x^7 + 4695835.4384x^6 - 51132198.6489x^5 + 368553169.4210x^4 - 1756821693.8998x^3 + 5334238927.1535x^2 - 9347057986.1372x + 7187117988.9412

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 9.1667

f(9.1667) = -140.9946(9.1667^9) + 9372.9061(9.1667^8) - 275476.2268(9.1667^7) + 4695835.4384(9.1667^6) - 51132198.6489(9.1667^5) + 368553169.4210(9.1667^4) - 1756821693.8998(9.1667^3) + 53
34238927.1535(9.1667^2) - 9347057986.1372(9.1667) + 7187117988.9412
f(9.1667) = -140.9946(457001015.6354) + 9372.9061(49854474.9621) - 275476.2268(5438650.2190) + 4695835.4384(593305.1391) - 51132198.6489(64723.9616) + 368553169.4210(7060.7701) - 175682169
3.8998(770.2630) + 5334238927.1535(84.0284) - 9347057986.1372(9.1667) + 7187117988.9412
f(9.1667) = -64434657016.1396 + 467281310630.4265 - 1498218841246.7285 + 2786063298145.4910 - 3309478463779.3877 + 2602269213446.8354 - 1353214805396.0356 + 448227503003.0322 - 85681676441
.5239 + 7187117988.9412
f(9.1667) = -665.0888

Hasil aproksimasi tidak akurat karena x berada di luar range[6.5670, 9.0000]
```



Solusi untuk tanggal 28/07/2022 dimana bentuk desimalnya adalah 7.9032

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = -140.9946x^9 + 9372.9061x^8 - 275476.2268x^7 + 4695835.4384x^6 - 51132198.6489x^5 + 368553169.4210x^4 - 1756821693.8998x^3 + 5334238927.1535x^2 - 9347057986.1372x + 7187117988.9412

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 7.9032

f(7.9032) = -140.9946(7.9032^9) + 9372.9061(7.9032^8) - 275476.2268(7.9032^7) + 4695835.4384(7.9032^6) - 51132198.6489(7.9032^5) + 368553169.4210(7.9032^4) - 1756821693.8998(7.9032^3) + 53
34238927.1535(7.9032^2) - 9347057986.1372(7.9032) + 7187117988.9412
f(7.9032) = -140.9946(120289231.9223) + 9372.9061(15220319.8606) - 275476.2268(1925842.6790) + 4695835.4384(243678.8489) - 51132198.6489(30832.9346) + 368553169.4210(3901.3228) - 175682169
3.8998(493.6384) + 5334238927.1535(62.4606) - 9347057986.1372(7.9032) + 7187117988.9412
f(7.9032) = -16960127300.5277 + 142658628299.1913 - 530523874633.8741 + 1144275774463.1284 - 1576555738287.9580 + 1437844895665.2532 - 867234612678.1558 + 333179605186.4145 - 73871668676.0
396 + 7187117988.9412
f(7.9032) = 26.3735
```



Analisis:

Pada studi kasus ini diberikan tabel yang berisi data tanggal dan jumlah kasus baru COVID-19. Tanggal kemudian akan diubah kedalam bentuk desimal dengan rumus sebagai berikut:

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

Tanggal dalam bentuk desimal akan menjadi x sedangkan jumlah kasus baru COVID-19 akan menjadi f(x) atau y. Pasangan (x,y) kemudian akan dimasukkan ke dalam program kami dan setelah itu akan didapat sebuah persamaan polinom seperti yang sudah terlihat pada gambar-gambar di atas. Melalui persamaan itu kemudian di aproksimasi jumlah kasus baru pada tanggal yang diberikan oleh soal. Tanggal-tanggal yang diberikan tentu saja harus diubah terlebih dahulu ke dalam bentuk desimal lalu baru dimasukkan ke dalam program kami. Berikut adalah data hasil perhitungan dari program kami:

a. Pada tanggal 16/07/2022, diprediksi jumlah kasus baru COVID-19 berjumlah 53.5366 jiwa.
b. Pada tanggal 10/08/2022. diprediksi jumlah kasus baru COVID-19 berjumlah 36.317 jiwa.
c. Pada tanggal 05/09/2022. diprediksi jumlah kasus baru COVID-19 berjumlah -665.0888 jiwa.
d. Pada tanggal 28/07/2022. diprediksi jumlah kasus baru COVID-19 berjumlah 26.3735 jiwa.

Jika dilihat jumlah kasus baru COVID-19 pada tanggal 05/09/2022 memberikan hasil yang cukup berbeda jauh dibandingkan yang lainnya. Hal tersebut dikarenakan tanggal tersebut melebihi range dari persamaan interpolasi yang terbentuk dimana hal tersebut juga sudah tertulis pada program kami (05/09/2022 lebih dari 31/08/2022).

