LAPORAN TUGAS BESAR 01

IF2123 ALJABAR LINEAR DAN GEOMETRI

Kelompok KAYDENJI

Disusun oleh:

Denise Felicia Tiowanni	(13522013)
Angelica Kierra Ninta Gurning	(13522048)
Kayla Namira Mariadi	(13522050)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2023

DAFTAR ISI

BAB I DESKRIPSI MASALAH.......... 3

BAB II TEORI SINGKAT.......... 4

BAB III IMPLEMENTASI PUSTAKA DAN PROGRAM.......... 13

BAB IV EKSPERIMEN.......... 24

BAB V PENUTUP.......... 48

DAFTAR PUSTAKA.......... 49

BAB I
DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. 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). Determinan juga dapat dihitung dengan metode reduksi baris dan kofaktor.

Di dalam Tugas Besar 1 ini, dibuat 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, *library* tersebut digunakan dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

BAB II
TEORI SINGKAT

A. Metode Eliminasi Gauss

Misal terdapat sebuah SPL dengan m buah persamaan dan n buah variabel x_1, x_2, …, x_n. SPL dapat memiliki solusi unik, banyak solusi, atau tidak memiliki solusi. SPL tersebut direpresentasikan dalam bentuk matriks *augmented*.

Metode Eliminasi Gauss adalah teknik untuk menyelesaikan sistem persamaan linear dengan menerapkan OBE pada matriks *augmented* hingga terbentuk matriks eselon baris. Persamaan lalu diselesaikan dengan teknik penyulihan mundur.

Matriks eselon baris memiliki beberapa sifat-sifat penting:

- Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama).
- Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
- Di dalam dua baris berurutan 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.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_n \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & * & * & \dots & * & * \\ 0 & 1 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

Gambar 1. Penyelesaian SPL dengan Metode Eliminasi Gauss

B. Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan adalah teknik untuk menyelesaikan sistem persamaan linear dengan mengubah matriks ke dalam bentuk matriks eselon baris tereduksi. Dengan metode ini, tidak diperlukan substitusi secara mundur untuk memperoleh nilai-nilai variabel, melainkan langsung diperoleh dari matriks *augmented* akhir (jika solusinya unik). Metode Eliminasi Gauss-Jordan memiliki 2 fase, yaitu:

- Fase maju (*forward phase*), yaitu menghasilkan nilai-nilai 0 di bawah 1 utama dengan eliminasi Gauss.
- Fase mundur (*backward phase*), yaitu menghasilkan nilai-nilai 0 di atas 1 utama.

Kedua fase ini dapat dilakukan secara bersamaan atau sekuensial.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix}$$

Gambar 2. Penyelesaian SPL dengan Metode Eliminasi Gauss-Jordan

C. Determinan

Pada matriks A berukuran n x n, determinannya dinotasikan sebagai berikut:

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

Gambar 3. Determinan matriks A

Misal terdapat matriks B yang diperoleh dari memanipulasi matriks A. Berlaku aturan determinan sebagai berikut:

1. Jika baris pada matriks A dikali dengan k, det(B) = k det(A)
2. Jika dua baris matriks A dipertukarkan, det(B) = - det(A)
3. Jika sebuah baris pada matriks A ditambahkan k kali baris lain, det(B) = det(A)

Determinan matriks dapat dicari dengan 2 metode:

1. Metode Reduksi Baris

   Determinan diperoleh dengan menerapkan OBE hingga diperoleh matriks segitiga.

$$[A] \overset{\text{OBE}}{\sim} [\text{matriks segitiga bawah}]$$

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

$$\text{maka } \det(A) = (-1)^p \, a'_{11} a'_{22} \ldots a'_{nn}$$

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

Gambar 4. Menentukan determinan matriks A dengan reduksi baris

2. Metode Ekspansi Kofaktor

   Untuk metode ini didefinisikan :

   M_{ij} = minor entri a_{ij} atau determinan submatriks yang elemen-elemennya tidak berada pada baris i dan kolom j

   $C_{ij} = (-1)^{i+j} M_{ij}$ atau kofaktor entri a_{ij}

   Untuk menghitung determinan matriks A, digunakan salah satu dari persamaan berikut.

Secara baris:

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

Secara kolom:

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

Gambar 5. Menentukan determinan matriks A dengan ekspansi kofaktor

D. Matriks Balikan

Untuk matriks persegi A dengan ukuran n x n, matriks balikan (invers) A, yang disimbolkan sebagai A^{-1}, didefinisikan sebagai matriks yang memenuhi persamaan AA^{-1} = $A^{-1}A$ = I, dimana I adalah matriks identitas dengan elemen 1 pada diagonal utamanya dan elemen 0 di seluruh tempat lainnya. Matriks identitas A, yang disimbolkan sebagai I, didefinisikan sebagai matriks yang memenuhi persamaan IA = A.

Suatu matriks A memiliki matriks invers jika dan hanya jika determinan A tidak sama dengan 0 (det(A) ≠ 0). Ada dua metode yang dapat digunakan untuk mendapatkan matriks invers A, yaitu Metode Eliminasi Gauss-Jordan dan Metode Matriks Adjoin. Namun, pada program kami, hanya dipakai satu buah metode, yaitu Metode Eliminasi Gauss-Jordan.

Metode Eliminasi Gauss-Jordan dapat digunakan untuk menghitung matriks invers A. Untuk matriks A berukuran n x n, matriks inversnya dapat ditemukan dengan melakukan eliminasi secara simultan pada matriks gabungan [A | I] hingga menjadi [I | A^{-1}]. Di sini, I adalah matriks identitas berukuran n x n.

E. Matriks Kofaktor

Misalkan A adalah suatu matriks berukuran n x n dan C_{ij} adalah kofaktor entri dari a_{ij}, maka matriks kofaktor dari A adalah

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

Gambar 6. Matriks Kofaktor

dimana $C_{ij} = (-\ 1)^{i+j} M_{ij}$, M_{ij} adalah minor entri a_{ij} atau determinan submatriks yang elemen-elemennya tidak terletak pada baris i dan kolom j

F. Matriks Adjoin

Matriks adjoin adalah suatu matriks yang diperoleh dengan men-transpose matriks kofaktor. Matriks transpose sendiri merupakan diperoleh dengan menukar baris dengan kolom sebuah matriks sehingga elemen yang awalnya berada di baris ke-i dan kolom ke-j sekarang berada di baris ke-j dan kolom ke-i.

G. Kaidah Cramer

Kaidah Cramer adalah salah satu cara untuk menemukan solusi dari suatu sistem persamaan linear yang menggunakan matriks. Metode ini digunakan untuk menemukan solusi sistem persamaan linear yang memiliki banyak variabel. Jika Ax = b adalah suatu persamaan linear dengan n buah persamaan dan n buah peubah sedemikian sehingga det(A) $\neq$ 0, maka sistem persamaan linear tersebut memiliki solusi yang unik yaitu sebagai berikut.

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

Gambar 7. Solusi SPL menggunakan Kaidah Cramer

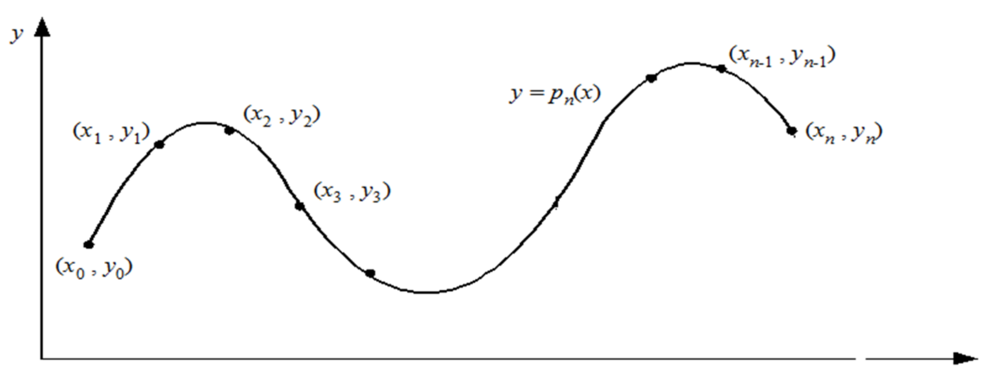
yang dalam hal ini, A_j adalah matriks yang diperoleh dengan mengganti entri pada kolom ke-j dari A dengan entri dari matriks

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

Gambar 8. Matriks B pada Kaidah Cramer

H. Interpolasi Polinom

Interpolasi polinom adalah interpolasi dengan mengasumsikan data yang dimiliki mengikuti pola polinomial berderajat satu (linier) atau berderajat tinggi. Misal diberikan n+1 buah titik berbeda yaitu (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). Polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, …, n adalah :



Gambar 9. Ilustrasi beberapa titik yang diinterpolasi secara polinomial.

