TUGAS BESAR 1
IF2123 ALJABAR LINEAR DAN GEOMETRI 2021/2022
SISTEM PERSAMAAN LINEAR, DETERMINAN, DAN APLIKASINYA

Kelompok 39:

Christine Hutabarat (1352005)
Jeremy S.O.N. Simbolon (13520042)
Angelica Winasta Sinisuka (13520097)

TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG

2021

BAB 1 DESKRIPSI MASALAH

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

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.

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 ada tiga: 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 10
   -3 7 8.3 11
   0.5 -10 -9 12

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. 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 x_{1i} , x_{2i} , ..., x_{ni} , nilai y_i , dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran (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 x_4 = -2, x_3 = 2s – t, x_2 = s, dan x_1 = 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 LANDASAN TEORI

1. Eliminasi Gauss

   Elminasi Gauss merupakan salah satu metode untuk mendapatkan solusi dari suatu persamaan linear menggunakan matriks dengan menerapkan Operasi Baris Elemneter (OBE) pada matriks augmented SPL sampai terbentuk matriks eselon baris. Contohnya seperti berikut

   $$\begin{aligned} 3x_1 + 4x_2 - 8x_3 &= 6 \\ 5 - 3x_2 - 6x_3 &= 2 \\ 4x_1 + 9x_2 - 7x_3 &= 3 \end{aligned} \qquad \longrightarrow \qquad \left[\begin{array}{ccc|c} 3 & 4 & -8 & 6 \\ 5 & -3 & -6 & 2 \\ 4 & 9 & -7 & 3 \end{array}\right]$$

   Dalam metode ini, dilakukan OBE terhadap matriks augmented sehingga didapat eselon baris, yaitu matriks segitiga atas dengan dengan elemen M_{ii} = 1 sebagai berikut

   $$\left[\begin{array}{ccc|c} 1 & 4/3 & -8/3 & 2 \\ 0 & 1 & -22/29 & 24/29 \\ 0 & 0 & 1 & -233/29 \end{array}\right]$$

   Kemudian diperoleh solusi SPL dari matriks eselon baris di atas adalah

   $$x_1 + \left(\frac{4}{3}\right) x_2 - \left(\frac{8}{3}\right) x_3 = 2 \quad \dots (1)$$

   $$x_2 - \left(\frac{22}{29}\right) x_3 = \frac{24}{29} \quad \dots (2)$$

   $$x_3 = -\frac{233}{29} \quad \dots (3)$$

   Solusi akhir setelah dilakukan teknik penyulihan mundur adalah

   $x_1 = -\frac{218}{187}$, $x_2 = -\frac{2}{17}$ $x_3 = -\frac{233}{187}$

2. Eliminasi Gauss-Jordan

   Metode eliminasi Gauss-Jordan merupakan lanjutan dari eliminasi Gauss dan proses yang dilakukan dalam kedua metode tersebut pun sama. Perbedaan dari kedua metode ini terletak pada hasil akhirnya, di mana metode eliminasi Gauss-Jordan memiliki hasil akhir berupa matriks eselon baris tereduksi sedangkan eliminasi Gauss berhenti saat telah terbentuk matriks eselon baris. Matriks eselon baris tereduksi memiliki sifat yang sama dengan matriks eselon baris, dengan tambahan sifat yaitu pada setiap kolom yang memiliki *leading one,* elemen lain dalam kolom tersebut bernilai nol. Berikut adalah beberapa contoh dari matriks eselon tereduksi.

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

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

3. Determinan Matriks

Determinan dari suatu matriks persegi dapat dicari dengan beberapa cara. Cara yang pertama adalah dengan melakukan ekspansi kofaktor. Dalam metode ini, determinan dari sebuah matriks didefinisikan sebagai hasil penjumlahan dari minor dan kofaktor yang saling bersesuaian. Sebagai contoh, diberikan matriks A yang berukuran 3x3.

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

Akan digunakan kolom pertama sebagai acuan. Maka det(A) dapat dihitung seperti sebagai berikut

$$\begin{aligned} \det(A) &= \begin{vmatrix} 3 & 1 & 0 \\ -2 & -4 & 3 \\ 5 & 4 & -2 \end{vmatrix} \\ &= 3\begin{vmatrix} -4 & 3 \\ 4 & -2 \end{vmatrix} - (-2)\begin{vmatrix} 1 & 0 \\ 4 & -2 \end{vmatrix} + 5\begin{vmatrix} 1 & 0 \\ -4 & 3 \end{vmatrix} \\ &= 3(-4) + 2(-2) + 5(3) \\ &= -12 - 4 + 15 \\ &= -1 \end{aligned}$$

Nilai determinan akan tetap sama, terlepas dari baris ataupun kolom yang dijadikan acuan dalam perhitungan.

Cara yang kedua adalah dengan melakukan operasi baris elementer dan mengubah matriks menjadi matriks segitiga atas. Pada matriks segitiga atas, semua elemen yang berada di bawah diagonal utama bernilai nol. Determinan kemudian diperoleh dengan cara mengalikan semua elemen yang berada pada diagonal utama. Akan dicari kembali determinan dari matriks A menggunakan metode yang kedua ini.

$$\begin{aligned} \det(A) &= \begin{vmatrix} 3 & 1 & 0 \\ -2 & -4 & 3 \\ 5 & 4 & -2 \end{vmatrix} \\ &= \begin{vmatrix} 3 & 1 & 0 \\ 0 & -\frac{10}{3} & 3 \\ 5 & 4 & -2 \end{vmatrix} && R2 + \frac{2}{3}R1 \\ &= \begin{vmatrix} 3 & 1 & 0 \\ 0 & -\frac{10}{3} & 3 \\ 0 & \frac{7}{3} & -2 \end{vmatrix} && R3 - \frac{5}{3}R1 \\ &= \begin{vmatrix} 3 & 1 & 0 \\ 0 & -\frac{10}{3} & 3 \\ 0 & 0 & \frac{1}{10} \end{vmatrix} && R3 + \frac{7}{10}R2 \\ &= 3 \times \left(-\frac{10}{3}\right) \times \frac{1}{10} \\ &= -1 \end{aligned}$$

4. Matriks Balikan

   Misalkan diberikan suatu matriks A. Matriks balikan dari A dapat dicari dengan menggunakan rumus

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

   Di mana det(A) merupakan determinan matriks A dan adj(A) merupakan matriks adjoin dari A. Matriks adjoin A adalah hasil transpose dari matriks kofaktor A. Berdasarkan rumus tersebut, dapat dilihat bahwa jika determinan matriks A bernilai nol, maka matriks A tidak dapat dibalikkan.

5. Matriks Kofaktor dan Adjoint Matriks

   Misalkan ada sebuah matiks A. Matriks ini berukuran $n \times n$ dan C_{ij} merupakan kofaktor dari a_{ij} (elemen matriks posisi ij). yang dituliskan dengan

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

   C_{ij} merupakan kofaktor elemen a_{ij} dan M_{ij} merupakan minor dari elemen a_{ij}. Dengan demikian didapatkan kofaktor matriks A missal dengan ukuran 3×3 adalah sebagai berikut

$$kof\ (A) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$
$$= \begin{bmatrix} M_{11} & -M_{12} & M_{13} \\ -M_{21} & M_{22} & -M_{23} \\ M_{31} & -M_{32} & M_{33} \end{bmatrix}$$

   Adjoint matriks A adalah transpose matriks kofaktor A.

6. Kaidah Cramer

   $Ax = B$ merupakan SPL dengan m buah persamaan dan n variabel seperti $x_1, x_2, x_3 \dots., x_n$ berbentuk:

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

   Dapat juga diubah dalam bentuk perkalian matriks

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}}_{\mathbf{b}}$$

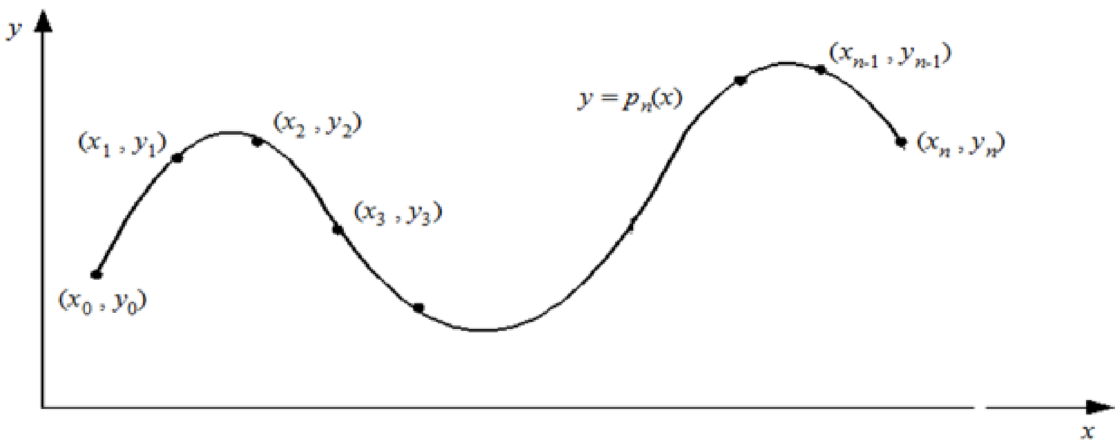
Apabila det(A) ≠ 0, maka SPL memiliki solusi unik, seperti berikut

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

Dimana A_i merupakan matriks yang didapat dengan menggantikan b pada kolom I matriks A.

7. 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 yi = 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) = $a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah p1(x) = a0 + a1x yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0, y_0), (x_1, y_1), dan (x2 , y2), 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 p3(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) = aa_0 + a_1x + a_2x^2 + ... + a_nx^n untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2, …, a_n

Interpolasi polinomial dimanfaatkan untuk mencari titik antara dari n buah titik menggunakan fungsi polinomial berderajat $n-1$. Setiap titik (x_n, y_n) pada persamaan polinomial akan memenuhi persamaan umum polinomial berikut

$$y_n = a_0 + a_1x_n + a_2x_n^2 + \cdots + a_{n-1}x_n^{n-1}$$

Untuk menentukan koefisien-koefisien x, dapat digunakan fakta bahwa titik-titik pada polinom tersebut juga memenuhi sistem persamaan garis berikut.

$$\begin{array}{l}
a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_{n-1}x_1^{n-1} = y_1 \\
a_0 + a_1x_2 + a_2x_2^2 + \cdots + a_{n-1}x_2^{n-1} = y_2 \\
\vdots \quad\;\; \vdots \quad\;\;\; \vdots \qquad\qquad\quad \vdots \qquad\quad \vdots \\
a_0 + a_1x_n + a_2x_n^2 + \cdots + a_{n-1}x_n^{n-1} = y_n
\end{array}$$

Sistem persamaan linier di atas dapat diselesaikan menggunakan metode eliminasi Gauss.

8. Regresi Linear Berganda

Regresi linear berganda adalah salah satu metode yang digunakan untuk mencari hubungan antara beberapa variabel independen x terhadap respons y. Model regresi linear berganda adalah sebagai berikut.

$$y_i = b_0 + b_1x_{1i} + b_2x_{2i} + \cdots + b_kx_{ki}$$

Dapat diturunkan bahwa titik-titik pada persamaan di atas memenuhi sistem persamaan berikut.

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

Sistem persamaan di atas dapat diselesaikan menggunakan metode eliminasi Gauss.

BAB 3 IMPLEMENTASI PROGRAM

Terdapat beberapa kelas yang digunakan berdasarkan kegunaannya dengan metode dan atribut sebagai berikut

3.1. Class Matrix

3.1.1 Atribut

`double[][] contents` `// Berisi elemen-elemen dari matriks`
`int row` `// Jumlah baris dari matriks`
`int col` `// Jumlah kolom dari matriks`

3.1.2 Method

`Matrix createMtr(int a, int b)` `// Membuat matriks kosong berukuran a x b`

`void readMtr (Matrix m)`
`// I.S. Menerima matriks yang sudah dialokasikan memorinya`
`// F.S. Mengembalikan matriks yang sudah terisi`
`// Proses: Pengisian matriks`

`void displayMtr (Matrix m)`
`// I.S. Menerima matrix`
`// F.S. Display ke screen matriksnya`

`int countZero (Matrix m, int i)` `// Mengembalikan jumlah nol di awal baris ke-i suatu matriks`
`boolean isSquare(Matrix m)` `// Mengembalikan true jika matriks m berukuran n x n, apabila tidak, menghasilkan false`
`int rowtoSwap (Matrix m, int i)` `// Mengembalikan baris di bawah baris ke-i yang jumlah leading zeronya lebih sedikit atau sama dengan i`

`void swapRow(Matrix m, int a, int b)`
`// I.S. Menerima bentuk matriks m yang sudah terisi dan variabel a dan b sebagai indeks baris`
`// F.S. Menukar baris m[a] dengan m[b]`

`void segitigaAtas (Matrix m, double[] det)`
`// I.S. Matriks merupakan matriks persegi dan menerima det untuk dilakukan perkalian -1 apabila terjadi swapRow satu kali`
`// F.S. Menghasilkan matriks segitiga atas`

double determinant (Matrix m) // Mengembalikan determinan dari matriks m
boolean isZeroRow (Matrix m, int i) // mengembalikan true jika semua elemen pada baris i bernilai nol, false jika tidak
boolean isZeroCol (Matrix m, int j) // mengembalikan true jika semua elemen pada kolom j bernilai nol, false jika tidak
Matrix signCofactorFunction(Matrix main) // mengembalikan matriks dengan tanda kofaktor yang telah disesuaikan
Matrix CofactorFunction(Matrix main) // Mengembalikan matrikx main yang berisi elemen kofaktornya

Matrix transpose(Matrix main)
// Menukar baris dengan kolom matriks sesuai dengan urutannya kemudian
//dikembalikan

void inverseFunction(Matrix main, Matrix inverse)
// I.S. Matriks main terisi dan bebas sizenya, Matriks inverse kosong d
//an memiliki size yang sama dengan matriks main
// F.S. Mengembalikan Matriks inverse yang sudah di-inverse dengan metode
//adjoin

void identity(Matrix m, Matrix mIdn)
// I.S. Matriks m terdefinisi dan tidak kosong, Matriks mIdn matriks koson
g yang memiliki size yang sama dengan matrix m
// F.S. Matrikx mIdn menjadi matriks identity

String crammer(Matrix m)
// Pesan berupa string dikembalikan berdasarkan hasil proses
// Proses: Membaca matriks kemudian diproses dengan metode crammer

void sortRow (Matrix m) // mengurutkan baris berdasarkan jumlah leading zero dari yang terkecil hingga terbesar

void gaussElim (Matrix m)
// I.S. Matriks m terdefinisi dan tidak kosong
// F.S. Dilakukan eliminasi Gauss pada matriks m, matriks m berupa matriks eselon

void gaussJordanElim (Matrix m)
// I.S. Matriks m terdefinisi dan tidak kosong
// F.S. Dilakukan eliminasi Gauss-Jordan pada matriks m, matriks m berupa matriks eselon tereduksi

String[] spl (Matrix m) // Membaca matriks augmented m kemudian mengembalikan hasil dari sistem persamaan linier

void inversegauss (Matrix m)
// I.S. Matriks m terdefinisi dan tidak kosong
// F.S. Matriks m telah di-inverse dengan metode eliminasi Gauss

double cofactordeterminant(Matrix m) // Menghasilkan determinan Matrix m dengan metode cofactor
Matrix multiplyMatrix(Matrix m1, Matrix m2) // Mengembalikan hasil perkalian matriks m1 dan m2
String matriksbalikan(Matrix m) // Mengembalikan String hasil proses matriks augmented m berupa solusi dari sistem persamaan linear

```
// Proses: Mengecek jika matriks m dapat dikerjakan dengan metode matriks balikan, jika iya, mengembalikan solusi, jika tidak, mengeluarkan pesan kesalahan
```

3.2 Class MatrixInterpolation

3.2.1 Method

```
Object[] createPolynomial(Matrix point_matrix, double num)
// Mengembalikan tuple yang berisi array string beranggotakan
   solusi-solusi spl serta string fungsi hasil substitusi num
```

3.3 Class MatrixRegression

3.3.1 Method

```
double calcEstimation(Matrix point_list, double[] estimate_list)
// Menyubstitusi elemen estimate_list yang bersesuaian ke dalam hasil
   regresi point_list dan mengembalikannya
```

3.4 Class MatrixFile

3.4.1 Method

```
Matrix readMatrixFile()
// membaca dan memproses matriks dari sebuah file bertipe .txt, kemudian
  mengembalikan objek Matrix
```

```
void displayMatrixFile(Matrix matrix)
// I.S. Diberikan Matrix matrix dan file target
// F.S. Matriks telah dituliskan ke dalam file target
```

```
Object[] readInterpolationFile()
// Membaca masukan pengguna dari file serta mengembalikan tuple berisikan
// matriks titik interpolasi dan nilai yang ingin ditaksir
```

```
void displayInterpolationFile(String result, String function_result)
// I.S. Diberikan hasil interpolasi dan file target
// F.S. Hasil interpolasi telah dituliskan ke dalam file target
```

```
Object[] readRegressionFile()
// Membawa masukan pengguna dari file serta mengembalikan tuple berisikan
// matriks titik-titik yang akan diregresikan dan array nilai-nilai yang
// akan digunakan dalam penaksiran
```

```
void displayRegressionFile(double estimate_result)
// I.S. Diberikan hasil regresi linier dan file target
// F.S. Hasil regresi linier telah dituliskan ke dalam file target
```

```
File ensureDirectoriesExist()
// Memastikan direktori serta file yang dituju dalam proses input output
// siap dipakai; mengembalikan path dari file yang dituju
```

```
void displaySPLResFile (String[] splRes)
// I.S. Diberikan hasil SPL dan file target
// F.S. Hasil SPL telah dituliskan ke dalam file target
```

```
void displayDetFile (double det)
// I.S. Diberikan nilai determinan matriks dan file target
// F.S. Determinan matriks telah dituliskan ke dalam file target
```

```
void displayCramer (Matrix m)
// I.S Diberikan hasil pesan yang dikeluarkan hasil metode Crammer
```

//F.S. Pesan proses metode crammer telah dituliskan ke dalam file target

void displaymatrixbalikan(Matrix m)
//I.S. Diberikan hasil pesan yang dikeluarkan hasil SPL metode matrix //balikan
//F.S. Pesan berupa solusi atau error telah dituliskan ke dalam file target

3.5 Class MatrixTerminal

3.5.1 Method

Matrix readMatrixTerminal() // membaca matriks melalui input terminal

void displayMatrixTerminal(Matrix matrix)
// I.S. Diberikan matriks matrix
// F.S. Matriks ditampilkan pada terminal

Object[] readInterpolationTerminal()
// Membaca masukan pengguna dari terminal serta mengembalikan tuple
// berisikan matriks titik interpolasi dan nilai yang ingin ditaksir

void displayInterpolationTerminal(String result, String function_result)
// I.S. Diberikan hasil interpolasi
// F.S. Hasil interpolasi ditampilkan pada terminal

Object[] readRegressionTerminal()
// Membawa masukan pengguna dari terminal serta mengembalikan tuple
// berisikan matriks titik-titik yang akan diregresikan dan array nilai-
// nilai yang akan digunakan dalam penaksiran

void displayRegressionTerminal(double estimate_result)
// I.S. Diberikan hasil regresi linier
// F.S. Hasil regresi liner ditampilkan pada terminal

void displaySPLResTerminal (String[] splRes)
// I.S. Diberikan hasil SPL
// F.S. Hasil SPL ditampilkan pada terminal

void displayDetTerminal (double det)
// I.S. Diberikan nilai determinan matriks
// F.S. Determinan matriks ditampilkan pada terminal

3.6 Class MatrixCalculator

3.6.1 Method

void main(String[] args)
// I.S. Sembarang
// F.S. Kalkulator selesai digunakan

void mainMenu()
// I.S. Sembarang
// F.S. Ditampilkan daftar operasi yang dapat dilakukan kalkulator

void linearMenu()
// I.S. Pengguna memilih menu untuk menyelesaikan sistem persamaan linier pada main menu
// F.S. Hasil SPL ditampilkan berdasarkan metode penyelesaian yang dipilih pengguna

void determinantMenu()
// I.S. Pengguna memilih menu untuk mencari determinan matriks
// F.S. Nilai determinan matriks dicari menggunakan metode pilihan pengguna

`void inverseMenu()`
`// I.S. Pengguna memilih menu untuk mencari matriks balikan`
`// F.S. Matriks balikan dicari dan ditampilkan berdasarkan metode pilihan pengguna`

`void polynomialMenu()`
`// I.S. Pengguna memilih menu untuk mencari hasil interpolasi polynomial`
`// F.S. Hasil interpolasi telah dicari dan ditampilkan/ditulis ke dalam file`

`void regressionMenu()`
`// I.S. Pengguna memilih menu untuk mencari hasil regresi linier`
`// F.S. Hasil regresi telah dicari dan ditampilkan/ditulis ke dalam file`

`int inputMenu()` `// Menampilkan pilihan metode input dan mengembalikan pilihan pengguna`
`int outputMenu()` `// Menampilkan pilihan metode output dan mengembalikan pilihan pengguna`

3.7 Garis Besar Program

Program ini dibuat dengan dasar mempunyai fitur untuk menghitung solusi sistem persamaan linear, determinan, inverse, interpolasi polinomial, dan regresi linear ganda. Saat memulai program, user akan masuk main menu, kemudian diarahkan untuk memilih dari metode apa yang diinginkan. untuk menyelesaikan masalah tertentu. User akan disuruh untuk memilih jenis input antara file dan keyboard. Setelah user memasukkan angkanya, program akan mengeluarkan hasil kalkulasi berdasarkan jenis permasalahannya.

BAB 4 EKSPERIMEN

<table>
<tr><th>No</th><th>Subno</th><th>Hasil Eksperimen</th><th>Hasil Analisis</th></tr>
<tr><td rowspan="3">1</td><td>a</td><td>Input choice (number) : 1
List of methods:
1. Gaussian Elimination
2. Gauss-Jordan Elimination
3. Matrix Inversion
4. Cramer's Rule
5. Back

Input method (number) : 2
Choice of input:
1. Terminal
2. File

Input choice (number): 2
Input file name: testcase1a
Choice of output:
1. Terminal
2. File

Output choice (number): 1
1.000000 0.000000 0.000000 0.666667 0.000000
0.000000 1.000000 0.000000 -2.666667 0.000000
0.000000 0.000000 1.000000 -1.000000 0.000000
0.000000 0.000000 0.000000 0.000000 1.000000

Sistem persamaan linear tidak memiliki solusi.</td><td>Dengan menggunakan metode Gauss elimination, SPL kasus 1a tidak memiliki solusi</td></tr>
<tr><td>b</td><td>List of methods:
1. Gaussian Elimination
2. Gauss-Jordan Elimination
3. Matrix Inversion
4. Cramer's Rule
5. Back

Input method (number) : 2
Choice of input:
1. Terminal
2. File

Input choice (number): 2
Input file name: testcase1b
Choice of output:
1. Terminal
2. File

Output choice (number): 1
1.000000 0.000000 0.000000 0.000000 -1.000000 3.000000
0.000000 1.000000 0.000000 0.000000 -2.000000 0.000000
0.000000 0.000000 0.000000 1.000000 -1.000000 -1.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

x1 = 3.000000 + x5
x2 = 2.000000x5
x3 = free
x4 = -1.000000 + x5
x5 = free</td><td>Dengan metode Gauss-Jordan Elimination, solusi SPL kasus 1b ialah

$x_1 = 3 + x_5$
$x_2 = 2x_5$
$x_3 = free$
$x_4 = -1 + x_5$
$x_5 = free$</td></tr>
<tr><td>c</td><td></td><td></td></tr>
</table>

		List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 1 Choice of input: 1. Terminal 2. File Input choice (number): 2 Input file name: testcase1c Choice of output: 1. Terminal 2. File Output choice (number): 1 0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 0.000000 0.000000 1.000000 0.000000 1.000000 -2.000000 0.000000 0.000000 0.000000 0.000000 1.000000 -1.000000 1.000000 x1 = free x2 = 1.000000 - x6 x3 = free x4 = -2.000000 - x6 x5 = 1.000000 + x6 x6 = free	Dengan metode Gauss Elimination, solusi SPL kasus 1c ialah $x_1 = free$ $x_2 = 1 - x_6$ $x_3 = free$ $x_4 = -2 - x_6$ $x_5 = 1 + x_6$ $x_6 = free$
	d1	List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 3 Choice of input: 1. Terminal 2. File Input choice (number): 2 ======================== Input Matrix to process the SPL AX = B by turning it into augmented matrix for example: 2x - 3y = 3 x + 2y = 5 Augemented matrix form 2X3(row x colomn) 2 -3 3 1 2 5 ======================== Input file name: testcase1da Choice of output: 1. Terminal 2. File Output choice (number): 1 x1 = 35.995433 x2 = -629.872733 x3 = 3359.152744 x4 = -7557.821735 x5 = 7557.616819 x6 = -2771.067465	Dengan metode Matriks Balikan, solusi SPL kasus 1d ialah $x_1 = 35.995433$ $x_2 = -629.872733$ $x_3 = 3359.152744$ $x_4 = -7557.821735$ $x_5 = 7557.616819$ $x_6 = -2771.067465$

	d2	List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 3 Choice of input: 1. Terminal 2. File Input choice (number): 2 ======================== Input Matrix to process the SPL AX = B by turning it into augmented matrix for example: 2x - 3y = 3 x + 2y = 5 Augemented matrix form 2X3(row x colomn) 2 -3 3 1 2 5 ========================= Input file name: testcase1db Choice of output: 1. Terminal 2. File Output choice (number): 1 x1 = 99.992465 x2 = -4949.530420 x3 = 79192.146526 x4 = -600531.200679 x5 = 2522205.430699 x6 = -6305460.309529 x7 = 9608252.007742 x8 = -8750318.817217 x9 = 4375136.161652 x10 = -923635.554759	Dengan metode Crammer, solusi SPL kasus 1d ialah $x_1 = 99.992465$ $x_2 = -4949.530420$ $x_3 = 79192.146526$ $x_4 = -600531.200679$ $x_5 = 2522205.430699$ $x_6 = -6305460.309529$ $x_7 = 9608252.007742$ $x_8 = -8750318.817217$ $x_9 = 4375136.161652$ $x_{10} = -923635.554759$
2	a	List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 2 Choice of input: 1. Terminal 2. File Input choice (number): 2 Input file name: testcase2a Choice of output: 1. Terminal 2. File Output choice (number): 1 1.000000 0.000000 0.000000 -1.000000 -1.000000 0.000000 1.000000 -2.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 x1 = -1.000000 + x4 x2 = 2.000000x3 x3 = free x4 = free	Dengan metode Gauss-Jordan Elimination, solusi SPL kasus 2a ialah $x_1 = -1.000000 + x_4$ $x_2 = 2.000000x_3$ $x_3 = free$ $x_4 = free$

<table>
<tr>
<td></td>
<td>b</td>
<td>List of methods:
1. Gaussian Elimination
2. Gauss-Jordan Elimination
3. Matrix Inversion
4. Cramer's Rule
5. Back

Input method (number) : 2
Choice of input:
1. Terminal
2. File

Input choice (number): 2
Input file name: testcase2b

Choice of output:
1. Terminal
2. File

Output choice (number): 1

1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 1.000000 0.000000 0.000000 2.000000
0.000000 0.000000 1.000000 0.000000 1.000000
0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000

x1 = 0.0
x2 = 2.000000
x3 = 1.000000
x4 = 1.000000</td>
<td>Dengan metode Gauss-Jordan Elimination, solusi SPL kasus 2b ialah

$x_1 = 0$
$x_2 = 2$
$x_3 = 1$
$x_4 = 1$</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>Input choice (number) : 1
List of methods:
1. Gaussian Elimination
2. Gauss-Jordan Elimination
3. Matrix Inversion
4. Cramer's Rule
5. Back

Input method (number) : 3
Choice of input:
1. Terminal
2. File

Input choice (number): 2
=========================
Input Matrix to process the SPL AX = B by turning it into augmented matrix
for example:
2x - 3y = 3
x + 2y = 5

Augemented matrix form 2X3(row x colomn)
2 -3 3
1 2 5
=========================
Input file name: testcase3a
Choice of output:
1. Terminal
2. File

Output choice (number): 1
x1 = -0.224324
x2 = 0.182432
x3 = 0.709459
x4 = -0.258108</td>
<td>Dengan metode matriks balikan, solusi SPL kasus 3a ialah

$x_1 = -0.224324$
$x_2 = 0.182432$
$x_3 = 0.709459$
$x_4 = -0.258108$</td>
</tr>
</table>

	b	Input choice (number) : 1 List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 1 Choice of input: 1. Terminal 2. File Input choice (number): 2 Input file name: testcase3b Choice of output: 1. Terminal 2. File Output choice (number): 1 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 Sistem persamaan linear tidak memiliki solusi.	Dengan metode eliminasi Gauss-Jordan, kasus 3b tidak memiliki solusi
4		Input choice (number) : 1 List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 2 Choice of input: 1. Terminal 2. File Input choice (number): 1 Input row size: 3 Input column size: 4 Input matrix: 40 -20 0 10 -20 40 -20 0 0 -20 40 10 Choice of output: 1. Terminal 2. File Output choice (number): 1 1.000000 0.000000 0.000000 0.500000 0.000000 1.000000 0.000000 0.500000 0.000000 0.000000 1.000000 0.500000 x1 = 0.500000 x2 = 0.500000 x3 = 0.500000	$x_1 = arus\ loop\ 1$ $x_2 = arus\ loop\ 2$ $x_3 = arus\ loop\ 3$ Diperoleh: $i_1 = x_1 = 0.5\ A$ $i_2 = x_1 - x_2 = 0\ A$ $i_3 = x_2 - x_3 = 0\ A$ $i_4 = x_2 = 0.5\ A$ $i_5 = x_3 = 0.5\ A$ $i_6 = x_2 = 0.5\ A$
5		List of methods: 1. Gaussian Elimination 2. Gauss-Jordan Elimination 3. Matrix Inversion 4. Cramer's Rule 5. Back Input method (number) : 3 Choice of input: 1. Terminal 2. File Input choice (number): 1 ======================== Input Matrix to process the SPL AX = B by turning it into augmented matrix for example: 2x - 3y = 3 x + 2y = 5 Augemented matrix form 2X3(row x colomn) 2 -3 3 1 2 5 ========================= Input row size: 3 Input column size: 4 Input matrix: -120 60 0 -1300 40 -80 0 0 80 20 -150 -200 Choice of output: 1. Terminal 2. File Output choice (number): 1 x1 = 14.444445 x2 = 7.222222 x3 = 10.000000	Dengan menggunakan matriks metode balikan, didapatkan hasil sebagai berikut $x_1 = 14.444445$ $x_2 = 7.222222$ $x_3 = 10$

<table>
<tr>
<td>6</td>
<td>a</td>
<td>Welcome to Matrix Calculator!

List of operations:
1. System of Linear Equations
2. Determinant of Matrix
3. Inverse of Matrix
4. Polynomial Interpolation
5. Multiple Linear Regression
6. Exit

Input choice (number) : 4
Choice of input:
1. Terminal
2. File

Input choice (number): 1
Input number point: 7
Input points:
0.1 0.003
0.3 0.067
0.5 0.148
0.7 0.248
0.9 0.370
1.1 0.518
1.3 0.697
Input value to be estimated: 0.2
Choice of output:
1. Terminal
2. File

Output choice (number): 1
-0.022976 x^0 + 0.239997 x^1 + 0.197413 x^2 + -0.000048 x^3 + 0.026112 x^4 + -0.000050 x^5 + 0.000014
f(0.2) = 0.0329613
Input value to be estimated: 0.55
Choice of output:
1. Terminal
2. File

Output choice (number): 1
-0.022976 x^0 + 0.239997 x^1 + 0.197413 x^2 + -0.000048 x^3 + 0.026112 x^4 + -0.000050 x^5 + 0.000014
f(0.55) = 0.17111908
Input value to be estimated: 0.85
Choice of output:
1. Terminal
2. File

Output choice (number): 1
-0.022976 x^0 + 0.239997 x^1 + 0.197413 x^2 + -0.000048 x^3 + 0.026112 x^4 + -0.000050 x^5 + 0.000014
f(0.85) = 0.33723658
Input value to be estimated: 1.28
Choice of output:
1. Terminal
2. File

Output choice (number): 1
-0.022976 x^0 + 0.239997 x^1 + 0.197413 x^2 + -0.000048 x^3 + 0.026112 x^4 + -0.000050 x^5 + 0.000014
f(1.28) = 0.6775446</td>
<td>Diperoleh:
$f(0.2)$
≈ 0.0329613
$f(0.55)$
≈ 0.17111908
$f(0.85)$
≈ 0.33723658
$f(1.28)$
≈ 0.6775446</td>
</tr>
</table>

b

```
List of operations:
1. System of Linear Equations
2. Determinant of Matrix
3. Inverse of Matrix
4. Polynomial Interpolation
5. Multiple Linear Regression
6. Exit

Input choice (number) : 4
Choice of input:
1. Terminal
2. File

Input choice (number): 2
Input file name: testcase6ba
Choice of output:
1. Terminal
2. File

Output choice (number): 1
7189658216471.242000 x^0 + -9350004058993.094000 x^1 + 5335755560850.102000 x^
2 + -1757276563260.703100 x^3 + 368640760971.040800 x^4 + -51143429292.710880
x^5 + 4696794214.320051 x^6 + -275528781.137900 x^7 + 9374584.419448 x^8 + -14
1018.352722 x^9
f(7.516129) = 53534.55078125
```

Prediksi 1 16 Juli 2021

```
Input value to be estimated: 7.322580
Choice of output:
1. Terminal
2. File

Output choice (number): 1
7189658216471.242000 x^0 + -9350004058993.094000 x^1 + 5335755560850.102000 x^
2 + -1757276563260.703100 x^3 + 368640760971.040800 x^4 + -51143429292.710880
x^5 + 4696794214.320051 x^6 + -275528781.137900 x^7 + 9374584.419448 x^8 + -14
1018.352722 x^9
f(7.32258) = 46499.46875
```

Prediksi 2 10 Agustus 2021

```
Input value to be estimated: 9.1666667
Choice of output:
1. Terminal
2. File

Output choice (number): 1
7189658216471.242000 x^0 + -9350004058993.094000 x^1 + 5335755560850.102000 x^
2 + -1757276563260.703100 x^3 + 368640760971.040800 x^4 + -51143429292.710880
x^5 + 4696794214.320051 x^6 + -275528781.137900 x^7 + 9374584.419448 x^8 + -14
1018.352722 x^9
f(9.1666667) = -664807.6796875
```

Prediksi 3 5 September 2021

```
Input value to be estimated: 10.032258
Choice of output:
1. Terminal
2. File

Output choice (number): 1
7189658216471.242000 x^0 + -9350004058993.094000 x^1 + 5335755560850.102000 x^
2 + -1757276563260.703100 x^3 + 368640760971.040800 x^4 + -51143429292.710880
x^5 + 4696794214.320051 x^6 + -275528781.137900 x^7 + 9374584.419448 x^8 + -14
1018.352722 x^9
f(10.032258) = -2.50827836875E8
```

Prediksi 4 1 Oktober 2021

Dengan menggunakan interpolasi polinom, didapatkan hasil untuk kasus 2b adalah sebagai berikut:

$f(7.516129) \approx 53534.55078125$
$f(7.32258) \approx 46499.46875$
$f(9.1666667) \approx -664807.6796$
$f(10.032258) \approx -2.50827837E8$

<table>
<tr>
<td></td>
<td>c</td>
<td>Welcome to Matrix Calculator!

List of operations:
1. System of Linear Equations
2. Determinant of Matrix
3. Inverse of Matrix
4. Polynomial Interpolation
5. Multiple Linear Regression
6. Exit

Input choice (number) : 4
Choice of input:
1. Terminal
2. File

Input choice (number): 1
Input number point: 6
Input points:
0 0
0.4 0.4188842
0.8 0.5071580
1.2 0.5609247
1.6 0.5836857
2.0 0.5766515
Input value to be estimated: 1
Choice of output:
1. Terminal
2. File

Output choice (number): 1
0.0x^0 + 2.035256 x^1 + -3.552673 x^2 + 3.237099 x^3 + -1.421257 x^4 + 0.236255 x^5
f(1.0) = 0.5346797</td>
<td>Diperoleh:
$f(x) \approx 0.236255x^5 - 1.421257x^4 + 3.237099x^3 - 3.552673x^2 + 2.035256x$</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Welcome to Matrix Calculator!

List of operations:
1. System of Linear Equations
2. Determinant of Matrix
3. Inverse of Matrix
4. Polynomial Interpolation
5. Multiple Linear Regression
6. Exit

Input choice (number) : 5
Choice of input:
1. Terminal
2. File

Input choice (number): 1
Input the total number of variables (x) and constants (y): 4
Input the number of data sample: 20
Input values:
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.8 29.03 0.82
54.9 70.9 29.37 0.95
Input values to be estimated:
50.0 76.0 29.30
Choice of output:
1. Terminal
2. File

Output choice (number): 1
Estimation result = 0.938473</td>
<td>Estimasi nilai NO ketika $Humid = 50$, $Temp = 76$, $P = 29.30$ adalah 0.938473</td>
</tr>
</table>

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Program ini dapat menerima input dari user untuk kalkulasi persoalan-persoalan seperti di atas dan mengeluarkan outputnya berupa pilihan file atau di terminal. Secara garis besar, program ini meliputi beberapa fitur seperti mencari solusi dari suatu persamaan linear menggunakan metode eliminasi Gauss-Jordan, Gauss, matriks balikan, dan crammer. Selain itu, program dapat mencari determinan suatu matriks menggunakan metode reduksi baris dan ekspansi kofaktor. Kemudian dapat mencari balikan suatu matrix menggunakan metode eliminasi Gauss-Jordan dan adjoint, mencari nilai suatu fungsi $y = f(x)$ yang tidak diketahui perumusannya secara jelas dengan pendekatan fungsi polinom pangkat n-1 untuk mendapatkan x menggunakan interpolasi polinom, dan mencari regresi linear berganda dari suatu fungsi.

Program diimplementasikan dalam bahasa Java dan sebagian besar interaksi dengan program hanya dapat dilakukan melalui terminal. Untuk memilih pilihan fitur yang tersedia, list fitur ditampilkan di layar pengguna kemudian pengguna memilih fitur yang ingin digunakan. Namun, untuk memasukkan data berupa matriks ke dalam program untuk diproses, pengguna mendapatkan pilihan lain yaitu dengan melalui file teks yang berisi matriks. Selain itu, untuk mendapatkan hasil perhitungan pengguna juga diminta untuk memilih metode luaran yang diinginkan. Terdapat dua metode untuk mendapatkan hasil perhitungan, yaitu dengan menyimpan hasil ke dalam file teks, atau dengan menampilkan hasil perhitungan pada terminal.

5.2. Refleksi dan Saran

Melalui pengerjaan tugas ini, kami mempelajari bahasa pemrograman yang sebelumnya tidak terlalu familier bagi kami, yaitu bahasa Java. Selain itu, kami juga belajar untuk mengimplementasikan serta memanfaatkan pengetahuan yang sudah kami miliki mengenai matriks dan sifat-sifatnya. Contoh studi kasus yang diberikan beserta dengan informasi dan petunjuk pengerjaan tugas ini cukup banyak dan beragam, sehingga membantu kami ketika menjalankan tes pada program. Beberapa contoh kasus juga memperjelas kegunaan nyata dari fitur-fitur yang telah dibuat dalam program.

Program ini masih dapat dikembangkan lagi, terutama pada antarmukanya. Pada program yang dibuat, interaksi pengguna dengan kalkulator dilakukan berbasis teks melalui terminal. Agar program menjadi lebih interaktif, dapat dibuat antarmuka grafis bagi pengguna. Hal ini diharapkan dapat lebih menarik perhatian dan memudahkan pengguna ketika menggunakan kalkulator.

REFERENSI

Anton, H., Rorres, C. and Kaul, A., 2019. *Elementary Linear Algebra : Applications Version*. Hoboken, New Jersey: Wiley.

Walpole, R.E., Myers, R.H., Myers, S.L. and Ye, K., 2016. *Probability & Statistics for Engineers & Scientists*. Essex, England: Pearson Education Limited.