LAPORAN TUGAS BESAR 01
ALJABAR GEOMETRI IF2123

KELOMPOK AMBATUCODE

Anggota:

1. Samy Muhammad Haikal 13522151
2. Muhammad Roihan 13522152
3. Chelvadinda 13522154

Institut Teknologi Bandung

2023/2024

DAFTAR ISI

DAFTAR GAMBAR

BAB I

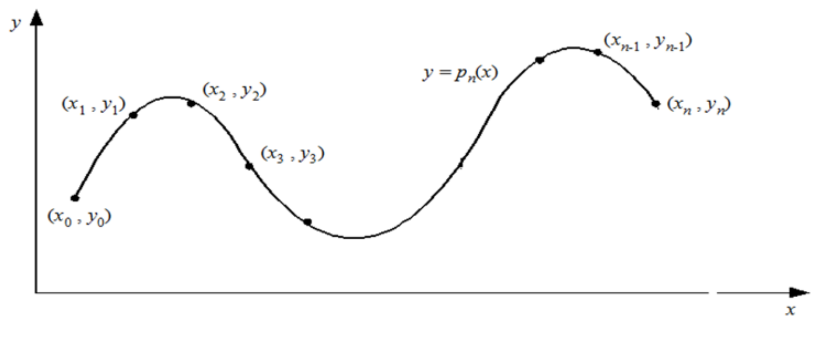
DESKRIPSI MASALAH

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

Di dalam Tugas Besar 1 ini akan 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, 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.

A. Interpolasi Polinom

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



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

Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$. adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. 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 $a_0, a_1, a_2, \ldots, a_n$,

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

Solusi sistem persamaan lanjar ini, yaitu nilai $a_0, a_1, \ldots, a_n$, 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

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

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

B. **Regresi Linier Berganda**

Regresi linier (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada persamaan jadi untuk menghitung regresi linier sederhana, terdapat persamaan umum dari regresi linier yang bisa digunakan untuk regresi linier 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 linier Regression* sebagai berikut:

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB II

TEORI SINGKAT

A. Eliminasi Gauss

Metode eliminasi Gauss adalah metode untuk operasi nilai-nilai dalam matriks untuk membuat matriks lebih sederhana. Metode eliminasi gauss dikembangkan dari metode eliminasi dengan cara menghilangkan atau mengurangi jumlah variabel untuk mendapatkan nilai variabel bebas.

Eliminasi Gauss dapat digunakan untuk menghitung rank dari suatu matriks, determinan matriks persegi, dan balikan dari suatu matriks yang determinannya bukan 0. Eliminasi Gauss juga digunakan untuk memecahkan sistem persamaan linier. Eliminasi Gauss mengubah persamaan linier menjadi bentuk matriks, kemudian diubah ke bentuk Eselon Baris melalui Operasi Baris Elementer (OBE) pada matriks yang meliputi: pertukaran dua baris, perkalian suatu baris dengan suatu konstanta, penambahan nilai pada suatu baris sebesar kelipatan baris lain untuk mendapatkan bentuk matriks segitiga atas berupa matriks eselon baris, ciri-cirinya sebagai berikut:

1. Jika suatu baris tidak seluruhnya terdiri atas nol. angka bukan nol pertama suatu baris adalah 1 (satu utama).
2. Jika ada baris yang hanya terdiri atas nol, baris-baris tersebut dikelompokkan bersama pada bagian bawah matriks.
3. Pada dua baris yang saling berurutan dan tidak seluruhnya 0, 1 utama pada baris yang bawah posisinya lebih ke kanan daripada 1 utama pada baris yang atas.