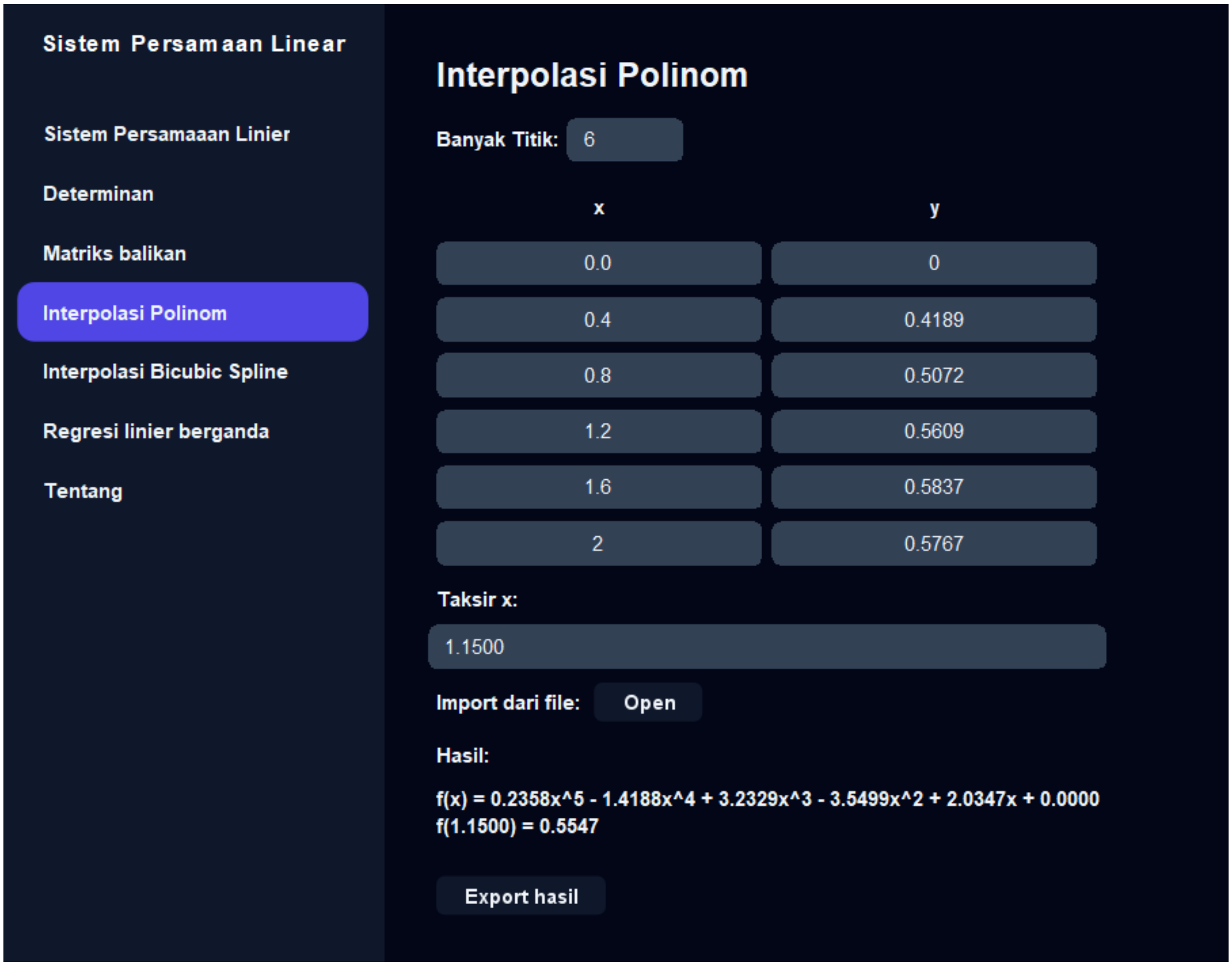
3. Akan disederhanakan fungsi f(x) berikut:

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

Dengan polinom interpolasi derajat n, dimana n = 5 di dalam selang [0,2]. Nilai x yang akan diambil adalah 0.0, 0.4, 0.8, 1.2, 1.6, dan 2. Dengan nilai f(x) berturut-turut adalah 0, 0.4189, 0.5072, 0.5609, 0.5837, dan 0.5767

Solusi:

```
--> Setelah didapat solusi dari SPL diatas maka bisa dibangun persamaan interpolasi sebagai berikut
f(x) = 0.2358x^5 - 1.4188x^4 + 3.2329x^3 - 3.5499x^2 + 2.0347x + 0.0000

--> Melalui persamaan interpolasi di atas akan ditafsir nilai dari f(x) dimana x = 1.1500

f(1.1500) = 0.2358(1.1500^5) - 1.4188(1.1500^4) + 3.2329(1.1500^3) - 3.5499(1.1500^2) + 2.0347(1.1500) + 0.0000
f(1.1500) = 0.2358(2.0114) - 1.4188(1.7490) + 3.2329(1.5209) - 3.5499(1.3225) + 2.0347(1.1500) + 0.0000
f(1.1500) = 0.4742 - 2.4815 + 4.9168 - 4.6947 + 2.3399 + 0.0000
f(1.1500) = 0.5547
```



Analisis:
Pada studi kasus ini diberikan sebuah fungsi f(x) dan kami diminta untuk mengubahnya menjadi persamaan polinom. Pada kasus ini kami mengambil polinom berderajat 5 kemudian mengambil 6 titik sampel pada rentang [0, 2]. Setelah diperoleh titik-titik tersebut kemudian titik-titik tersebut dimasukan ke dalam program kami dan kemudian akan diperoleh persamaan interpolasi polinom seperti yang terlihat pada foto di atas. Kami juga memasukan nilai x = 1.15 untuk diaproksimasi nilai f(x) nya. Pada program kami diperoleh f(1.15) = 0.5547 sedangkan jika f(1.15) dihitung secara manual dengan menggunakan rumus f(x) yang pertama (yang ada pada soal) maka akan diperoleh nilai f(1.15) = 0.5559. Jika dilihat pada kedua hasil f(1.15) tersebut hanya terdapat delta yang cukup kecil hal itu berarti bahwa persamaan polinom yang dibentuk oleh program kami sudah cukup akurat.