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

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

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

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

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

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

I. Regresi Linear Berganda

Regresi Linear adalah metode lain untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu:

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

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

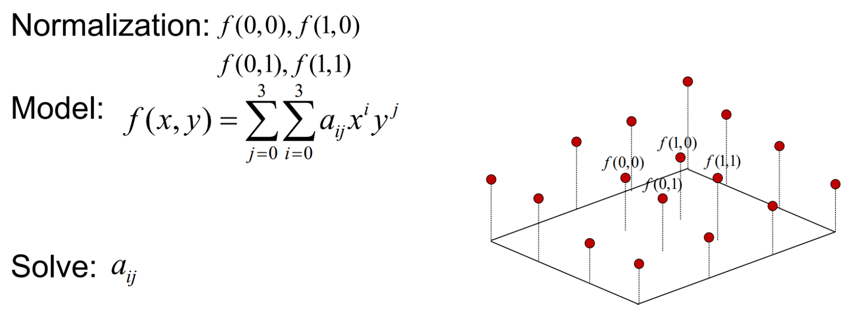
$$\begin{aligned}
nb_0 + b_1 \sum_{i=1}^{n} x_{1i} \quad + b_2 \sum_{i=1}^{n} x_{2i} \quad + \cdots + b_k \sum_{i=1}^{n} x_{ki} \quad &= \sum_{i=1}^{n} y_i \\
b_0 \sum_{i=1}^{n} x_{1i} + b_1 \sum_{i=1}^{n} x_{1i}^2 \quad + b_2 \sum_{i=1}^{n} x_{1i}x_{2i} + \cdots + b_k \sum_{i=1}^{n} x_{1i}x_{ki} &=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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

J. Interpolasi *Bicubic Spline*

Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. *Bicubic spline interpolation* melibatkan konsep *spline* dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. Pendekatan ini menciptakan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat daripada metode interpolasi linear.

Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.



Gambar 8. Pemodelan interpolasi *bicubic spline*.

Selain melibatkan model dasar, juga digunakan model turunan berarah dari kedua sumbu, baik terhadap sumbu x, sumbu y, mapun keduanya. Persamaan polinomial yang digunakan adalah sebagai berikut.

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

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

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

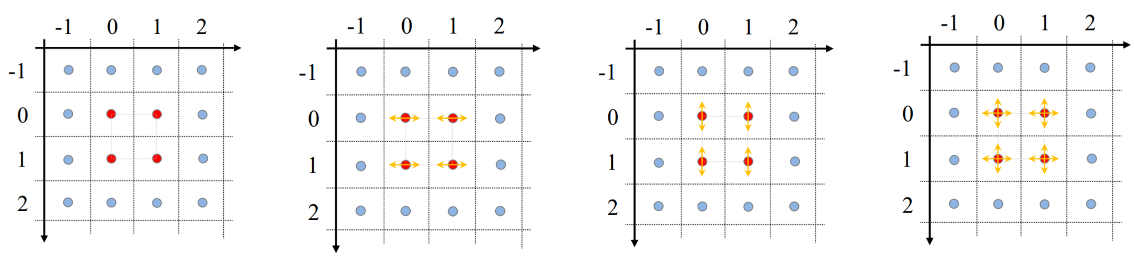
$$f_{xy}(x,y)=\sum_{j=0}^{3}\sum_{i=0}^{3}a_{ij}ijx^{i-1}y^{j-1}$$

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

$$y = Xa$$

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

Elemen pada matriks X adalah nilai dari komponen koefisien a_{ij} yang diperoleh dari persamaan fungsi maupun persamaan turunan. Nilai dari vektor a dicari dari persamaan $y = Xa$, lalu vektor a tersebut digunakan sebagai nilai variabel dalam $f(x, y)$, sehingga terbentuk fungsi interpolasi bicubic sesuai model.



Gambar 9. Nilai fungsi yang akan di interpolasi pada titik merah, turunan berarah terhadap sumbu x, terhadap sumbu y, dan keduanya (kiri ke kanan).

BAB III

IMPLEMENTASI PUSTAKA DAN PROGRAM

A. *Package* Functions

1. bicubicInterpolation.java

   Class ini berisi fungsi dan prosedur yang akan digunakan untuk melakukan *bicubic spline interpolation*.

Fungsi / Prosedur	**Deskripsi**
public static Scanner scan	Berisi variabel scanner untuk menampung input user
public static double[][] createY(double[][] m)	Membuat matrix y berukuran 16 x 1 yang merupakan *resize* dari matrix input berukuran 4 x 4
public static double koefFungsi(int x, int y, int i, int j)	Menghasilkan koefisien *aij* pada ekspansi sigma untuk persamaan $f(x,y)$
public static double koefTurunanX(int x, int y, int i, int j)	Menghasilkan koefisien *aij* pada ekspansi sigma untuk persamaan turunan berarah $f_x(x,y)$
public static double koefTurunanY(int x, int y, int i, int j)	Menghasilkan koefisien *aij* pada ekspansi sigma untuk persamaan turunan berarah $f_y(x,y)$
public static double koefTurunanXY(int x, int y, int i, int j)	Menghasilkan koefisien *aij* pada ekspansi sigma untuk persamaan turunan berarah $f_{xy}(x,y)$
public static double[][] createX()	Membuat matrix X berukuran 16x16 berisi koefisien *aij* dari ekspansi sigma tiap fungsi
public static double[][] inverseX (double[][] A)	Menghasilkan balikan dari matriks X
public static double bicubicSpline (double[][] m, double x, double y)	Menghasilkan aproksimasi nilai dari fungsi pada titik x dan y dengan *bicubic spline interpolation*

2. Interpolasi.java

   Class ini berisi fungsi dan prosedur yang akan digunakan untuk melakukan interpolasi polinomial.

Fungsi / Prosedur	Deskripsi
public static double[] solutionInterpolasi(double[][] m)	Mencari hasil interpolasi
public static double estimate(double[] s, double[] x)	Mencari perkiraan titik interpolasi
public static double[][] inputInterpolasi()	Membaca matriks dari keyboard untuk interpolasi
public static double[] inputTaksiran()	Membaca nilai yang ingin ditaksir
public static void hasilInterpolasi(double[] s)	Prosedur untuk menampilkan hasil interpolasi
public static void hasilEstimateInter(double x, double[] taksiran)	Fungsi untuk menampilkan hasil interpolasi dengan format yang sesuai

3. MainFunctions.java

   Class ini berisi fungsi dan prosedur yang akan digunakan pada ***package*** Main.

Fungsi / Prosedur	Deskripsi
public static Scanner scan	Berisi variabel scanner untuk menampung input user
public static void Gauss(double[][] matriks)	Memanggil dan melaksanakan fungsi Gauss untuk mencari solusi SPL
public static void GaussJordan(double[][] matriks)	Memanggil dan melaksanakan fungsi Gauss-Jordan untuk mencari solusi SPL
public static double[][] SPLBalikan(double[][] A, double[][] B)	Memanggil dan melaksanakan fungsi SPL Balikan untuk mencari solusi SPL
public static void Cramer(double[][] matriks)	Memanggil dan melaksanakan fungsi Cramer untuk mencari solusi SPL

public static void redBaris(double[][] matriks)	Memanggil dan melaksanakan fungsi untuk mencari determinan dengan reduksi baris
public static void ekspansi(double[][] matriks)	Memanggil dan melaksanakan fungsi untuk mencari determinan dengan ekspansi kofaktor
public static void inverseM(double[][] matriks)	Memanggil dan melaksanakan fungsi untuk mencari invers dari suatu matriks
public static void InterpolasiKeyboard(double[] m, double[] x)	Memanggil dan melaksanakan fungsi interpolasi saat sumber inputnya adalah keyboard
public static void InterpolasiFile(double[][] matrix, double[] x)	Memanggil dan melaksanakan fungsi interpolasi saat sumber inputnya adalah file
public static void InterpolasiSpline(double[][] matrixInput)	Memanggil dan melaksanakan fungsi interpolasi *spline*

4. regresiLinearBerganda.java

   Class ini berisi fungsi dan prosedur yang akan digunakan untuk melakukan regresi linear berganda.

