LAPORAN TUGAS BESAR

Sistem Persamaan Linier, Determinan, dan Aplikasinya

Ditujukan untuk memenuhi salah satu tugas besar mata kuliah IF2123 Aljabar Linier dan Geometri pada Semester I Tahun Akademik 2021/2022

Disusun oleh:

Sarah Azka Arief (K2)	**13520083**
Nelsen Putra (K3)	**13520130**
Rania Dwi Fadhilah (K3)	**13520142**

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2021

DAFTAR ISI

BAB I

DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Kelompok 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 untuk 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, *library* tersebut digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

Spesifikasi program yang harus kami buat adalah sebagai berikut.

1. Program dapat menerima masukan (input) baik dari *keyboard* maupun membaca masukan dari *file text*. Untuk SPL, masukan dari *keyboard* adalah m, n, koefisien a_{ij}, dan b_i. Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

```
3 4.5 2.8 10 12
-3 7 8.3 11 −4
0.5 -10 -9 12 0
```

2. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari *keyboard* adalah *n* dan koefisien a_{ij}. Masukan dari *file* berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

```
3 4.5 2.8 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 II

TEORI SINGKAT

1. Metode Eliminasi Gauss & Gauss Jordan

SPL dapat diselesaikan melalui banyak metode, 2 diantaranya adalah eliminasi Gauss dan Gauss Jordan. Eliminasi Gauss ini mulai populer setelah digunakan oleh Carl Friedrich Gauss. Sedangkan, eliminasi Gauss Jordan dikembangkan dan dipopulerkan oleh Willhelm Jordan melalui bukunya. Kedua metode ini memanfaatkan operasi baris elementer (OBE). OBE yang dapat dilakukan pada matriks augmented, antara lain:

1. Mengalikan sebuah baris dengan konstanta tidak nol.
2. Menukar dua buah baris.
3. Menambahkan sebuah baris dengan kelipatan baris lainnya.

Dengan menerapkan OBE pada matriks *augmented*, dapat dihasilkan sebuah matriks eselon baris maupun matriks eselon baris tereduksi. Apabila hasil akhirnya adalah matriks eselon baris, maka metode eliminasi yang digunakan adalah metode eliminasi Gauss, dengan contoh hasil sebagai berikut :

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

Pada matriks yang telah menjadi eselon baris tersebut, kita dapat mencari nilai dari tiap variabel menggunakan teknik penyulihan mundur. Contoh penyelesaian kasusnya adalah sebagai berikut :

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

diperoleh persamaan-persamaan linier sbb:

$x_1 + 3/2x_2 - 1/2x_3 = 5/2$ (i)
$x_2 + 1/2x_3 = 7/2$ (ii)
$x_3 = 3$ (iii)

Selesaikan dengan teknik penyulihan mundur sbb:

(iii) $x_3 = 3$
(ii) $x_2 + 1/2x_3 = 7/2 \rightarrow x_2 = 7/2 - 1/2(3) = 2$
(i) $x_1 + 3/2x_2 - 1/2x_3 = 5/2 \rightarrow x_1 = 5/2 - 3/2\ (2) - 1/2\ (3) = 1$

Solusi: $x_1 = 1,\ x_2 = 2,\ x_3 = 3$

Apabila hasil akhir dari OBE pada matriks augmented adalah eselon baris tereduksi, maka metode eliminasi yang digunakan adalah metode eliminasi Gauss Jordan. Pada metode ini, eliminasi terdiri dari dua fase, yaitu fase maju (eliminasi Gauss) untuk menghasilkan nilai-nilai 0 di bawah 1 utama dan fase mundur untuk menghasilkan nilai-nilai 0 di atas satu utama. Contoh hasil eliminasi Gauss Jordan adalah sebagai berikut :

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

Pada eliminasi Gauss Jordan, tidak perlu dilakukan teknik penyulihan mundur untuk memperoleh nilai dari tiap variabel karena bisa langsung didapatkan dari matriks *augmented* akhir. Contoh penyelesaian kasusnya adalah sebagai berikut.

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

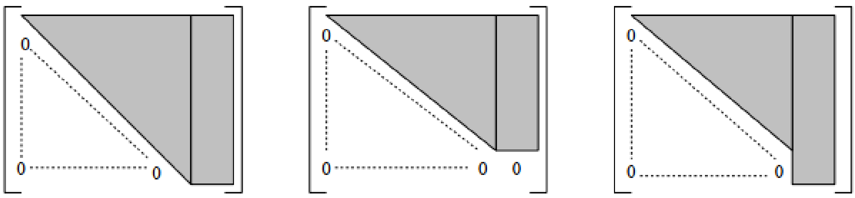
Diperoleh persamaan-persamaan berikut:

$x_1 - x_3 = 0 \rightarrow x_1 = x_3$

$x_2 + x_3 = 0 \rightarrow x_2 = -x_3$

$x_4 = 0$

Terdapat tiga kemungkinan solusi yang bisa didapatkan dari sistem persamaan linear, yaitu solusi unik, solusi banyak, dan tidak ada solusi. Untuk menentukan tipe solusi dari sebuah SPL, kita dapat melihat hasil akhir matriks *augmented* setelah proses eliminasi.



Solusi unik Solusi banyak Tidak ada solusi

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 3 & -2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

2. Determinan

Determinan adalah nilai skalar yang dapat diperoleh dari sebuah matriks N x N (matriks persegi). Determinan dari sebuah matriks dapat digunakan untuk menentukan balikan/invers dari suatu matriks, yang nantinya nilai tersebut berguna untuk pencarian solusi dari sistem persamaan linier. Contoh determinan adalah sebagai berikut.

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

Terdapat berbagai cara yang dapat dilakukan untuk mencari nilai determinan dari sebuah matriks, antara lain dengan menggunakan metode reduksi baris dan menggunakan metode ekspansi kofaktor. Mencari nilai determinan dengan menggunakan metode reduksi baris diimplementasikan dengan menerapkan operasi baris elementer (OBE) pada matriks terkait hingga terbentuk matriks segitiga atas atau matriks segitiga bawah. Apabila matriks telah berbentuk segitiga atas atau matriks segitiga bawah, determinan dapat dicari dengan mengalikan semua elemen yang berada di diagonal utamanya.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\det(A) = a_{11}a_{22}\,a_{33}a_{44}$$

Terdapat beberapa aturan yang digunakan untuk mencari nilai determinan yang diambil dari manipulasi matriks. Sebagai contoh, matriks B = manipulasi matriks A, maka determinan B dapat dicari dengan memanipulasi nilai determinan A dengan syarat sebagai berikut.

- $A \xrightarrow{\text{Kalikan sebuah baris dengan k}} B$, maka det(B) = k det(A)
- $A \xrightarrow{\text{Pertukarkan dua baris}} B$, maka det(B) = –det(A)
- $A \xrightarrow[\text{dengan k kali baris yang lain}]{\text{Sebuah baris ditambahkan}} B$, maka det(B) = det(A)

Untuk metode dengan ekspansi kofaktor, determinan ditentukan dengan penjumlahan dari perkalian elemen-elemen yang bersesuaian pada suatu baris/kolom tertentu dengan kofaktor yang bersesuaian dengan elemen-elemen tersebut. Selain itu, terdapat beberapa teorema yang perlu diperhatikan tentang determinan matriks, antara lain:

1. Jika A mengandung sebuah baris / kolom nol, maka det(A) = 0
2. Jika A^T = matriks transpose A, maka $\det(A^T) = \det(A)$
3. Jika A = BC, maka det(A) = det(B)det(C)
4. Sebuah matriks hanya mempunya balikan jika & hanya jika det A $\neq 0$
5. $\det(A^{-1}) = \frac{1}{\det(A)}$

3. Matriks Kofaktor dan Adjoin

Matriks kofaktor merupakan matriks yang mengandung nilai kofaktor dari tiap elemen yang terdapat pada matriks. Nilai kofaktor sendiri adalah nilai yang didapat dengan mengabaikan kolom dan baris yang mengandung elemen yang ditinjau. Misalkan A adalah matriks N x N dan C_{ij} adalah kofaktor entri a_{ij}, maka matriks kofaktor dari A adalah sebagai berikut :

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

Untuk mencari nilai C_{ij}, digunakan rumus $C_{ij} = (-1)^{i+j}M_{ij}$ dengan M_{ij} merupakan matriks minor, yaitu determinan dari elemen matriks selain elemen yang berada di baris dan kolom yang sama dengan elemen yang ditinjau. Misalnya, minor elemen pada baris ke-i dan kolom ke-j merupakan hasil penghitungan determinan submatriks dengan mengabaikan baris ke-i dan kolom ke-j pada matriks awal. Contoh nilai matriks minor adalah sebagai berikut :

$$M_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$$

Matriks kofaktor ini nantinya dapat digunakan untuk mencari nilai determinan dari sebuah matriks. Caranya adalah dengan menjumlahkan perkalian elemen matriks dengan elemen kofaktornya yang bersesuaian dalam suatu baris atau kolom tertentu. Oleh karena itu, ketika menghitung minor untuk mencari kofaktor dari suatu matriks, penghitungan kofaktor ini akan bersifat rekursif jika kita menggunakan metode kofaktor pula pada proses penghitungan determinan dari submatriksnya. Sifat rekursif ini memiliki basis ketika matriks/submatriks-nya berisi satu elemen atau berukuran 1 x 1. Determinan dari matriks/submatriks berukuran 1 x 1 tersebut adalah elemen yang berada di dalamnya. Contoh dari matriks kofaktor adalah sebagai berikut.

$$\text{matriks kofaktor: } \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix}$$

4. **Matriks Adjoin**

Matriks adjoin merupakan matriks yang dihasilkan dari matriks kofaktor yang telah ditranspose. Matriks ini dapat digunakan untuk mencari invers dari sebuah matriks. Caranya adalah dengan mengalikan matriks adjoin dengan 1/determinan matriks asli. Determinannya bisa dicari dengan menggunakan metode reduksi baris ataupun metode ekspansi kofaktor. Matriks adjoin yang merupakan transpose dari matriks kofaktor, artinya terdapat penukaran elemen C_{ij} pada matriks kofaktor menjadi elemen C_{ji} pada matriks adjoinnya. Sebagai contoh, berikut ini merupakan adjoin matriks yang didapat dari transpose matriks kofaktor di sampingnya.

$$\text{matriks kofaktor: } \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix} \quad \text{adj}(A) = \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{bmatrix}$$

5. **Matriks Balikan**

Matriks balikan (*inverse)* dari matriks A adalah A^{-1} yang apabila dikalikan kembali dengan matriks aslinya akan menghasilkan matriks identitas. (A^{-1} A = I). Matriks hanya memiliki *inverse* apabila berbentuk persegi dan merupakan matriks non-*singular*, yaitu matriks yang memiliki nilai determinan bukan 0. Matriks balikan dapat dicari dengan memanfaatkan determinan dan adjoin melalui rumus berikut :

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

Selain itu, matriks *inverse* ini juga bisa dihitung dengan menggunakan eliminasi Gauss-Jordan. Pada matriks A dengan ukuran N x N, matriks balikannya dapat dicari dengan cara meng-augmentasi-kan matriks A dengan matriks identitasnya, lalu dilakukan proses OBE dengan metode eliminasi Gauss-Jordan pada matriks *augmented* tersebut sehingga bagian kiri matriks *augmented* akan berbentuk matriks identitas. Proses tersebut secara sederhana dirumuskan pada gambar di bawah ini.

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

Berdasarkan karakteristik dari matriks balikan yang telah dipaparkan di atas, bahwa matriks balikan dari suatu matriks apabila dikalikan dengan matriks itu sendiri akan menghasilkan sebuah matriks identitas, kita dapat mencari solusi SPL pada Ax = B dengan mengalikan invers A dengan kedua ruas dari persamaan linier tersebut. Dengan langkah tersebut, akan didapatkan $A^{-1}Ax = A^{-1}B \Leftrightarrow Ix = A^{-1}B \Leftrightarrow x = A^{-1}B$.

6. **Kaidah Cramer**

Pada aljabar linear, Kaidah Cramer adalah suatu formula untuk solusi dari suatu sistem persamaan linear. Cara ini valid apabila sistem memiliki solusi yang unik. Solusi diekspresikan dalam bentuk determinan dari koefisien matriks dan matriks-matriks yang didapatkan dari penukaran terhadap setiap satu kolom dengan vektor kolom yang berada pada sisi kanan dari persamaan.

Suatu sistem dengan n jumlah persamaan linear direpresentasikan dalam matriks bentuk perkalian sebagai berikut:

$$A\mathbf{x} = \mathbf{b}$$

Pada persamaan di atas, matrix A (n x n) memiliki determinan yang bukan nol dan vector $x = (x1, \ldots, xn)^T$ adalah vektor kolom dari variabel-variabel. Maka, berdasarkan kaidah Cramer, pada kasus ini sistem memiliki solusi unik yang nilainya didapatkan melalui:

$$x_i = \frac{\det(A_i)}{\det(A)} \qquad i = 1, \ldots, n$$

dengan Ai adalah matriks yang didapat dengan mengubah kolom ke-i pada matriks A dengan vektor kolom b.

7. **Interpolasi Polinom**

Interpolasi polinom adalah interpolasi dari kumpulan data yang diberikan oleh polinomial dengan derajat serendah mungkin yang melewati titik-titik kumpulan data. Jika terdapat suatu interpolasi polinom dalam bentuk

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0.$$

maka kalimat p menginterpolasi titik-titik kumpulan data menandakan bahwa

$$p(x_i) = y_i \qquad \text{for all } i \in \{0, 1, \ldots, n\}.$$

sehingga jika kita mensubstitusi persamaan p(x), kita akan mendapatkan suatu sistem berisi persamaan-persamaan linear dalam koefisien a_k . Apabila sistem diubah dalam bentuk matriks-vektor maka akan didapat:

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & \dots & x_1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & x_n^{n-2} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Untuk mendapatkan solusi dari sistem tersebut, dapat digunakan penyelesaian dengan metode eliminasi Gauss yang akan menghasilkan a_0 , …, a_n .

8. Regresi Linier Berganda

Regresi Linear adalah suatu metode yang kerap digunakan untuk menaksir suatu nilai. Rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda adalah

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

Normal Estimation Equation for Multiple Linear Regression dapat digunakan untuk mendapatkan nilai dari setiap dari βi sehingga membentuk sistem persamaan linear seperti berikut:

$$\begin{array}{lllllll} n\hat{\beta}_0 & +\hat{\beta}_1 \sum_{i=1}^{n} x_{i1} & +\hat{\beta}_2 \sum_{i=1}^{n} x_{i2} & +\cdots & +\hat{\beta}_k \sum_{i=1}^{n} x_{ik} & = & \sum_{i=1}^{n} y_i \\ \hat{\beta}_0 \sum_{i=1}^{n} x_{i1} & +\hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 & +\hat{\beta}_2 \sum_{i=1}^{n} x_{i1}x_{i2} & +\cdots & +\hat{\beta}_k \sum_{i=1}^{n} x_{i1}x_{ik} & = & \sum_{i=1}^{n} x_{i1}y_i \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ \hat{\beta}_0 \sum_{i=1}^{n} x_{ik} & +\hat{\beta}_1 \sum_{i=1}^{n} x_{ik}x_{i1} & +\hat{\beta}_2 \sum_{i=1}^{n} x_{ik}x_{i2} & +\cdots & +\hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 & = & \sum_{i=1}^{n} x_{ik}y_i \end{array}$$

Metode eliminasi Gauss dapat kemudian digunakan untuk menyelesaikan sistem persamaan linear di atas.

BAB III

IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM JAVA

Dalam Tugas Besar 1, src folder kami berisikan sebuah file Java yaitu Main.java dan sebuah folder bernama 'Functions'. Classpath dikonfigurasi agar input berada pada folder src dan output pada folder bin sehingga semua file pada folder Functions yang merupakan bagian dari package Functions diawali dengan kalimat 'package Functions'. Folder 'Functions' ini memuat 9 file Java, di antaranya Matrix.java, Cramer.java, Invers.java, Interpolasi.java, Func.java, Menu.java, Gaussian.java, Regresi.java, dan Determinant.java Berikut ini merupakan rincian nama serta deskripsi (atribut dan metode) dari setiap file tersebut :

1. Class Matrix (file name: Matrix.java)

1.1 Atribut

`contents`	array 2D bertipe double yang digunakan untuk menyimpan elemen dari Matrix
`rows`	integer yang menyimpan jumlah baris dalam Matrix
`cols`	integer yang menyimpan jumlah kolom dalam Matrix

1. 2 Method

`Matrix (int nRow, int nCol)`
`{ I.S. Jumlah kolom & baris matriks sudah terdefinisi }`
`{ F.S. Mengembalikan jumlah kolom dan baris matriks & semua elemen matriks diisi oleh 0 }`
	Konstruktor pada class Matrix menerima dua buah integer (nRow dan nCol) yang akan dijadikan jumlah rows dan cols dari Matrix tersebut dan mengisi semua elemen pada Matrix tersebut dengan angka 0.

2. Class Cramer (file name: Cramer.java)

2.1 Method

`String[] solveSPL (Matrix M)`
`{ I.S. Matrix terisi & tidak kosong }`
`{ F.S. Mengembalikan solusi dari sebuah SPL dengan memanfaatkan kaidah Cramer }`
	Menerima input berupa matrix dan mengembalikan array bertipe string yang diisi dengan hasil dari penyelesaian persamaan melalui kaidah cramer dengan menggunakan metode rowRed dari class Determinant. Apabila terdapat kasus penekanan tombol cancel atau input tidak valid mengembalikan array of String yang kosong (null).

3. Class Invers (file name: Invers.java)

3.1 Method

`String[] solveSPL (Matrix M)`
`{ I.S. Matrix terisi & tidak kosong }`
`{ F.S. Mengembalikan solusi dari sebuah SPL }`
	Menyelesaikan sebuah SPL dengan menggunakan metode invers dengan memanfaatkan metode gaussJordan. Apabila terdapat kasus penekanan tombol cancel atau input tidak valid, mengembalikan array of String yang kosong (null).

`Matrix adjoint (Matrix M)`
`{ I.S. Matrix terisi, tidak kosong, dan berbentuk persegi }`
`{ F.S. Mengembalikan inverse dari matriks dengan metode kofaktor }`
	Menerima matriks dan mengembalikan matriks berupa inversenya dengan memanfaatkan metode kofaktor. Apabila terdapat kasus penekanan tombol cancel atau input tidak valid, mengembalikan matriks dengan jumlah row lebih besar 1.

`Matrix gaussJordan (Matrix M)`
`{ I.S. Matrix terisi, tidak kosong, dan berbentuk persegi }`
`{ F.S. Mengembalikan inverse dari matriks dengan metode Gauss-Jordan }`
	Menerima matriks dan mengembalikan matriks berupa inversenya dengan memanfaatkan metode Gauss-Jordan. Apabila terdapat kasus penekanan tombol cancel atau input tidak valid, mengembalikan matriks dengan jumlah row lebih besar 1.

4. Class Interpolasi (file name: Interpolasi.java)

4.1 Method

`String[] solveInterpolasi(Matrix m, double x)`
`{ I.S. Matrix sudah terisi & x sudah terdefinisi }`
`{ F.S. Hasil interpolasi dari matriks dengan x }`
	Menerima matriks dengan jumlah baris n+1 dan double selaku titik x yang akan ditaksir kemudian mencari persamaan polinom n dari matriks tersebut serta mencari taksiran dengan memanfaatkan persamaan tersebut dan juga eliminasi Gauss dan mengembalikan array of String berisi hasilnya. Jika menekan tombol cancel/*input* tidak valid, mengembalikan array of String yang kosong (null).

5. Class Func (file name: Func.java)

5.1 Method

`Matrix inputMatrix (int inputType)`
`{ I.S. Tipe matriks sudah terdefinisi }`
`{ F.S. Menerima input matriks yang sesuai dengan tipe yang diminta oleh` *user* `}`
	Menerima integer sebagai tipe operasi kalkulator (determinan, invers, penyelesaian SPL, regresi, dan interpolasi) yang digunakan untuk menentukan alur serta ketentuan input yang sesuai untuk matriks setiap operasi dan mengembalikan sebuah matrix dengan ukuran baris, ukuran kolom, dan isi elemen yang sesuai dengan input user. Apabila user menekan cancel atau memasukkan input yang tidak valid maka akan mengembalikan matrix kosong dengan ukuran row dan column berupa 0 0.

`Matrix readMatrix ()`
`{ I.S. File .txt sudah tersedia }`
`{ F.S. Menyimpan matriks yang disimpan pada sebuah file }`
	Meminta input nama file dari user, membaca file .txt dengan nama sesuai input user, dan menyimpan isi file tersebut dalam sebuah matrix yang dikembalikan melalui fungsi ini. Apabila file dengan nama sesuai input user tidak ditemukan, maka akan keluar error panel yang mengatakan bahwa file tidak bisa dibaca dan user akan kembali ke menu utama. Apabila user menekan cancel maka akan mengembalikan matriks kosong dengan jumlah row 0 dan jumlah column 0.

`void writeMatrix (Matrix m)`
`{ I.S. Matriks sudah terdefinisi }`
`{ F.S. Menyimpan matriks dalam sebuah file .txt dengan nama sesuai input` *filename* `dari` *user* `}`
	Menerima matrix/string dan meminta input berupa filename dari user. Matrix/string akan ditulis dan disimpan dalam file txt dengan nama sesuai input filename dari user. Akan keluar panel yang menjelaskan apakah save berhasil atau tidak (error

	panel). Apabila user menekan tombol cancel, maka akan mengembalikan matriks kosong dengan row 0 dan column 0.

`void writeMatrix (String m)`
`{ I.S. Matriks sudah terdefinisi }`
`{ F.S. Menyimpan matriks dalam sebuah file .txt dengan nama sesuai input *filename* dari *user* }`
	Menuliskan matrix sebagai string.

`void displayMatrix (Matrix m)`
`{ I.S. Matriks terdefinisi & tidak kosong }`
`{ F.S. Matriks ditampilkan pada layar }`
	Menerima matrix dan menuliskan setiap elemennya pada panel.

`double getElmt (Matrix m, int i, int j)`
`{ I.S. i,j, dan matriks terdefinisi & tidak kosong }`
`{ F.S. Mengambil elemen dari matriks yang berada pada baris i dan kolom j }`
	Getter yang menerima matrix dan dua integer (baris dan kolom) untuk mengakses elemen dari matrix tersebut yang terletak pada baris dan kolom tersebut dan mengembalikan elemen tersebut.

`void setElmt (Matrix m, int i, int j, double fill)`
`{ I.S. i,j, fill, dan juga matriks terdefinisi & tidak kosong }`
`{ F.S. Mengisi elemen dari matriks yang berada pada baris i dan kolom j dengan fill }`
	Setter yang menerima matrix, dua integer (baris dan kolom), dan satu double (elemen baru). Elemen dari matrix pada baris dan kolom tersebut diisi elemen baru.

`int getLastIdxRow (Matrix m)`
`{ I.S. Matriks terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan indeks terakhir baris matriks }`
	Menerima matrix dan mengembalikan indeks terakhir baris matrix.

`int getLastIdxCol (Matrix m)`
`{ I.S. Matriks terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan indeks terakhir kolom matriks }`
	Menerima matrix dan mengembalikan indeks terakhir kolom matrix.

`void multiplyOBE (Matrix m, int row, double multiplier)`
`{ I.S. Matriks, row, dan multiplier sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks yang semua elemen baris rownya sudah dikali dengan multiplier }`
	Menerima matrix, integer (baris), dan double (multiplier) dan melakukan perkalian terhadap baris pada matrix tersebut dengan multiplier

`void divideOBE (Matrix m, int row, double divider)`
`{ I.S. Matriks, row, dan divider sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks yang semua elemen baris rownya sudah dibagi dengan divider }`
	Menerima matrix, integer (baris), dan double (divider) dan melakukan pembagian terhadap baris pada matrix tersebut dengan divider.

`void switchOBE (Matrix m, int row1, int row2)`
`{ I.S. Matriks, row1, dan row2 sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks yang semua elemen baris row1-nya sudah ditukar dengan baris row2 }`
	Menerima matrix dan dua integer (baris satu dan baris dua) kemudian menukar elemen pada baris satu dengan elemen pada baris dua.

`void addOBE (Matrix m, int row1, int row2, double multiplier)`
`{ I.S. Matriks, row1, dan row2, dan multiplier sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks yang semua elemen baris row1-nya sudah dijumlahkan dengan baris row2 yang sudah dikalikan dengan multiplier }`
	Menerima matrix, dua integer (baris satu dan baris dua), serta double (multiplier) kemudian menambah baris satu dengan hasil perkalian baris dua dengan multiplier.

`boolean isSizeEqual (Matrix m1, Matrix m2)`
`{ I.S. Kedua matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan true apabila ukuran matriks m1 sama dengan matriks m2 }`
	Mencari tahu apakah ukuran matriks m1 sama dengan matriks m2 atau bukan.

`boolean isEqual (Matrix m1, Matrix m2)`
`{ I.S. Kedua matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan true apabila ukuran matriks dan semua elemen matriks m1 sama dengan matriks m2 }`
	Mencari tahu apakah sebuah matriks m2 merupakan matriks yang sama persis dengan matriks m1 atau buksn.

`boolean isSquare (Matrix m)`
`{ I.S. Matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan true apabila matriks berbentuk persegi (jumlah baris = jumlah kolom) }`
	Mencari tahu apakah sebuah matriks berbentuk persegi atau bukan.

`Matrix copyMatrix (Matrix m)`
`{ I.S. Matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan salinan matriks m }`
	Membuat salinan dari sebuah matriks.

`Matrix add (Matrix m1, Matrix m2)`
`{ I.S. Kedua matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Melakukan operasi penjumlahan terhadap semua elemen matriks m1 dengan matriks m2 }`
	Melakukan operasi penjumlahan pada semua elemen dari m1 dengan m2 yang berada pada posisi yang sama.

`Matrix subtract (Matrix m1, Matrix m2)`
`{ I.S. Kedua matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Melakukan operasi pengurangan terhadap semua elemen matriks m1 dengan matriks m2 }`
	Melakukan operasi pengurangan pada semua elemen dari m1 dengan m2 yang berada pada posisi yang sama.

`Matrix multiply (Matrix m1, Matrix m2)`
`{ I.S. Kedua matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Melakukan operasi perkalian terhadap semua elemen matriks m1 dengan matriks m2 }`
	Melakukan operasi perkalian pada semua elemen dari m1 dengan m2 yang berada pada posisi yang sama.

`int nbElmt (Matrix m)`
`{ I.S. Matriks sudah terisi & tidak kosong }`
`{ F.S. Mengembalikan jumlah elemen pada matriks }`
	Mencari jumlah elemen yang terdapat pada sebuah matriks.

`Matrix makeMinor (Matrix m, int rowAcuan, int colAcuan)`
`{ I.S. Matriks, rowAcuan, dan colAcuan sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks minor }`
	Menerima matriks, int baris acuan dan kolom acuan kemudian mengembalikan matriks minornya.

`int makeSgtgAtas (Matrix m)`
`{ I.S. Matriks sudah terisi & tidak kosong }`
`{ F.S. Mengubah matriks menjadi matriks segitiga atas dan mengembalikan banyaknya proses pertukaran baris }`
	Menerima matriks dan mengubahnya menjadi matriks segitiga atas kemudian mengembalikan jumlah proses pertukaran baris yang dilakukan.

`Matrix transpose (Matrix m)`
`{ I.S. Matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks transpose }`
	Menerima matriks dan mengembalikan matriks transposenya.

`Matrix makeCofactor (Matrix m)`
`{ I.S. Matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks kofaktor }`
	Menerima matriks dan mengembalikan matriks kofaktornya.

`Matrix ZeroNRound (Matrix m)`
`{ I.S. Matriks sudah terdefinisi & tidak kosong }`
`{ F.S. Mengembalikan matriks yang semua angkanya sudah dibulatkan 4 angka di belakang koma dan tidak ada angka (-0) }`
	Melakukan operasi pembulatan (4 angka di belakang koma) dan mengubah angka (-0) menjadi 0.

`int findLeadingOne (Matrix m, int row)`
`{ I.S. Matriks dan row sudah terisi & tidak kosong }`
`{ F.S. Mengembalikan index dari angka 1 pertama pada baris}`
	Mencari index dari angka 1 pertama pada sebuah baris. Apabila tidak ditemukan, maka index = -1.

6. Class Menu (file name: Menu.java)

Class yang berisi fungsi public static void bernama user yang berfungsi sebagai *menu* sederhana dari kalkulator sebagai interface user.

Alur dari menu dimulai pada main menu yang menunjukkan pilihan 1 sampai 6. Apabila user memberi input 6 atau menekan tombol cancel, program akan berhenti (system exit). Apabila user memberi input 1 hingga 5, user akan diarahkan ke menu berikutnya sesuai dengan input pada main menu. User akan diminta untuk memasukkan matriks baik dari file maupun input manual.

Matriks hasil input user akan diolah sesuai operasi yang dipilih oleh user dan hasilnya akan ditampilkan pada panel. Pada operasi determinan dan inverse, output disimpan dalam bentuk akhir berupa matriks sementara output dari operasi lainnya disimpan dalam bentuk array of String, Setelah hasil operasi ditampilkan pada terminal, user diberi pilihan untuk menyimpan hasil tersebut dalam file txt dengan nama file sesuai input user atau tidak.

Pada fungsi user, terdapat tiga parameter yakni resultDet berupa double yang digunakan untuk menyimpan hasil determinan, mnew berupa Matrix yang digunakan untuk menyimpan hasil inverse, dan resultApprox berupa array of string yang digunakan untuk menyimpan hasil operasi penyelesaian SPL, regresi, dan interpolasi. Parameter digunakan sebagai penanda output yang tidak valid sekaligus sebagai pemicu return.

7. Class Gaussian (file name: Gaussian.java)

7.1 Method

`Matrix Gauss (Matrix M)`
`{ I.S. Matrix sudah terisi & tidak kosong }`
`{ F.S. Matrix diubah menjadi matriks eselon baris }`
	Mengubah matriks menjadi matriks eselon baris.

`Matrix GaussJordan (Matrix M)`
`{ I.S. Matrix sudah terisi & tidak kosong }`
`{ F.S. Matrix diubah menjadi matriks eselon baris tereduksi }`
	Mengubah matriks menjadi matriks eselon baris tereduksi.

`double[] UniqueSPL (Matrix M)`
`{ I.S. Matrix sudah terisi & tidak kosong }`
`{ F.S. Solusi dari persamaan SPL yang dapat diperoleh dari matriks }`
	Mencari solusi (x1, x2, x3, …) dari sebuah matriks.

`char getLetter (int i)`
`{ I.S. i terdefinisi }`
`{ F.S. Mengembalikan sebuah karakter }`
	Digunakan untuk menyelesaikan persamaan parametrik yang membutuhkan variabel.

`boolean IdxValid (double[] checkIdx, int idx)`
`{ I.S. idx terdefinisi dan checkIdx sudah terisi }`
`{ F.S. Mengembalikan` *`true`* `apabila idx bukan merupakan bagian dari double array }`
	Mengembalikan valid apabila idx tidak terdapat dalam sebuah array double.

`boolean StrValid (String[] checkStr, int idx)`
`{ I.S. idx terdefinisi dan checkStr sudah terisi }`
`{ F.S. Mengembalikan` *`true`* `elemen checkStr dengan index idx adalah "" }`
	Mengembalikan valid apabila array string dengan index idx belum terisi oleh kata apapun.

`String[] SolveSPL (Matrix M, String SPLtype)`
`{ I.S. Matrix terisi dan SPLtype diasumsikan selalu valid (Gauss/Gauss Jordan) }`
`{ F.S. Solusi SPL dari matriks }`
	Menyelesaikan sebuah SPL dalam bentuk matriks eselon baris/eselon baris tereduksi sesuai tipe solusinya (solusi unik, solusi banyak, atau tidak memiliki solusi).

8. Class Regresi (file name: Regresi.java)

8.1 Method

`String[] SolveRegresi (Matrix m)`
`{ I.S. Matrix sudah terisi & tidak kosong }`
`{ F.S. Hasil regresi dari matrix }`
	Mencari hasil regresi dari sebuah matriks dengan memanfaatkan penyelesaian SPL metode Gauss.

9. Class Determinant (file name: Determinant.java)

9.1 Method

`double cofExp (Matrix M)`
`{ I.S. Matrix terisi & tidak kosong }`
`{ F.S. Mengembalikan determinan matriks }`
	Metode menghitung determinan matriks dengan ekspansi kofaktor. Apabila terdapat kasus penekanan tombol cancel atau input tidak valid, mengembalikan NaN double.

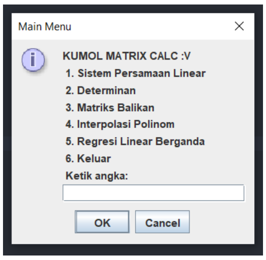
`double rowRed (Matrix M)`
`{ I.S. Matrix terisi & tidak kosong }`
`{ F.S. Mengembalikan determinan matriks }`
	Metode menghitung determinan matriks dengan reduksi baris pada matriks. Apabila terdapat kasus penekanan tombol cancel atau input tidak valid, mengembalikan NaN double.

BAB IV

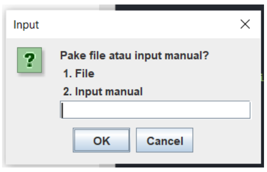
EKSPERIMEN

1. Interface

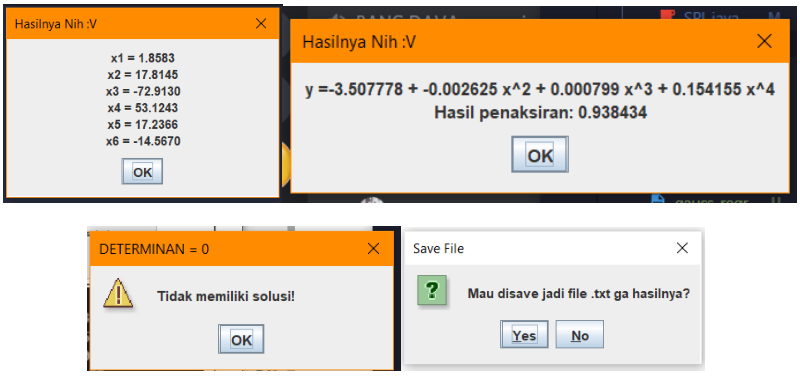
a. Menu Utama



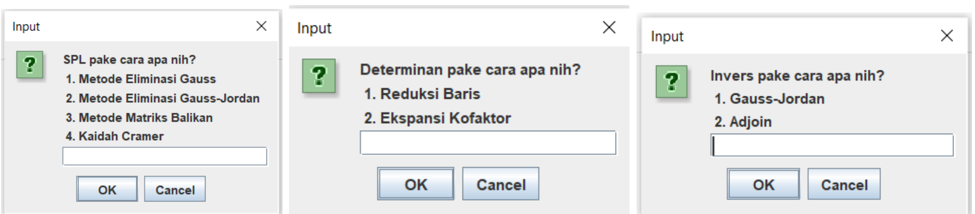
b. Input file



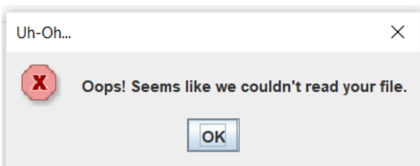
c. Output hasil



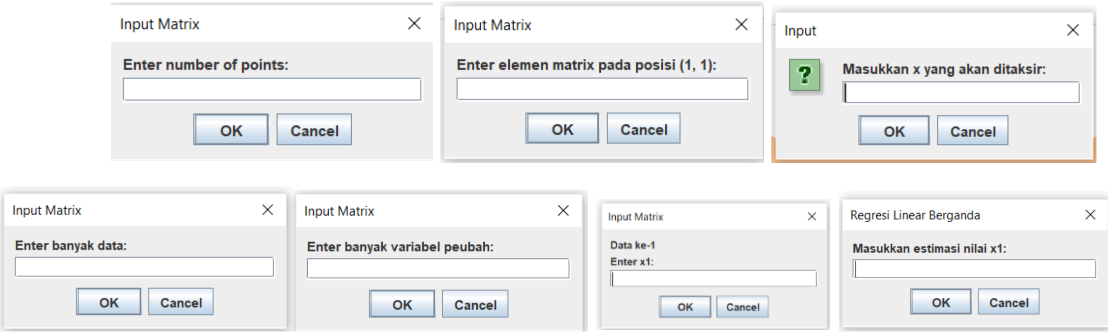
d. Input pilihan cara



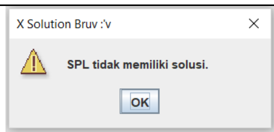
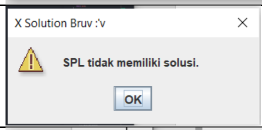
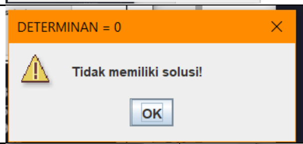
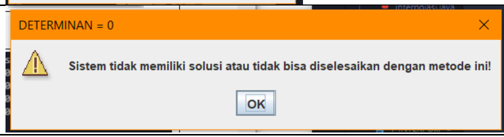
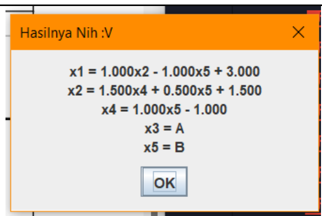
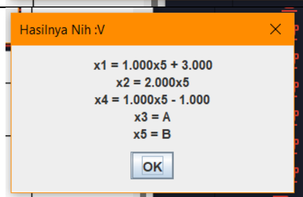
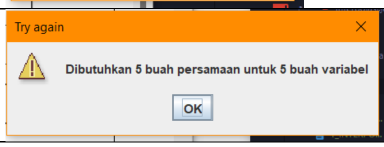
e. Error

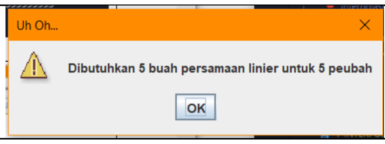
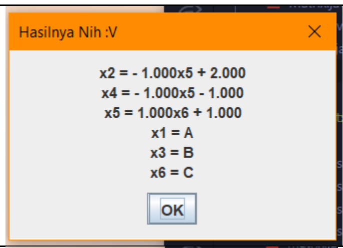
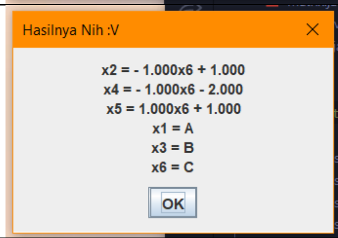
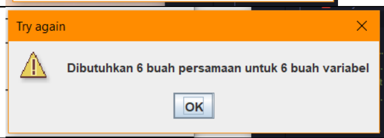
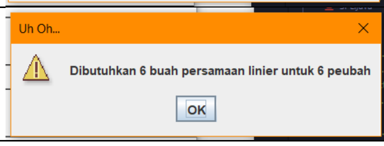
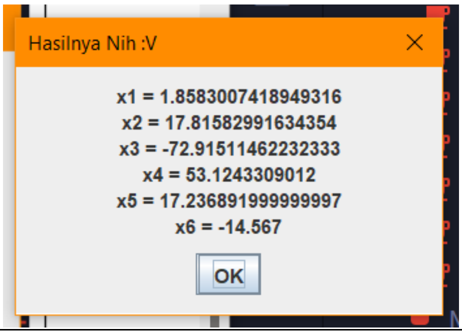


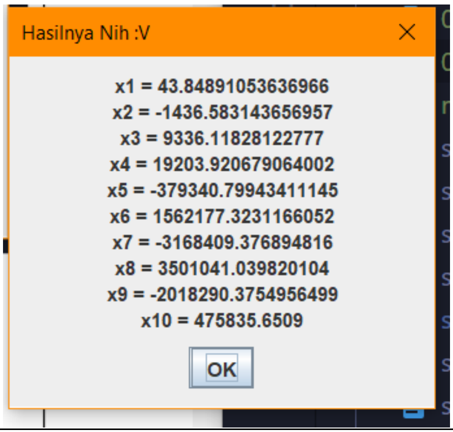
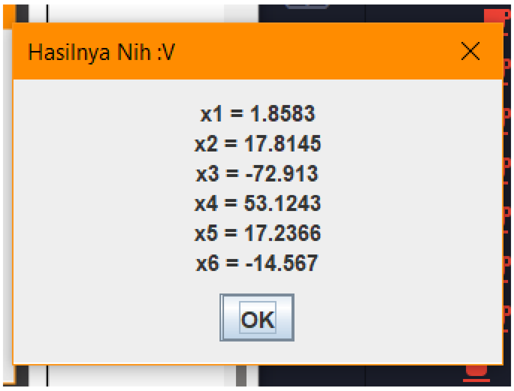
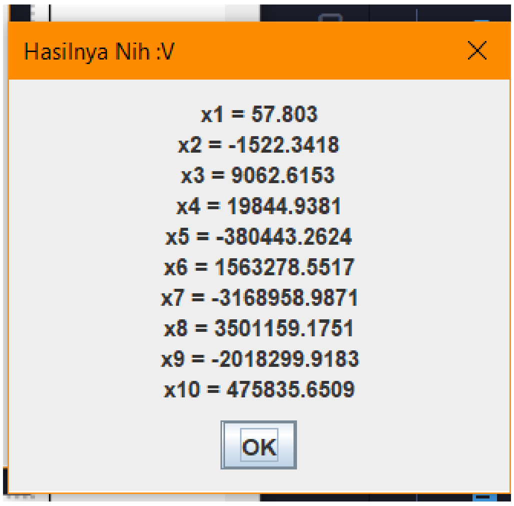
f. Input data

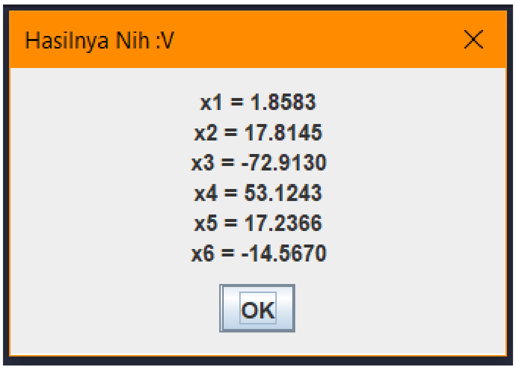


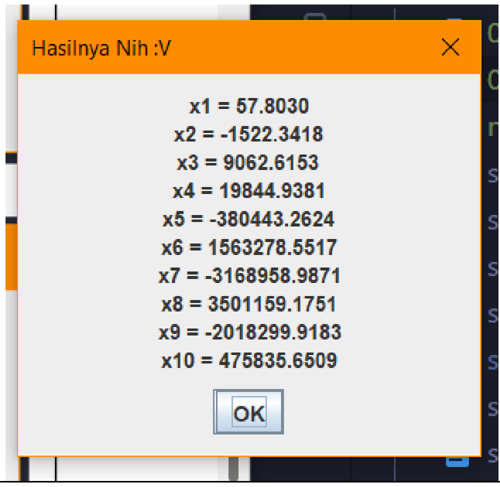
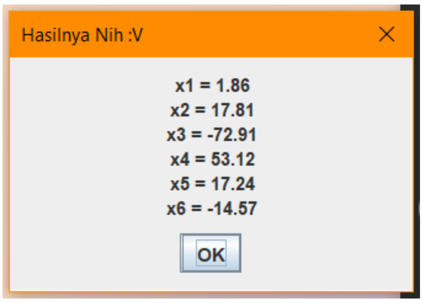
2. Studi Kasus

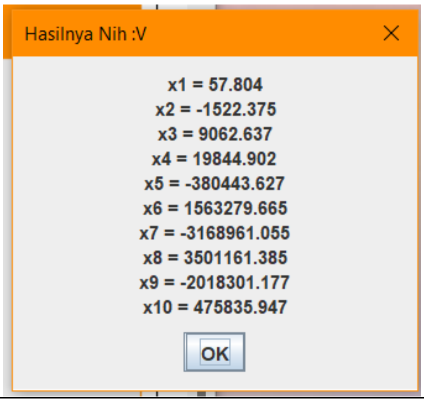
No	Soal	Metode	Hasil
1a	$A=\begin{bmatrix}1&1&-1&-1\\2&5&-7&-5\\2&-1&1&3\\5&2&-4&2\end{bmatrix},\quad b=\begin{bmatrix}1\\-2\\4\\6\end{bmatrix}$	Gauss	
		Gauss Jordan	
		Matriks Balikan	
		Cramer	
Analisis: Solusi dari SPL ini sama pada keempat metode, yaitu tidak memiliki solusi alias determinannya 0 (kaidah cramer dan matriks balikan) dan terdapat baris berisi angka 0 kecuali angka paling ujung kanan baris tersebut (eliminasi Gauss dan Gauss Jordan).			
1b	$A=\begin{bmatrix}1&-1&0&0&1\\1&1&0&-3&0\\2&-1&0&1&-1\\-1&2&0&-2&-1\end{bmatrix},\quad b=\begin{bmatrix}3\\6\\5\\-1\end{bmatrix}$	Gauss	
		Gauss Jordan	
		Matriks Balikan	

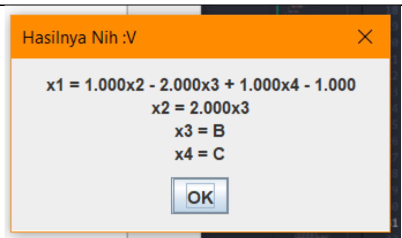
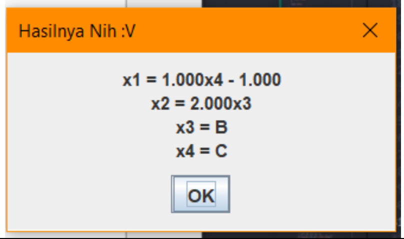
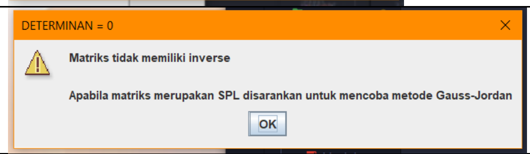
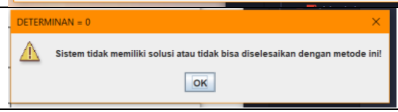
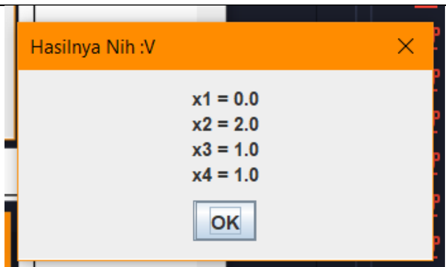
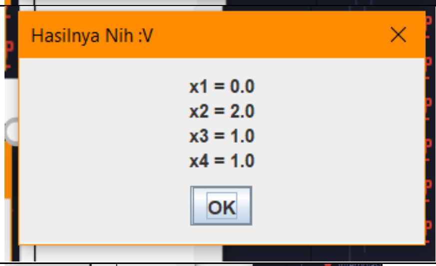
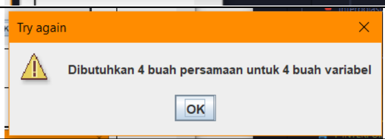
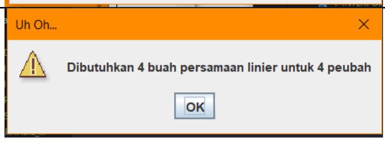
		Cramer	
Analisis: Penyelesaian SPL tidak dapat dilakukan melalui kaidah cramer atau matriks balikan karena matriks A bukan matriks persegi. Melalui eliminasi Gauss dan Gauss Jordan, didapatkan penyelesaian SPL dalam bentuk persamaan parametrik seperti pada gambar.			
1c	$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$	Gauss	
		Gauss Jordan	
		Matriks Balikan	
		Cramer	
Analisis: Penyelesaian SPL tidak dapat dilakukan melalui kaidah cramer atau matriks balikan karena matriks A bukan matriks persegi. Melalui eliminasi Gauss dan Gauss-Jordan, didapatkan penyelesaian SPL dalam bentuk solusi unik.			
1d	$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ Coba n = 6 dan n = 10	Gauss	N = 6 

			N = 10 
		Gauss Jordan	N = 6  N = 10 

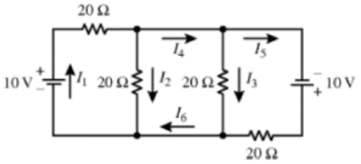
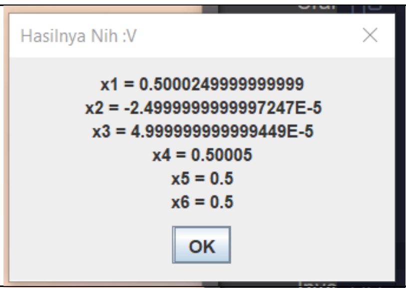
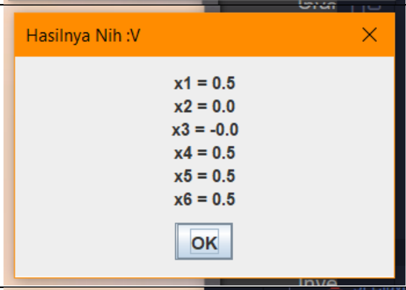
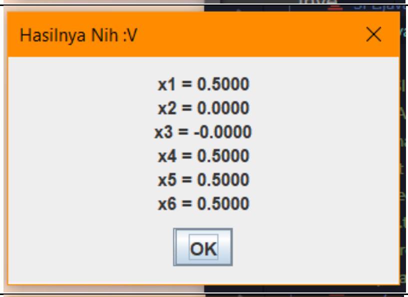
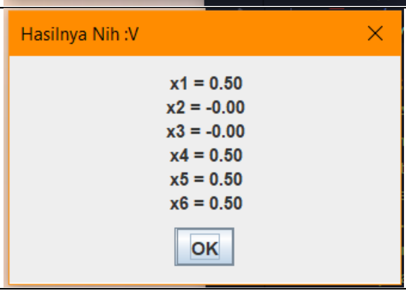
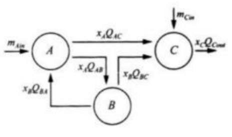
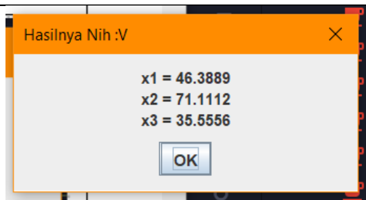
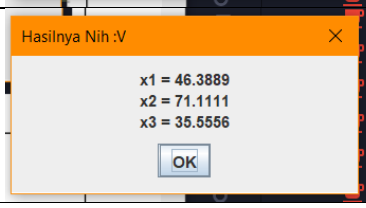
<table>
<tr><td></td><td></td><td>Matriks Balikan</td><td>N = 6


N = 10

</td></tr>
<tr><td></td><td></td><td>Cramer</td><td>N = 6


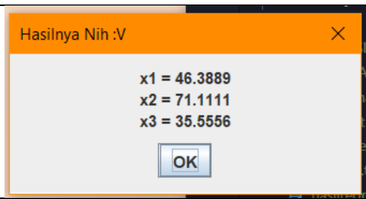
N = 10

</td></tr>
<tr><td colspan="4">Analisis: SPL memiliki jawaban berupa solusi unik yang dapat diselesaikan oleh keempat metode.</td></tr>
</table>

<table>
<tr><td rowspan="4">2a</td><td rowspan="4">$$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$$</td><td>Gauss</td><td>
</td></tr>
<tr><td>Gauss Jordan</td><td>
</td></tr>
<tr><td>Matriks Balikan</td><td>
</td></tr>
<tr><td>Cramer</td><td>
</td></tr>
<tr><td colspan="4">Analisis: Soal ini tidak dapat diselesaikan oleh metode Cramer dan matriks balikan karena determinan dari matriks adalah 0 (sifatnya inkonsisten atau bersolusi banyak). Sebagai alternatif, metode Gauss dan Gauss Jordan dapat digunakan untuk menyelesaikan SPL ini dengan solusi berupa parametrik.</td></tr>
<tr><td rowspan="4">2b</td><td rowspan="4">$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$$</td><td>Gauss</td><td>
</td></tr>
<tr><td>Gauss Jordan</td><td>
</td></tr>
<tr><td>Matriks Balikan</td><td>
</td></tr>
<tr><td>Cramer</td><td>
</td></tr>
</table>

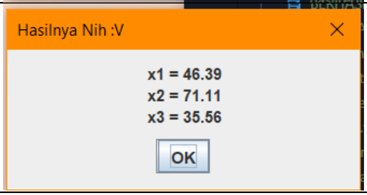
<table>
<tr><td colspan="4">Analisis: Soal tidak dapat diselesaikan oleh matriks balikan dan Cramer karena matriks tidak berbentuk persegi. Dengan menggunakan metode Gauss dan Gauss-Jordan, ditemukan jawaban berupa solusi unik.</td></tr>
<tr><td rowspan="4">3a</td><td rowspan="4">$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$</td><td>Gauss</td><td>Hasilnya Nih :V
x1 = -0.22431395175000002
x2 = 0.182445254
x3 = 0.70942212
x4 = -0.2581
OK</td></tr>
<tr><td>Gauss Jordan</td><td>Hasilnya Nih :V
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
OK</td></tr>
<tr><td>Matriks Balikan</td><td>Hasilnya Nih :V
x1 = -0.2243
x2 = 0.1824
x3 = 0.7095
x4 = -0.2581
OK</td></tr>
<tr><td>Cramer</td><td>Hasilnya Nih :V
x1 = -0.22
x2 = 0.18
x3 = 0.71
x4 = -0.26
OK</td></tr>
<tr><td colspan="4">Analisis: Keempat metode penyelesaian SPL dapat digunakan untuk menjawab soal no 3a dengan nilai taksiran yang cenderung sama.</td></tr>
<tr><td rowspan="4">3b</td><td rowspan="4">$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$$</td><td>Gauss</td><td>X Solution Bruv :v
SPL tidak memiliki solusi.
OK</td></tr>
<tr><td>Gauss Jordan</td><td>X Solution Bruv :v
SPL tidak memiliki solusi.
OK</td></tr>
<tr><td>Matriks Balikan</td><td>Try again
Dibutuhkan 9 buah persamaan untuk 9 buah variabel
OK</td></tr>
<tr><td>Cramer</td><td>Uh Oh...
Dibutuhkan 9 buah persamaan linier untuk 9 peubah
OK</td></tr>
</table>

Analisis: Soal tidak dapat diselesaikan menggunakan matriks balikan dan kaidah cramer, namun bisa diselesaikan dengan menggunakan metode Gauss dan Gauss-Jordan. Jawaban dari SPL ini bertipe tidak bersolusi.			
4		Gauss	
		Gauss Jordan	
		Matriks Balikan	
		Cramer	
Analisis: Soal dapat diselesaikan dengan menggunakan keempat metode penyelesaian SPL dengan jawaban bertipe solusi unik. Jawaban dari keempat metode relatif sama.			
5	 Dengan laju volume Q dalam m^3/s dan input massa m_{in} dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut: A: $m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$ B: $Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0$ C: $m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0$ 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.	Gauss	
		Gauss Jordan	

Matriks Balikan



Cramer



Analisis: Soal dapat dijawab menggunakan keempat metode penyelesaian SPL dengan tipe jawaban berupa solusi unik. Nilai dari x1, x2, dan x3 yang didapat oleh keempat metode cenderung sama.

6a

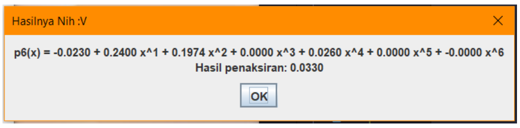
x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
$f(x)$	0.003	0.067	0. 148	0.248	0.370	0.518	0.697

$x = 0.2$
$x = 0.55$
$x = 0.85$
$x = 1.28$

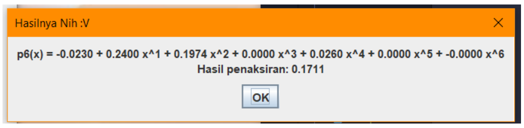
$f(x) = ?$

Interpolasi

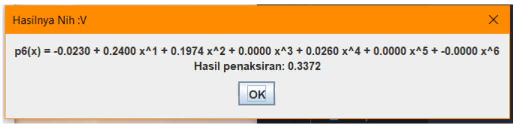
x = 0.2



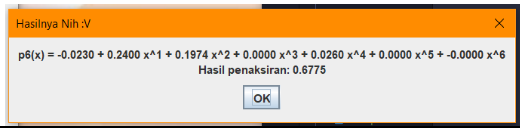
X = 0.55



X = 0.85



X = 1.28



Analisis: Taksiran nilai bisa didapatkan dengan memanfaatkan fitur interpolasi.

6b

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2021	6,567	12.624
30/06/2021	7	21.807
08/07/2021	7,258	38.391
14/07/2021	7,451	54.517
17/07/2021	7,548	51.952
26/07/2021	7,839	28.228
05/08/2021	8,161	35.764
15/08/2021	8,484	20.813
22/08/2021	8,709	12.408
31/08/2021	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 17/06/2021 (dibaca: 17 Juni 2021) diperoleh tanggal(desimal) sebagai berikut:

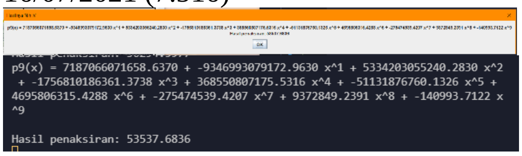
Tanggal(desimal) = 6 + (17/30) = 6,567

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

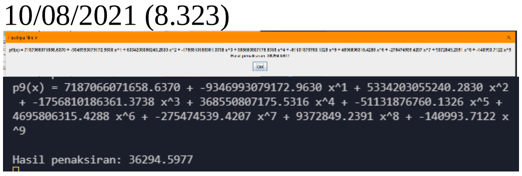
a. 16/07/2021
b. 10/08/2021
c. 05/09/2021
d. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2021.

Interpolasi

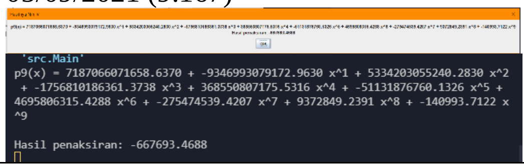
16/07/2021 (7.516)



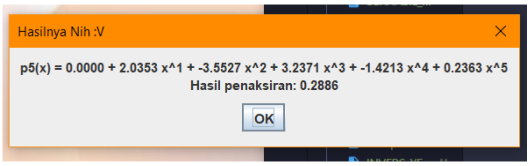
10/08/2021 (8.323)

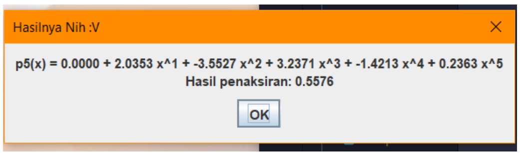


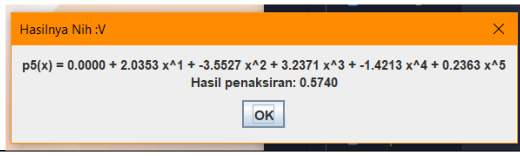
05/09/2021 (9.167)



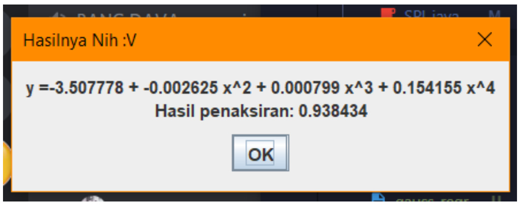
Analisis: Taksiran nilai bisa didapatkan dengan memanfaatkan fitur interpolasi. Kami mencantumkan versi di terminal apabila gambar pada panel terlalu kecil untuk dibaca.

<table>
<tr>
<td>6c</td>
<td>Sederhanakan fungsi
$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$
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.</td>
<td>Interpolasi</td>
<td>X = 0.2


X = 1.173


X = 1.85

</td>
</tr>
<tr>
<td colspan="4">Analisis: Taksiran nilai bisa didapatkan dengan memanfaatkan fitur interpolasi.</td>
</tr>
<tr>
<td>7</td>
<td>Diberikan sekumpulan data sesuai pada tabel berikut ini.
Table 12.1: Data for Example 12.1
<table>
<tr><th>Nitrous Oxide, y</th><th>Humidity, x_1</th><th>Temp., x_2</th><th>Pressure, x_3</th><th>Nitrous Oxide, y</th><th>Humidity, x_1</th><th>Temp., x_2</th><th>Pressure, x_3</th></tr>
<tr><td>0.90</td><td>72.4</td><td>76.3</td><td>29.18</td><td>1.07</td><td>23.2</td><td>76.8</td><td>29.38</td></tr>
<tr><td>0.91</td><td>41.6</td><td>70.3</td><td>29.35</td><td>0.94</td><td>47.4</td><td>86.6</td><td>29.35</td></tr>
<tr><td>0.96</td><td>34.3</td><td>77.1</td><td>29.24</td><td>1.10</td><td>31.5</td><td>76.9</td><td>29.63</td></tr>
<tr><td>0.89</td><td>35.1</td><td>68.0</td><td>29.27</td><td>1.10</td><td>10.6</td><td>86.3</td><td>29.56</td></tr>
<tr><td>1.00</td><td>10.7</td><td>79.0</td><td>29.78</td><td>1.10</td><td>11.2</td><td>86.0</td><td>29.48</td></tr>
<tr><td>1.10</td><td>12.9</td><td>67.4</td><td>29.39</td><td>0.91</td><td>73.3</td><td>76.3</td><td>29.40</td></tr>
<tr><td>1.15</td><td>8.3</td><td>66.8</td><td>29.69</td><td>0.87</td><td>75.4</td><td>77.9</td><td>29.28</td></tr>
<tr><td>1.03</td><td>20.1</td><td>76.9</td><td>29.48</td><td>0.78</td><td>96.6</td><td>78.7</td><td>29.29</td></tr>
<tr><td>0.77</td><td>72.2</td><td>77.7</td><td>29.09</td><td>0.82</td><td>107.4</td><td>86.8</td><td>29.03</td></tr>
<tr><td>1.07</td><td>24.0</td><td>67.7</td><td>29.60</td><td>0.95</td><td>54.9</td><td>70.9</td><td>29.37</td></tr>
</table>
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.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.
$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$
$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$
$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$
$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$</td>
<td>Regresi Linier Berganda</td>
<td>
</td>
</tr>
<tr>
<td colspan="4">Analisis: Taksiran nilai bisa didapatkan dengan memanfaatkan fitur regresi linier berganda.</td>
</tr>
</table>

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

I. Kesimpulan

Telah berhasil diimplementasikan sebuah objek berupa matriks sesuai dengan definisi matriks pada umumnya yang dipelajari dalam mata kuliah IF2123 Aljabar Linear dan Geometri. Hal mengenai matriks yang berhasil diimplementasi dalam program ini meliputi:

a. Sifat matriks: matriks persegi, matriks identitas, matriks eselon
b. Transformasi elementer: pertukaran baris, perkalian baris dengan konstanta, penambahan baris dengan kelipatan baris lainnya
c. Transformasi kompleks: reduksi eselon (gauss dan gauss jordan), inversi, transposisi
d. Properti: baris, kolom, determinan, minor/kofaktor

Semua implementasi ini kemudian berhasil digunakan untuk menyelesaikan metode yang ada di dalam spesifikasi, di antaranya penyelesaian Sistem Persamaan Linear dengan metode eliminasi Gauss, Gauss Jordan, matriks inversi, dan kaidah *Cramer*. Selanjutnya metode Sistem Persamaan Linear (SPL) ini digunakan sebagai dasar dalam mencari koefisien polinom dan fungsi linear yang tepat berturut-turut dalam metode interpolasi dan regresi linear berganda.

Solusi dari SPL sendiri dapat berupa solusi banyak, solusi unik, maupun tidak memiliki solusi sama sekali. Keefektifan metode penyelesaian SPL yang digunakan bergantung pada matriks/SPL yang diterima. Untuk metode matriks balikan dan kaidah *Cramer*, matriks *augmented* kiri harus berukuran N x N atau jumlah persamaan pada SPL harus sama dengan jumlah variabel yang dicari. Oleh karena itu, untuk mencari solusi dari sebuah SPL, metode yang paling efektif dan fleksibel untuk digunakan adalah metode eliminasi Gauss atau Gauss-Jordan. Hal ini juga terbukti dari studi kasus yang telah dilakukan dimana hampir semua kasus dapat diselesaikan dengan kedua metode tersebut.

Nilai taksiran dari suatu interpolasi polinom maupun regresi linear berganda dapat diselesaikan dengan mengaplikasikan SPL. Fitur ini terdapat pada program kalkulator Java yang kami buat. Terdapat 5 fitur yang dapat digunakan pada program kalkulator Java kami, antara lain:

a. Menyelesaikan SPL menggunakan metode eliminasi Gauss, eliminasi Gauss-Jordan, matriks balikan, dan kaidah *Cramer*.
b. Mencari determinan dari sebuah matriks dengan metode ekspansi kofaktor dan reduksi baris.
c. Mencari matriks balikan dengan memanfaatkan metode Gauss-Jordan dan adjoin.
d. Menaksir nilai dari suatu interpolasi polinom
e. Menyelesaikan permasalahan regresi linear berganda dengan memanfaatkan metode Gauss dan *normal equation*.

Fitur ini dapat digunakan pada program kalkulator kami yang sudah menerapkan GUI untuk mempermudah interaksi antara *user* dengan program.

Dengan demikian, kelompok menyimpulkan bahwa dengan mengerjakan Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester 1 Tahun 2021/2022 ini, dapat diketahui bahwa penyelesaian sistem persamaan linear (SPL) dapat dilakukan dengan membuat sebuah program dengan menggunakan bahasa pemrograman Java yang mengimplementasikan metode Gauss, Gauss-Jordan, matriks balikan, dan kaidah *Cramer*.

II. Saran

Tugas Besar 1 IF2123 Aljabar Linier dan Geometri Semester 1 Tahun 2021/2022 menjadi salah satu proses pembelajaran bagi kelompok dalam menerapkan ilmu-ilmu yang diajarkan pada kuliah maupun melakukan eksplorasi materi secara mandiri. Berikut ini merupakan sejumlah saran dari kelompok untuk pihak-pihak yang ingin melakukan atau mengerjakan hal serupa.

a. Program yang diminta adalah program dengan menggunakan bahasa pemrograman Java, yakni salah satu bahasa pemrograman yang belum dikuasai oleh ketiga anggota kelompok yang terlibat dalam pengerjaan tugas besar ini. Dengan demikian, kelompok merekomendasikan agar disediakan waktu yang cukup untuk melakukan eksplorasi terkait bahasa pemrograman yang digunakan sebelum mengimplementasikannya ke dalam sebuah program. Hal ini akan meningkatkan efektivitas kerja tim dalam pembuatan suatu program.
b. Setiap struktur data memiliki keunggulan dan kelemahannya masing - masing sehingga membuat struktur data tertentu unggul dan tepat untuk digunakan dalam situasi yang sesuai dengannya. Matriks memiliki banyak properti, sifat, dan operasi dengan kegunaannya masing - masing yang menjadi keunggulannya dibandingkan dengan struktur data yang lain ketika diimplementasikan dalam *Object Oriented Programming* (OOP). Kami sangat menyarankan untuk mengimplementasi matriks dalam struktur berorientasi objek. Berbagai sifat, atribut, serta operasi maupun transformasi yang dimilikinya dapat direalisasikan dengan rapi dalam struktur berorientasi objek ini. Pemakaian matriks dalam program tentunya dilakukan berulang kali dan tidak sedikit sehingga bertambah satu alasan untuk menggunakan matriks sebagai objek dalam sebuah program.
c. Penting bagi kelompok untuk memiliki strategi serta distribusi tugas yang baik. Ketika membuat program dalam sebuah tim, kesamaan cara menulis kode serta kemampuan untuk menulis komentar menjadi hal yang sangat penting. Hal ini diperlukan agar memudahkan anggota kelompok dalam menyatukan dan melanjutkan sebuah program. Kemampuan tersebut tentunya didukung juga dengan adanya *version control system* yang baik yang dapat digunakan oleh *programmer* dalam membuat sebuah program secara bersama-sama. Untuk itu, kami sangat menyarankan ‘GitHub’ untuk digunakan sebagai *version control system* dalam pengerjaan tugas-tugas besar pada mata kuliah IF2123 ini, maupun pada pembuatan program yang lainnya.

III. Refleksi

Matriks merupakan sebuah struktur data yang termasuk ke dalam salah satu struktur aljabar. Struktur data ini memiliki bermacam-macam kegunaan berkaitan dengan properti, sifat, dan operasi yang dibawa atau dimiliki oleh matriks ini sendiri. Dalam mengerjakan Tugas Besar

1 IF2123 Aljabar Linier dan Geometri Semester 1 Tahun 2021/2022 ini, banyak hal dan sudut pandang yang diperoleh dan dipelajari mengenai matriks. Timbul apresiasi yang besar dalam diri anggota kelompok kepada semua orang yang terlibat dalam bidang studi aljabar linear, khususnya dalam hal studi matriks. Keberadaan matriks serta atribut-atribut yang dimilikinya memiliki dampak dan manfaat yang signifikan, terutama dalam menyelesaikan suatu masalah yang melibatkan sistem persamaan linear. Dengan begitu, kelompok menyadari bahwa orang-orang tersebut memiliki peran yang patut dihargai sebagai bagian dari kelompok yang telah melahirkan maupun mengkaji ilmu dan penerapan matriks itu sendiri.

Kegunaan matriks sebagian besar dapat dijumpai pada bidang keilmuan aljabar linear. Akan tetapi, perlu diketahui bahwa bidang keilmuan ini beririsan dengan berbagai bidang keilmuan lain yang ada. Bidang keilmuan yang tidak terlepas dari matriks, misalnya statistika, elektronika, informatika, mesin, kimia, dan lain sebagainya. Hal ini disebabkan karena bidang-bidang tersebut berurusan dengan teknik pengolahan data, melibatkan kalkulasi bilangan besar, sistem persamaan yang banyak, memperkirakan sebuah nilai atau sebuah fungsi dari informasi data yang ada, dan lain sebagainya. Sifat matriks yang sarat akan kegunaannya tentu membuat salah satu ilmu dari bidang aljabar linear dan geometri ini patut untuk dipelajari serta ditekuni agar dapat diimplementasikan di kemudian hari.

REFERENSI

Docs.oracle.com. (2020). Java™ Platform, Standard Edition 7. Diakses pada 22 September 2021, dari https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html

Geeksforgeeks.org. (2016, 10 Juli). System.exit() in Java. Diakses pada 25 September 2021, dari https://www.geeksforgeeks.org/system-exit-in-java/

Geeksforgeeks.org. (2018, 6 September). Different Ways of Reading a Text File in Java. Diakses pada 24 September 2021, dari https://www.geeksforgeeks.org/different-ways-reading-text-file-java/

Geeksforgeeks.org. (2020, 20 November). How to Set Class Path in Java?. Diakses pada 20 September 2021, dari https://www.geeksforgeeks.org/how-to-set-classpath-in-java/

Geeksforgeeks.org. (2021, 16 Juli). Program for Gauss-Jordan Elimination Method. Diakses pada 23 September 2021, dari https://www.geeksforgeeks.org/program-for-gauss-jordan-elimination-method/

Informatika.stei.itb.ac.id/~rinaldi.munir. (2021). Algeo #4 Tiga Kemungkinan Solusi SPL. Diakses pada 28 September 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-04-Tiga-Kemungkinan-Solusi-SPL.pdf

Informatika.stei.itb.ac.id/~rinaldi.munir. (2021). Algeo #5 Sistem Persamaan Linier (Metode Eliminasi Gauss Jordan). Diakses pada 28 September 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf

Informatika.stei.itb.ac.id/~rinaldi.munir. (2021). Algeo #8 Determinan (bagian 1). Diakses pada 24 September 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf

Informatika.stei.itb.ac.id/~rinaldi.munir. (2021). Algeo #9 Determinan (bagian 2). Diakses pada 24 September 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf

Jackrutorial.com. (2018, 20 Juni). Java Split String by Space and Newline. Diakses pada 28 September 2021, dari https://www.jackrutorial.com/2018/06/java-split-string-by-space-and-newline.html

Java2blog.com. (2021). 7 Ways to Format Double to 2 Decimal Places in Java. Diakses pada 26 September 2021, dari https://java2blog.com/format-double-to-2-decimal-places-java/

Javapoint.com. (2021). Java Convert Double to String. Diakses pada 29 September 2021, dari https://www.javatpoint.com/java-double-to-string

Sanfoundry.com. (2021). Java Program to Implement Gauss Jordan Elimination. Diakses pada 22 September 2021, dari https://www.sanfoundry.com/java-program-implement-gauss-jordan-elimination/

Sanfoundry.com. (2021). Java Program to Implement Gaussian Elimination Algorithm. Diakses pada 22 September 2021, dari https://www.sanfoundry.com/java-program-gaussian-elimination-algorithm/

Techiedelight.com. (2021). Read a Text File Using FileReader in Java. Diakses pada 24 September 2021, dari https://www.techiedelight.com/read-text-file-using-filereader-java/