C. Studi Kasus Regresi Linear Berganda

1. Akan diselesaikan studi kasus no 6 yang memberikan data sebagai berikut:

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.

Pada soal juga diminta untuk mengestimasi nilai NO apabila humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Solusi:

```
Normal Estimation Equation for Multiple Linear Regression:
20.000x0 + 863.100x1 + 1530.400x2 + 587.840x3 = 19.4204
863.100x0 + 54876.890x1 + 67000.090x2 + 25283.395x3 = 779.4774
1530.400x0 + 67000.090x1 + 117912.320x2 + 44976.867x3 = 1483.4374
587.840x0 + 25283.395x1 + 44976.867x2 + 17278.509x3 = 571.1224

f(x) = -3.508 - 0.003x1 + 0.001x2 + 0.154x3

Taksiran:
f(50.000,76.000,29.300) = 0.938
```

Analisis:

Semua data yang diberikan kemudian akan dimasukkan ke dalam program kami. Di dalam program kami kemudian data tersebut akan diubah ke dalam bentuk *Normal Estimation Equation for Multiple Linear Regression* dan setelah itu akan diselesaikan dengan pendekatan matriks metode GaussJordan. Setelah itu akan terbentuk persamaan regresi linear berganda seperti yang tertulis pada foto di atas. Kemudian program kami juga akan mengestimasi nilai NO berdasarkan nilai kelembapan, suhu, dan tekanan yang diberikan. Berdasarkan program kami di dapat nilai NO adalah 0.938.

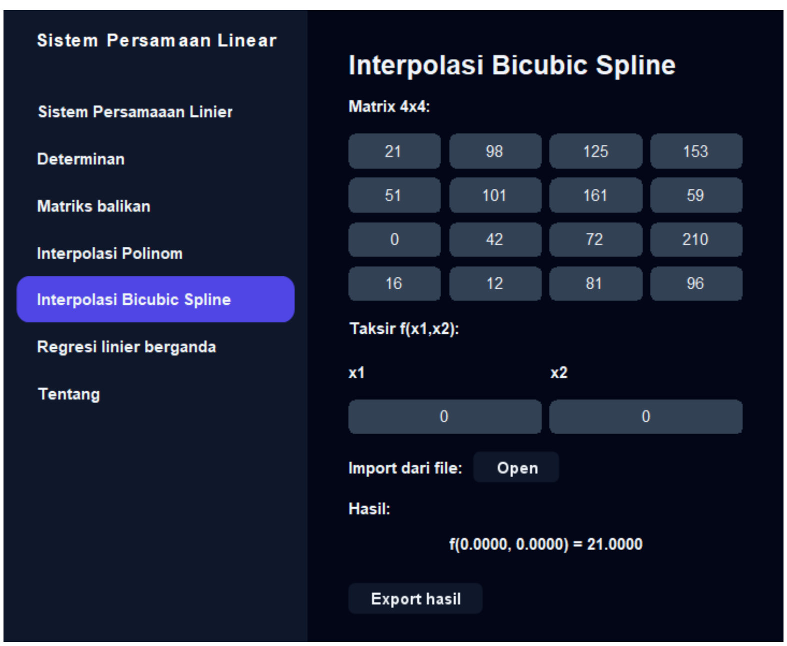
D. Studi Kasus Interpolasi Bicubic Spline