Fungsi / Prosedur	**Deskripsi**
public static Scanner scan	Berisi variabel scanner untuk menampung input user
public static double[] solutionReg(double[][] mat)	Mencari nilai beta
public static double estimateReg(double[] s, double[] x)	Mencari taksiran nilai x
public static double[] inputTaksiran(double[][] m)	Menerima input nilai yang ingin ditaksir
public static double[][] inputReg()	Menerima input titik sampel
public static double[][] matrixReg(double[][] a)	Mencari matriks normal estimation equation

public static double addOneCol(double[][] m, int col)	Menjumlahkan semua elemen dalam satu kolom
public static double multiplyEl(double[][] m, int colSelector, int colAcuan)	Mendapatkan hasil kali pengubah x
public static void hasilRLB(double[] s)	Prosedur untuk menampilkan hasil regresi linear berganda
public static void hasilEstimateRLB(double result, double[] taksiran)	Fungsi untuk menampilkan hasil taksiran regresi linear berganda dengan format yang sesuai

5. SPL.java

   Class ini berisi fungsi dan prosedur yang akan digunakan terkait penyelesaian sistem persamaan linear.

Fungsi / Prosedur	**Deskripsi**
public static Scanner scan	Berisi variabel scanner untuk menampung input user
public static void recursiveGauss (double[][] m)	Prosedur yang dipanggil sebagai bentuk rekursif untuk mencari matriks eselon baris
public static void eselonbaris(double[][] m)	Mencari matriks eselon baris
public static void GaussJ(double[][] m)	Mencari matriks eselon baris tereduksi
public static double[][] getGauss(double[][] matriks)	Dipakai untuk menulis solusi Gauss ke file
public static double[][] kofaktor(double[][] m, int size, int col,int start)	Mencari determinan dengan metode kofaktor
public static double determinan(double[][]m)	Mencari determinan dengan kofaktor kolom pertama
public static double[][] adjoin(double[][] m)	Mencari invers matriks dengan menggunakan adjoin

public static double[][] subMatrix(double[][] m, int size, int startRow, int startCol)	Mencari kofaktor
public static double[][] inverse(double[][] m)	Mencari balikan dari matriks
public static boolean noInv(double[][] m)	Mencari tahu apakah matriks memiliki balikan atau tidak
public static double[][] kaidahCramer(double[][] m)	Mencari solusi SPL dengan Kaidah Cramer
public static double[][] getMatrixA()	Mendapatkan matriks A untuk menyelesaikan SPL dengan Matriks Balikan
public static double[][] getMatrixB()	Mendapatkan matriks B untuk menyelesaikan SPL dengan Matriks Balikan
public static double[][] SPLBalikan(double[][] A, double[][] B)	Mencari solusi SPL dengan Matriks Balikan
public static int idxFirstRowNot0(double[][] m, int idxcol)	Mencari indeks baris yang elemen pertamanya tidak nol
public static int FirstElementNot0(double[][] m, int startrowIdx, int colIdx)	Mengecek indeks elemen pertama yang tidak nol
public static double detReduksiBaris(double[][] m)	Mencari determinan dengan metode Reduksi Baris
public static String[] parameter(double[][] m, boolean gaussJordan)	Mencari solusi parametrik dari SPL yang bersolusi banyak
public static String[] parameterFile(double[][] m, boolean gaussJordan)	Dipakai untuk menulis solusi SPL yang berbentuk parametrik ke file

6. Hilbert.java

Class ini berisi fungsi dan prosedur yang akan digunakan terkait mencari solusi SPL matriks Hilbert dengan Kaidah Cramer.

Fungsi / Prosedur	Deskripsi
public static double[][] copyMatrixHilbert(double[][] mIn)	Menyalin matriks Hilbert
public static double detHilbert(double[][] mIn)	Mencari determinan matriks Hilbert
public static double[][] kaidahCramerBert(double[][] m)	Mencari solusi SPL matriks Hilbert dengan Kaidah Cramer

B. *Package* Main

1. Main.java

   Main merupakan program utama yang menampilkan menu permasalahan yang dapat diselesaikan oleh kalkulator, diantaranya:

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

   dimana pengguna diminta untuk memasukkan pilihan permasalahan yang ingin diselesaikan. Program ini akan berhenti saat pengguna menginput 7 (opsi keluar).

C. *Package* Matrix

1. matrixIO.java

   Class ini berisi fungsi dan prosedur yang akan digunakan terkait input dan output matriks.

Fungsi / Prosedur	Deskripsi
public static Scanner scan	Berisi variabel scanner untuk menampung input user
public static double[][]	Membaca matriks dari keyboard (A

readMatrixSPL()	dan B)
public static double[][] readMatrixKeyboard()	Membaca matriks augmented dari keyboard
public static double[][] readMatrixDet()	Membaca matriks dari keyboard untuk determinan dan inverse
public static String inputFile()	Meminta nama dari file txt
public static double[][] fileToMatrix(String path, int type)	Membaca matriks dari file
public static double[] getTaksiran(String path)	Mendapatkan nilai ab dari file untuk *bicubic spline interpolation*
public static int getCol(String path)	Mendapatkan banyak nCol dari file
public static int[] countRowCol(String path)	Mendapatkan banyak nRow dan nCol dari file
public static int[] countRowColSC(String path)	Mendapatkan banyak nRow dan nCol dari file yang memiliki taksiran
public static int countLine(String path)	Menghitung jumlah baris dalam file
public static void displayMatrix (double[][] m)	Menampilkan matriks pada layar
public static void createFile(String path)	Membuat file baru
public static String matrixString(double[][] m)	Mengubah matrix ke string
public static double[][] splHilbert(int n)	Membuat matriks Hilbert

2. matrixOP.java

   Class ini berisi fungsi dan prosedur yang akan digunakan untuk melakukan operasi terhadap matriks.

Fungsi / Prosedur	**Deskripsi**
public static int getRow(double[][] m)	Mendapatkan jumlah baris dari suatu matriks

public static int getCol(double[][] m)	Mendapatkan jumlah kolom dari suatu matriks
public static int getLastIdxRow(double[][] m)	Mendapatkan indeks baris terakhir dari suatu matriks
public static int getLastIdxCol(double[][] matrix)	Mendapatkan indeks kolom terakhir dari suatu matriks
public static void copyMatrix(double[][] mIn, double[][] mOut, int rowMin, int rowMax, int colMin, int colMax)	Menyalin nilai mIn ke mOut dari range rowmin-rowmax dan colmin-colmax
public static double[][] addMatrix (double[][] m1, double[][] m2)	Mencari jumlah dari elemen-elemen m1 dan m2
public static double[][] substractMatrix (double[][] m1, double[][] m2)	Mencari selisih dari elemen-elemen m1 dan m2
public static double[][] multiplyMatrixMatrix (double[][] m1, double[][] m2)	Mengalikan dua buah matriks
public static double[][] multiplyByConst (double[][] m, int x)	Mengalikan elemen matriks dengan suatu konstanta
public static boolean isMatrixEqual (double[][] m1, double[][] m2)	Mengecek apakah suatu matriks sama
public static boolean isMatrixNotEqual (double[][] m1, double[][] m2)	Mengecek apakah suatu matriks tidak sama
public static boolean isMatrixSizeEqual (double[][] m1, double[][] m2)	Mengecek apakah ukuran suatu matriks sama
public static int countElmt(double[][] m)	Menghitung banyak elemen suatu matriks
public static boolean isSquare (double[][] m)	Mengecek apakah suatu matriks merupakan matriks persegi
public static boolean isSymmetric (double[][] m)	Mengecek apakah suatu matriks simetri

public static boolean isIdentity(double[][] m)	Mengecek apakah suatu matriks merupakan matriks identitas
public static double[][] transpose(double[][] m)	Men-transpose matriks
public static double[][] kali_baris (double[][] m, int row, int k)	Mengalikan baris dengan suatu konstanta
public static double[][] tukar_baris (double[][] m, int row1, int row2)	Menukar baris dari suatu matriks
public static double[][] tambah_baris(double[][] m, int row_asal, int row_yangdiubah)	Menambah baris ke baris lain
public static double[][] kurang_baris(double[][] m, int row_asal, int row_yangdiubah)	Mengurangi baris dari baris lain
public static boolean noSolusi (double[][] m)	Mengecek apakah baris terakhir dari suatu matriks semuanya nol kecuali pada kolom terakhirnya
public static boolean noSolusiJordan (double[][] m, int col, int i)	Mengecek apakah baris terakhir dari suatu matriks semuanya nol kecuali pada kolom terakhirnya (untuk Gauss-Jordan)
public static boolean Nol (double[][] m)	Mengecek apakah baris terakhir dari suatu matriks semuanya nol
public static boolean barisNol(double[][] m, int col, int i)	Mengecek apakah dalam 1 baris ada elemen yang nol
public static boolean KolomNol(double[][] m, int idxCol)	Mengecek apakah dalam 1 kolom ada yang elemennya nol semua
public static void pindahBarisNol (double[][] m, int row, int col)	Memindahkan baris nol ke paling bawah matriks
public static void matriksNoSolusi (double[][] m)	Membuat matriks default yang tidak memiliki solusi
public static int satuUtama(double[][] m, int row, int j)	Mencari indeks satu utama
public static double[][]	Menghapus baris yang isinya