Untuk mendapatkan solusi dari sistem persamaan linier, suatu sistem persamaan linier memiliki 3 kemungkinan hasil:

1. Solusi tunggal / unik.
2. Banyak solusi
3. Tidak memiliki solusi.

Pemecahan solusi dari sistem persamaan linier dengan metode eliminasi Gauss memerlukan pembentukan matriks *augmented* sebagai langkah awal. Matriks *augmented* ini terdiri dari koefisien persamaan dan hasilnya. Selanjutnya, dilakukan operasi baris elementer untuk menghasilkan matriks eselon baris dan metode substitusi balik digunakan untuk menentukan nilai dari setiap variabel.

B. Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan berasal dari metode eliminasi Gauss yang kemudian mengalami pengembangan. Metode eliminasi Gauss-Jordan menghasilkan matriks eselon baris tereduksi yang memiliki sifat yang sama dengan matriks eselon baris, namun dengan tambahan satu sifat penting, yaitu setiap kolom yang berisi 1 utama memiliki nilai 0 pada semua baris dan kolom lainnya. Metode eliminasi Gauss-Jordan dapat digunakan untuk menyelesaikan suatu sistem persamaan linear dengan menggunakan operasi baris elementer yang sama seperti yang diterapkan dalam metode eliminasi Gauss.

Eliminasi gauss-jordan efektif untuk menyelesaikan sistem persamaan linier yang sederhana, namun untuk sistem persamaan linier yang cenderung besar lebih efektif menggunakan metode eliminasi gauss.

C. Determinan

Determinan adalah nilai real yang dihitung berdasarkan nilai elemen-elemennya, menurut rumus tertentu yang ditulis dengan simbol det (A) atau |A|. Sifat-sifat determinan yaitu

1. Jika determinan matriks = 0, maka matriks tersebut disebut “matriks singular”
2. Ketika dua baris atau kolom dari matriks A dipertukarkan, determinannya akan berubah tanda
3. Jika sebuah matriks A dapat diubah menjadi matriks B dengan operasi baris dasar (seperti pertukaran dua baris atau mengalikan satu baris dengan suatu skalar) maka det A = det B.

Determinan dapat ditentukan dengan menggunakan metode reduksi baris dan ekspansi kofaktor. Untuk metode reduksi dilakukan dengan cara melakukan operasi matriks elementer sampai diperoleh matriks segitiga (segitiga atas atau segitiga bawah), dengan persamaan:

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

keterangan:
p = jumlah pergantian baris
k = perkalian baris-baris matriks dengan k

Untuk metode ekspansi kofaktor dapat dilakukan menggunakan persamaan:

$$\det(A) = ad - bc, untuk\ matriks\ A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Determinan ditentukan dengan menjumlahkan perkalian dari elemen pada kolom/baris tersebut dengan kofaktor kolom/baris yang bersangkutan, tidak peduli yang dipilih adalah kolom atau baris, dinotasikan dengan:

1. Secara baris

$$\det(A) = a_{i1}C_{i1} + a_{i2}C_{i2} + \cdots + a_{in}C_{in}$$

2. Secara kolom

$$\det(A) = a_{1j}C_{1j} + a_{2j}C_{2j} + \cdots + a_{nj}C_{nj}$$

keterangan:

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

D. Matriks Balikan

Invers matriks adalah kebalikan (invers) dari sebuah matriks yang apabila matriks tersebut dikalikan dengan inversnya, akan menjadi matriks identitas. Invers matriks dilambangkan dengan A-1. Suatu matriks dikatakan memiliki invers jika determinan dari matriks tersebut tidak sama dengan nol.