1. Akan diselesaikan persoalan studi kasus no 7 dengan masukan data berbentuk matriks sebagai berikut:

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

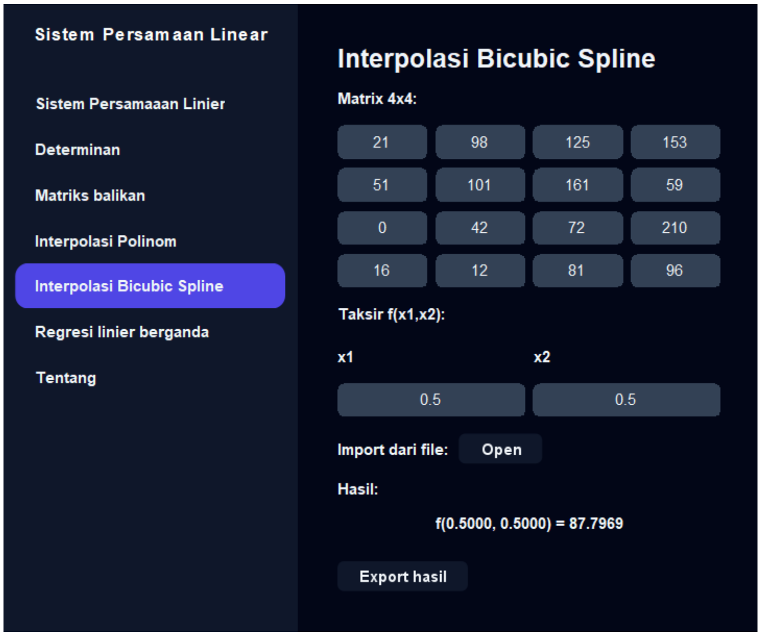
Solusi untuk a= 0, dan b = 0

```
--> Melalui nilai konstanta a yang telah didapat akan di cari nilai f(x,y) dengan x = 0.0000 dan y = 0.0000
--> f(x,y) = Σi=0,1,2,3 Σj=0,1,2,3 ((aij)(x^i)(y^j))
--> Proses perhitungan nilai f(x,y) adalah sebagai berikut

f(0.0000, 0.0000) = (21.0000)(0.0000)^0(0.0000)^0 + (51.0000)(0.0000)^1(0.0000)^0 + (28.0000)(0.0000)^2(0.0000)^0 + (-2.0000)(0.0000)^3(0.0000)^0 + (0.0000)(0.0000)^0(0.0
000)^1 + (16.0000)(0.0000)^1(0.0000)^1 + (82.0000)(0.0000)^2(0.0000)^1 + (-56.0000)(0.0000)^3(0.0000)^1 + (240.0000)(0.0000)^0(0.0000)^2 + (217.0000)(0.0000)^1(0.0000)^2
+ (-1295.0000)(0.0000)^2(0.0000)^2 + (709.0000)(0.0000)^3(0.0000)^2 + (-136.0000)(0.0000)^0(0.0000)^3 + (-123.0000)(0.0000)^1(0.0000)^3 + (888.0000)(0.0000)^2(0.0000)^3 +
 (-487.0000)(0.0000)^3(0.0000)^3
f(0.0000, 0.0000) = (21.0000)(1.0000)(1.0000) + (51.0000)(0.0000)(1.0000) + (28.0000)(0.0000)(1.0000) + (-2.0000)(0.0000)(1.0000) + (0.0000)(1.0000)(0.0000) + (16.0000)(0
.0000)(0.0000) + (82.0000)(0.0000)(0.0000) + (-56.0000)(0.0000)(0.0000) + (240.0000)(1.0000)(0.0000) + (217.0000)(0.0000)(0.0000) + (-1295.0000)(0.0000)(0.0000) + (709.00
00)(0.0000)(0.0000) + (-136.0000)(1.0000)(0.0000) + (-123.0000)(0.0000)(0.0000) + (888.0000)(0.0000)(0.0000) + (-487.0000)(0.0000)(0.0000)
f(0.0000, 0.0000) = 21.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000 + 0.0000
f(0.0000, 0.0000) = 21.0000
```



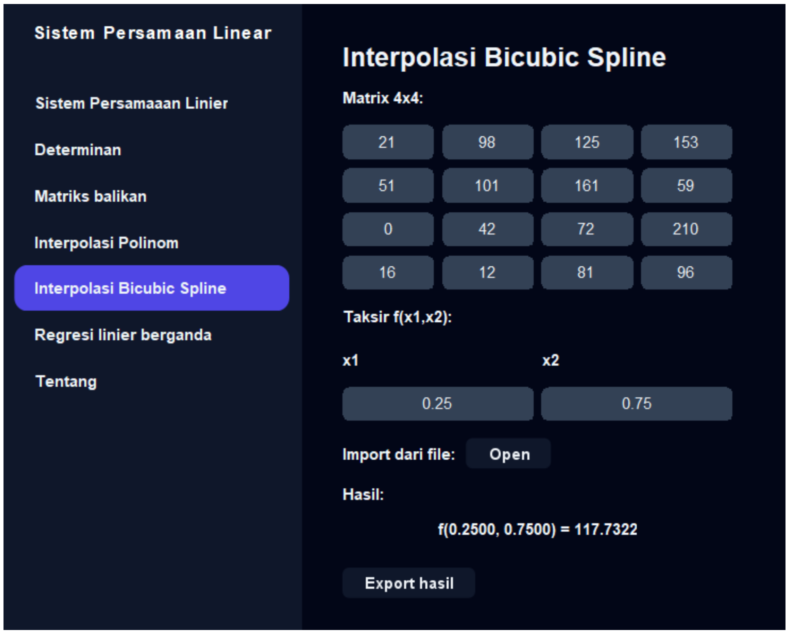
Solusi untuk a = 0.5 dan b = 0.5

```
--> Melalui nilai konstanta a yang telah didapat akan di cari nilai f(x,y) dengan x = 0.5000 dan y = 0.5000
--> f(x,y) = Σi=0,1,2,3 Σj=0,1,2,3 ((aij)(x^i)(y^j))
--> Proses perhitungan nilai f(x,y) adalah sebagai berikut

f(0.5000, 0.5000) = (21.0000)(0.5000)^0(0.5000)^0 + (51.0000)(0.5000)^1(0.5000)^0 + (28.0000)(0.5000)^2(0.5000)^0 + (-2.0000)(0.5000)^3(0.5000)^0 + (0.0000)(0.5000)^0(0.5
000)^1 + (16.0000)(0.5000)^1(0.5000)^1 + (82.0000)(0.5000)^2(0.5000)^1 + (-56.0000)(0.5000)^3(0.5000)^1 + (240.0000)(0.5000)^0(0.5000)^2 + (217.0000)(0.5000)^1(0.5000)^2
+ (-1295.0000)(0.5000)^2(0.5000)^2 + (709.0000)(0.5000)^3(0.5000)^2 + (-136.0000)(0.5000)^0(0.5000)^3 + (-123.0000)(0.5000)^1(0.5000)^3 + (888.0000)(0.5000)^2(0.5000)^3 +
 (-487.0000)(0.5000)^3(0.5000)^3
f(0.5000, 0.5000) = (21.0000)(1.0000)(1.0000) + (51.0000)(0.5000)(1.0000) + (28.0000)(0.2500)(1.0000) + (-2.0000)(0.1250)(1.0000) + (0.0000)(1.0000)(0.5000) + (16.0000)(0
.5000)(0.5000) + (82.0000)(0.2500)(0.5000) + (-56.0000)(0.1250)(0.5000) + (240.0000)(1.0000)(0.2500) + (217.0000)(0.5000)(0.2500) + (-1295.0000)(0.2500)(0.2500) + (709.00
00)(0.1250)(0.2500) + (-136.0000)(1.0000)(0.1250) + (-123.0000)(0.5000)(0.1250) + (888.0000)(0.2500)(0.1250) + (-487.0000)(0.1250)(0.1250)
f(0.5000, 0.5000) = 21.0000 + 25.5000 + 7.0000 - 0.2500 + 0.0000 + 4.0000 + 10.2500 - 3.5000 + 60.0000 + 27.1250 - 80.9375 + 22.1563 - 17.0000 - 7.6875 + 27.7500 - 7.6094
f(0.5000, 0.5000) = 87.7969
```



