Laporan

Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Sistem Persamaan Linier, Determinan, dan Aplikasinya

Disusun Oleh:

Fransiskus Davin Anwari	13520025
Kent Liusudarso	13520069
Rizky Akbar Asmaran	13520111

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

DAFTAR ISI

BAB 1

DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Kami sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}b$), 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, kami diminta membuat satu atau lebih library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, kami menggunakan library tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

1.1 Spesifikasi Tugas

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

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

   Spesifikasi program adalah sebagai berikut:

   1. Program dapat menerima masukan (input) baik dari keyboard maupun membaca masukan dari file text. Untuk SPL, masukan dari keyboard adalah 3 m, n, koefisien aij, dan bi. 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 aij. Masukan dari file berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya

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

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0), (x1, y1), ..., (xn, yn), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

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

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

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

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

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.

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

9. Bahasa program yang digunakan adalah Java.

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

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

```
MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
```

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

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

Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB 2

TEORI SINGKAT

2.1 Eliminasi Gauss

Eliminasi Gauss adalah sebuah metode yang dimanfaatkan untuk memecahkan sistem persamaan liner dengan mempresentasikan (mengubah) menjadi sebuah bentuk matriks, dan matriks tersebut diubah kebentuk "Eselon Baris" dengan menggunakan Operasi Baris Elementer dan diselesaikan dengan subtitusi balik. Sistem Eliminasi ini dibuat dan ditemukan oleh tokoh bernama *Carl Friedrich Gauss.* Algoritma yang dibuat bertujuan untuk menghasilkan sebuah matriks eselon, yaitu sebuah matriks yang dimulai oleh angka 1 pada setiap barisnya dengan kolomnya yang menjorok kedalam.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1n} & d_1 \\ 0 & c_{22} & c_{23} & \cdots & c_{2n} & d_2 \\ 0 & 0 & c_{33} & \cdots & c_{3n} & d_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & c_{nn} & d_n \end{bmatrix}$$

2.2 Eliminasi Gauss-Jordan

Metode Elminasi Gauss-Jordan adalah pengembangan dari metode eliminasi Gauss, akan tetapi matriks *augmented* pada sisi kiri diubah menjadi matriks eselon baris tereduksi. Bentuk eselon baris tereduksi merupakan sebuah bentuk matriks eselon baris yang lebih disederhanakan sehingga pencarian solusi dapat lebih mudah ditemukan. Eselon baris tereduksi bersifat 1 utama yang hanya memiliki angka 1 pada setiap bagian kolom dan barisnya.

$$\begin{bmatrix} a11 & a12 & \cdots & a1n & b1 \\ a21 & a22 & & a2n & b2 \\ \vdots & & \ddots & & \vdots \\ an1 & an2 & \cdots & ann & bn \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \cdots & 0 & d1 \\ 0 & 1 & & 0 & d2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & dn \end{bmatrix}$$

2.3 Determinan

Determinan adalah jumlah hasil kali elementer bertanda dari suatu matriks. Determinan dapat diartikan sebagai jumlah semua dari hasil entri-entri matriks yang tidak berada pada baris atau kolom yang sama. Determinan dapat dinotasikan sebagai det(A). Determinan dari sebuah matriks dapat dicari dengan Operasi Baris Elementer dan Ekspansi Kofaktor.

2.3.1 Metode Ekspansi Kofaktor

Jika A adalah suatu matriks persegi (nxn), maka minor entri a_{ij} dinyatakan sebagai M_{ij} dan didefinisikan menjadi determinan submatriks yang tersisa setelah baris ke i dan kolom ke j dihilangkan dari A. Bilangan $(-1)^{i+j}M_{ij}$ dinyatakn sebagai C_{ij} dan disebut sebagai *Kofaktor* dari entri a_{ij}. Kofaktor dan minor dari suatu elemen a_{ij} hanya berbeda pada tandanya, dimana C_{ij} = C_{ij} + $M_{ij.}$ Terdapat cara yang lebih baik dan lebih cepat untuk menentukan tanda yang menghubungkan C_{ij} dan M_{ij} berada pada baris ke i dan kolom ke j dengan menggunakan susunan berikut:

$$\begin{bmatrix} + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Berdasarkan matriks tanda diatas, maka didapatkan kofaktor:

$C_{11} = M_{11}$, $C_{21} = -M_{21}$, $C_{43} = - M_{43}$, …, $C_{ij} = (-1)^{i+j}M_{ij}$.

Secara matematis, determinan matriks dengan ordo n x n dapat dihitung dengan ekspansi kofaktor yang dapat ditulis sebagai berikut:

$$\det(A) = \sum_{i=0}^{n}(-1)^{i+j}|M_{ij}| \text{ atau } |A| = \sum_{i=0}^{n} a_{ij}C_{ij}$$

2.3.2 Metode Operasi Baris Elementer

Terdapat beberapa langkahb untuk mendapatkan determinan dengan operasi baris elementer sebagai berikut:

1. Lakukan operasi baris dengan memperhatikan tiga hal sebagai berikut:

   a. Jika A’ adalah matriks yang dihasilkan dari pertukaran dua baris matriks A maka det(A’) = –det(A)

   b. Jika A’ adalah matriks yang dihasilkan dari perkalian matriks dengan suatu konstanta k maka det(A’) = kdet(A)

   c. Jika A’ adalah matriks yang dihasilkan dari penjumlahan hasil kali dari baris satu ke baris yang lain pada matriks A maka det(A’) = det(A).

2. Operasi baris berhenti jika matriks yang dihasilkan adalah matriks segitiga atas atau segitiga bawah.

$$A = \begin{matrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{matrix} \qquad B = \begin{matrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{matrix}$$

Matriks A adalah matriks segitiga atas dan matriks B adalah matriks segitiga bawah.

3. Determinan diperoleh dengan mengalikan semua entri pada diagonal utama dari matriks segitiga atas atau segitiga bawah

2.4 Matriks Balikan

Matriks balikan atau biasa disebut invers matriks adalah kebalikan dari sebuah matriks. Invers memiliki syarat yaitu harus merupakan matriks persegi (berukuran n x n) dan determinan yang tidak nol. Untuk mendapatkan matriks balikan terdapat metode adjoin dan metode eliminasi Gauss Jordan.

2.5 Matriks Kofaktor

Kofaktor adalah hasil perkalian minor dengan suatu angka yang besarnya mengikuti aturan $(-1)^{i+j}$ dimana I adalah baris dan j adalah kolom. Kofaktor suatu elemen baris ke-I dan kolom ke-j dari matriks A dilambangkan dengan C_{ij}.

$C_{ij} = (-1)^{I+j} M_{ij}$

Matriks Kofaktor adalah matriks yang terdiri dari kofaktor matriks itu sendiri. Susunan matriks kofaktor tetap mengikuti susunan kofaktor – kofaktornya.

Contoh:

$$S = \begin{matrix} 3 & 1 & -4 \\ 2 & 5 & 6 \\ 1 & 4 & 8 \end{matrix}$$

Minor dari entri a_{11} adalah:

$$M_{11} = \begin{matrix} 3 & 1 & -4 \\ 2 & 5 & 6 \\ 1 & 4 & 8 \end{matrix} = \begin{vmatrix} 5 & 6 \\ 4 & 8 \end{vmatrix} = 16$$

Kofaktor dari entri a_{11} adalah

$C_{11} = (-1)^{1+1} M_{11} = M_{11} = 16$

Minor dari entri a_{32} adalah:

$$M_{32} = \begin{matrix} 3 & 1 & -4 \\ 2 & 5 & 6 \\ 1 & 4 & 8 \end{matrix} = \begin{vmatrix} 5 & -4 \\ 2 & 6 \end{vmatrix} = 26$$

Kofaktor dari entri a_{32} adalah

$C_{32} = (-1)^{3+2} M_{32} = M_{32} = -26$

Perhatikan kofaktor dan minor dari suatu elemen aij hanya berbeda dalam tandanya dimana

$C_{ij} = M_{ij}$

Cara untuk menentukan apakah tanda – atau + yang digunakan adalah dengan menentukan fakta bahwa tanda yang berkaitan dengan Cij dan Mij berada dalam baris ke-i dan kolom ke-j dari susunan "papan catur" berikut:

$$\begin{bmatrix} + & - & + & \cdots \\ - & + & - & \cdots \\ + & - & + & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

2.6 Matriks Adjoin

Adjoin matriks merupakan transpose dari matriks kofaktor. Adjoin A dapat dinotasikan dengan Adj(A). Transpose merupakan pertukaran elemen pada baris menjadi kolom. Adjoin matriks digunakan dalam menentukan invers matriks. Berikut merupakan salah satu contoh matriks Adjoin

$$\begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix} \qquad \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$$

2.7 Kaidah Cramer

Jika *AX* = *B* adalah sebuah sistem persamaan linear dengan A matriks persegi dan det(A) bukan 0, sedangkan maka sistem tersebut mempunyai penyelesaian tunggal (unik).

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ dan } B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Untuk mencari penyelesaian menggunakan kaidah Crammer, didasarkan pada determinan matriks A. bila determinan matriks A bukan 0 maka akan diperoleh suatu penyelesaiank, dan penyelesaian itu adalah:

$$x_1 = \frac{\det(A_1)}{\det(A)},\ x_2 = \frac{\det(A_2)}{\det(A)},\ x_3 = \frac{\det(A_3)}{\det(A)}, \cdots,\ x_n = \frac{\det(A_n)}{\det(A)}$$

Dengan A_i adalah matriks yang diperoleh dengan menggantikan elemen-elemen kolom ke j dari A dengan elemen-elemen matriks B.

2.8 Interpolasi Polinom

Interpolasi polinomial digunakan untuk mencari titik-titik antara dari n buah titik P1(X1, Y1), P2(X2, Y2), P3(X3, Y3), …., P_n (X_n, Y_n) dengan menggunakan pendekatan fungsi polinomial berpangkat n-1:

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

Dengan memasukkan nilai dari setiap titik ke dalam persamaan polinomial diatas dan diperoleh persamaan simultan dengan n persamaan dan n variabel bebas:

$$y_1 = a_0 + a_1 x_1 + a_1 {x_1}^2 + a_1 {x_1}^3 + \cdots + a_{n-1} {x_1}^{n-1}$$

$$y_2 = a_0 + a_1 x_2 + a_1 {x_2}^2 + a_1 {x_2}^3 + \cdots + a_{n-1} {x_2}^{n-1}$$

$$y_3 = a_0 + a_1 x_3 + a_1 {x_3}^2 + a_1 {x_3}^3 + \cdots + a_{n-1} {x_3}^{n-1}$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$y_n = a_0 + a_1 x_n + a_1 {x_n}^2 + a_1 {x_n}^3 + \cdots + a_{n-1} {x_n}^{n-1}$$

2.9 Regresi Linear Berganda

Regresi Linear Berganda menjelaskan bagaimana varibel respon tunggal Y bergantung secara linier pada sejumlah varibel prediktor. Meskipun, sudah ada rumus untuk menghitung regresi linier sederhana, terdapat rumus umum dari regersi linear yang bisa digunakan untuk regresi linier berganda, yaitu:

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

Untuk mendapatkan nilai dari setiap βi dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:

$$\begin{array}{cccccccccc}
n\beta_0 & + & \beta_1 \sum_{i=1}^{n} x_{1i} & + & \beta_2 \sum_{i=1}^{n} x_{2i} & + & \dots & + & \beta_k \sum_{i=1}^{n} x_{ki} & = & \sum_{i=1}^{n} y_i \\
\beta_0 \sum_{i=1}^{n} x_{1i} & + & \beta_1 \sum_{i=1}^{n} {x_{1i}}^2 & + & \beta_2 \sum_{i=1}^{n} x_{1i} x_{2i} & + & \dots & + & \beta_k \sum_{i=1}^{n} x_{1i} x_{ki} & = & \sum_{i=1}^{n} x_{1i} y_i \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
\beta_0 \sum_{i=1}^{n} x_{ki} & + & \beta_1 \sum_{i=1}^{n} x_{ki} x_{1i} & + & \beta_2 \sum_{i=1}^{n} x_{ki} x_{2i} & + & \dots & + & \beta_k \sum_{i=1}^{n} {x_{ki}}^2 & = & \sum_{i=1}^{n} x_{ki} y_i
\end{array}$$

Kemudian, sistem persamaan linier tersebut dapat diselesaikan dengan metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode Matriks Balikan, maupun Kaidah Cramer. Sehingga nantinya persamaan regresi linier berganda diatas dapat digunakan untuk menentukan variable respon tunggal Y yang bergantung secara linier pada sejumlah (i) variable prediktor.

BAB 3

IMPLEMENTASI

3.1 Implementasi Matriks

Public void bacaMatriks () I.S: menerima input berupa jumlah baris dan kolom F.S: mengembalikan matriks dengan jumlah baris dan kolom yang di input
Public void tulisMatriks() I.S: matriks sembarang F.S: Menuliskan matriks ke layar
Public double[][] tambahKolom(double m[][]) I.S: Matriks sembarang F.S: Mengembalikan matriks dengan kolom bertambah 1
Public double getElm(int baris, int kolom) I.S: Matriks sembarang F.S: mengembalikan elemen pada baris dan kolom tertentu
Public void salinMatriks(double m[][], double temp[][], int n) I.S: Matriks (M) terdefinisi dan sembarang F.S: Matriks (M) tersalin.
Public double[][] swapBaris(int baris1, int baris2, double m[][]) I.S: Matriks (M) terdefinisi dan sembarang, baris1 dan baris2 dalam range ukuran baris M F.S: Baris ke baris1 dan baris2 bertukar posisi
Public double[][] removeLastRow(double m[][]) I.S: Matriks sembarang F.S: Mengembalikan matriks dengan baris terakir dihapus

3.2 Implementasi Elminasi Gauss dan Gauss Jordan

Public double [][] splGauss() I.S: Matriks sembarang F.S: Matriks yang sudah dimanipulasi menjadi matriks eselon baris
Public String[] solveGauss() I.S: Matriks eselon baris atau matriks eselon baris tereduksi F.S: Array of string berisi solusi dari SPL matriks yang diberikan, atau tulisan matriks tidak ada jawaban jika matriks tidak memiliki solusi
Public double [][] splGaussJordan() I.S: Matriks sembarang F.S: Matriks yang sudah dimanipulasi menjadi matriks eselon baris tereduksi

3.3 Implementasi Determinan

Static void kofaktor(double matriks[][], double temp[][], int p, int q, int n) I.S: Sebuah Matriks(M) berukuran nxn terdefinisi dan sembarang. F.S: mengembalikan matriks kofaktor Proses: prosedur akan menyalin elemen matriks yang tidak terdapat pada baris ke-a dan ke kolom ke-b.
Public static double KofaktorDet(double m[][], int n) I.S: Sebuah Matriks(M) berukuran nxn terdefinisi dan sembarang.

F.S: mengembalikan hasil perhitungan determinan matriks M dengan ekspansi kofaktor
Public double reduksiBaris(double matriks[][], int n) I.S: Matriks (M) terdefinisi dan sembarang F.S: Matriks (M) menjadi bentuk segitiga atas
Public double ReduksiDet(double matrisk[][], int n) I.S: Sebuah matriks (M) berukuran n x n terdefinisi dan sembarang. F.S: Determinan M dituliskan pada layar atau dalam sebuah file Proses: Mengubah matriks M ke dalam bentuk matriks segitiga, kemudian mengalikan semua elemen pada diagonal utamanya.
Public double[][] swapBaris(int baris1, int baris2, double m[][]) I.S: Matriks (M) terdefinisi dan sembarang, baris1 dan baris2 dalam range ukuran baris M F.S: Baris ke baris1 dan baris2 bertukar posisi
Public void salinMatriks(double matriks[][], double temp[][], int n) I.S: Matriks (M) terdefinisi dan sembarang F.S: Matriks (M) tersalin.

3.4 Implementasi Inverse Matriks

Public double[][] inverseMatriksGauss() I.S: Sebuah matriks (M) berukuran n x n terdefinisi dan sembarang. F.S: Matriks yang sudah ter inverse
Public String splInverse() I.S: matriks A berukuran n x n dan B berukuran n x 1 F.S: solusi SPL dari matriks yang diberikan

3.5 Implementasi Kaidah Cramer

Public String[] splKramer() I.S: Matriks dengan n baris dan n+1 kolom F.S: Array of String berisi solusi dari SPL Matriks yang diberikan

3.6 Implementasi Interpolasi Polinom

Public double[][] polinomMatriks() I.S: Matriks yang berisi titik-titik yang ingin dicari tahu persamaan polinomnya F.S: Matriks berisi persamaan lanjar dari data titik-titik yang diberikan
Public double solvePolinom(double n) I.S: Persamaan lanjar dari polinomMatriks() yang telah dijadikan matriks eselon baris dan x yang ingin ditafsir jawabannya F.S: Jawaban dari x yang ditafsir dengan solusi dari persamaan lanjar
Public String printPolinomsolution() I.S: Persamaan lanjar dari polinomMatriks() yang telah dijadikan matriks eselon baris F.S: Persamaan polinom yang didapat dari persamaan lanjar yang diberikan

3.7 Implementasi Regresi Linier Berganda

Public double[][] Regresi(int perubah, int persamaan, Matriks m1) I.S: perubah pada persamaan regresi dan banyaknya persamaan yang ada, dan matriks berisi semua persamaan yang ada pada data regresi linier berganda

F.S: Matriks SPL setelah dilakukan *Normal Estimation Equation for Multiple Linear Regression*
Public double solveRegresi(double[] n) I.S: Matriks dari setelah *Normal Estimation Equation for Multiple Linear Regression* yang telah dijadikan matriks eselon baris dan array of double berisi nilai x1 sampai xn yang ingin ditafsir F.S: Jawaban dari Regresi Linear Berganda dengan nilai x yang diinput
Public double printRegresisolution() I.S: Matriks dari setelah *Normal Estimation Equation for Multiple Linear Regression* yang telah dijadikan matriks eselon baris F.S: Persamaan y yang didapat dari SPL matriks yang diinput

3.8 Implementasi Manipulasi File

Public void saveSplSolution(String[] solution, String filename) I.S: Terdapat array of String yang didapatkan dari penyelesaian SPL, dan sebuah string yang nantinya berfungsi sebagai nama file yang akan di save F.S: Terbuat file txt dengan nama sesuai yang diinput di ../test yang berisi solusi dari SPL, jika sudah ada nama file yang sama akan menampilkan pesan "File already exist."
Public void bacaMatriksfile(String text, int baris, int kolom) I.S: Terdapat matrix dengan baris dan kolom yang telah ditentukan dan string sebagai nama file yang ingin dibaca F.S: Matrix dengan isi sesuai dengan text yang diinginkan, error jika tidak ada file yang dimaksud
Public void saveMatrix(String filename) I.S: Terdapat matrix sebarang dan string yang nantinya berfungsi sebagai nama file yang akan di save F.S: Terbuat file txt dengan nama sesuai yang diinput di ../test yang berisi matrix yang diinput, jika sudah ada nama file yang sama akan menampilkan pesan "File already exist."
Public void saveStringSolution(String solution, String filename) I.S: Terdapat string berisi solusi kasus dan string yang nantinya berfungsi sebagai nama file yang akan di save F.S: Terbuat file txt dengan nama sesuai yang diinput di ../test yang berisi string yang diinput, jika sudah ada nama file yang sama akan menampilkan pesan "File already exist."

BAB 4

EKSPERIMEN

4.1 SPL

1. Test Case 1

input	output

input

```
Masukkan baris: 4
Masukkan kolom: 5

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6
Masukkan baris: 4
Masukkan kolom: 6

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1
```

```
Masukkan baris: 3
Masukkan kolom: 7

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1
Masukkan baris: 6

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
1 0.5 0.333 0.25 0.2 0.166667 1
0.5 0.333 0.25 0.2 0.166667 0.142857 0
0.333 0.25 0.2 0.166667 0.142857 0.125 0
0.25 0.2 0.166667 0.142857 0.125 0.11111 0
0.2 0.166667 0.142857 0.125 0.11111 0.1 0
0.166667 0.142857 0.125 0.11111 0.1 0.0909090909 0
```

output

```
Hasil matriks:
1.0 1.0 -1.0 -1.0 1.0
0.0 1.0 -1.6666666666666667 -1.0 -1.3333333333333333
-0.0 -0.0 1.0 -1.0 1.0
0.0 0.0 0.0 0.0 1.0
Sistem persamaan linear tidak ada solusi.
x1 = null
x2 = null
x3 = null
x4 = null
[Ljava.lang.String;@568db2f2
```

```
Hasil matriks:
1.0 0.0 0.0 0.0 -1.0 3.0
-0.0 1.0 -0.0 0.0 -2.0 0.0
0.0 0.0 0.0 1.0 -1.0 -1.0
x1 = 3.0 + 1.0  e
x2 = 0.0 + 2.0  e
x3 = c
x4 = -1.0 + 1.0  e
x5 = e
Hasil matriks:
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
-0.0 -0.0 -0.0 -0.0 1.0 -1.0 1.0
x1 = a
x2 = 1.0 - 1.0  f
x3 = c
x4 = -2.0 - 1.0  f
x5 = 1.0 + 1.0  f
x6 = f
```

```
x0=68.935718824619 47
x1=-647.5765002964532
x2=195.19107588734127
x3=5990.574943806644
x4=-11418.591828699036
x5=5861.559473928665
```

<table>
<tr><td>Masukkan baris: 10

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
1 0.5 0.333 0.25 0.2 0.166667 0.142857 0.125 0.11111 0.1 1
0.5 0.333 0.25 0.2 0.166667 0.142857 0.125 0.11111 0.1 0.0909090909 0
0.333 0.25 0.2 0.166667 0.142857 0.125 0.11111 0.1 0.0909090909 0.0833333 0
0.25 0.2 0.166667 0.142857 0.125 0.11111 0.1 0.0909090909 0.0833333 0.076923 0
0.2 0.166667 0.142857 0.125 0.11111 0.1 0.0909090909 0.0833333 0.076923 0.07142857142 0
0.166667 0.142857 0.125 0.11111 0.1 0.0909090909 0.0833333 0.076923 0.07142857142 0.066666 0
0.142857 0.125 0.11111 0.1 0.0909090909 0.0833333 0.076923 0.07142857142 0.066666 0.0625 0
0.125 0.11111 0.1 0.0909090909 0.0833333 0.076923 0.07142857142 0.066666 0.0625 0.05555555 0
0.11111 0.1 0.0909090909 0.0833333 0.076923 0.07142857142 0.066666 0.0625 0.05882352941 0.05555555 0
0.1 0.0909090909 0.0833333 0.076923 0.07142857142 0.066666 0.0625 0.05882352941 0.05555555 0.05263 0</td>
<td>x1 = -3.5781337575118206
x2 = 157.23754969367857
x3 = -672.8388213611532
x4 = 288.5900471095863
x5 = 1617.3549762999303
x6 = -626.6092253808076
x7 = -3013.42830388028
x8 = 2756.7208791881067
x9 = -496.26204748895225
x10 = -1.2984744022423482</td></tr>
<tr><td colspan="2">Note: Test case 1a menggunakan metode eliminasi Gauss.
Test case 1b menggunakan metode eliminasi Gauss-Jordan.
Test case 1c menggunakan metode eliminasi Gauss
Untuk test case 1d menggunakan metode invers dan kaidah kramer agar tidak tertulis hasil manipulasi matriks yang terlalu besar</td></tr>
</table>

2. Test Case 2

<table>
<tr><th>input</th><th>output</th></tr>
<tr><td>Masukkan baris: 4
Masukkan kolom: 5

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
Masukkan baris: 6
Masukkan kolom: 5

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
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</td>
<td>Hasil matriks:
1.0 -1.0 2.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
x1 = -1.0
x2 = 0.0 + 2.0 c
x3 = c
x4 = d

Hasil matriks:
1.0 0.0 4.0 0.0 4.0
0.0 1.0 0.0 4.0 6.0
0.0 0.0 1.0 0.0 1.0
0.0 0.0 0.0 1.0 1.0
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0</td></tr>
<tr><td colspan="2">Note: Untuk test case 2a hasil matriks hanya 2 baris karena baris yang lainnya bukan baris efisien (semuanya berelemen 0), sama halnya dengan test case 2b, yang terdapat 2 baris tidak efisien.</td></tr>
</table>

3. Test case 3

input

```
Masukkan baris: 4
Masukkan kolom: 5

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
8 0 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
```

```
Masukkan baris: 12
Masukkan kolom: 10

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
0 0 0 0 0 0 1 1 1 13
0 0 0 1 1 1 0 0 0 15
1 1 1 0 0 0 0 0 0 8
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.04289 0.75 0.04289 0 0.04289 0 0 3.81
0 0 1 0 0 1 0 0 1 18
0 1 0 0 1 0 0 1 0 12
1 0 0 1 0 0 1 0 0 6
0.04289 0.75 0.61396 0 0.04289 0.75  0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04
```

output

```
Hasil matriks:
1.0 4.5 -0.5 -1.0 0.5
-0.0 1.0 -0.19444444444444445 -0.2777777777777778 0.1111111111111111
0.0 0.0 1.0 -0.18867924528301885 0.7547169811320754
0.0 0.0 0.0 1.0 -0.25882352941176456
x1 = -0.19999999999999996
x2 = 0.17647058823529413
x3 = 0.7058823529411764
x4 = -0.25882352941176456
[Ljava.lang.String;@568db2f2
```

```
Hasil matriks:
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0
0.0 1.0 3.65684 1.0 3.65684 1.0 3.65684 1.0 0.0 57.24
0.0 0.0 1.0 0.0 1.0 17.48659361156447 1.0 17.48659361156447 14.31475868500816 344.8356260200513
0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 15.0
-0.0 -0.0 -0.0 -0.0 1.0 -1.615462680971933766E17 -5.530060054440778E15 -1.6707632815163414E17 -1.3777398906904456E17 -3.1558965052728571E18
-0.0 -0.0 -0.0 -0.0 -0.0 1.0 0.0342320507900167 1.0342320507900167 0.8528453841233499 19.535558093945756
0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 13.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 17.0 13.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -16.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 532.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1792.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -4432.0
Sistem persamaan linear tidak ada solusi.
x1 = null
x2 = null
x3 = null
x4 = null
x5 = null
x6 = null
x7 = null
x8 = null
x9 = null
```

Note: Kedua test case menggunakan metode eliminasi Gauss.

4. Test case 4

input

```
Masukkan baris: 3
Masukkan kolom: 4

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
4 -2 0 1
1 -2 1 0
0 -2 4 1
```

output

```
Hasil matriks:
1.0 -0.5 0.0 0.25
-0.0 1.0 -0.6666666666666666 0.16666666666666666
0.0 0.0 1.0 0.49999999999999994
x1 = 0.49999999999999994
x2 = 0.4999999999999999
x3 = 0.49999999999999994
```

Note: Test case menggunakan metode eliminasi Gauss

5. Test case 5

input	output

<table>
<tr><td>Masukkan baris: 3
Masukkan kolom: 4

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
120 -60 0 1300
40 -80 0 0
-80 -20 150 200</td><td>Hasil matriks:
1.0 0.25 -1.875 -2.5
-0.0 1.0 -0.8333333333333334 -1.1111111111111112
0.0 0.0 1.0 10.0
x1 = 14.444444444444445
x2 = 7.222222222222223
x3 = 10.0
[Ljava.lang.String;@6d03e736</td></tr>
<tr><td colspan="2">Note: Mencari jawaban menggunakan metode eliminasi Gauss.</td></tr>
</table>

4.2 Determinan

1. Test Case 1

<table>
<tr><th>input</th><th>output</th></tr>
<tr><td>Masukkan besar matriks: 3
Membaca text(1) atau input(2) :
2
Isi matriks:
2 4 8
4 2 5
6 7 3</td><td>Isi matriks:
2 4 8
4 2 5
6 7 3
Determinan adalah 142.0</td></tr>
<tr><td colspan="2">Note: Menggunakan metode Ekspansi Kofaktor</td></tr>
</table>

2. Test Case 2

<table>
<tr><th>input</th><th>output</th></tr>
<tr><td>Masukkan besar matriks: 5
Membaca text(1) atau input(2) :
2
Isi matriks:
2 4 6 3 1
2 5 7 4 2
5 7 3 6 2
2 5 7 3 2
5 2 6 2 6</td><td>Isi matriks:
2 4 6 3 1
2 5 7 4 2
5 7 3 6 2
2 5 7 3 2
5 2 6 2 6
Determinan adalah 202.0</td></tr>
<tr><td colspan="2">Note: Menggunakan Metode Reduksi Baris</td></tr>
</table>

4.3 Matriks Balikan

input	output

Masukkan besar matriks: 3 1. Membaca File 2. Input Matriks -> Pilihan: 2 Isi matriks: 3 2 -1 1 6 3 2 -4 0	Hasil Inverse: 0.1875 0.0625 0.1875 0.09375 0.03125 -0.15625000000000003 -0.25 0.25 0.25
Note: Matriks balikan menggunakan metode gauss	

input	output
Masukkan besar matriks: 3 1. Membaca File 2. Input Matriks -> Pilihan: 2 Isi matriks: 2 4 1 3 0 2 5 4 3	Determinan = 0, matriks tidak memiliki balikan. Hasil Inverse: 2.0 4.0 1.0 3.0 0.0 2.0 5.0 4.0 3.0
Note: Matriks singular memiliki determinan = 0, sehingga tidak memiliki balikan	

4.4 Interpolasi Polinom

input	output
Banyak persamaan: 7 1. Membaca File 2. Input Matriks -> Pilihan: 2 Isi matriks: 0.1 0.003 0.3 0.067 0.5 0.148 0.7 0.248 0.9 0.37 1.1 0.518 1.3 0.697	Nama file: kasuspolinom1 Dicari f(x) untuk x = 0.2 f(x) = -0.022976562500000014 x^0 + 0.24000000000000246 x^1 + 0.19739583333333196 x^2 + 3.4409860425904843E-14 x^3 + 0.026041666666624073 x^4 + 2.5501338598675655E-14 x^5 + -5.910407676398067E-15 x^6 f(x) = 0.032960937500000002 untuk x = 0.2 Nama file: kasuspolinom1 Dicari f(x) untuk x = 0.55 f(x) = -0.022976562500000014 x^0 + 0.24000000000000246 x^1 + 0.19739583333333196 x^2 + 3.4409860425904843E-14 x^3 + 0.026041666666624073 x^4 + 2.5501338598675655E-14 x^5 + -5.910407676398067E-15 x^6 f(x) = 0.17111865234374998 untuk x = 0.55 Nama file: kasuspolinom1 Dicari f(x) untuk x = 0.85 f(x) = -0.022976562500000014 x^0 + 0.24000000000000246 x^1 + 0.19739583333333196 x^2 + 3.4409860425904843E-14 x^3 + 0.026041666666624073 x^4 + 2.5501338598675655E-14 x^5 + -5.910407676398067E-15 x^6 f(x) = 0.33723583984375005 untuk x = 0.85 Nama file: kasuspolinom1 Dicari f(x) untuk x = 1.28 f(x) = -0.022976562500000014 x^0 + 0.24000000000000246 x^1 + 0.19739583333333196 x^2 + 3.4409860425904843E-14 x^3 + 0.026041666666624073 x^4 + 2.5501338598675655E-14 x^5 + -5.910407676398067E-15 x^6 f(x) = 0.6775418375000001 untuk x = 1.28
Note: Untuk kasus interpolasi polinom menggunakan metode eliminasi gauss **Untuk lebih mudah dilihat jawaban ditulis lagi:** - f(x) = 0.03296 untuk x = 0.2 - f(x) = 0.17112 untuk x = 0.55 - f(x) = 0.33723 untuk x = 0.84 - f(x) = 0.67754 untuk x = 1.28	

<table>
<tr><th>input</th><th>output</th></tr>
<tr><td>Banyak persamaan: 10

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
6.567 12624
7 21807
7.258 38391
7.451 54517
7.548 51952
7.839 28228
8.161 35764
8.484 20813
8.709 12408
9 10534
Dicari f(x) untuk x =</td><td>Nama file:
kasuspolinom2
Dicari f(x) untuk x = 7.51612903226
f(x) = 7.187066071658637E12 x^0 + -9.346993079172963E12 x^1 + 5.334203055240283E12 x^2 + -1.7568101863613738E12 x^3 + 3.685508071755316E11 x^4 + -5.1131876760132576E10 x^5 + 4.695806315428787E9 x^6 + -2.7547453942066926E8 x^7 + 9372849.23910132 x^8 + -140993.71224863594 x^9
f(x) = 53532.501953125 untuk x = 7.51612903226

Nama file:
kasuspolinom2
Dicari f(x) untuk x = 8.32258064516
f(x) = 7.187066071658637E12 x^0 + -9.346993079172963E12 x^1 + 5.334203055240283E12 x^2 + -1.7568101863613738E12 x^3 + 3.685508071755316E11 x^4 + -5.1131876760132576E10 x^5 + 4.695806315428787E9 x^6 + -2.7547453942066926E8 x^7 + 9372849.23910132 x^8 + -140993.71224863594 x^9
f(x) = 36315.421875 untuk x = 8.32258064516