deleteZeroRows(double[][] matriks)	semuanya nol (untuk kasus dimana dia solusi unik tapi ada zero rows dibawah)
Public static double[] hasilSPLGauss (double[][] matriks)	Mengeluarkan hasil SPL Gauss yaitu berupa solusi x1 = …, x2 = …, dst
public static double[] hasilSPLGaussJordan (double[][] matriks)	Mengeluarkan hasil SPL Gauss-Jordan yaitu berupa solusi x1 = …, x2 = …, dst
public static void hasilSPLInverse (double[][] matriks)	Mengeluarkan hasil SPL dengan metode Matriks Balikan yaitu berupa solusi x1 = …, x2 = …, dst
public static double[] getGaussMatrix(double[][] matriks)	Mendapatkan hasil Gauss dalam bentuk matriks
public static double[] hasilSPLGaussFile (double[][] matriks)	Mengeluarkan hasil SPL Gauss yaitu berupa solusi x1 = …, x2 = …, dst untuk digunakan pada output file
public static double[] hasilSPLGaussJordanFile (double[][] matriks)	Mengeluarkan hasil SPL Gauss-Jordan yaitu berupa solusi x1 = …, x2 = …, dst untuk digunakan pada output file
public static double[][] getMatrixA()	Mendapatkan matriks A
public static double[][] getMatrixB()	Mendapatkan matriks B
public static double[][] getABalikan(double [][] m)	Mendapatkan matriks A dari matriks augmented
public static double[][] getBBalikan(double [][] m)	Mendapatkan matriks B dari matriks augmented

3. outputFile.java

   Class ini berisi fungsi dan prosedur yang akan digunakan untuk meng-output solusi dalam bentuk file.

Fungsi / Prosedur	**Deskripsi**
public static String getPathOut(String name)	Mendapatkan directory untuk file
public static boolean isNoSolution	Mengecek apakah SPL memiliki

	solusi
public static boolean isSolutionBanyak (double[] s)	Mengecek apakah SPL memiliki banyak solusi
public static void fileInterpolasi(double[] taksiran, double[][] matrix)	Membuat file baru yang berisi solusi dari persoalan Interpolasi
public static void fileGauss(double[][] matrix)	Membuat file baru yang berisi solusi dari SPL Gauss
public static void fileGaussJordan(double[][] matrix)	Membuat file baru yang berisi solusi dari SPL Gauss-Jordan
public static void fileRLB(double[]s,double x,double[] taksiran)	Membuat file baru yang berisi solusi dari persoalan Regresi Linear Berganda
public static void fileDeterminan(double[][] matrix, int type)	Membuat file baru yang berisi determinan dari matriks input
public static void fileInverse(double[][] matrix)	Membuat file baru yang berisi balikan dari matriks input
public static void fileBicubic(double[][] matrix , double[] taksiran)	Membuat file baru yang berisi solusi dari persoalan Interpolasi Bicubic Spline
public static void fileCrammer(double[][] matrix)	Membuat file baru yang berisi solusi dari SPL Kaidah Cramer
public static void fileSPLInverse(double[][] solution)	Membuat file baru yang berisi solusi dari SPL dengan Matriks Balikan
public static void fileSPLInverseNO()	Membuat file baru yang berisi pesan SPL tidak dapat dikerjakan menggunakan SPL Inverse
Public static void fileCrammerHilbert	

BAB IV
EKSPERIMEN

*Untuk persoalan SPL, determinan, matriks balikan, interpolasi, dan regresi, kolom tabel di sebelah kanan merepresentasikan program yang membaca input dari keyboard, dan di sebelah kiri dari file. File output dapat diakses pada folder src/test/output.

1. Solusi SPL Ax=b, untuk kasus

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

<table>
<tr><th colspan="2">ELIMINASI GAUSS</th></tr>
<tr>
<td><pre>Matriks eselon baris:
1.0 1.0 -1.0 -1.0 1.0
0.0 1.0 -1.0 -1.6666666666666667 -0.6666666666666666
0.0 0.0 1.0 -1.0 1.0
0.0 0.0 0.0 0.0 1.0
Matriks tidak memiliki solusi.</pre></td>
<td><pre>-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 1.000 -1.000 -1.000 1.000
0.000 1.000 -1.000 -1.667 -0.667
0.000 0.000 1.000 -1.000 1.000
0.000 0.000 0.000 0.000 1.000
Matrix di atas tidak memiliki solusi.</pre></td>
</tr>
<tr><th colspan="2">ELIMINASI GAUSS JORDAN</th></tr>
<tr>
<td><pre>Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.6666666666666667 0.0
0.0 1.0 0.0 -2.666666666666667 0.0
0.0 0.0 1.0 -1.0 0.0
0.0 0.0 0.0 0.0 1.0
Matriks tidak memiliki solusi.</pre></td>
<td><pre>-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.667 0.000
0.000 1.000 0.000 -2.667 0.000
0.000 0.000 1.000 -1.000 0.000
0.000 0.000 0.000 0.000 1.000
Matrix di atas tidak memiliki solusi.</pre></td>
</tr>
<tr><th colspan="2">KAIDAH CRAMER</th></tr>
</table>

```
Determinan matriks bernilai 0 sehingga tidak dapat menggunakan kaidah Cramer.
Apakah anda ingin menyimpan jawaban?
```

```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
1.000 1.000 -1.000 -1.000 1.000
2.000 5.000 -7.000 -5.000 -2.000
2.000 -1.000 1.000 3.000 4.000
5.000 2.000 -4.000 2.000 6.000
Solusi:
Matrix di atas tidak bisa diselesaikan dengan kaidah cramer
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nama file lengkap dengan .txt: soal1a.txt
Matriks tidak memiliki inverse, sehingga tidak memiliki solusi.
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
Determinan matriks = 0, sehingga matriks tidak memiliki inverse.
```

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

ELIMINASI GAUSS

```
Matriks eselon baris:
1.0 -1.0 0.0 0.0 1.0 3.0
0.0 1.0 0.0 1.0 -3.0 -1.0
0.0 0.0 0.0 1.0 -1.0 -1.0
0.0 0.0 0.0 0.0 0.0 0.0
Matriks memiliki banyak solusi.
x1 = 3.000 + 1.000x5
x2 = 2.000x5
x3 = c
x4 = -1.000 + 1.000x5
x5 = e
```

```
-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 -1.000 3.000
0.000 1.000 0.000 0.000 -2.000 0.000
0.000 0.000 0.000 1.000 -1.000 -1.000
0.000 0.000 0.000 0.000 0.000 0.000
Matrix memiliki banyak solusi.
x1 = 3.000 + 1.000x5
x2 = 2.000x5
x3 = c
x4 = -1.000 + 1.000x5
x5 = e
```

ELIMINASI GAUSS JORDAN

```
Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.0 -1.0 3.0
0.0 1.0 0.0 0.0 -2.0 0.0
0.0 0.0 0.0 1.0 -1.0 -1.0
0.0 0.0 0.0 0.0 0.0 0.0
Matriks memiliki banyak solusi.
x1 = 3.000 + 1.000x5
x2 = 2.000x5
x3 = c
x4 = -1.000 + 1.000x5
x5 = e
Apakah anda ingin menyimpan jawaban?
1. Yes
2. No
```

```
-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 -1.000 3.000
0.000 1.000 0.000 0.000 -2.000 0.000
0.000 0.000 0.000 1.000 -1.000 -1.000
0.000 0.000 0.000 0.000 0.000 0.000
Matrix memiliki banyak solusi.
x1 = 3.000 + 1.000x5
x2 = 2.000x5
x3 = c
x4 = -1.000 + 1.000x5
x5 = e
```

KAIDAH CRAMER

```
Masukkan nama file lengkap dengan .txt: soal1b.txt
Tidak dapat menggunakan kaidah Cramer.
```

```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
1.000 -1.000 0.000 0.000 1.000 3.000
1.000 1.000 0.000 -3.000 0.000 6.000
2.000 -1.000 0.000 1.000 -1.000 5.000
-1.000 2.000 0.000 -2.000 -1.000 -1.000
Solusi:
Matrix di atas tidak bisa diselesaikan dengan kaidah cramer
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nama file lengkap dengan .txt: soal1b.txt
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.
```