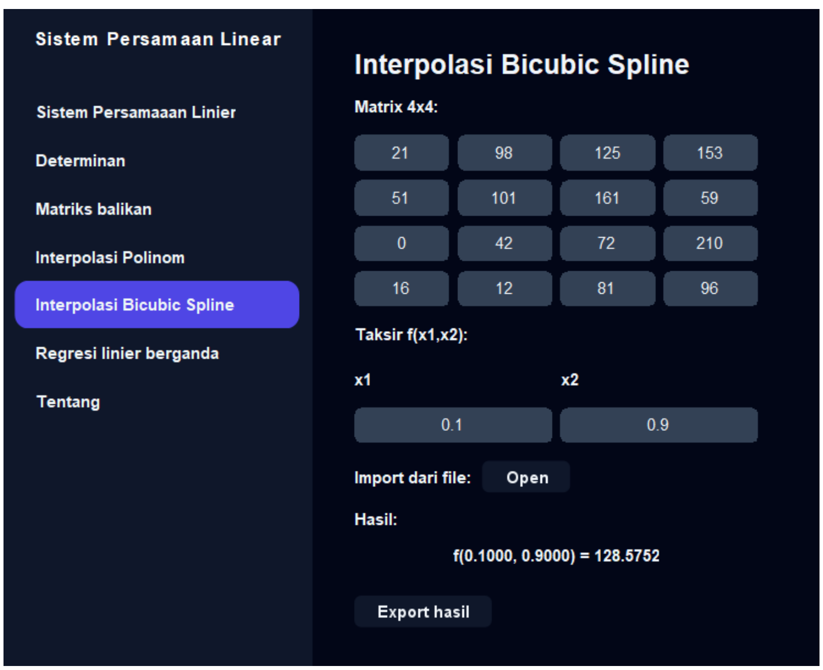
Solusi untuk a = 0.25 dan b = 0.75

```
--> Melalui nilai konstanta a yang telah didapat akan di cari nilai f(x,y) dengan x = 0.2500 dan y = 0.7500
--> f(x,y) = Σi=0,1,2,3 Σj=0,1,2,3 ((aij)(x^i)(y^j))
--> Proses perhitungan nilai f(x,y) adalah sebagai berikut

f(0.2500, 0.7500) = (21.0000)(0.2500)^0(0.7500)^0 + (51.0000)(0.2500)^1(0.7500)^0 + (28.0000)(0.2500)^2(0.7500)^0 + (-2.0000)(0.2500)^3(0.7500)^0 + (0.0000)
(0.2500)^0(0.7500)^1 + (16.0000)(0.2500)^1(0.7500)^1 + (82.0000)(0.2500)^2(0.7500)^1 + (-56.0000)(0.2500)^3(0.7500)^1 + (240.0000)(0.2500)^0(0.7500)^2 + (21
7.0000)(0.2500)^1(0.7500)^2 + (-1295.0000)(0.2500)^2(0.7500)^2 + (709.0000)(0.2500)^3(0.7500)^2 + (-136.0000)(0.2500)^0(0.7500)^3 + (-123.0000)(0.2500)^1(0.
7500)^3 + (888.0000)(0.2500)^2(0.7500)^3 + (-487.0000)(0.2500)^3(0.7500)^3
f(0.2500, 0.7500) = (21.0000)(1.0000)(1.0000) + (51.0000)(0.2500)(1.0000) + (28.0000)(0.0625)(1.0000) + (-2.0000)(0.0156)(1.0000) + (0.0000)(1.0000)(0.7500)
 + (16.0000)(0.2500)(0.7500) + (82.0000)(0.0625)(0.7500) + (-56.0000)(0.0156)(0.7500) + (240.0000)(1.0000)(0.5625) + (217.0000)(0.2500)(0.5625) + (-1295.000
0)(0.0625)(0.5625) + (709.0000)(0.0156)(0.5625) + (-136.0000)(1.0000)(0.4219) + (-123.0000)(0.2500)(0.4219) + (888.0000)(0.0625)(0.4219) + (-487.0000)(0.015
6)(0.4219)
f(0.2500, 0.7500) = 21.0000 + 12.7500 + 1.7500 - 0.0313 + 0.0000 + 3.0000 + 3.8438 - 0.6563 + 135.0000 + 30.5156 - 45.5273 + 6.2314 - 57.3750 - 12.9727 + 23
.4141 - 3.2102
f(0.2500, 0.7500) = 117.7322
```