Untuk menentukan invers dari sebuah matriks, terdapat dua aturan berdasarkan ordonya, yaitu ordo 2x2 dan ordo 3x3. Invers matriks persegi dengan ordo 2x2 dapat dicari dengan cara sebagai berikut:

$A^{-1} = \frac{1}{|A|} \times Adj\ A$, dengan syarat $|A| \neq 0$

Jika $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, maka $A^{-1} = \frac{1}{ad-bc} \times \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$, dengan $|A| \neq 0$

Gambar 2.1 Invers matriks 2 x 2

Untuk mencari invers matriks pada ordo 3x3 atau lebih, dapat digunakan metode eliminasi Gauss Jordan. Secara sistematis, eliminasi Gauss Jordan dapat dinyatakan sebagai berikut:

$$[A|I] \rightarrow [I|A^{-1}]$$

Gambar 2.2 Invers matriks 3 x 3

E. Matriks Kofaktor

Matriks kofaktor merupakan matriks dengan elemen a_{ij} bernilai $(-1)^{i+j}M_{ij}$ dengan M_{ij} merupakan determinan dari submatriks yang mengecualikan baris ke-i dan kolom ke-j. Jika matriks kofaktor di transpose akan menghasilkan matriks adjoin.

F. Matriks Adjoin

Matriks adjoin merupakan transpose dari matriks kofaktor. Adjoin sering disingkat dengan Adj. Misalkan matriks A, maka adjoin A ditulis Adj (A). Transpose sendiri maksudnya adalah pertukaran elemen pada baris menjadi kolom atau kolom menjadi baris. Adjoin matriks digunakan dalam menentukan invers matriks.

G. Kaidah *Cramer*

Kaidah Cramer merupakan suatu metode untuk menemukan solusi dari suatu sistem persamaan linear. Untuk sistem persamaan linear $Ax = B$ yang terdiri dari n persamaan linier dengan n peubah (variabel) dimana $\det(A) \neq 0$, maka sistem persamaan linear tersebut memiliki solusi unik, yakni

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

Apabila determinan matriks koefisien pada sistem persamaan linier bernilai 0, yang memberi arti bahwa aturan Cramer tidak berlaku, maka itu menandakan bahwa sistem tersebut tidak konsisten (tidak memiliki penyelesaian atau memiliki banyak penyelesaian).

H. Interpolasi Polinom

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk melakukan interpolasi dari nilai yang diketahui atau ekstrapolasi (prediksi) dari nilai diluar rentang data yang diketahui.

Jika terdapat n+1 buah titik berbeda, kita dapat memperkirakan sebuah polinomial $Pn(x)$ yang memenuhi dan menginterpolasi (melewati) n+1 buah titik tersebut, sehingga $\forall$ titik (xi , yi) dengan $i \in \{0 \ldots n\}$ maka $yi = Pn (xi)$. Polinom ini dapat kita tuliskan sebagai:

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n,$$

n merupakan pangkat tertinggi dari polinom. Untuk mendapatkan nilai-nilai dari $\{a0, a1, a2, \ldots , an\}$ kita dapat mensubstitusikan setiap titik (xi , yi) ke dalam polinom $yi = Pn (xi)$ sehingga akan terbentuk SPL yang nantinya akan memiliki solusi unik. Solusi dari SPL ini dapat dicari dengan menggunakan metode gauss, metode gauss-jordan, dan kaidah cramer.

I. Interpolasi *Bicubic Spline*

Metode interpolasi bicubic spline merupakan metode matematika yang digunakan untuk membangun titik-titik baru dalam batas-batas sekumpulan titik yang diketahui. titik-titik baru ini adalah nilai fungsi dari fungsi interpolasi (spline) yang terdiri dari potongan-potongan polinomial dengan potongan-potongan yang lebih halus dan digabungkan secara bersamaan. Metode interpolasi bicubic spline menghasilkan citra output yang lebih tajam, sedikit kabur dan dapat menghindari efek kotak-kotak.