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

ELIMINASI GAUSS

```
Matriks eselon baris:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 0.0 0.0 0.0 1.0 -1.0 1.0
Matriks memiliki banyak solusi.
x1 = a
x2 = 1.000 - 1.000x6
x3 = c
x4 = -2.000 - 1.000x6
x5 = 1.000 + 1.000x6
x6 = f
```

```
-----HASIL GAUSS-----
Hasil Eselon Baris:
0.000 1.000 0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 1.000 0.000 1.000 -2.000
0.000 0.000 0.000 0.000 1.000 -1.000 1.000
Matrix memiliki banyak solusi.
x1 = a
x2 = 1.000 - 1.000x6
x3 = c
x4 = -2.000 - 1.000x6
x5 = 1.000 + 1.000x6
x6 = f
```

ELIMINASI GAUSS JORDAN

```
Matriks eselon baris tereduksi:
0.0 1.0 0.0 0.0 0.0 1.0 1.0
0.0 0.0 0.0 1.0 0.0 1.0 -2.0
0.0 0.0 0.0 0.0 1.0 -1.0 1.0
Matriks memiliki banyak solusi.
x1 = a
x2 = 1.000 - 1.000x6
x3 = c
x4 = -2.000 - 1.000x6
x5 = 1.000 + 1.000x6
x6 = f
```

```
-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
0.000 1.000 0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 1.000 0.000 1.000 -2.000
0.000 0.000 0.000 0.000 1.000 -1.000 1.000
Matrix memiliki banyak solusi.
x1 = a
x2 = 1.000 - 1.000x6
x3 = c
x4 = -2.000 - 1.000x6
x5 = 1.000 + 1.000x6
x6 = f
```

KAIDAH CRAMER

```
Masukkan nama file lengkap dengan .txt: soal1c.txt
Tidak dapat menggunakan kaidah Cramer.
```

```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
0.000 1.000 0.000 0.000 1.000 0.000 2.000
0.000 0.000 0.000 1.000 1.000 0.000 -1.000
0.000 1.000 0.000 0.000 0.000 1.000 1.000
Solusi:
Matrix di atas tidak bisa diselesaikan dengan kaidah cramer
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nama file lengkap dengan .txt: soal1c.txt
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.
```

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

Matriks Hilbert dengan n = 6

ELIMINASI GAUSS

```
Matriks eselon baris:
1.0 0.5 0.3333333333333333 0.25 0.2 0.16666666666666666 1.0
0.0 1.0 1.0666666666666667 0.9999999999999999 0.9142857142857141 0.8333333333333333 -3.9999999999999996
0.0 0.0 1.0 1.5000000000000013 1.714285714285721 1.7857142857142927 30.000000000000206
0.0 0.0 0.0 1.0 1.9999999999999005 2.77777777777766 -139.9999999999921
0.0 0.0 0.0 0.0 1.0 2.500000000000325 629.999999999344
0.0 0.0 0.0 0.0 0.0 1.0 -2772.0000002617608
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

```
-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 0.500 0.333 0.250 0.200 0.167 1.000
0.000 1.000 1.067 1.000 0.914 0.833 -4.000
0.000 0.000 1.000 1.500 1.714 1.786 30.000
0.000 0.000 0.000 1.000 2.000 2.778 -140.000
0.000 0.000 0.000 0.000 1.000 2.500 630.000
0.000 0.000 0.000 0.000 0.000 1.000 -2772.000
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

ELIMINASI GAUSS JORDAN

```
Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.0 0.0 0.0 36.00000000101636
0.0 1.0 0.0 0.0 0.0 0.0 -630.0000000309496
0.0 0.0 1.0 0.0 0.0 0.0 3360.0000002177785
0.0 0.0 0.0 1.0 0.0 0.0 -7560.000000581748
0.0 0.0 0.0 0.0 1.0 0.0 7560.000000654647
0.0 0.0 0.0 0.0 0.0 1.0 -2772.0000002617608
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

```
-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 0.000 0.000 36.000
0.000 1.000 0.000 0.000 0.000 0.000 -630.000
0.000 0.000 1.000 0.000 0.000 0.000 3360.000
0.000 0.000 0.000 1.000 0.000 0.000 -7560.000
0.000 0.000 0.000 0.000 1.000 0.000 7560.000
0.000 0.000 0.000 0.000 0.000 1.000 -2772.000
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

KAIDAH CRAMER

```
Masukkan nilai n untuk matrix Hilbert: 6
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
1.000 0.500 0.333 0.250 0.200 0.167 1.000
0.500 0.333 0.250 0.200 0.167 0.143 0.000
0.333 0.250 0.200 0.167 0.143 0.125 0.000
0.250 0.200 0.167 0.143 0.125 0.111 0.000
0.200 0.167 0.143 0.125 0.111 0.100 0.000
0.167 0.143 0.125 0.111 0.100 0.091 0.000
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nilai n untuk matrix Hilbert: 6
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
x1 = 36.000
x2 = -630.000
x3 = 3360.000
x4 = -7560.000
x5 = 7560.000
x6 = -2772.000
```

Matriks Hilbert untuk n = 10

ELIMINASI GAUSS

```
0.0 0.0 0.0 0.0 1.0 2.500000000000325 4.090909090908357 5.568181818181584 6.853146853145942 7.930069930068012 629.999999999344
0.0 0.0 0.0 0.0 0.0 1.0 3.0000000001336744 5.65384615417665 8.615384615920608 11.63076923155209 -2772.0000002617608
0.0 0.0 0.0 0.0 0.0 0.0 1.0 3.500000002310575 7.466666676277195 12.600000019428336 12012.000047844705
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 3.9999999196694342 9.529411575381463 -51480.00040078561
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 4.499995093094009 218789.23231633098
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -923655.110237102
Solusi:
x1 = 99.997
x2 = -4949.736
x3 = 79194.355
x4 = -600548.503
x5 = 2522273.711
x6 = -6305621.497
x7 = 9608484.824
x8 = -8750520.734
x9 = 4375232.696
x10 = -923655.110
```

```
-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 0.500 0.333 0.250 0.200 0.167 0.143 0.125 0.111 0.100 1.000
0.000 1.000 1.067 1.000 0.914 0.833 0.762 0.700 0.646 0.600 -4.000
0.000 0.000 1.000 1.500 1.714 1.786 1.786 1.750 1.697 1.636 30.000
0.000 0.000 0.000 1.000 2.000 2.778 3.333 3.712 3.960 4.112 -140.000
0.000 0.000 0.000 0.000 1.000 2.500 4.091 5.568 6.853 7.930 630.000
0.000 0.000 0.000 0.000 0.000 1.000 3.000 5.654 8.615 11.631 -2772.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 3.500 7.467 12.600 12012.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 4.000 9.529 -51480.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 4.500 218789.232
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 -923655.110
Solusi:
x1 = 99.997
x2 = -4949.736
x3 = 79194.355
x4 = -600548.503
x5 = 2522273.711
x6 = -6305621.497
x7 = 9608484.824
x8 = -8750520.734
x9 = 4375232.696
x10 = -923655.110
```

ELIMINASI GAUSS JORDAN

```
Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 99.99696425561706
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -4949.73612323511
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 79194.35458281764
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 -600548.5029340824
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 2522273.7114662006
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 -6305621.497321621
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 9608484.823872551
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 -8750520.734160174
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 4375232.696094496
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -923655.110237102
Solusi:
x1 = 99.997
x2 = -4949.736
x3 = 79194.355
x4 = -600548.503
x5 = 2522273.711
x6 = -6305621.497
x7 = 9608484.824
x8 = -8750520.734
x9 = 4375232.696
x10 = -923655.110
```

```
-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 99.997
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -4949.736
0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 79194.355
0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 -600548.503
0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 2522273.711
0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 -6305621.497
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 9608484.824
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 -8750520.734
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 4375232.696
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 -923655.110
Solusi:
x1 = 99.997
x2 = -4949.736
x3 = 79194.355
x4 = -600548.503
x5 = 2522273.711
x6 = -6305621.497
x7 = 9608484.824
x8 = -8750520.734
x9 = 4375232.696
x10 = -923655.110
```