Solusi untuk a = 0.1 dan b = 0.9

```
--> Melalui nilai konstanta a yang telah didapat akan di cari nilai f(x,y) dengan x = 0.1000 dan y = 0.9000
--> f(x,y) = Σi=0,1,2,3 Σj=0,1,2,3 ((aij)(x^i)(y^j))
--> Proses perhitungan nilai f(x,y) adalah sebagai berikut

f(0.1000, 0.9000) = (21.0000)(0.1000)^0(0.9000)^0 + (51.0000)(0.1000)^1(0.9000)^0 + (28.0000)(0.1000)^2(0.9000)^0 + (-2.0000)(0.1000)^3(0.9000)^0 + (0.0000)
(0.1000)^0(0.9000)^1 + (16.0000)(0.1000)^1(0.9000)^1 + (82.0000)(0.1000)^2(0.9000)^1 + (-56.0000)(0.1000)^3(0.9000)^1 + (240.0000)(0.1000)^0(0.9000)^2 + (21
7.0000)(0.1000)^1(0.9000)^2 + (-1295.0000)(0.1000)^2(0.9000)^2 + (709.0000)(0.1000)^3(0.9000)^2 + (-136.0000)(0.1000)^0(0.9000)^3 + (-123.0000)(0.1000)^1(0.
9000)^3 + (888.0000)(0.1000)^2(0.9000)^3 + (-487.0000)(0.1000)^3(0.9000)^3
f(0.1000, 0.9000) = (21.0000)(1.0000)(1.0000) + (51.0000)(0.1000)(1.0000) + (28.0000)(0.0100)(1.0000) + (-2.0000)(0.0010)(1.0000) + (0.0000)(1.0000)(0.9000)
 + (16.0000)(0.1000)(0.9000) + (82.0000)(0.0100)(0.9000) + (-56.0000)(0.0010)(0.9000) + (240.0000)(1.0000)(0.8100) + (217.0000)(0.1000)(0.8100) + (-1295.000
0)(0.0100)(0.8100) + (709.0000)(0.0010)(0.8100) + (-136.0000)(1.0000)(0.7290) + (-123.0000)(0.1000)(0.7290) + (888.0000)(0.0100)(0.7290) + (-487.0000)(0.001
0)(0.7290)
f(0.1000, 0.9000) = 21.0000 + 5.1000 + 0.2800 - 0.0020 + 0.0000 + 1.4400 + 0.7380 - 0.0504 + 194.4000 + 17.5770 - 10.4895 + 0.5743 - 99.1440 - 8.9667 + 6.47
35 - 0.3550
f(0.1000, 0.9000) = 128.5752
```



Analisis:
Pada studi kasus ini diberikan sebuah matriks yang berisi nilai dari fungsi f, fx, fy, dan fxy yang sesuai dengan ketentuan yang diminta. Matriks tersebut kemudian dimasukkan ke dalam program kami dan menghasilkan persamaan interpolasi bicubic spline. Melalui persamaan itu akan diaproksimasi suatu nilai f(a, b) berdasarkan nilai a dan b yang diberikan oleh soal. Dari program kami didapatkan data sebagai berikut:

Di saat a = 0, b = 0, f(0, 0) = 0
Di saat a = 0.5, b = 0.5, f(0.5, 0.5) = 87.7969
Di saat a = 0.25, b = 0.75, f(0.25, 0.75) = 117.7322
Di saat a = 0.1, b = 0.9, f(0.1, 0.9) = 128.5752

Jika dilihat nilai untuk f(0,0) sebenarnya juga merupakan masukan dari user. Dimana pada masukan user f(0,0) = 21 dan hal tersebut sudah sama dengan apa yang program kami hitung.

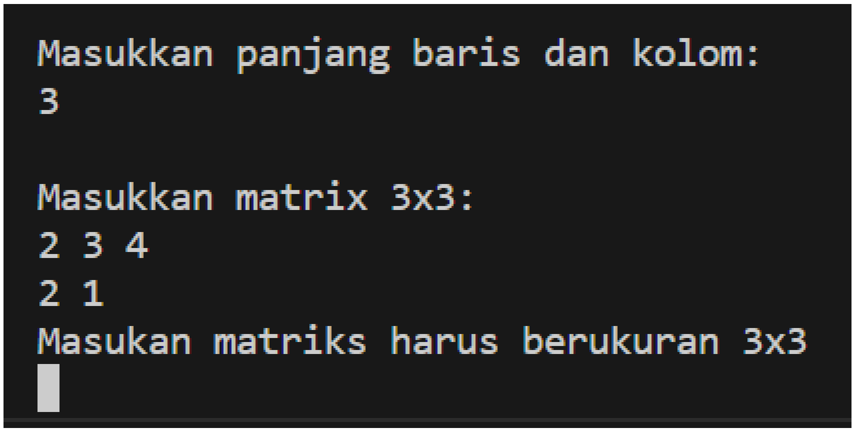
E. Studi Kasus Input invalid

Berikut ini contoh kasus untuk input yang invalid.

