Laporan Tugas Besar I

IF2123 ALJABAR LINEAR DAN GEOMETRI

Kelompok 14 - Geprek Mumbul

Muhammad Alfansya	13523005
Muhammad Raihan Nazhim Oktana	13523021
Dzaky Aurellia Fawwaz	13523065

Institut Teknologi Bandung

2023

BAB I - Deskripsi Masalah

I. Sistem Persamaan Linier (SPL)

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

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

Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

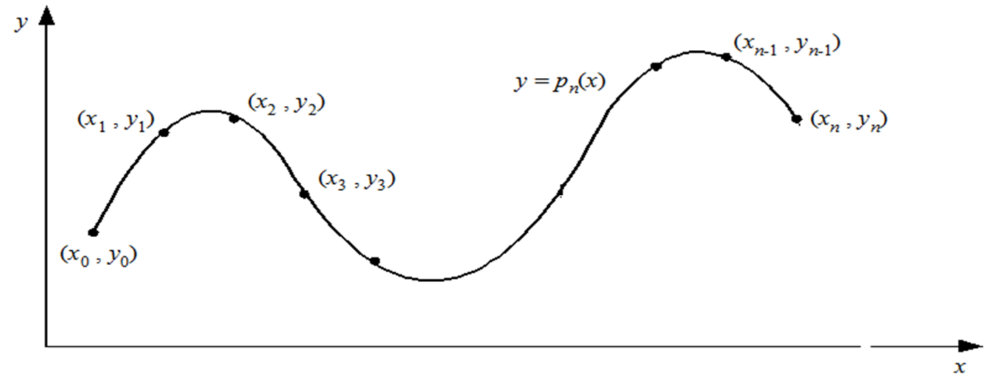
Di dalam Tugas Besar 1 ini, Anda diminta membuat satu atau lebih *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n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

Beberapa tulisan cara membuat library di Java:

1. https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/
2. https://developer.ibm.com/tutorials/j-javalibrary/

II. Interpolasi Polinomial

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$. Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $i = 0, 1, 2, …, n$.



Gambar 2. 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, …, a_n,

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

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

$$
\begin{aligned}
a_0 + 8.0a_1 + 64.00a_2 &= 2.0794 \\
a_0 + 9.0a_1 + 81.00a_2 &= 2.1972 \\
a_0 + 9.5a_1 + 90.25a_2 &= 2.2513
\end{aligned}
$$

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: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

III. Regresi Berganda

Regresi (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Pada tugas besar ini, anda diminta untuk membuat 2 jenis regresi yaitu Regresi Linier Berganda dan Regresi Kuadratik Berganda.

1. Regresi Linier Berganda

   Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat 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 setiap β_i dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:

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

2. Regresi Kuadratik Berganda

   Dalam kasus ini, proses mengubah data-data dalam regresi kuadratik berganda cukup berbeda dengan Regresi Linier Berganda. Bentuk persamaan dari regresi kuadratik ada 3, yaitu:

   a. Variabel Linier: Variabel dengan derajat satu seperti X, Y, dan Z
   b. Variabel Kuadrat: Variabel dengan derajat dua seperti X^2
   c. Variabel Interaksi: 2 Variabel dengan derajat satu yang dikalikan dengan satu sama lain seperti XY, YZ, dan XZ

   Setiap n-peubah, jumlah variabel linier, kuadrat, dan interaksi akan berbeda-beda. Perhatikan contoh regresi kuadratik 2 variabel peubah sebagai berikut!

$$\begin{pmatrix} N & \sum u_i & \sum v_i & \sum u_i^2 & \sum u_i v_i & \sum v_i^2 \\ \sum u_i & \sum u_i^2 & \sum u_i v_i & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i v_i^2 \\ \sum v_i & \sum u_i v_i & \sum v_i^2 & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum v_i^3 \\ \sum u_i^2 & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i^4 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 \\ \sum u_i v_i & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 & \sum u_i v_i^3 \\ \sum v_i^2 & \sum u_i v_i^2 & \sum v_i^3 & \sum u_i^2 v_i^2 & \sum u_i v_i^3 & \sum v_i^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i u_i \\ \sum y_i v_i \\ \sum y_i u_i^2 \\ \sum y_i u_i v_i \\ \sum y_i v_i^2 \end{pmatrix}$$

N menandakan jumlah peubah, terdapat 2 variabel linier yaitu u_i dan v_i, 2 variabel kuadrat yaitu u_i^2 dan v_i^2, dan 1 variabel interaksi yaitu uv. Untuk setiap n-peubah, akan terdapat 1 konstan N (Terlihat di bagian atas kiri gambar), n variabel linier, n variabel kuadrat, dan C_2^n variabel linier (dengan syarat n > 1). Tentu dengan bertambahnya peubah n, ukuran matriks akan bertumbuh lebih besar dibandingkan regresi linier berganda tetapi solusi tetap bisa didapat dengan menggunakan SPL.

Kedua model regresi yang dijadikan sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

IV. Bicubic Spline Interpolation

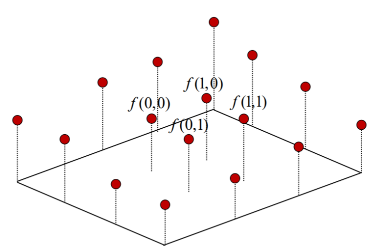
Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. *Bicubic spline interpolation* melibatkan konsep *spline* dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. 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.

Normalization: $f(0,0), f(1,0)$
$f(0,1), f(1,1)$

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

Solve: a_{ij}



Gambar 3. 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.

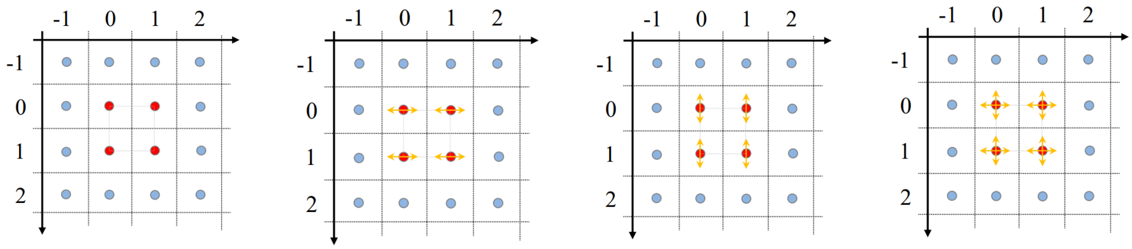
$$y = Xa$$

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

Perlu diketahui bahwa elemen pada matriks *X* adalah nilai dari setiap komponen koefisien a_{ij} yang diperoleh dari persamaan fungsi maupun persamaan turunan yang telah dijelaskan sebelumnya. Sebagai contoh, elemen matriks *X* pada baris 8 kolom ke 2 adalah koefisien dari a_{10} pada ekspansi sigma untuk $f_x(1, 1)$ sehingga diperoleh nilai konstanta 1 x 1^{1-1} x 1^0 = 1, sesuai dengan isi matriks *X*.