KAIDAH CRAMER

```
Masukkan nilai n untuk matrix Hilbert: 10
Solusi:
x1 = 99.992
x2 = -4949.596
x3 = 79193.356
x4 = -600537.912
x5 = 2522224.470
x6 = -6305485.557
x7 = 9608261.783
x8 = -8750305.214
x9 = 4375119.566
x10 = -923630.235
```

```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
1.000 0.500 0.333 0.250 0.200 0.167 0.143 0.125 0.111 0.100 1.000
0.500 0.333 0.250 0.200 0.167 0.143 0.125 0.111 0.100 0.091 0.000
0.333 0.250 0.200 0.167 0.143 0.125 0.111 0.100 0.091 0.083 0.000
0.250 0.200 0.167 0.143 0.125 0.111 0.100 0.091 0.083 0.077 0.000
0.200 0.167 0.143 0.125 0.111 0.100 0.091 0.083 0.077 0.071 0.000
0.167 0.143 0.125 0.111 0.100 0.091 0.083 0.077 0.071 0.067 0.000
0.143 0.125 0.111 0.100 0.091 0.083 0.077 0.071 0.067 0.062 0.000
0.125 0.111 0.100 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.000
0.111 0.100 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.056 0.000
0.100 0.091 0.083 0.077 0.071 0.067 0.062 0.059 0.056 0.053 0.000
Solusi:
x1 = 99.992
x2 = -4949.596
x3 = 79193.356
x4 = -600537.912
x5 = 2522224.470
x6 = -6305485.557
x7 = 9608261.783
x8 = -8750305.214
x9 = 4375119.566
x10 = -923630.235
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nilai n untuk matrix Hilbert: 10
Solusi:
x1 = 99.997
x2 = -4949.736
x3 = 79194.355
x4 = -600548.503
x5 = 2522273.711
x6 = -6305621.497
x7 = 9608484.824
x8 = -8750520.734
x9 = 4375232.696
x10 = -923655.110
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
x1 = 99.997
x2 = -4949.736
x3 = 79194.355
x4 = -600548.503
x5 = 2522273.711
x6 = -6305621.497
x7 = 9608484.824
x8 = -8750520.734
x9 = 4375232.696
x10 = -923655.110
```

2. Matrix Augmented

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

ELIMINASI GAUSS

<table>
<tr><td>Matriks eselon baris:
1.0 -1.0 2.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
Matriks memiliki banyak solusi.
x1 = -1.000 + 1.000x4
x2 = 2.000x3
x3 = c
x4 = d</td><td>-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 0.000 0.000 -1.000 -1.000
0.000 1.000 -2.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
Matrix memiliki banyak solusi.
x1 = -1.000 + 1.000x4
x2 = 2.000x3
x3 = c
x4 = d</td></tr>
<tr><td colspan="2">ELIMINASI GAUSS JORDAN</td></tr>
<tr><td>Matriks eselon baris tereduksi:
1.0 0.0 0.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
Matriks memiliki banyak solusi.
x1 = -1.000 + 1.000x4
x2 = 2.000x3
x3 = c
x4 = d</td><td>-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 -1.000 -1.000
0.000 1.000 -2.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
Matrix memiliki banyak solusi.
x1 = -1.000 + 1.000x4
x2 = 2.000x3
x3 = c
x4 = d</td></tr>
<tr><td colspan="2">KAIDAH CRAMER</td></tr>
<tr><td>Masukkan nama file lengkap dengan .txt: soal2a.txt
Determinan matriks bernilai 0 sehingga tidak dapat menggunakan kaidah Cramer.</td><td>-----HASIL SPL DENGAN KAIDAH CRAMMER-----
1.000 -1.000 2.000 -1.000 -1.000
2.000 1.000 -2.000 -2.000 -2.000
-1.000 2.000 -4.000 1.000 1.000
3.000 0.000 0.000 -3.000 -3.000
Solusi:
Matrix di atas tidak bisa diselesaikan dengan kaidah cramer</td></tr>
<tr><td colspan="2">SPL DENGAN MATRIX BALIKAN</td></tr>
<tr><td>Masukkan nama file lengkap dengan .txt: soal2a.txt
Determinan matriks = 0, sehingga matriks tidak memiliki inverse.</td><td>-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
Determinan matriks = 0, sehingga matriks tidak memiliki inverse.</td></tr>
</table>

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

ELIMINASI GAUSS	
Matriks eselon baris: 1.0 0.0 4.0 0.0 4.0 0.0 1.0 0.0 -1.5 0.5 0.0 0.0 1.0 0.0 1.0 0.0 0.0 0.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 Solusi: x1 = 0.000 x2 = 2.000 x3 = 1.000 x4 = 1.000	-----HASIL GAUSS----- Hasil Eselon Baris: 1.000 0.000 4.000 0.000 4.000 0.000 1.000 0.000 -1.500 0.500 0.000 0.000 1.000 0.000 1.000 0.000 0.000 0.000 1.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 Solusi: x1 = 0.000 x2 = 2.000 x3 = 1.000 x4 = 1.000
ELIMINASI GAUSS JORDAN	

<table>
<tr><td>Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 2.0
0.0 0.0 1.0 0.0 1.0
0.0 0.0 0.0 1.0 1.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
Solusi:
x1 = 0.000
x2 = 2.000
x3 = 1.000
x4 = 1.000</td><td>-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 2.000
0.000 0.000 1.000 0.000 1.000
0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
Solusi:
x1 = 0.000
x2 = 2.000
x3 = 1.000
x4 = 1.000</td></tr>
<tr><td colspan="2">KAIDAH CRAMER</td></tr>
<tr><td>Masukkan nama file lengkap dengan .txt: soal2b.txt
Tidak dapat menggunakan kaidah Cramer.</td><td>-----HASIL SPL DENGAN KAIDAH CRAMMER-----
2.000 0.000 8.000 0.000 8.000
0.000 1.000 0.000 4.000 6.000
-4.000 0.000 6.000 0.000 6.000
0.000 -2.000 0.000 3.000 -1.000
2.000 0.000 -4.000 0.000 -4.000
0.000 1.000 0.000 -2.000 0.000
Solusi:
Matrix di atas tidak bisa diselesaikan dengan kaidah cramer</td></tr>
<tr><td colspan="2">SPL DENGAN MATRIX BALIKAN</td></tr>
<tr><td>Masukkan nama file lengkap dengan .txt: soal2b.txt
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.</td><td>-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.</td></tr>
</table>

3. **SPL berbentuk**

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

ELIMINASI GAUSS

```
Matriks eselon baris:
1.0 0.125 0.375 0.25 0.0
0.0 1.0 0.5652173913043478 -0.43478260869565216 0.6956521739130435
0.0 0.0 1.0 -0.19480519480519481 0.7597402597402597
0.0 0.0 0.0 1.0 -0.2581081081081081
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

```
-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 0.125 0.375 0.250 0.000
0.000 1.000 0.565 -0.435 0.696
0.000 0.000 1.000 -0.195 0.760
0.000 0.000 0.000 1.000 -0.258
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

ELIMINASI GAUSS JORDAN

```
Masukkan nama file lengkap dengan .txt: soal3a.txt
Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.0 -0.2243243243243243
0.0 1.0 0.0 0.0 0.18243243243243246
0.0 0.0 1.0 0.0 0.7094594594594594
0.0 0.0 0.0 1.0 -0.2581081081081081
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

```
-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 -0.224
0.000 1.000 0.000 0.000 0.182
0.000 0.000 1.000 0.000 0.709
0.000 0.000 0.000 1.000 -0.258
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

KAIDAH CRAMER

```
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
8.000 1.000 3.000 2.000 0.000
2.000 9.000 -1.000 -2.000 1.000
1.000 3.000 2.000 -1.000 2.000
1.000 0.000 6.000 4.000 3.000
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nama file lengkap dengan .txt: soal3a.txt
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

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

ELIMINASI GAUSS

```
Matriks eselon baris:
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0
0.0 1.0 3.65684 1.0 3.65684 1.0 3.65684 1.0 0.0 57.24
0.0 0.0 1.0 0.0 1.0 17.48659361156447 0.0 16.48659361156447 13.31475868500816 331.8356260200513
0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 15.0
0.0 0.0 0.0 0.0 1.0 -1.6154626809719338E17 0.0 -1.615462680971934E17 -1.3224392901460379E17 -3.0840057245651267E18
0.0 0.0 0.0 0.0 0.0 1.0 0.0 1.0000000000000002 0.8186133333333333 19.090541433675536
0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 13.0
0.0 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 0.0 0.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Matriks tidak memiliki solusi.
```

```
-----HASIL GAUSS-----
Hasil Eselon Baris:
1.000 1.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 8.000
0.000 1.000 3.657 1.000 3.657 1.000 3.657 1.000 0.000 57.240
0.000 0.000 1.000 0.000 1.000 17.487 0.000 16.487 13.315 331.836
0.000 0.000 0.000 1.000 1.000 1.000 0.000 0.000 0.000 15.000
0.000 0.000 0.000 0.000 1.000 -161546268097193376.000 0.000 -161546268097193408.000 -132243929014603792.000 -3084005724565126700.000
0.000 0.000 0.000 0.000 0.000 1.000 0.000 1.000 0.819 19.091
0.000 0.000 0.000 0.000 0.000 0.000 1.000 1.000 1.000 13.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
Matrix di atas tidak memiliki solusi.
```

ELIMINASI GAUSS JORDAN

```
Masukkan nama file lengkap dengan .txt: soal3b.txt
Matriks eselon baris tereduksi:
1.0 0.0 0.0 0.0 0.0 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 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 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 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0 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 0.0 0.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Matriks tidak memiliki solusi.
```

```
-----HASIL GAUSS JORDAN-----
Hasil Eselon Baris:
1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
Matrix di atas tidak memiliki solusi.
```

KAIDAH CRAMER

```
Masukkan nama file lengkap dengan .txt: soal3b.txt
Tidak dapat menggunakan kaidah Cramer.
Apakah anda ingin menyimpan jawaban?
```

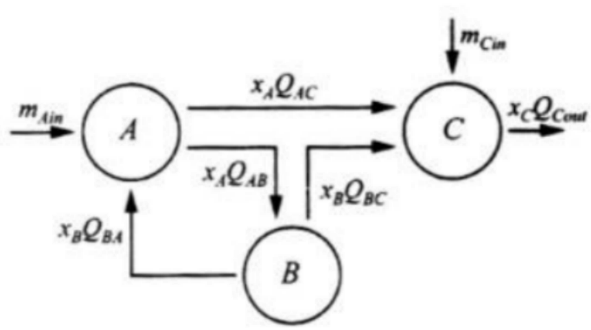
```
-----HASIL SPL DENGAN KAIDAH CRAMMER-----
0.000 0.000 0.000 0.000 0.000 0.000 1.000 1.000 1.000 13.00
0.000 0.000 0.000 1.000 1.000 1.000 0.000 0.000 0.000 15.00
1.000 1.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 8.000
0.000 0.000 0.043 0.000 0.043 0.750 0.043 0.750 0.614 14.79
0.000 0.250 0.914 0.250 0.914 0.250 0.914 0.250 0.000 14.31
0.614 0.750 0.043 0.750 0.043 0.000 0.043 0.000 0.000 3.810
0.000 0.000 1.000 0.000 0.000 1.000 0.000 0.000 1.000 18.00
0.000 1.000 0.000 0.000 1.000 0.000 0.000 1.000 0.000 12.00
1.000 0.000 0.000 1.000 0.000 0.000 1.000 0.000 0.000 6.000
0.043 0.750 0.614 0.000 0.043 9.750 0.000 0.000 0.043 10.51
0.914 0.250 0.000 0.250 0.914 0.250 0.000 0.000 0.914 16.13
Solusi:
Matrix di atas tidak bisa diselesaikan dengan kaidah cramer
```

SPL DENGAN MATRIX BALIKAN

```
Masukkan nama file lengkap dengan .txt: soal3b.txt
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.
```

```
-----HASIL SPL DENGAN MATRIX BALIKAN-----
Solusi:
Matriks tidak berukuran n x n, sehingga inverse tidak dapat dicari.
```

4. **Soal**



ELIMINASI GAUSS

Matriks eselon baris: 1.0 -0.5 0.0 10.833333333333334 0.0 1.0 2.5 -17.77777777777778 0.0 0.0 1.0 -10.0 Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000	-----HASIL GAUSS----- Hasil Eselon Baris: 1.000 -0.500 0.000 10.833 0.000 1.000 2.500 -17.778 0.000 0.000 1.000 -10.000 Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000
ELIMINASI GAUSS JORDAN	
Matriks eselon baris tereduksi: 1.0 0.0 0.0 14.444444444444445 0.0 1.0 0.0 7.222222222222221 0.0 0.0 1.0 -10.0 Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000	-----HASIL GAUSS JORDAN----- Hasil Eselon Baris: 1.000 0.000 0.000 14.444 0.000 1.000 0.000 7.222 0.000 0.000 1.000 -10.000 Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000
KAIDAH CRAMER	
Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000	-----HASIL SPL DENGAN KAIDAH CRAMMER----- -120.000 60.000 0.000 -1300.000 40.000 -80.000 0.000 0.000 80.000 20.000 150.000 -200.000 Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000
SPL DENGAN MATRIX BALIKAN	

Masukkan nama file lengkap dengan .txt: soal4.txt Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000	-----HASIL SPL DENGAN MATRIX BALIKAN----- Solusi: x1 = 14.444 x2 = 7.222 x3 = -10.000

5. **Studi Kasus Interpolasi Polinom**

    a. Program menerima masukan nilai x yang akan ditentukan nilai fungsi f(x) nya dengan interpolasi.

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

X = 0.2
Hasil interpolasi: f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023 Hasil taksiran: f(0.200)= 0.033
-----HASIL INTERPOLASI----- f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023 Hasil taksiran: f(0.200)= 0.033
X = 0.55
Hasil interpolasi: f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023 Hasil taksiran: f(0.550)= 0.171
-----HASIL INTERPOLASI----- f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023 Hasil taksiran: f(0.550)= 0.171
X = 0.85

```
Hasil interpolasi:
f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023
Hasil taksiran:
f(0.850)= 0.337
```

```
-----HASIL INTERPOLASI-----
f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023
Hasil taksiran:
f(0.850)= 0.337
```

X = 1.28

```
Hasil interpolasi:
f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023
Hasil taksiran:
f(1.280)= 0.678
```

```
-----HASIL INTERPOLASI-----
f(x)= -0.000x^6 + 0.000x^5 + 0.026x^4 + 0.000x^3 + 0.197x^2 + 0.240x - 0.023
Hasil taksiran:
f(1.280)= 0.678
```

b. Berikut data 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

Data tanggal dalam bentuk desimal memanfaatkan interpolasi polinomial untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

16/07/2022

```
Hasil interpolasi:
f(x)= -141038.743x^9 + 9376036.418x^8 - 275574683.925x^7 + 4697639766.821x^6 - 51153430767.727x^5 + 368719537926.256x^4 - 1757689745151.114x^3 + 533714
7086320.320x^2 - 9352734583385.234x + 7192036706500.712
Hasil taksiran:
f(7.516)= 53533.357
```

```
-----HASIL INTERPOLASI-----
f(x)= -141038.743x^9 + 9376036.418x^8 - 275574683.925x^7 + 4697639766.821x^6 - 51153430767.727x^5 + 368719537926.256x^4 - 17576897451
Hasil taksiran:
f(7.516)= 53533.357
```

10/08/2022

```
Hasil interpolasi:
f(x)= -141038.743x^9 + 9376036.418x^8 - 275574683.925x^7 + 4697639766.821x^6 - 51153430767.727x^5 + 368719537926.256x^4 - 1757689745151.114x^3 + 533714
7086320.320x^2 - 9352734583385.234x + 7192036706500.712
Hasil taksiran:
f(8.323)= 36295.137
```

```
-----HASIL INTERPOLASI-----
f(x)= -141038.743x^9 + 9376036.418x^8 - 275574683.925x^7 + 4697639766.821x^6 - 51153430767.727x^5 + 368719537926.256x^4 - 1757689745151.114x^3 + 5337147086320.320x^2 - 9352734583385.234x + 7192036706500.712
Hasil taksiran:
f(8.323)= 36295.137
```

05/09/2022

```
Hasil interpolasi:
f(x)= -141038.743x^9 + 9376036.418x^8 - 275574683.925x^7 + 4697639766.821x^6 - 51153430767.727x^5 + 368719537926.256x^4 - 1757689745151.114x^3 + 533714
7086320.320x^2 - 9352734583385.234x + 7192036706500.712
Hasil taksiran:
f(9.167)= -667741.953
```

```
-----HASIL INTERPOLASI-----
f(x)= -141038.743x^9 + 9376036.418x^8 - 275574683.925x^7 + 4697639766.821x^6 - 51153430767.727x^5 + 368719537926.256x^4 - 17576897451
Hasil taksiran:
f(9.167)= -667741.953
```

18/11/2022

```
Hasil interpolasi:
f(x)= 0.236x^5 - 1.421x^4 + 3.237x^3 - 3.553x^2 + 2.035x
Hasil taksiran:
f(11.600)= 28486.173
```

```
-----HASIL INTERPOLASI-----
f(x)= 0.236x^5 - 1.421x^4 + 3.237x^3 - 3.553x^2 + 2.035x
Hasil taksiran:
f(11.600)= 28486.173
```

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

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

f(0.5)
Hasil interpolasi: f(x)= 0.236x^5 - 1.421x^4 + 3.237x^3 - 3.553x^2 + 2.035x Hasil taksiran: f(0.500)= 0.453
-----HASIL INTERPOLASI----- f(x)= 0.236x^5 - 1.421x^4 + 3.237x^3 - 3.553x^2 + 2.035x Hasil taksiran: f(0.500)= 0.453

Catatan : karena dilakukan rounding, output akan terlihat 0.000 seperti angkat 0, tetapi hasil interpolasi sesungguhnya merupakan angka yang mendekati nilai 0. (Contoh kasus 5A)

6. Regresi Linear Berganda

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.

f(0.5)

```
Hasil regresi linear berganda:
f(x) = -3.508 - 0.003x1 + 0.001x2 + 0.154x3
Hasil taksiran:
f(50.000,75.000,29.300)= 0.938
```

```
-----HASIL REGRESI LINEAR BERGANDA-----
f(x)= -3.508 - 0.003x1 + 0.001x2 + 0.154x3
Hasil taksiran:
f(50.000,75.000,29.300)= 0.938
```

*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.

7. Bicubic Spline Interpolation

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

***f*(0, 0)**

```
Masukkan nama file lengkap dengan .txt: soalBicubic1.txt
f(0.0,0.0)= 21.0
Apakah anda ingin menyimpan jawaban?
```

```
-----HASIL BICUBIC SPLINE INTERPOLATION-----
21.000 98.000 125.000 153.000
51.000 101.000 161.000 59.000
0.000 42.000 72.000 210.000
16.000 12.000 81.000 96.000
Hasil taksiran matrix di atas adalah:
f(0.000,0.000)= 21.0
```

f(0.5, 0.5)

```
Masukkan nama file lengkap dengan .txt: soalBicubic2.txt
f(0.5,0.5)= 87.796875
```

```
-----HASIL BICUBIC SPLINE INTERPOLATION-----
21.000 98.000 125.000 153.000
51.000 101.000 161.000 59.000
0.000 42.000 72.000 210.000
16.000 12.000 81.000 96.000
Hasil taksiran matrix di atas adalah:
f(0.500,0.500)= 87.796875
```

f(0.25, 0.75)

```
Masukkan nama file lengkap dengan .txt: soalBicubic3.txt
f(0.25,0.75)= 82.148193359375
```

```
-----HASIL BICUBIC SPLINE INTERPOLATION-----
21.000 98.000 125.000 153.000
51.000 101.000 161.000 59.000
0.000 42.000 72.000 210.000
16.000 12.000 81.000 96.000
Hasil taksiran matrix di atas adalah:
f(0.250,0.750)= 82.148193359375
```

f(0.1, 0.9)

```
Masukkan nama file lengkap dengan .txt: soalBicubic4.txt
f(0.1,0.9)= 91.27126700000001
```

```
-----HASIL BICUBIC SPLINE INTERPOLATION-----
21.000 98.000 125.000 153.000
51.000 101.000 161.000 59.000
0.000 42.000 72.000 210.000
16.000 12.000 81.000 96.000
Hasil taksiran matrix di atas adalah:
f(0.100,0.900)= 91.27126700000001
```

*Untuk *bicubic spline interpolation*, masukan hanya 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).

8. Persoalan Determinan

Kasus Matriks Memiliki Determinan

```
test > input > adadet.txt
1    0 1 5
2    3 -6 9
3    2 6 1
```

METODE REDUKSI BARIS

`******************** Metode Reduksi Baris ********************`
`1. Keyboard`
`2. File`
 	
`Masukan pilihan input: 2`
 	
`INPUT SOURCE: FILE`
`Masukkan nama file lengkap dengan .txt: adadet.txt`
`Determinan: 165.000`

`-----HASIL DETERMINAN DENGAN REDUKSI BARIS-----`
`3.000 -6.000 9.000`
`0.000 10.000 -5.000`
`0.000 0.000 5.500`
`Determinan untuk matrix di atas adalah: 165.0`

METODE EKSPANSI KOFAKTOR

```
****************** Metode Ekspansi Kofaktor ******************
1. Keyboard
2. File

Masukan pilihan input: 2

INPUT SOURCE: FILE
Masukkan nama file lengkap dengan .txt: adadet.txt
Determinan: 165.000
```

```
-----HASIL DETERMINAN DENGAN EXPANSI KOFAKTOR-----
0.000 1.000 5.000
3.000 -6.000 9.000
2.000 6.000 1.000
Determinan untuk matrix di atas adalah: 165.0
```

*Untuk metode ekspansi kofaktor, baris dan kolom yang dijadikan acuan untuk menentukan determinan adalah baris dan kolom ke-1

Kasus Matriks Tidak Memiliki Determinan

```
test > input > nodet.txt
1 2 3 4
5 6 7 8
9 10 11 12
```

*matriks tidak simetri sehingga tidak memiliki determinan

METODE REDUKSI BARIS

```
INPUT SOURCE: FILE
Masukkan nama file lengkap dengan .txt: nodet.txt
Matriks di atas tidak memiliki determinan.
```

```
-----HASIL DETERMINAN DENGAN REDUKSI BARIS-----
1.000 2.000 3.000 4.000
5.000 6.000 7.000 8.000
9.000 10.000 11.000 12.000
Matrix di atas tidak memiliki determinan.
```

METODE EKSPANSI KOFAKTOR

```
****************** Metode Ekspansi Kofaktor ******************
1. Keyboard
2. File

Masukan pilihan input: 2

INPUT SOURCE: FILE
Masukkan nama file lengkap dengan .txt: nodet.txt
Matriks di atas tidak memiliki determinan.
```

```
-----HASIL DETERMINAN DENGAN EXPANSI KOFAKTOR-----
1.000 2.000 3.000 4.000
5.000 6.000 7.000 8.000
9.000 10.000 11.000 12.000
Matrix di atas tidak memiliki determinan.
```

9. Persoalan Invers

Kasus Matriks Memiliki Invers

```
test > input > adainverse.txt
3 1 0
2 1 1
6 2 2
```

```
*************************************************************
********************** MATRIKS BALIKAN **********************
*************************************************************
1. Keyboard
2. File

Masukan pilihan input: 2

INPUT SOURCE: FILE
Masukkan nama file lengkap dengan .txt: adainverse.txt
0.0 -1.0 0.5
1.0 3.0 -1.5
-1.0 0.0 0.5
```

```
-----HASIL INVERSE MATRIX-----
3.000 1.000 0.000
2.000 1.000 1.000
6.000 2.000 2.000
Inverse untuk matrix di atas adalah:
0.000 -1.000 0.500
1.000 3.000 -1.500
-1.000 -0.000 0.500
```

Kasus Matriks Tidak Memiliki Invers

```
test > input > noinverse.txt
-24 6 3
1 2 4
-3 0 -1
```

```
****************************************************************
*********************** MATRIKS BALIKAN ************************
****************************************************************
1. Keyboard
2. File

Masukan pilihan input: 2

INPUT SOURCE: FILE
Masukkan nama file lengkap dengan .txt: noinverse.txt
Matriks di atas tidak memiliki invers.
```

```
-----HASIL INVERSE MATRIX-----
-24.000 6.000 3.000
1.000 2.000 4.000
-3.000 0.000 -1.000
Matrix di atas tidak memiliki inverse.
```

BAB V
PENUTUP

A. Kesimpulan

Setelah mendapatkan materi dari kuliah Aljabar Linier dan Geometri IF2123, kami dapat mengimplementasikan materi tersebut untuk berbagai penyelesaian aljabar linear dalam suatu program bahasa Java. Implementasi tersebut digunakan untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, determinan, matriks balikan, interpolasi, serta regresi linear berganda.

Melalui tugas besar ini, kami lebih memahami penyelesaian masalah dengan menggunakan matriks, seperti menaksir nilai suatu fungsi menggunakan interpolasi bikubik dan polinom, serta menentukan hasil regresi linear berganda.

B. Saran

Saran penulis untuk kedepannya adalah agar masukan dan luaran untuk beberapa masalah sebaiknya lebih diperjelas dalam spek. Hal ini agar memudahkan penulis mengecek apakah program yang dibuat sudah benar/belum.

C. Refleksi

Hikmah yang kami dapatkan dari tugas besar ini yaitu terasahnya kemampuan *problem solving*, *computational thinking,* dekomposisi masalah, komunikasi, serta kerjasama. Kami juga bisa mengetahui lebih dalam mengenai pemrograman dengan bahasa java.

D. Link Repository

Link repository tugas besar 1 mata kuliah IF2123 Aljabar Linier dan Geometri kelompok 13 adalah sebagai berikut : https://github.com/denoseu/Algeo01-22013

DAFTAR PUSTAKA

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/algeo23-24.htm

https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

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