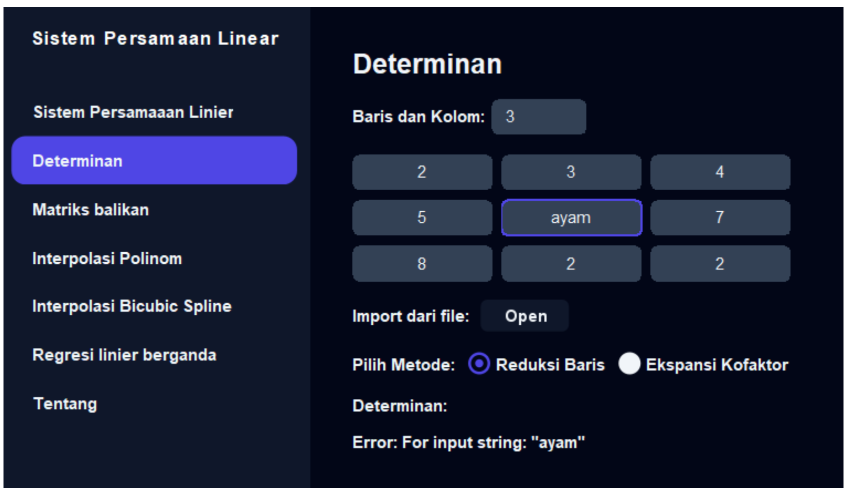
Kasus input diluar range pilihan menu atau bukan bilangan bulat.

```
[Welcome to Library Matrix]

MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi linier berganda
7. Perbesaran foto
8. Keluar

Pilih menu (Angka):
9
Masukan harus dalam range 1 sampai 8
-5
Masukan harus dalam range 1 sampai 8
ayam
Masukan harus bilangan bulat
b3b3k
Masukan harus bilangan bulat
```