Nilai dari vektor *a* dapat dicari dari persamaan $y = Xa$, lalu vektor *a* tersebut digunakan sebagai nilai variabel dalam $f(x, y)$, sehingga terbentuk fungsi interpolasi bicubic sesuai model. Tugas Anda pada studi kasus ini adalah membangun persamaan $f(x, y)$ yang akan digunakan untuk melakukan interpolasi berdasarkan nilai $f(a, b)$ dari masukan matriks

4 x 4. Nilai masukan a dan b berada dalam rentang [0, 1]. Nilai yang akan diinterpolasi dan turunan berarah disekitarnya dapat diilustrasikan pada titik berwarna merah pada gambar di bawah.



Gambar 4. Nilai fungsi yang akan di interpolasi pada titik merah, turunan berarah terhadap sumbu x, terhadap sumbu y, dan keduanya (kiri ke kanan).

Untuk studi kasus ini, buatlah matriks X menggunakan persamaan yang ada (tidak *hardcode*) serta carilah invers matriks X dengan *library* yang telah kalian buat dalam penyelesaian masalah. Berikut adalah sebuah tautan yang dapat dijadikan referensi.

SPESIFIKASI TUGAS

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 invers, metode yang digunakan ada 2 yaitu menggunakan OBE dan Matriks Adjoin

4. 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
```

5. 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.
6. 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 $x_1 = t$).
7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah

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

dan untuk regresi adalah

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

untuk kasus regresi kuadratik, variabel boleh menggunakan x_1, x_2, dan lain-lain tetapi perlu dijelaskan variabel tersebut merepresentasikan apa. Contoh

x_1 = X
.
x_3 = X²
.
x_5 = XY
[Persamaan dan Solusi]

8. 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 $f(0.5, 0.5)$.

9. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam *file*.
10. Bahasa program yang digunakan adalah Java. Anda bebas untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20/21).
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 dan kuadratik berganda
    7. Interpolasi Gambar (Bonus)
    8. 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, 3, dan 6

Bab II - Teori Singkat

2.1 Metode Eliminasi Gauss

Metode Eliminasi merupakan metode penyelesaian persamaan linear dengan mengubah menjadi bentuk matrix lalu mengubahnya menjadi bentuk eselon baris menggunakan operasi baris elementer (OBE) sehingga mempermudah metode substitusi untuk mencari solusi persamaannya.

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

2.2 Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan merupakan lanjutan dari Metode Gauss yaitu melanjutkan dari eselon baris dan mengubah nya hingga menjadi eselon baris tereduksi. Eselon baris tereduksi mempermudah mencari solusi karena tidak diperlukan lagi metode substitusi

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

2.3 Determinan

Determinan adalah nilai yang dapat dihitung dari unsur unsur matrix persegi. Dengan mengetahui determinan suatu matrix bisa diketahui sifat sifat dari matrix tersebut. Nilai determinan bisa didapatkan dengan berbagai metode, antara lain metode ekspansi kofaktor, metode OBE, metode sarrus dan metode sarrus

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{matrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 5 & 0 & 4 \\ 8 & 2 & -3 \end{bmatrix}$$

$$\det(A) = a_{12}C_{12} + a_{22}C_{22} + a_{32}C_{32}$$

$$\det(A) = -(-1)\begin{vmatrix} 5 & 4 \\ 8 & -3 \end{vmatrix} + 0\begin{vmatrix} 3 & 2 \\ 8 & -3 \end{vmatrix} - 2\begin{vmatrix} 3 & 2 \\ 5 & 4 \end{vmatrix}$$

2.4 Matriks Balikan

Matriks balikan adalah sebuah matriks yang jika dikalikan dengan matriks asli nya menghasilkan matriks identitas. Matrix identitas bisa didapat dengan beberapa metode, antara lain dengan rumus $M^{-1} = \frac{1}{det\ M} Adj(A)$ dan dengan metode Gauss Jordan $(A|I) \xrightarrow{GJ} (I|A^{-1})$

2.5 Matriks Kofaktor

Matriks kofaktor merupakan matriks yang terdiri dari kofaktor-kofaktor matriks itu sendiri. Jadi, misalkan terdapat suatu matriks A, maka matriks kofaktor A merupakan matriks yang terdiri dari kofaktor-kofaktor dari matriks A. Susunan elemen matriks kofaktor juga mengikuti susunan (letak) kofaktor-kofaktornya.

2.6 Matriks Adjoin

Matrix adjoin merupakan transpose dari matriks kofaktor. Sedangkan Transpos merupakan pertukaran elemen pada baris menjadi kolom dan kolom menjadi baris. Matrix Adjoin sering dipakai untuk menentukan nilai invers matrix.

2.7 Kaidah Cramer

Kaidah Cramer merupakan salah satu metode untuk mencari solusi dari suatu SPL. Metode Cramer berlaku jika jumlah variabel yang tidak diketahui sama dengan jumlah persamaannya.Metode ini dapat dilakukan dengan mencari determinan dari matriks koefisien yang salah satu kolom nya diganti dengan kolom matriks konstanta dan kemudian dibagi dengan determinan matriks koefisien tersebut, lakukan hal yang sama secara berulang dengan mengganti kolom dari kolom pertama ke kolom terakhir untuk mencari masing masing nilai solusi variabel yang dicari.

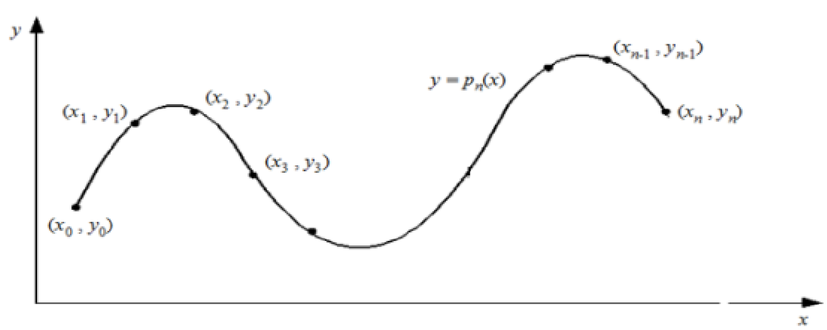
$$A = \begin{bmatrix} -1 & 2 & -3 \\ 2 & 0 & 1 \\ 3 & -4 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} \mathbf{1} & 2 & -3 \\ \mathbf{0} & 0 & 1 \\ \mathbf{2} & -4 & 4 \end{bmatrix} \quad A_2 = \begin{bmatrix} -1 & \mathbf{1} & -3 \\ 2 & \mathbf{0} & 1 \\ 3 & \mathbf{2} & 4 \end{bmatrix} \quad A_3 = \begin{bmatrix} -1 & 2 & \mathbf{1} \\ 2 & 0 & \mathbf{0} \\ 3 & -4 & \mathbf{2} \end{bmatrix}$$

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

2.8 Interpolasi Polinomial

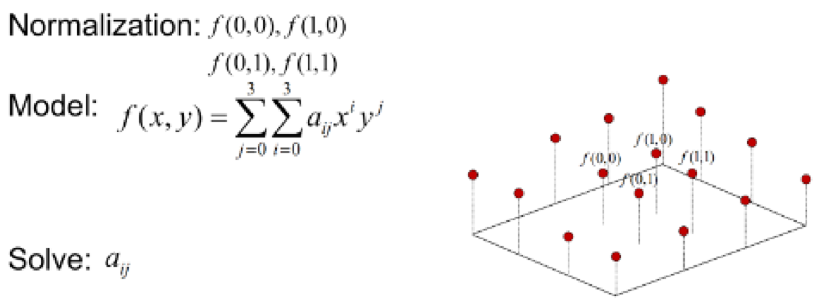
Metode Interpolasi Polinom adalah metode yang dipakai untuk menaksir nilai yang tidak diketahui dengan cara mengambil beberapa data nilai yang diketahui dan mengolahnya menjadi suatu polinom yang akan melewati semua titik titik yang diketahui. Untuk menggunakan metode interpolasi polinom dibutuhkan data seminimalnya n + 1 titik untuk dapat membuat polinom interpolasi $P_n(x)$, setelah polinom ditemukan dapat ditaksir nilai y di sembarang titik di dalam selang $[x_0, x_n]$



2.9 Interpolasi Bicubic Spline

Metode Interpolasi Bicubic merupakan metode yang digunakan untuk mengaproksimasi nilai fungsi diantara titik titik data yang diketahui. Metode ini melibatkan konsep spline dan konstruksi serangkaian polinomial kubik. Dengan metode ini kita bisa melakukan pendekatan untuk menciptakan permukaan yang lebih halus dan kontinu. Salah satu penerapan interpolasi bicubic untuk scaling image, dengan menggunakan metode ini kita bisa memperbesar gambar dan membuat gambar lebih jernih dari sebelumnya.

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.



2.10 Regresi Linier Berganda

Metode Regresi Linear berganda merupakan metode yang dapat menaksir nilai berdasarkan data data yang diproses terlebih dahulu. Tidak seperti Metode Interpolasi Polinom yang wajib menginterpolasi semua titik titik data yang diberikan dan membuat rumus polinom yang bisa berderajat lebih dari satu . Di Metode regresi ini kita hanya mengambil “jalur tengah” dari data - data yang diberikan. Dalam metode regresi linear berganda terdapat variabel bebas dan variable terikat. Variabel bebas menjelaskan hubungan/relasi/pengaruhnya terhadap variabel terikat , seperti namanya nilai dari variabel ini bebas sedangkan variabel terikat merupakan variabel yang terikat oleh variabel bebas.

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

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

2.11 Regresi Kuadratik Berganda

Regresi Kuadratik Berganda adalah salah satu bentuk regresi non-linear yang digunakan untuk memodelkan hubungan antara variabel dependen dan beberapa variabel independen, tetapi dengan tambahan komponen kuadratik (pangkat dua) dari variabel-variabel independen. Teknik ini digunakan ketika hubungan antara variabel dependen dan variabel independen tidak linear, melainkan melibatkan interaksi atau pengaruh kuadratik.

$$\begin{pmatrix} N & \sum u_i & \sum v_i & \sum u_i^2 & \sum u_i v_i & \sum v_i^2 \\ \sum u_i & \sum u_i^2 & \sum u_i v_i & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i v_i^2 \\ \sum v_i & \sum u_i v_i & \sum v_i^2 & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum v_i^3 \\ \sum u_i^2 & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i^4 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 \\ \sum u_i v_i & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 & \sum u_i v_i^3 \\ \sum v_i^2 & \sum u_i v_i^2 & \sum v_i^3 & \sum u_i^2 v_i^2 & \sum u_i v_i^3 & \sum v_i^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i u_i \\ \sum y_i v_i \\ \sum y_i u_i^2 \\ \sum y_i u_i v_i \\ \sum y_i v_i^2 \end{pmatrix}$$

Bab III - Implementasi

3.1 File Main.java

Entry point dari program kalkulator matrix 'Geprek Mumbul'. File ini mengatur navigasi menu utama dan sub-menu kemudian menjalankan fungsi sesuai keinginan user.

- Atribut

  (Tidak ada)
- Konstruktor

  (Tidak ada)
- Method

Method	Deskripsi
public static void runSPL()	Memanggil Menu SPL
public static void runDeterminan()	Memanggil Menu Determinant
public static void runMatrixBalikan()	Menu matrix balikan
public static void runInterPolinom()	Memanggil menu interpolinom
public static void runInterBicub()	Memanggil menu bicubic.
public static void runRegresi()	Memanggil menu regresi
public static void main()	Entry point keseluruhan program. Menampilkan menu gambaran besar program.

3.2 Folder matrix

3.2.1 File matrix.java

Menyediakan berbagai metode operasi matrix, mempresentasikan matrix dengan ukuran yang dapat disesuaikan.

- Atribut

Atribut	Deskripsi
private int col	Menyimpan jumlah kolom pada matriks.
private int row	Menyimpan jumlah baris pada matriks.
private double[][] data	Menyimpan elemen-elemen dari matriks dalam bentuk array dua dimensi.
private static Scanner input	Scanner statis yang digunakan untuk membaca input dari pengguna.

- Konstruktor

Konstruktur	Deskripsi
public Matrix(int row, int col)	Menginisialisasi objek `Matrix` dengan jumlah baris dan kolom yang diberikan.

- Method

Method	Deskripsi
public int getRow()	Mengembalikan jumlah baris matriks.
public int getCol()	Mengembalikan jumlah kolom matriks.
public int getLastRowIdx()	Mengembalikan indeks baris terakhir.
public int getLastColIdx()	Mengembalikan indeks kolom terakhir.
public double getElmt(int row, int col)	Mengembalikan elemen pada baris dan kolom yang ditentukan.
public Matrix getColElmts(int col)	Mengembalikan matriks yang berisi elemen dari kolom tertentu.
public Matrix getRowElmts(int row)	Mengembalikan matriks yang berisi elemen dari baris tertentu.
public void deleteLastRow()	Menghapus baris terakhir dari matriks.

public void setElmt(int row, int col, double elmt)	Mengatur elemen pada posisi tertentu di matriks.
public static Matrix readMatrix(int row, int col)	Membaca matriks dengan ukuran yang ditentukan dari input pengguna.
public void rowSwap(Matrix m, int rows1, int rows2)	Menukar dua baris dalam matriks.
public static Matrix readMatrixFromKeyboard()	Membaca matriks dari input keyboard (baris dan kolom bebas).
public static Matrix readMatrixSquareFromKeyboard()	Membaca matriks persegi dari input keyboard.
public void displayMatrix()	Menampilkan matriks ke konsol.
public boolean isSquareMatrix()	Mengecek apakah matriks adalah matriks persegi.
public boolean isSymmetric()	Mengecek apakah matriks adalah matriks simetris.
public boolean isMatrixSizeEqual(Matrix Matrix2)	Mengecek apakah dua matriks memiliki ukuran yang sama.
public boolean isMatrixEqual(Matrix Matriks2)	Mengecek apakah dua matriks sama.
public boolean isIdentity()	Mengecek apakah matriks adalah matriks identitas.
public String toString()	Mengembalikan representasi string dari matriks.
public Matrix copyMatrix()	Mengembalikan salinan dari matriks saat ini.
public Matrix transpose()	Mengembalikan transpose dari matriks.
public static Matrix multiplyMatrix(Matrix matrix1, Matrix matrix2)	Mengalikan dua matriks.

3.2.2 File InverseIdentity.java

Membuat matriks identitas dan menghitung invers matriks dengan metode eliminasi Gauss-Jordan

- Atribut
- Konstruktor
- Method

Method	Deskripsi
public static Matrix createIdentityMatrix(int size)	Membuat dan mengembalikan sebuah matriks identitas dengan ukuran yang ditentukan (size x size).
public static Matrix inversIdentity(Matrix matrix)	Mengembalikan invers dari matriks persegi (n x n) menggunakan metode eliminasi Gauss-Jordan.

3.2.3 File InverseAdjoin.java

Melakukan operasi pada matriks seperti membuat kofaktor, matriks adjoin, dan invers adjoin .

- Atribut
  (Tidak ada)
- Konstruktor
  (Tidak ada)
- Method

Method	Deskripsi
public static Matrix Cofactor(Matrix matrix)	Membuat dan mengembalikan matriks kofaktor dari matriks awal.
public static Matrix Adjoin(Matrix matrix)	Membuat dan mengembalikan matriks adjoin dari matriks awal.
public static Matrix inversAdjoin(Matrix matrix)	Mengembalikan invers adjoin dari matriks awal dengan menggunakan determinan.

3.2.4 File Gauss.java

Melakukan operasi pada matriks seperti membuat kofaktor, matriks adjoin, dan invers adjoin menggunakan metode determinan

- Atribut
  (Tidak ada)
- Konstruktor
  (Tidak ada)
- Method

Method	Deskripsi
public static Matrix gauss(Matrix matrix)	Mengubah matriks ke dalam bentuk eselon baris dengan eliminasi Gauss, memastikan diagonal utama bernilai 1 dan elemen di bawahnya bernilai 0.

3.2.5 File GaussJordan.java

Mengubah matrix ke bentuk eselon tereduksi dengan metode Gauss Jordan.

- Atribut
  (Tidak ada)
- Konstruktor
  (Tidak ada)
- Method

Method	Deskripsi
public static Matrix gaussJordan(Matrix matrix)	Mengubah matriks ke dalam bentuk eselon baris tereduksi menggunakan metode eliminasi Gauss-Jordan.

3.2.6 File Determinant.java

- Atribut

Atribut	Deskripsi

private static final Scanner scanner	Scanner statis yang digunakan untuk membaca input dari pengguna.

- Konstruktor
  (Tidak ada)
- Method

Method	Deskripsi
public static void menuDeterminant()	Menampilkan menu pilihan input untuk perhitungan determinan (dari keyboard atau file).
public static void runDeterminantFromFile(String fileName)	Membaca matriks dari file dan menghitung determinan menggunakan metode yang dipilih. Menyimpan hasil ke file jika diinginkan.
public static void runDeterminantFromKeyboard()	Mengambil input matriks dari keyboard dan menghitung determinan menggunakan metode yang dipilih. Menyimpan hasil ke file jika diinginkan.
public static double DeterminantCOFACTOR(Matrix matrix)	Menghitung determinan dengan menggunakan ekspansi kofaktor.
public static double DeterminantCOFACTORPass(Matrix matrix, int row, int col)	Menghitung determinan matriks bagian dengan mengabaikan baris dan kolom tertentu (kofaktor).
public static double DeterminantOBE(Matrix matrix)	Menghitung determinan dengan menggunakan metode Operasi Baris Elementer (OBE).

3.3 Folder function

3.3.1 File BicubicSpline.java

Menghitung interpolasi bicubic menggunakan

- Atribut

Atribut	Deskripsi
private static final Scanner scanner	Scanner statis yang digunakan untuk membaca input dari pengguna.

- Konstruktor
  (Tidak ada)
- Method

Method	Deskripsi
public static void menuBicubic()	Menampilkan menu input untuk interpolasi bicubic, menerima input dari keyboard atau file, dan menampilkan hasil aproksimasi f(x, y).
public static Matrix convertTo16x1(Matrix inputMatrix)	Mengonversi matriks input menjadi matriks 16x1 yang digunakan dalam perhitungan interpolasi bicubic.
public static double function(double x, double y, Matrix MatrixY)	Menghitung aproksimasi f(x, y) berdasarkan matriks koefisien 16x1.
public static Matrix bicubicMatrix()	Membangun matriks bicubic 16x16 yang digunakan dalam proses interpolasi.

3.3.2 File InterpolasiPolinom.java

Melakukan interpolasi Polinom. Kelas ini memungkinkan pengguna memberi input data yang kemudian bisa dihasilkan aproksimasi f(x,y) sesuai permintaan user.

- Atribut

Atribut	Deskripsi

private double[] coefficients	Menyimpan koefisien polinomial hasil dari perhitungan interpolasi.
private static final Scanner scanner	Scanner statis yang digunakan untuk membaca input dari pengguna.

- Konstruktor
  (Tidak ada)

- Method

Method	Deskripsi
public static void menuInterpolasi()	Menampilkan menu untuk interpolasi polinom dengan pilihan input dari keyboard atau file.
public double[] getCoefficients()	Mengembalikan koefisien polinomial hasil dari interpolasi.
public void calculateCoefficient(Matrix data)	Menghitung koefisien polinomial berdasarkan titik-titik data yang diberikan.