J. Regresi Linier Berganda

Regresi linier berganda merupakan model persamaan yang menjelaskan hubungan satu variabel tak bebas/ response (Y) dengan dua atau lebih variabel bebas/ predictor (X1, X2,…Xn). Tujuan dari uji regresi linier berganda adalah untuk memprediksi nilai variabel tak bebas/ response (Y) apabila nilai-nilai variabel bebasnya (X1, X2,..., Xn) diketahui. Disamping itu juga untuk dapat mengetahui bagaimanakah arah hubungan variabel tak bebas dengan variabel - variabel bebasnya. Persamaan regresi linier berganda secara matematik diekspresikan oleh :

$$Y = a + b1\ X1 + b2\ X2 + \ldots + bn\ Xn$$

yang mana :
Y = variabel tak bebas (nilai variabel yang akan diprediksi)
a = konstanta b1,b2,…,
bn = nilai koefisien regresi X1,X2,…,
Xn = variabel bebas

Mencari atau menentukan b_0, b_1, b_2, b_3, …, b_n berarti mencari atau menentukan solusi dari sistem persamaan linier (SPL).

BAB III

PROGRAM DAN IMPLEMENTASI

A. Garis Besar Program

1. *Folder lib*

a. Matrix.java

Class ini digunakan untuk mengimplementasikan matriks yang akan digunakan di class lain dan program utama.

- Atribut

Atribut	Deskripsi
`private int baris`	Jumlah baris yang digunakan pada matrix, berupa integer
`private int kolom`	Jumlah kolomyang digunakan pada matrix, berupa integer
`private double[][] Matrix`	Elemen-elemen dari matrix, berupa double

- Konstruktor

Konstruktor	Deskripsi
`public Matrix(int baris,int kolom)`	Membuat suatu matrix dengan baris=baris dan kolom=kolom

- Method

Method	Deskripsi
`private static Matrix keyboard()`	Berfungsi untuk membuat Matrix dari input keyboard
`public static Matrix inputFile()`	Berfungsi untuk membuat Matrix dari input suatu file
`public static Matrix inputMatrix()`	Melakukan prosedur input matrix dengan memanfaatkan method keyboard() dan inputFile()

`public static void printMatrix(Matrix M)`	Melakukan prosedur print Matriks
`public static int getBaris(Matrix M)`	Mengembalikan jumlah baris dari suatu matriks
`public static int getKolom(Matrix M)`	Mengembalikan jumlah kolom dari suatu matriks
`public static double getElmt(Matrix M,int i,int j)`	Mengembalikan elemen M[i,j]
`public static void inputElmt(Matrix M,int i,int j,double value)`	Memasukkan elemen value ke M[i,j]
`public static Matrix dotPMatrix(Matrix M, double k)`	Mengembalikan hasil perkalian dot matriks M dengan skalar k
`public void dotPMatrix(double k)`	versi non-static dari dotPMatrix
`public static Matrix crossMatrix(Matrix M, Matrix N)`	Mengembalikan hasil kali silang Matrix M dan N
`public static Matrix minor(Matrix M, int i, int j)`	Mengembalikan minor matriks dari elemen M[1,j]
`public void tukarBaris(int Baris1, int Baris2)`	Melakukan prosedur tukar baris
`public void plusBaris(int Baris1, int Baris2, double k)`	Melakukan prosedur penambahan baris
`public void minBaris(int Baris1, int Baris2, double k)`	Melakukan prosedur penguranganbaris

`public static void copyMatrix(Matrix M1, Matrix M2)`	Melakukan prosedur salin matrix dari M1 ke M2
`public static Matrix getA(Matrix M)`	Mengembalikan matrix A dari bentuk Ax=b
`public static Matrix getB(Matrix M)`	Mengembalikan matrix b dari bentuk Ax=b

b. Determinan.java

Menghitung determinan matriks dengan 2 metode, yaitu reduksi baris dan ekspansi kofaktor.

- **Atribut**

  -
- **Konstruktor**

  -
- **Method**

Method	**Deskripsi**
`public static double detReduksi(Matrix M)`	Mencari determinan dengan metode reduksi baris
`public static double detKofaktor(Matrix M)`	Mencari determinan dengan metode kofaktor
`public static String[] detReduksi()`	Ubah jawaban dari detReduksi menjadi string
`public static String[] detKofaktor()`	Ubah jawaban dari detKofaktor menjadi string
`public static void ansDet()`	Driver, melakukan prosedur pencarian determinan dengan memanfaatkan method-method lain

c. Invers.java

Menghitung matriks balikan dengan 2 metode, yaitu reduksi baris dan adjoin.

- **Atribut**

Atribut	**Deskripsi**

`private String[] ansinv`	menyimpan jawaban yang sudah diubah menjadi string dengan format yang diminta
`int nEff`	Menyimpan baris efisien dari list string
`String[]x`	menyimpan jawaban yang sudah diubah menjadi string tanpa formatting

- **Konstruktor**

Konstruktor	**Deskripsi**
`public Invers()`	Membuat ADT Invers

- **Method**

Method	**Deskripsi**
`public static Matrix identitas(int i)`	Mengembalikan Matrix identitas ixi
`public static Matrix konkat(Matrix M1, Matrix M2)`	Menyambung 2 Matriks yang mempunyai baris yang sama
`public Matrix invers(Matrix M0)`	Melakukan prosedur invers matrix dengan metode reduksi baris
`public static Matrix transpose(Matrix M)`	Melakukan prosedur transpose Matrix
`public Matrix adjoin(Matrix M)`	Melakukan prosedur invers matrix dengan metode adjoin
`private static Matrix keyboard()`	Melakukan prosedur input matrix dari keyboard dengan format yang disesuaikan
`public static Matrix inputMatrix()`	Melakukan prosedur input matrix dengan format yang disesuaikan
`public static void`	Driver Invers Melakukan

`inversMatrix()`	prosedur invers matrix dengan memanfaatkan method-method yang ada

d. SPL.java

- **Atribut**

Atribut	**Deskripsi**
`private String[] ans`	menyimpan jawaban yang sudah diubah menjadi string dengan format yang diminta
`int nEff`	Menyimpan baris efisien dari list string
`String[]x`	menyimpan jawaban yang sudah diubah menjadi string tanpa formatting

- **Konstruktor**

Konstruktor	**Deskripsi**
`public SPL()`	Membuat ADT SPL

- **Method**

Method	**Deskripsi**
`public static Matrix eselon(Matrix M)`	Melakukan prosedur OBE hingga mendapat matriks eselon baris
`public static Matrix eselonRed(Matrix M)`	Melakukan prosedur OBE hingga mendapat matriks eselon baris tereduksi
`public void manySolution(Matrix M)`	Melakukan prosedur penyelesaian spl untuk solusi banyak
`public void Gauss(Matrix M)`	Melakukan prosedur penyelesaian SPL dengan metode Gauss

`public void Gauss_Jordan(Matrix M)`	Melakukan prosedur penyelesaian SPL dengan metode Gauss Jordan
`public void splInvers()`	Melakukan prosedur penyelesaian SPL dengan metode Invers
`public void cramer()`	Melakukan prosedur penyelesaian SPL dengan metode Cramer
`public static void ansSPL()`	Driver SPL, Melakukan prosedur penyelesaian SPL dengan memanfaatkan method-method yang ada

e. **Interpolasi.java**

Class ini digunakan untuk menyelesaikan persoalan interpolasi polinomial.

- **Atribut**

  -
- **Konstruktor**

  -
- **Method**

Method	**Deskripsi**
`public static void solveInterpolasi()`	Melakukan prosedur penyelesaian interpolasi

f. **Bicubic.java**

Class ini dugunakan untuk menyelesaikan persoalan terkait Bicubic Spline Interpolation.

- **Atribut**

  -
- **Konstruktor**

  -
- **Method**

Method	**Deskripsi**
`private static void`	Melakukan input data untuk

`inputBicubic(Matrix M,Matrix M2)` `solveInterpolasi()`	persoalan Interpolasi
`public static void solveBicubic()`	Melakukan prosedur penyelesaian persoalan interpolasi

g. **Regresi.java**

Class ini digunakan untuk menyelesaikan persoalan yang berkaitan dengan regresi linear berganda.

- **Atribut**

  -

- **Konstruktor**

  -

- **Method**

Method	**Deskripsi**
`public static double[] gaussReg(Matrix M)`	Mengembalikan nilai-nilai x1,x2,... dari Matrix M dengan metode gauss
`public static void regresiLinearBerganda()`	Melakukan prosedur penyelesaian persoalan regresi linear

2. **Folder src**

a. **Main.java**

Program yang digunakan untuk mengimplementasikan pustaka yang telah dibuat.

- **Atribut**

  -

- **Konstruktor**

  -

- **Method**

Method	**Deskripsi**
`public static void`	Prosedur utama untuk menjalankan program

`main(String[] args)`	

B. *Input* dan *Output*

Masukan (input) yang diterima program berupa masukan *console* dan file .txt. Berikut keterangannya:

a. Sistem Persamaan Linier

Melalui *console* diperlukan jumlah baris dan kolom matriks, lalu memasukkan matriks yang berjumlah baris × kolom. Melalui file tidak diperlukan masukan jumlah baris dan kolom.

b. Determinan

Melalui *console* diperlukan jumlah baris dan kolom matriks, lalu memasukkan matriks yang berjumlah baris × kolom. Melalui file tidak diperlukan masukan jumlah baris dan kolom.

c. Interpolasi polinom

Melalui *console* diperlukan masukan nilai n yaitu derajat polinom dan sejumlah $n + 1$ titik uji. Selain itu, diperlukan masukan nilai x untuk menaksir nilai f(x). Melalui file tidak diperlukan masukan nilai n, namun hanya memerlukan titik (x, y), diakhiri dengan satu buah nilai x untuk menaksir nilai f(x).

d. Interpolasi *bicubic spline*

Input hanya menggunakan file. Diperlukan titik (x, y) diakhiri dengan satu nilai x untuk menaksir nilai f(x).

e. Regresi linier berganda

Melalui *console* diperlukan input banyak peubah dan banyak sampel yang bertipe integer, serta diperlukan pula input matriks yang berdimensi sampel × peubah+1 serta data yang akan ditaksir. Melalui file diperlukan input sejumlah n baris sampel tanpa menspesifikasikan jumlah n dan diperlukan input data yang akan ditaksir.

Keluaran (output) yang dihasilkan program diuraikan sebagai berikut:

a. Sistem Persamaan Linier

Keluaran yang dihasilkan berupa nilai setiap variabel pada sistem persamaan linear, baik merupakan hasil tunggal/unik ataupun dalam bentuk parametrik, atau berupa keterangan bahwa matriks tidak memiliki solusi.

b. Determinan

Keluaran yang dihasilkan berupa nilai determinan suatu matriks.

c. Interpolasi polinom
   Keluaran yang dihasilkan berupa hasil polinom dan nilai y hasil taksiran.

d. Interpolasi *bicubic spline*
   Keluaran yang dihasilkan berupa hasil polinom dan nilai y hasil taksiran.

e. Regresi linier berganda
   Keluaran yang dihasilkan berupa persamaan regresi dan taksiran hasil data yang menjadi masukan.

BAB IV

EKSPERIMEN

4.1. Tentukan Penyelesaian SPL berikut:

a. Test case 1

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

Jawaban:

```
X1 : 1.733333
X2 : -0.433333
X3 : 0.400000
X4 : -0.100000
```

b. Test case 2

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

Jawaban :

```
X1 = 4.0
X2 = 2.0
X3 = P1

X4 =
X5 = 1.0
```

c. Test case 3

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

Jawaban :

```
X1 = P1

X2 = 1.0 - P2
X3 = P3

X4 = -2.0 - P2
X5 = 1.0 + P2
X6 = P2
```

d. Test case 4

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

H adalah matrix Hilbert. Coba untuk n=6 dan n=10

n=6 :

```
X1 : 36,000000
X2 : -630,000000
X3 : 3360,000000
X4 : -7560,000001
X5 : 7560,000001
X6 : -2772,000000
```

n=10:

```
X1 : 99,992465
X2 : -4949,557030
X3 : 79193,253929
X4 : -600537,483868
X5 : 2522224,827346
X6 : -6305486,515134
X7 : 9608263,464827
X8 : -8750306,935238
X9 : 4375120,511596
X10 : -923630,451075
```

4.2. SPL berbentuk matriks augmented

a. Test case 1

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

Jawaban:

```
X1 = -1.0 + P1
X2 = 2.0P2
X3 = P2
X4 = P1
```

b. Test case 2

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

Jawaban:

```
X1 : 0.000000
X2 : 2.000000
X3 : 1.000000
X4 : 1.000000
```

4.3. SPL berbentuk

a. Test case 1

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

Jawaban:

```
X1 : -0.224324
X2 : 0.182432
X3 : 0.709459
X4 : -0.258108
```

b. Test case 2

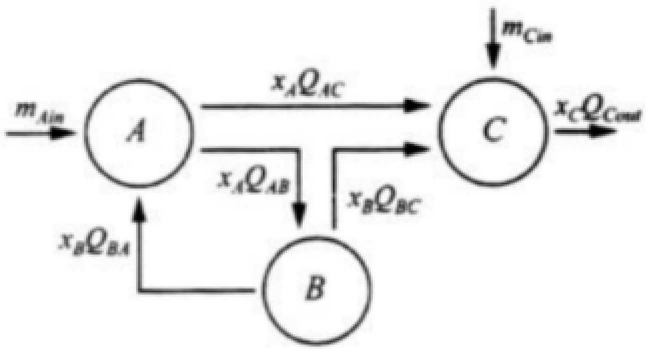
$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\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

Jawaban:

```
X1 : 3,002653
X2 : -1,004720
X3 : 6,002068
X4 : 2,997517
X5 : 5,004720
X6 : 6,997762
X7 : -0,000170
X8 : 8,000000
X9 : 5,000170
```

4.4

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:

$$
\begin{aligned}
&\text{A:} \quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0 \\
&\text{B:} \quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0 \\
&\text{C:} \quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0
\end{aligned}
$$

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.

Jawaban:

```
X1 : 14,444444
X2 : 7,222222
X3 : 10,000000
```

4.5 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) = -0.000000x^6 + 0.000000x^5 + 0.026042x^4 + 0.000000x^3 + 0.197396x^2 + 0.240000x^1 + -0.022977x^0, f(0.2) = 0.0329605072
```

$x = 0.55$

```
f(x) = -0.000000x^6 + 0.000000x^5 + 0.026042x^4 + 0.000000x^3 + 0.197396x^2 + 0.240000x^1 + -0.022977x^0, f(0.55) = 0.1711182957625
```

$x = 0.85$

```
f(x) = -0.000000x^6 + 0.000000x^5 + 0.026042x^4 + 0.000000x^3 + 0.197396x^2 + 0.240000x^1 + -0.022977x^0, f(0.85) = 0.3372356967625
```

$x = 1.28$

```
f(x) = -0.000000x^6 + 0.000000x^5 + 0.026042x^4 + 0.000000x^3 + 0.197396x^2 + 0.240000x^1 + -0.022977x^0, f(1.28) = 0.6775425678515199
```

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

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

Sebagai contoh, untuk tanggal 17/06/2022 (dibaca: 17 Juni 2022) diperoleh tanggal(desimal) sebagai berikut:

Tanggal (desimal) = 6 + (17/30) = 6,567

Gunakanlah data di atas dengan memanfaatkan **interpolasi polinomial** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

a. 16/07/2022

```
Masukkan nama file: 5b1.txt
f(x) = -141120.310600x^9 + 9381759.266086x^8 + -275752903.603842x^7 + 4700873047.890322x^6 + -51191089915.822266x^5 + 369011568500.298100x^4 + -1759197443156.986300x^3 + 5342144
345318.836000x^2 + -9362383549278.719000x^1 + 7200305831156.062500x^0, f(7.51612903) = 53564.46875
```

b. 10/08/2022

```
f(x) = -141120.310600x^9 + 9381759.266086x^8 + -275752903.603842x^7 + 4700873047.890322x^6 + -51191089915.822266x^5 + 369011568500.298100x^4 + -1759197443156.986300x^3 + 5342144
345318.836000x^2 + -9362383549278.719000x^1 + 7200305831156.062500x^0, f(8.32258065) = 36396.671875
```

c. 05/09/2022

```
f(x) = -141120.310600x^9 + 9381759.266086x^8 + -275752903.603842x^7 + 4700873047.890322x^6 + -51191089915.822266x^5 + 369011568500.298100x^4 + -1759197443156.986300x^3 + 5342144
345318.836000x^2 + -9362383549278.719000x^1 + 7200305831156.062500x^0, f(9.16666667) = -664774.421875
Apakah hasil Interpolasi ingin anda simpan ?
```

d. Masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.
13/01/2022 (TTL Samy)

```
Masukkan nama file: 5b4.txt
f(x) = -141120.310600x^9 + 9381759.266086x^8 + -275752903.603842x^7 + 4700873047.890322x^6 + -51191089915.822266x^5 + 369011568500.298100x^4 + -1759197443156.986300x^3 + 5342144
345318.836000x^2 + -9362383549278.719000x^1 + 7200305831156.062500x^0, f(1.41935484) = 8.817800499072214E11
Apakah hasil Interpolasi ingin anda simpan ?
```

b. 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 $h = (2 - 0)/5 = 0.4$.
titik yang diambil {0,0.4,0.8,1.2,1.6,2.0}

```
Masukkan nama file: 5c.txt
f(x) = 0.236256x^5 + -1.421265x^4 + 3.237114x^3 + -3.552684x^2 + 2.035259x^1 + 0.000000x^0, f(1.0) = 0.5346799999999998
Apakah hasil Interpolasi ingin anda simpan ?
```

4.6

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.

Persamaan linear & Hasil Taksiran menggunakan data baru :

```
f(x) = -3.5077781408835103 + -0.002624990745878327X1 + 7.989410472218274E-4X2 + 0.15415503019830143X3
f(50.0, 76.0, 29.3) = 0.9384342262216645
```

4.7

4. Studi Kasus Interpolasi *Bicubic Spline*

Diberikan matriks input dengan bentuk sebagai berikut. Format matriks masukan bukan mewakili nilai matriks, tetapi mengikuti format masukan pada bagian “Spesifikasi Tugas” nomor 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(0.0,0.0) = 21.0`

$f(0.5, 0.5)$ = `f(0.5,0.5) = 87.796875`

$f(0.25, 0.75)$ = `f(0.25,0.75) = 49.698486328125`

$f(0.1, 0.9)$ = `f(0.1,0.9) = 28.969083000000005`

BAB V

KESIMPULAN DAN SARAN

Dalam SPL terdapat beberapa cara penyelesain yaitu metode Gauss, Gauss Jordan, matriks balikan, dan cramer. Untuk perhitungan determinan dapat menggunakan dua cara yaitu reduksi baris dan ekspansi kofaktor.Metode metode tersebut juga dapat bermanfaat selain dalam penyelesaian SPL seperti, untuk melakukan interpolasi polinomial, regresi linear berganda, dan bicubic spline interpolation

Saran

1. Uji kasus sebaiknya diberikan jawaban untuk memastikan program yang dibuat sudah benar.
2. Untuk soal bonus sebaiknya diberikan penjelasan yang lebih jelas khususnya terkait image processing.