Nama file:
kasuspolinom2
Dicari f(x) untuk x = 9.16666666667
f(x) = 7.187066071658637E12 x^0 + -9.346993079172963E12 x^1 + 5.334203055240283E12 x^2 + -1.7568101863613738E12 x^3 + 3.685508071755316E11 x^4 + -5.1131876760132576E10 x^5 + 4.695806315428787E9 x^6 + -2.7547453942066926E8 x^7 + 9372849.23910132 x^8 + -140993.71224863594 x^9
f(x) = -664791.40625 untuk x = 9.16666666667

Nama file:
kasuspolinom2
Dicari f(x) untuk x = 8
f(x) = 7.187066071658637E12 x^0 + -9.346993079172963E12 x^1 + 5.334203055240283E12 x^2 + -1.7568101863613738E12 x^3 + 3.685508071755316E11 x^4 + -5.1131876760132576E10 x^5 + 4.695806315428787E9 x^6 + -2.7547453942066926E8 x^7 + 9372849.23910132 x^8 + -140993.71224863594 x^9
f(x) = 27752.4375 untuk x = 8.0</td></tr>
<tr><td colspan="2">Note: Untuk kasus interpolasi polinom menggunakan metode eliminasi gauss
Untuk lebih mudah dilihat jawaban ditulis lagi:
- f(x) = 53533 untuk x = 7.5161
- f(x) = 36316 untuk x = 8.3225
- f(x) = -664791 untuk x = 9.16667
- f(x) = 27753 untuk x = 8 (31/7/2021)</td></tr>
</table>

<table>
<tr><th>input</th><th>output</th></tr>
<tr><td>Banyak persamaan: 6

1. Membaca File
2. Input Matriks
-> Pilihan: 2

Isi matriks:
0.0 0
0.4 0.418
0.8 0.507
1.2 0.561
1.6 0.583
2.0 0.576</td><td>Dicari f(x) untuk x = 1
f(x) = 3.7730235602495554E-17 x^0 + 2.0286250000000114 x^1 + -3.5382812500000576 x^2 + 3.228515625000092 x^3 + -1.4208984375000573 x^4 + 0.23681640625001218 x^5
f(x) = 0.5347773437500007 untuk x = 1.0</td></tr>
<tr><td colspan="2">Note: Untuk kasus interpolasi polinom menggunakan metode eliminasi gauss</td></tr>
</table>

4.5 Regresi Linear Berganda

Input/output

```
Banyak peubah: 3
Banyak persamaan: 20

1. Membaca File
2. Input Matriks
-> Pilihan: 1

Nama file: 7
20.0 863.0999999999999 1530.2000000000003 587.8399999999999 19.42
863.0999999999999 54876.89 66997.51 25283.395 779.4769999999999
1530.2000000000003 66997.51 117885.40000000001 44970.98899999999 1483.2169999999996
587.8399999999999 25283.395 44970.98899999999 17278.508600000005 571.1219000000001
y = -3.504210787558894 + -0.002624728214054177 x1 + 7.822236313393433E-4 x2 + 0.15407706190809164 x3 + ϵ
Dicari y untuk:
x1: 50
x2: 76
x3: 29.30
y = 0.9384597116272726
```

Note: Test case diselesaikan dengan melakukan metode eliminasi Gauss.

BAB 5

KESIMPULAN

5.1 Kesimpulan

Matriks merupakan sekumpulan bilangan yang disusun secara baris dan kolom. Terdapat banyak jenis matriks dan operasi yang dapat dilakukan pada matriks. Program tentang matriks yang dibuat dengan Java oleh kelompok kami pada tugas besar ini dapat mempermudah untuk menemukan hasil dari operasi – operasi matriks yang tersedia seperti SPL, determinan, inverse, cramer, interpolasi polinom, dan regresi liner berganda.

5.2 Saran

1. Memberikan contoh struktur program pada Java agar lebih mudah dalam koordinasi pengerjaan program.
2. Memberi test case yang memiliki jawaban agar tidak memakan waktu kebingungan ketika mencoba kode
3. Program akan jauh lebih baik apabila dapat menggunakan GUI agar *user friendly*.

5.3 Refleksi

1. Perlu lebih mendalami tentang penggunaan github, karena sering terjadi tabrakan saat satu anggota kelompok sedang koding dan anggota lainnya meng-*push* kodingannya.
2. Meningkatkan komunikasi antar anggota kelompok untuk meminimalisir terjadinya miskomunikasi.
3. Program berhasil dibuat sesuai dengan spesifikasi dan telah berhasil dilakukan uji coba dengan beberapa testcase.

DAFTAR REFERENSI

Dr. Ir. Rinaldi Munir, MT. (2021). Aljabar Linier dan Geometri [Presentasi PowerPoint]. Diakses dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/algeo21-22.htm

W3schools. Java Tutorial. Diakses pada 17 September 2021, dari https://www.w3schools.com/java/default.asp

W3schools. Java Create and Write To Files. Diakses pada 25 September 2021, dari https://www.w3schools.com/java/java_files_create.asp

dicoding. Memulai Pemograman dengan Java. Diakses pada 19 September 2021, dari **https://www.dicoding.com/academies/60**