public double estimateY(double x)	Menghitung nilai estimasi f(x) berdasarkan koefisien polinomial yang telah dihitung.
public String coefficientsToEquation()	Menghasilkan representasi string dari persamaan polinomial berdasarkan koefisien.
public void saveInterpolToFile(String fileName)	Menyimpan hasil persamaan polinomial ke dalam file dengan nama yang diberikan.
public String generateOutputString(double x, double yApprox)	Menghasilkan string keluaran yang berisi persamaan polinomial dan hasil estimasi f(x).
public static void runInterpolationFromFile(String fileName)	Melakukan interpolasi polinom dengan mengambil titik-titik data dari file.

public static void runInterpolationFromKeyboard( )	Melakukan interpolasi polinom dengan menerima titik-titik data dari input keyboard.

3.3.3 File RegresiKuadratik.java

Menjalankan perhitungan regresi kuadratik.

- Atribut

Atribut	Deskripsi
private static final Scanner scanner	Objek Scanner yang digunakan untuk menerima input dari pengguna.

- Konstruktor
  (Tidak ada)

- Method

Method	Deskripsi
public static void menuRegresiKuadratik()	Menampilkan menu untuk input regresi kuadratik dari keyboard atau file.
public static void runRegKuadratikFromFile(String fileName)	Membaca data dari file dan menjalankan regresi kuadratik. Hasil regresi serta taksiran nilai fungsi pada titik xk ditampilkan dan disimpan ke file.
public static void runRegKuadratikFromKeyboard()	Menerima input regresi kuadratik dari keyboard. Hasil regresi serta taksiran nilai fungsi pada titik xk ditampilkan dan disimpan ke file.

public static Matrix AugmentMatrix(Matrix x, double[] y, int n, int m)	Membentuk matriks augmented yang digunakan untuk menyelesaikan sistem persamaan linier regresi kuadratik.
public static void DisplayEquationReg(double[] koefisien, int n)	Menampilkan persamaan regresi dalam bentuk f(x) = b0 + b1x1 + b2x2 + ...
public static double estimateYXkReg(double[] koefisien, double[] xk)	Menghitung nilai taksiran fungsi f(xk) menggunakan koefisien yang diperoleh dari hasil regresi.
public static String coefficientsToEquationReg(double[] koefisien)	Mengonversi koefisien menjadi persamaan dalam bentuk string.
public static String generateOutputStringReg(double yApprox, double[] koefisien)	Menghasilkan output string yang mencakup persamaan regresi dan nilai fungsi pada titik xk.

3.3.4 File RegresiLinear.java

Menghitung regresi linier berganda. Kelas ini menerima input dari keyboard atau file, kemudian kelas ini dapat memungkinkan program untuk melakukan perhitungan regresi linier

- Atribut

Atribut	Deskripsi
private static final Scanner scanner	Scanner statis yang digunakan untuk membaca input dari pengguna.

- Konstruktor
  (Tidak ada)

- Method

Method	Deskripsi
public static void menuRegresiLinier()	Menampilkan menu input untuk regresi linier dari keyboard atau file.
public static Matrix multipleLinearRegression(Matrix matrixX, Matrix matrixY)	Menghitung solusi regresi linier berganda berdasarkan matriks X dan Y.
public static void runRLFromKeyboard()	Mengambil input dari keyboard untuk melakukan regresi linier berganda.
public static void runRLFromFile(String fileName)	Mengambil input dari file untuk melakukan regresi linier berganda.

3.4 Folder utils

3.4.1 File Menu.java

Menampilkan menu tanpa pemrosesan data atau perhitungan lebih lanjut. Method di file ini bertindak seperti ASCII

- Atribut
  (Tidak ada)

- Konstruktor
  (Tidak ada)
- Method

Method	Deskripsi
public static void welcome()	Menampilkan pesan selamat datang di Kalkulator Matrix Geprek Mumbul.
public static void menu()	Menampilkan menu utama untuk memilih jenis operasi yang ingin dilakukan, seperti SPL, determinan, interpolasi, dan regresi.
public static void subMenuSPL()	Menampilkan submenu untuk memilih metode penyelesaian SPL.

public static void subMenuDeterminan()	Menampilkan submenu untuk memilih metode perhitungan determinan.
public static void subMenuMatriksBalikan()	Menampilkan submenu untuk memilih metode perhitungan matriks balikan.
public static void subMenuRegresi()	Menampilkan submenu untuk memilih jenis regresi yang diinginkan.
public static void menuInput()	Menampilkan pilihan untuk memasukkan data dari keyboard atau file.
public static void subMenuSaveFile()	Menanyakan kepada pengguna apakah hasil ingin disimpan ke file.
public static void credit()	Menampilkan kredit dan nama-nama anggota kelompok.

3.4.2 File SavetoFile.java

Bertanggung jawab untuk menyimpan output ke dalam file..

- Atribut

Atribut	Deskripsi
private static final String BASE_DIRECTORY	Lokasi folder default untuk menyimpan file hasil, yang diatur ke "../test/output".

- Konstruktor
  (Tidak ada)

- Method

Method	Deskripsi
public static void saveResultToFile(String content, String fileName)	Menyimpan string konten ke file di folder output. Jika file belum ada, akan dibuat file baru.

3.4.3 File ReadFile.java

Membaca data dari file teks dan mengonversinya menjadi objek matriks.

- Atribut

Atribut	Deskripsi
private static final String BASE_DIRECTORY	Lokasi folder default untuk file input, yang diatur ke "../test/input/".

- Konstruktor

  (Tidak ada)
- Method

Method	Deskripsi
public static Matrix readMatrixFromFile(String fileName)	Membaca data matriks dari file teks dan mengonversinya menjadi objek matriks.

3.4.4 File SPL.java

- Atribut

Atribut	Deskripsi
private static final Scanner scanner	Scanner statis yang digunakan untuk membaca input dari pengguna.

- Konstruktor
  (Tidak ada)

- Method

Method	Deskripsi
public static void menuSPL()	Menampilkan menu input untuk perhitungan SPL dari keyboard atau file.
public static void runSPLFromFile(String fileName)	Mengambil matriks dari file dan menyelesaikan SPL dengan metode yang dipilih.
public static void runSPLFromKeyboard()	Mengambil input matriks dari keyboard dan menyelesaikan SPL dengan metode yang dipilih.
public static void SPLinvers(Matrix matrix)	Menyelesaikan SPL menggunakan metode invers.
public static Matrix replaceColumn(Matrix matrix, double[] konstanta, int k)	Mengganti kolom k pada matriks dengan array konstanta untuk digunakan dalam metode Cramer.
public static double[] kaidahCramer(Matrix matrix, double[] konstanta)	Menghitung solusi SPL menggunakan kaidah Cramer.
public static void SPLcramer(Matrix matrixC)	Menyelesaikan SPL menggunakan kaidah Cramer.
public static int SolutionType(Matrix matrix)	Mengidentifikasi jenis solusi (unik, tak terbatas, atau tidak ada solusi) dari SPL.
public static double[] Substitute(Matrix matrix, double[] X)	Melakukan substitusi mundur untuk mendapatkan solusi SPL setelah eliminasi.
public static void solveParametric(Matrix matrix)	Menyelesaikan SPL dengan solusi parametrik.
public static void SPLgauss(Matrix matrix)	Menyelesaikan SPL menggunakan eliminasi Gauss.
public static void SPLgaussJordan(Matrix matrix)	Menyelesaikan SPL menggunakan eliminasi Gauss-Jordan.

Bab IV - Eksperimen

4.1 SPL

4.1.1 Test Case 1

Temukan solusi SPL Ax = b, berikut :

1) Test Case 1 a

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

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi tunggal : X1 = 1,4000 ; X2 = -2,6000 ; X3 = -1,1000 ; X4 = -0,1000
------------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi tunggal : X1 = 1,4000 ; X2 = -2,6000 ; X3 = -1,1000 ; X4 = -0,1000
------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
Solusi : x1 = 1,4000 ; x2 = -2,6000 ; x3 = -1,1000 ; x4 = -0,1000
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
Solusi : x1 = 1,4000 ; x2 = -2,6000 ; x3 = -1,1000 ; x4 = -0,1000
------------------------------------------------------------
```

2) Test Case 1 b

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

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi parametrik :
X1 = 3,0000 + 1,0000b
X2 = 0,0000 + 2,0000b
X3 = a
X4 = -1,0000 + 1,0000b
X5 = b
------------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi parametrik :
X1 = 3,0000 + 1,0000b
X2 = 0,0000 + 2,0000b
X3 = a
X4 = -1,0000 + 1,0000b
X5 = b
------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
SPL tersebut tidak dapat diselesaikan dengan metode invers.
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
SPL tersebut tidak dapat diselesaikan dengan metode cramer!
------------------------------------------------------------
```

3) Test Case 1 c

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

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi parametrik :
X1 = a
X2 = 1,0000 - 1,0000c
X3 = b
X4 = -2,0000 - 1,0000c
X5 = 1,0000 + 1,0000c
X6 = c
------------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi parametrik :
X1 = a
X2 = 1,0000 - 1,0000c
X3 = b
X4 = -2,0000 - 1,0000c
X5 = 1,0000 + 1,0000c
X6 = c
------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
SPL tersebut tidak dapat diselesaikan dengan metode invers.
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
SPL tersebut tidak dapat diselesaikan dengan metode cramer!
------------------------------------------------------------
```

4) Test case 1 d

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

Kasus n = 6

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi tunggal : X1 = 35,6558 ; X2 = -619,2082 ; X3 = 3282,7840 ; X4 = -7351,4848 ; X5 = 7323,5567 ; X6 = -2676,9191
-----------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi tunggal : X1 = 35,6558 ; X2 = -619,2082 ; X3 = 3282,7840 ; X4 = -7351,4848 ; X5 = 7323,5567 ; X6 = -2676,9191
-----------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
Solusi : x1 = 35,6558 ; x2 = -619,2082 ; x3 = 3282,7840 ; x4 = -7351,4848 ; x5 = 7323,5567 ; x6 = -2676,9191
-----------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
Solusi : x1 = 35,6558 ; x2 = -619,2075 ; x3 = 3282,7802 ; x4 = -7351,4763 ; x5 = 7323,5482 ; x6 = -2676,9159
-----------------------------------------------------------
```

Kasus n = 10

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi tunggal : X1 = 12,0218 ; X2 = -77,1603 ; X3 = 124,5157 ; X4 = 57,3643 ; X5 = -268,2617 ; X6 = 34,5441 ; X7 = 226,4496 ; X8 = -7
4,6534 ; X9 = 35,0669 ; X10 = -72,9123
-----------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi tunggal : X1 = 12,0218 ; X2 = -77,1603 ; X3 = 124,5157 ; X4 = 57,3643 ; X5 = -268,2617 ; X6 = 34,5441 ; X7 = 226,4496 ; X8 = -74,6534 ; X9 = 35,0669 ; X10 = -72,9123
-----------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
Solusi : x1 = 12,0218 ; x2 = -77,1603 ; x3 = 124,5157 ; x4 = 57,3643 ; x5 = -268,2617 ; x6 = 34,5441 ; x7 = 226,4496 ; x8 = -74,6534 ;
 x9 = 35,0669 ; x10 = -72,9123
-----------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
Solusi : x1 = 12,0218 ; x2 = -77,1603 ; x3 = 124,5157 ; x4 = 57,3643 ; x5 = -268,2617 ; x6 = 34,5441 ; x7 = 226,4496 ; x8 = -74,6534 ;
 x9 = 35,0669 ; x10 = -72,9123
-----------------------------------------------------------
```

4.1.2 Test Case 2

SPL berbentuk matrix augmented

1) Test Case 2 a

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

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi parametrik :
X1 = -1,0000 + 1,0000b
X2 = 0,0000 + 2,0000a
X3 = a
X4 = b
-----------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi parametrik :
X1 = -1,0000 + 1,0000b
X2 = 0,0000 + 2,0000a
X3 = a
X4 = b
------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
SPL tersebut tidak dapat diselesaikan dengan metode invers.
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
SPL tersebut tidak dapat diselesaikan dengan metode cramer!
------------------------------------------------------------
```

2) Test case 2 b

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

a. Gauss

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi tunggal : X1 = 0,0000 ; X2 = 2,0000 ; X3 = 1,0000 ; X4 = 1,0000
------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
SPL tersebut tidak dapat diselesaikan dengan metode invers.
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
SPL tersebut tidak dapat diselesaikan dengan metode cramer!
------------------------------------------------------------
```

4.1.3 Test Case 3

1) Test Case 3 a

$$8x_1 + x_2 + 3x_3 + 2x_4 = 0$$

$$2x_1 + 9x_2 - x_3 - 2x_4 = 1$$

$$x_1 + 3x_2 + 2x_3 - x_4 = 2$$

$$x_1 + 6x_3 + 4x_4 = 3$$

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi tunggal : X1 = -0,2243 ; X2 = 0,1824 ; X3 = 0,7095 ; X4 = -0,2581
--------------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi tunggal : X1 = -0,2243 ; X2 = 0,1824 ; X3 = 0,7095 ; X4 = -0,2581
--------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
Solusi : x1 = -0,2243 ; x2 = 0,1824 ; x3 = 0,7095 ; x4 = -0,2581
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
Solusi : x1 = -0,2243 ; x2 = 0,1824 ; x3 = 0,7095 ; x4 = -0,2581
------------------------------------------------------------
```

2) Test Case 3 b

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

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
SPL tersebut tidak memiliki solusi.
-------------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
SPL tersebut tidak memiliki solusi.
-------------------------------------------------------------
```

c. Balikan

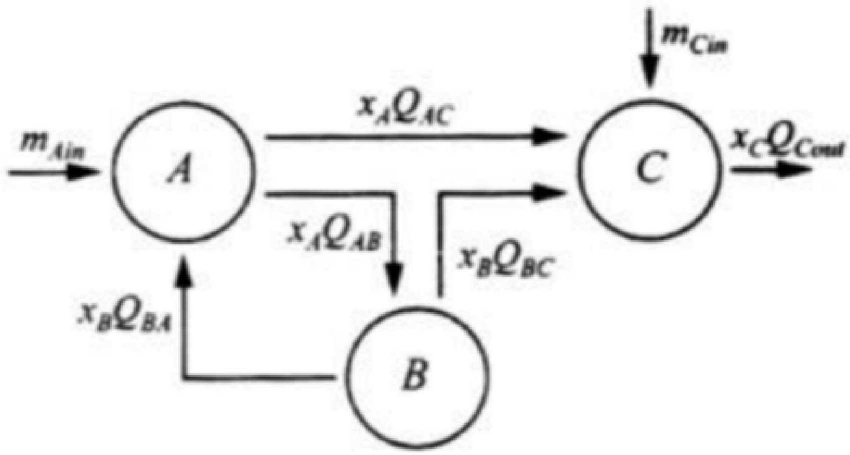
```
### PROSES PENGERJAAN SPL METODE INVERS ###
SPL tersebut tidak dapat diselesaikan dengan metode invers.
-------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
SPL tersebut tidak dapat diselesaikan dengan metode cramer!
-------------------------------------------------------------
```

4.1.4 Test Case 4

Lihatlah sistem reaktor pada gambar berikut.



Dengan laju volume Q dalam m^3/s dan input massa min dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

Tentukan solusi x_A, x_B, x_C dengan menggunakan parameter berikut : 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.

a. Gauss

```
### PROSES PENGERJAAN SPL METODE GAUSS ###
Solusi tunggal : X1 = 14,4444 ; X2 = 7,2222 ; X3 = 10,0000
------------------------------------------------------------
```

b. Gauss Jordan

```
### PROSES PENGERJAAN SPL METODE GAUSS JORDAN ###
Solusi tunggal : X1 = 14,4444 ; X2 = 7,2222 ; X3 = 10,0000
------------------------------------------------------------
```

c. Balikan

```
### PROSES PENGERJAAN SPL METODE INVERS ###
Solusi : x1 = 14,4444 ; x2 = 7,2222 ; x3 = 10,0000
------------------------------------------------------------
```

d. Cramer

```
### PROSES PENGERJAAN SPL METODE CRAMER ###
Solusi : x1 = 14,4444 ; x2 = 7,2222 ; x3 = 10,0000
------------------------------------------------------------
```

4.2 Interpolasi Polinom

4.2.1 Test Case 5

1.) Test Case 5a

Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai *x* yang akan dicari nilai fungsi *f(x)*.

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

Lakukan pengujian pada nilai-nilai berikut:

$x = 0.2$ $f(x) = ?$
$x = 0.55$ $f(x) = ?$
$x = 0.85$ $f(x) = ?$
$x = 1.28$ $f(x) = ?$

Hasil Test Case 5a :

f(0,2) =

```
f(x) = -0,0000x^6 + 0,0000x^5 + 0,0260x^4 + 0,0000x^3 + 0,1974x^2 + 0,2400x + 0,0230, f(0,2000) = 0,0330
```

f(0.55) =

```
f(x) = -0,0000x^6 + 0,0000x^5 + 0,0260x^4 + 0,0000x^3 + 0,1974x^2 + 0,2400x + 0,0230, f(0,5500) = 0,1711
```

f(0.85) =

```
f(x) = -0,0000x^6 + 0,0000x^5 + 0,0260x^4 + 0,0000x^3 + 0,1974x^2 + 0,2400x + 0,0230, f(0,8500) = 0,3372
```

f(1.28) =

```
f(x) = -0,0000x^6 + 0,0000x^5 + 0,0260x^4 + 0,0000x^3 + 0,1974x^2 + 0,2400x + 0,0230, f(1,2800) = 0,6775
```

2.) Test Case 5b

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

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

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

Gunakanlah data di atas dengan memanfaatkan interpolasi polinomial untuk melakukan prediksi jumlah kasus baru Covid-19 pada 16/07/2022.

Hasil Test Case 5b :

```
f(x) = -140993,7122x^9 + 9372849,2391x^8 + 275474539,4207x^7 + 4695806315,4288x^6 + 51131876760,1328x^5 + 368550807175,5350x^4 + 1756810186361,3564x^3 + 5334203055240,5780x^2 + 9346993
079173,4380x + 7187066071661,2010, f(7,5161) = 53562,1875
```

3.) Test Case 5c
Sederhanakan fungsi *f(x)* yang memenuhi kondisi

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

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.*

Hasil Test Case 5c :

```
f(x) = 0,2363x^5 + 1,4213x^4 + 3,2371x^3 + 3,5527x^2 + 2,0353x, f(4,0000) = 36,5558
```

4.3 Regresi Berganda

4.3.1 Test Case 6

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

1.) Hasil Regresi Linear Berganda`

```
F(X1 , X2 , X3) = -3,5078 - 0,0026 X1 + 0,0008 X2 + 0,1542 X3
F(X1 = 50,0000 ; X2 = 76,0000 ; X3 = 29,3000) = 0,9384
------------------------------------------------------------
```

2.) Hasil Regresi Linear Kuadratik

```
Persamaan regresi: f(x) = -0,0000 + 0,0028x1 + 0,0008x2 - -0,0063x3 - -0,0000x4 + 0,0001x5 - -0,0008x6 - -0,0004x7 + 0,0010x8 + 0,0005x9
f(xk) = 1.025863324132304
```

4.4 Interpolasi Bicubic Spine

4.4.1 Test Case 7

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

Tentukan nilai:

$f(0, 0) = ?$

$f(0.5, 0.5) = ?$

$f(0.25, 0.75) = ?$

$f(0.1, 0.9) = ?$

1.) Test Case 7a

```
f(0,0000, 0,0000) = 21,0000
```

2.) Test Case 7 b

```
f(0,5000, 0,5000) = 87,7969
```

3.) Test Case 7c

```
f(0,2500, 0,7500) = 117,7322
```

4.) Test Case 7d

```
f(0,1000, 0,9000) = 2,5725
```

4.5 Determinan

Hitunglah Determinan dari Matrix berikut

4.5.1 Test Case 0a

21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96

a.) Metode Kofaktor

```
### PROSES PENGERJAAN METODE EKSPANSI KOFAKTOR ###
Determinan = 6781746,0000
```

b.) Metode OBE

```
### PROSES PENGERJAAN METODE OBE (OPERASI BARIS ELEMENTER) ###
Determinan = 6781746,0000
-----------------------------------------------------------------
```

4.5.2 Test Case 0b

1 3 5 9
1 3 1 7
4 3 9 7
5 2 0 9

a.) Metode Kofaktor

```
### PROSES PENGERJAAN METODE EKSPANSI KOFAKTOR ###
Determinan = -376,0000
```

b.) Metode OBE

```
### PROSES PENGERJAAN METODE OBE (OPERASI BARIS ELEMENTER) ###
Determinan = -376,0000
```

4.6 Invers

4.6.1 Test Case 0a

1 8 4 3
5 7 1 9
6 6 0 5
2 7 3 1

a.) Metode Kofaktor

```
Menggunakan Metode Adjoin...
Matrix Invers:
-7.166666666666667 4.166666666666667 -4.833333333333333 8.166666666666666
7.583333333333333 -4.583333333333333 5.416666666666667 -8.583333333333334
-12.75 7.75 -9.25 14.75
-0.5 0.5 -0.5 0.5
```

b.) Metode Identitas

```
Menggunakan Metode Identitas...
Matrix Invers:
-7.166666666666694 4.166666666666681 -4.833333333333352 8.166666666666696
7.583333333333367 -4.583333333333352 5.416666666666689 -8.58333333333337
-12.750000000000053 7.750000000000028 -9.250000000000036 14.750000000000059
-0.5000000000000058 0.5000000000000033 -0.500000000000004 0.5000000000000064

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

4.6.2 Test Case 0b

1 2 3
4 5 6
7 8 9

```
Menggunakan Metode Identitas...
Error: Matriks tidak dapat di-inverskan karena pivot 0 dan tidak ada baris untuk ditukar.
```

4.7 Menu

Menu Utama

```
              Selamat Datang di Kalkulator Matrix Geprek Mumbul
---------------------------------------------------------------------------------
                                MENU
----------------------------------------------------------------------
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Interpolasi Bicubic Spline
6. Regresi Linier dan Kuadratik Berganda
7. Keluar

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

Menu Input

```
                  MENU INPUT
------------------------------------------------------------
1. Masukan dari Keyboard
2. Masukan dari File
------------------------------------------------------------
```

Menu metode determinan

```
                    PILIH METODE
------------------------------------------------------------
1. Determinan Ekspansi Kofaktor
2. Determinan OBE
------------------------------------------------------------
```

Menu metode SPL

```
                    PILIH METODE
------------------------------------------------------------
1. SPL Invers
2. SPL Cramer
3. SPL Gauss
4. SPL Gauss Jordan
------------------------------------------------------------
```

Menu Metode Regresi

```
                MAU REGRESI APA?
------------------------------------------------------------
------------------------------------------------------------
1. Regresi Linear
2. Regresi Kuadratik
------------------------------------------------------------
```

Bab V - Kesimpulan

5.1 Kesimpulan

Mata kuliah IF2123 Aljabar Linear dan Geometri memberikan pemahaman mendalam tentang berbagai metode penyelesaian sistem persamaan linear seperti Eliminasi Gauss, Gauss-Jordan, metode Cramer, dan penggunaan matriks balikan. Dengan pengetahuan yang diperoleh, kami mampu menyelesaikan berbagai permasalahan yang melibatkan matriks, termasuk mencari determinan dan matriks balikan

Selain itu, kami juga mengimplementasikan konsep-konsep tersebut dalam program menggunakan bahasa pemrograman, pada tubes kali ini Java, yang mampu menyelesaikan sistem persamaan linear, melakukan regresi linear berganda serta regresi kuadratik berganda, menerapkan konsep interpolasi polinom dan bikubik interpolation spine..

5.2 Saran dan Refleksi

1. Pengerjaan tubes sebaiknya lebih jauh dari deadline sehingga hasil lebih maksimal
2. Pengecekan test case juga sebaiknya dilakukan jauh dari deadline
3. Koordinasi dan kerjasama antar anggota bisa lebih dimanfaatkan dan ditingkatkan

Daftar Pustaka

Informatika.stei.itb.ac.id. (2024). Sistem persamaan linier (Bagian 1: Metode eliminasi Gauss). Diakses pada 1 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf

Informatika.stei.itb.ac.id. (2024). Sistem persamaan linier (Bagian 3: Metode eliminasi Gauss-Jordan). Diakses pada 1 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-05-Sistem-Persamaan-Linier-2-2023.pdf

Informatika.stei.itb.ac.id. (2024). Sistem persamaan linier (Bagian 1:Tiga Kemungkinan Solusi Persamaan linear). Diakses pada 1 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf

Informatika.stei.itb.ac.id. (2023). Determinan (Bagian 1). Diakses pada 1 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-08-Determinan-bagian1-2023.pdf

Informatika.stei.itb.ac.id. (2023). Determinan (Bagian 2). Diakses pada 1 Oktober 2024, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-09-Determinan-bagian2-2023.pdf

LAMPIRAN

Link Repository : https://github.com/WwzFwz/Algeo01-23005

Link Video : https://youtu.be/YJ0Fr11_7eM?si=mQEKAeD1M49xVHNj