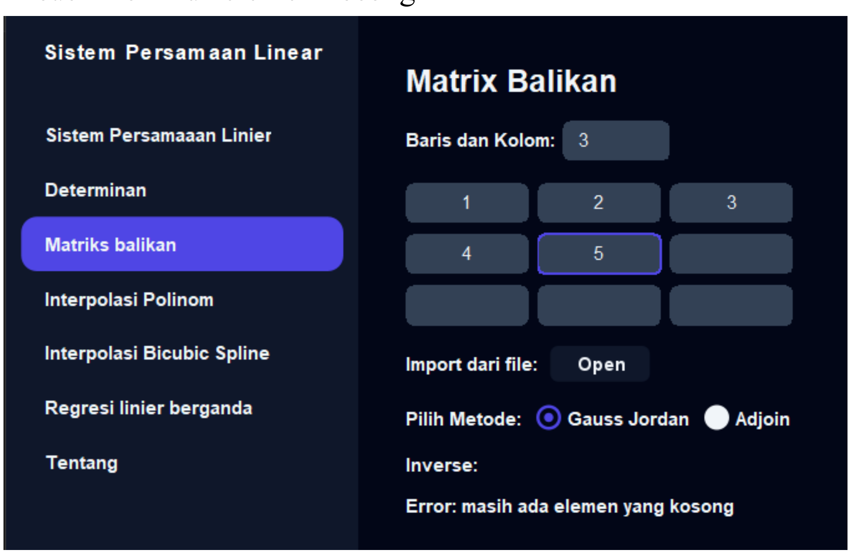
Kasus input matriks tidak sesuai ukuran



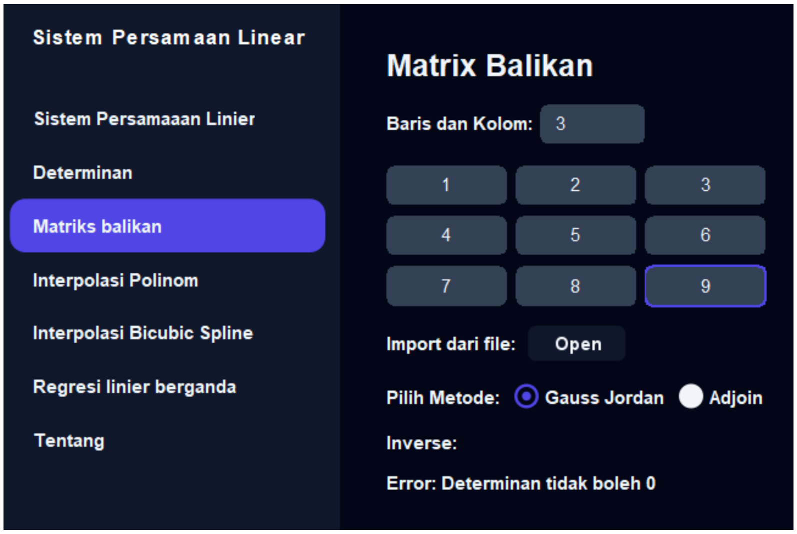
Kasus input tidak sesuai



Kasus masih ada elemen kosong

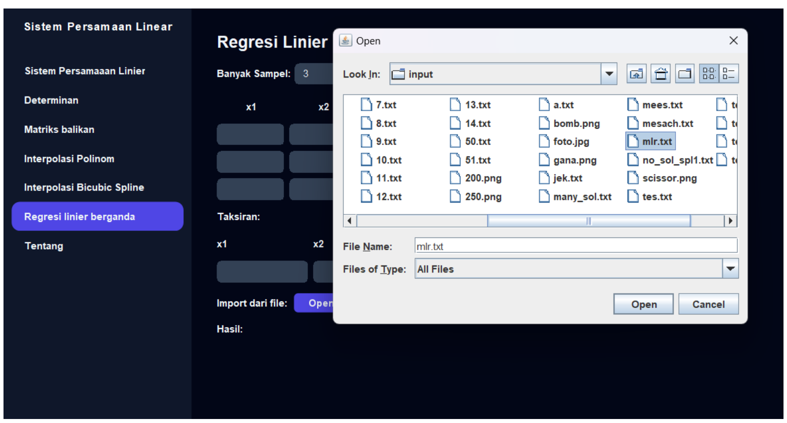


Kasus terjadi hal yang dilarang ditengah perhitungan.

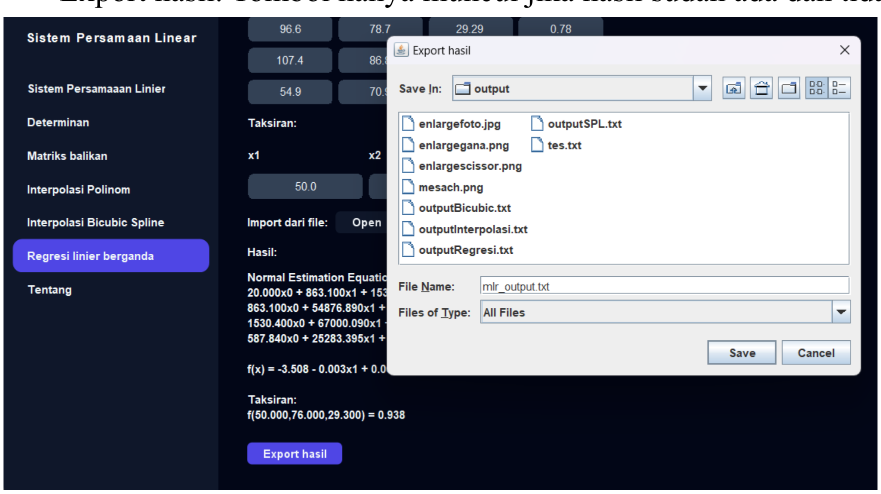


F. Input/Output File

Import file. Hasil import akan dimasukkan menjadi input.



Export hasil. Tombol hanya muncul jika hasil sudah ada dan tidak ada error.



BAB V KESIMPULAN

A. Kesimpulan

Berbagai permasalahan matematis dapat diselesaikan dengan sebuah program. Misalnya seperti sistem persamaan linier. Solusi dari masalah ini dapat diselesaikan dengan berbagai metode yang memiliki kelebihan dan kekurangannya tersendiri. Untuk menyelesaikannya kita memerlukan pustaka matrix yang di dalamnya ada implementasi untuk hal seperti determinan dan inverse. Dari pustaka sistem persamaan linier, kita dapat memanfaatkannya untuk menyelesaikan permasalahan lain seperti interpolasi polinom, interpolasi bicubic spline, regresi linier berganda, dan perbesaran foto. Agar lebih interaktif, kita juga bisa membuat GUI untuk implementasi dari pustaka ini.

B. Saran

Saran dari penulis kepada asisten-asisten mata kuliah ini adalah untuk lebih memperjelas input dari masing-masing poin pada spesifikasi dan juga luaran dari masing-masing spesifikasi.

C. Komentar

Menurut kami, tugas besar ini sangat bermanfaat sebab memaksa kami untuk eksplorasi banyak hal, tidak hanya membuat matriks, memproses matriks, mengimplementasi matriks pada suatu persoalan lainnya, tetapi juga membuat kami mengeskplorasi bagaimana resize gambar dapat bekerja.

Tim ini juga sangat responsif dan komunikatif terhadap setiap permasalah yang dihadapi oleh suatu anggota, kelompok Eskrim… TERBAIK!

D. Refleksi

Melakukan testing sebaiknya dilakukan lebih awal karena akan ditemukan hal-hal di luar ekspektasi. Struktur class dan fungsi sebaiknya dipikirkan dari awal agar tidak jadi perubahan yang akan memakan banyak waktu nantinya.

DAFTAR REFERENSI

Munir, Rinaldi. (2022). IF2123 Aljabar Geometri - Semester I Tahun 2023/2024. Institut Teknologi Bandung. Diakses pada 5 Oktober 2023, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm