Laporan Tugas Besar I
IF2123 Aljabar Linier dan Geometri

Oleh:

Kelompok BINGUNG

Anggota:

Vionie Novencia Thanggestyo (13520006)

Wesly Giovano (13520071)

Vieri Mansyl (13520092)

SEKOLAH TINGGI ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021/2022

DAFTAR ISI

BAB I

DESKRIPSI MASALAH

1.1 Latar Belakang

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

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

1.2 Spesifikasi Tugas

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

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

1.3 Spesifikasi Program

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 aij. Masukan dari file berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

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

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (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 isinyadipersilakan 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

2.1 Matriks dan Operasi Baris Elementer

Matriks adalah susunan bilangan yang disusun dalam baris dan kolom. Matriks dapat direpresentasikan seperti berikut.

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Gambar 2.1.1. Representasi matriks

Matriks persegi adalah matriks yang memiliki jumlah baris dan kolom yang sama. Matriks persegi yang seluruh elemen diagonal utama bernilai 1 dan yang bukan elemen diagonal utama bernilai 0 disebut matriks identitas yang dilambangkan dengan *I*.

Matriks eselon baris adalah matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya bernilai nol. 1 utama adalah bilangan tidak nol pertama di dalam baris tersebut.

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

Gambar 2.1.2. Contoh matriks eselon baris

Matriks eselon baris tereduksi adalah matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya bernilai nol, serta elemen di bawah dan di atas 1 utama bernilai nol.

$$\begin{bmatrix} 1 & * & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{atau} \quad \begin{bmatrix} 0 & 1 & 0 & 0 & * \\ 0 & 0 & 1 & 0 & * \\ 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Gambar 2.1.3. Contoh matriks eselon baris tereduksi

Operasi-operasi dasar matriks meliputi penjumlahan dua buah matriks, perkalian skalar dengan matriks, perkalian dua buah matriks, dan transpose matriks.

1. Penjumlahan dua buah matriks dinyatakan sebagai $C = A + B = [c_{ij}]$, dengan $c_{ij} = a_{ij} + b_{ij}$.
2. Perkalian skalar dengan matriks dinyatakan sebagai $B = k \cdot A = [b_{ij}]$, dengan $b_{ij} = k \cdot a_{ij}$.

3. Perkalian dua buah matriks dinyatakan sebagai C = A × B = [c_{ij}], dengan $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{in}b_{nj}$.
4. Transpose matriks dinyatakan sebagai B = A^T = [b_{ij}], dengan $b_{ij} = a_{ji}$.

Operasi baris elementer adalah operasi yang dapat dilakukan pada baris matriks. Operasi baris elementer meliputi tiga hal:
1. Perkalian suatu baris dengan konstanta tidak nol.
2. Pertukaran dua buah baris.
3. Penambahan suatu baris dengan kelipatan baris lainnya.

2.2 Determinan

Determinan merupakan sebuah nilai skalar yang dapat dihitung dari suatu matriks persegi. Determinan suatu matriks A ditulis sebagai det(A) , det A , atau |A| , dan direpresentasikan sebagai berikut.

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

Gambar 2.2.1. Penulisan determinan matriks A

Misalkan matriks $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, determinan matriks A yang berukuran 2 x 2 bernilai :

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

Untuk matriks berukuran 3 x 3 , determinan matriks dihitung sebagai berikut.

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$
$$= aei + bfg + cdh - ceg - bdi - afh.$$

Gambar 2.2.2. Determinan matriks A berukuran 3x3

Dengan membentuk matriks seitiga atas maupun bawah, determinan dari matriks berukuran N x N dapat dihitung dari perkalian diagonal utama matriks tersebut, yaitu :

$$\det(A) = a_{11}a_{22} \ldots a_{(n-1)(n-1)}a_{nn}$$

Determinan dari matriks A yang berukuran N x N yang melewati proses OBE akan bernilai

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

dengan p menyatakan banyaknya operasi pertukaran baris di dalam OBE serta k_1 s.d. k_n menyatakan perkalian baris-baris matriks di dalam OBE.

Determinan matriks A berukuran N x N juga dapat dihitung menggunakan metode ekspansi kofaktor, dengan M_{ij} = minor entri a_{ij} (submatiks dari matriks A yang elemen-elemennya tidak berada pada baris i dan kolom j), serta $C_{ij} = (-1)^{i+j}M_{ij}$ = kofaktor entri a_{ij}.

Dengan menggunakan metode ekspansi kofaktor ,determinan matriks A berukuran N x N dapat dihitung sebagai berikut.

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

Secara baris

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

Secara kolom

Gambar 2.2.3. determinan matriks A berukuran N x N dengan metode ekspansi kofaktor

2.3 Matriks Balikan

Suatu matriks memiliki balikan apabila merupakan matriks persegi dan determinan ≠ 0.

2.3.1 Metode Matriks Adjoin

Untuk matriks berordo 2x2, balikan A dapat dihitung dengan cara singkat sebagai berikut.

$$A^{-1} = \frac{1}{det(A)}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Untuk matriks berordo 3x3, adjoin matriks didapatkan dari transpose kofaktor matriks. Misalkan matriks A adalah

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Kofaktor dari matriks A dapat dicari, yaitu $C_{11} = \begin{vmatrix} e & f \\ h & i \end{vmatrix}$, $C_{12} = \begin{vmatrix} d & f \\ g & i \end{vmatrix}$, $C_{13} = \begin{vmatrix} d & e \\ g & h \end{vmatrix}$, dan seterusnya sehingga membentuk matriks kofaktor

$$A_C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

Kofaktor tersebut kemudian ditranspose hingga didapat adjoin matriks A sebagai berikut.

$$adj(A) = (A_C)^T = \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix}$$

Sehingga, untuk matriks berordo 3x3, balikan matriks dapat dicari dengan:

$$A^{-1} = \frac{1}{det(A)} adj(A) = \frac{1}{det(A)} \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix}$$

2.3.2 Metode Eliminasi Gauss-Jordan

Misalkan A adalah matriks persegi berukuran 3×3.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Matriks balikan A dapat dicari dengan melakukan operasi baris elementer (eliminasi Gauss-Jordan) pada matriks augmented (A|I) menjadi (I|A^{-1}).

$$(A|I) = \begin{bmatrix} a & b & c & 1 & 0 & 0 \\ d & e & f & 0 & 1 & 0 \\ g & h & i & 0 & 0 & 1 \end{bmatrix} \longrightarrow (I|A^{-1}) = \begin{bmatrix} 1 & 0 & 0 & a' & b' & c' \\ 0 & 1 & 0 & d' & e' & f' \\ 0 & 0 & 1 & g' & h' & i' \end{bmatrix}$$

Sehingga diperoleh A^{-1} yaitu

$$A^{-1} = \begin{bmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{bmatrix}$$

2.4 Sistem Persamaan Linier

Sistem persamaan linier merupakan sekumpulan persamaan linier (suku tertingginya 1) yang memiliki m buah persamaan dan n buah variabel (dapat dinyatakan sebagai bentuk A**x** = **b**), sehingga dapat ditulis sebagai berikut.

a.)
$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \vdots \qquad \vdots \qquad\quad\; \vdots \quad &\;\; \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

b.)
$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

c.) $$[A \mid \mathbf{b}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right]$$

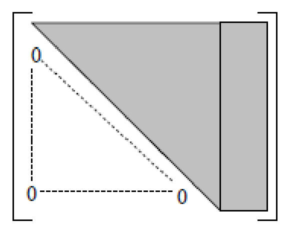
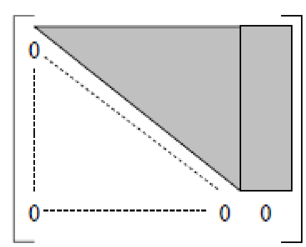
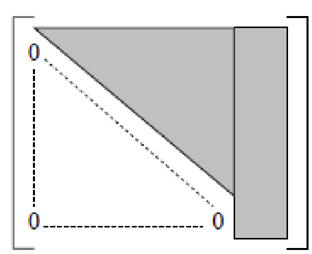
Gambar 2.4.1. a.) Sistem Persamaan Linier b.) SPL dalam representasi matriks c.) SPL dalam representasi matriks *augmented*

Dengan menyatakan suatu SPL ke dalam bentuk matriks *augmented*, SPL tersebut dapat diselesaikan dengan 2 metode, yaitu :

a. metode Gauss : mengubah matriks menjadi bentuk matriks eselon baris
b. metode Gauss Jordan : mengubah matriks menjadi bentuk matriks eselon baris tereduksi

Kedua metode tersebut akan menghasilkan matriks baru, sedemikian sehingga persamaan-persamaan pada SPL dapat diselesaikan dengan metode substitusi mundur.

Terdapat 3 kemungkinan solusi pada SPL , ialah :

Solusi unik / tunggal	
Solusi banyak / tak terhingga Seluruh elemen pada suatu baris bernilai 0.	
Tanpa solusi Seluruh elemen kecuali pada kolom terakhir pada suatu baris bernilai 0.	

Apabila suatu SPL memiliki bentuk A**x**=**b** , Sistem Persamaan Linier Homogen memiliki bentuk A**x**=**b** dengan **b** = **0** . Terdapat 2 jenis solusi pada SPL Homogen , ialah :

a. solusi trivial : solusi dimana seluruh variabel x bernilai 0
b. solusi non-trivial : solusi lain selain seluruh variabel x bernilai 0

Suatu SPL sembarang A**x**=**b** disebut konsisten apabila sekurangnya memiliki 1 solusi , dan disebut inkonsisten apabila tidak memiliki solusi.

Suatu SPL A**x**=**b** dengan n buah persamaan dan n buah peubah dapat pula diselesaikan dengan metode matriks balikan dan kaidah Cramer jika dan hanya jika det(A) ≠ 0.

SPL dengan kaidah Cramer memiliki solusi unik yaitu

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

yang dalam hal ini A_j adalah matriks yang diperoleh dengan mengganti entri pada kolom ke-j dari A dengan entri dari matriks **b**.

Selain ketiga metode diatas, SPL juga dapat diselesaikan dengan menggunakan matriks balikan .Misalkan SPL A**x** = b. Kalikan kedua ruas persamaan dengan A^{-1}

$$(A^{-1})A\mathbf{x} = (A^{-1})b$$

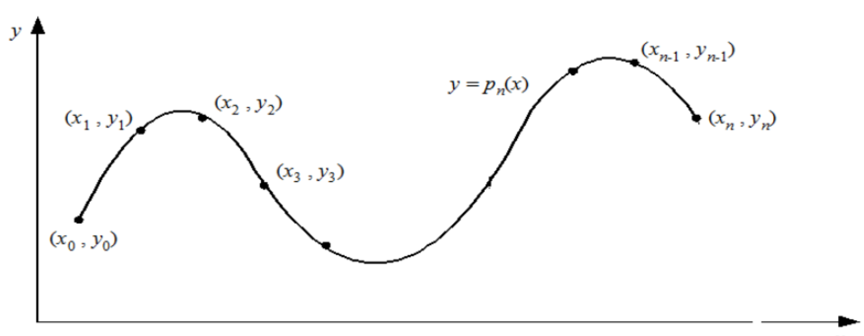
$$I\mathbf{x} = A^{-1}b$$

$$\mathbf{x} = A^{-1}b$$

Jadi, solusi SPL A**x**=b adalah $x = A^{-1}b$

2.5 Interpolasi Polinomial

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



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

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x_0 , y_0), (x_1 , y_1), ..., (x_n , y_n). adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, (x_0 , y_0) dan (x_1 , y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0 , y_0), (x_1 , y_1), dan (x_2 , y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0 , y_0), (x_1 , y_1), (x_2 , y_2), dan (x_3 , y_3), polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (x_i , y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx_n$ untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2, …, a_n, yaitu

$$a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n = y_1$$

$$\ldots$$

$$a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n = y_n$$

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

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan $a_0 = 0.6762$, $a_1 = 0.2266$, dan $a_2 = -0.0064$. Polinom interpolasi yang melalui ketiga buah 2 titik tersebut adalah $p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$. Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

2.6 Regresi Linier Berganda

Regresi Linier Berganda merupakan model regresi linier dengan melibatkan lebih dari 1 variabel bebas (disebut prediktor) untuk memprediksi hasil dari variabel terikat. Tujuan dari regresi linier berganda adalah untuk memodelkan hubungan linier antara variabel bebas (independen) dan variabel terikat (dependen). Sebuah variabel dependen jarang dijelaskan oleh hanya satu variabel sehingga dengan regresi linier berganda memungkinkan seseorang untuk memprediksi suatu variabel berdasarkan informasi yang diketahui dari banyak variabel lain.

Regresi linier berganda didasarkan pada asumsi bahwa ada hubungan linier antara variabel dependen dan independent, serta mengasumsi tidak adanya korelasi besar antar variabel independen.

Formula dari regresi linier berganda berbentuk

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i$$

dengan :

y_i = variabel terikat

x_i = variabel bebas

β_0 = y-intercept (konstanta)

β_p = koefisien untuk setiap variabel bebas

ϵ = residu / error

Untuk mencari nilai dari setiap β , dapat digunakan metode *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut.

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

Gambar 2.6.1. Metode *Normal Estimation Equation for Multiple Linear Regression*

BAB III

IMPLEMENTASI

Program yang dibuat dalam bahasa pemrograman Java mengimplementasikan berbagai fungsionalitas ke dalam dua buah file, yaitu package Matrix.jar sebagai library penyelesaian operasi dan Main.java sebagai program utama yang akan dijalankan. File Main.java tersebut kemudian dikompilasi bersama package Matrix.jar menjadi file bytecode Main.class dengan jdk-17.

3.1 Matrix.jar

File ini berfungsi sebagai library sekaligus menyimpan class Matrix yang terbungkus dalam package Matrix. Class Matrix memiliki 3 atribut dan 33 method yang sebagian bersifat private (tidak dapat diakses dari Main.java).

Atribut class Matrix meliputi:

- `int row;` berfungsi untuk menyimpan jumlah baris matrix,
- `int col;` berfungsi untuk menyimpan jumlah kolom matrix, dan
- `double[][] Mat = new double[100][100];` berfungsi untuk menyimpan matrix dengan alokasi maksimum 100 baris dan 100 kolom.

Method class Matrix meliputi:

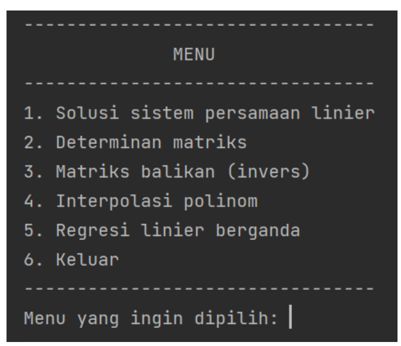
1. `Matrix()` berfungsi sebagai konstruktor matrix dengan menginisialisasi semua nilai pada Mat menjadi 0.
2. `getCopy() → Matrix` berfungsi untuk mendapatkan copy dari sebuah objek Matrix dan mereturnnya ke dalam suatu Matrix baru.
3. `readMatrixKeyboard()` berfungsi untuk membaca dan menyalin ke dalam atribut Mat sebuah objek Matrix dari keyboard sesuai dengan atribut row dan col yang telah terdefinisi.
4. `readMatrixFile(String pathname)` berfungsi untuk membaca dan menyalin ke dalam objek Matrix dari file dengan `pathname` sebagai parameter.
5. `displayMatrix(PrintWriter output)` berfungsi untuk menyalin sebuah objek Matrix ke layar terminal/console ataupun ke file, sesuai dengan parameter `output` yang dipanggil.
6. `displayMatrixSolution(PrintWriter output)` berfungsi untuk menyalin solusi SPL yang disimpan dalam objek Matrix ke layar atau file sesuai dengan parameter `output` yang dipanggil. Hasil solusi memiliki bentuk `xn = k0 + k1 a + k2 b + k3 c + ...` dengan k adalah konstanta/koefisien dan a,b,c,... adalah parameter. Parameter dibentuk dengan method `col2p(int i)`.
7. `isColZero(int row,int col) → integer` berfungsi untuk menentukan baris yang memiliki nilai pada kolom ke - col.

8. `isLastRowZero() → boolean` berfungsi untuk menentukan apakah seluruh elemen baris terakhir bernilai 0 atau tidak
9. `changeplace(int row,int col) → boolean` berfungsi untuk menukar baris dengan elemen terdefinisi pada kolom ke – col.
10. `bagi1utama(int row,int col)` berfungsi melakukan konversi elemen menjadi 1 utama disesuaikan pada baris ke-row tersebut.
11. `makeZero(int row, int col , int pass)` berfungsi melakukan konversi elemen menjadi 0 yang berada di bawah 1 utama disesuaikan pada seluruh row.
12. `getRowMain(int i) → int` menentukan row yang mengandung 1 utama sebagai eksistensi dari variabel x_i.
13. `getValue() → Matrix` berfungsi membentuk matriks yang menanpung nilai dari tiap variabel x_i.
14. `optimizeMat()` berfungsi untuk mengoptimasi matriks M x N (dengan M < N-1) menjadi matriks berukuran N x (N+1).
15. `gauss()` berfungsi untuk menkonversi matriks augmented menjadi matriks eselon baris.
16. `gaussJordan()` berfungsi untuk menkonversi matriks augmented menjadi matriks eselon baris tereduksi.
17. `solveGauss() → Matrix` berfungsi menyelesaikan SPL yang telah berbentuk matriks augmented menggunakan metode `gauss()`.
18. `solveGaussJordan() → Matrix` berfungsi menyelesaikan SPL yang telah berbentuk matriks augmented menggunakan metode `gaussJordan()`.
19. `solveInverse() → Matrix` berfungsi untuk menyelesaikan SPL dengan matriks balikan.
20. `solveCramer() → Matrix` berfungsi untuk menyelesaikan SPL dalam bentuk augmented dengan n buah peubah dan menghasilkan suatu objek Matrix baru yang menyimpan solusi SPL. Method ini memanfaatkan `getCopy()` agar dapat mengganti entri tiap kolom pada Matrix yang ingin diselesaikan.
21. `detReduction() → double` berfungsi menghitung determinan dari matriks persegi dengan menggunakan OBE.
22. `getCofactor(double[][] mat, double[][] temp, int rows, int cols, int dims)` berfungsi untuk mendapatkan kofaktor dari matriks mat dan memasukkan kofaktornya ke matriks temp.
23. `detCofactor(double [][]mat, int dim) → double` berfungsi untuk menghitung determinan dari matriks persegi dengan menggunakan metode ekspansi kofaktor.
24. `MatrixInvalid() → Matrix` berfungsi untuk mengembalikan matriks yang semua elemennya berisi –999.
25. `IsMatrixInvalid() → boolean` berfungsi untuk mengecek apakah suatu matriks adalah matriks invalid
26. `InverseAdjoin() → Matrix` berfungsi untuk menghitung matriks balikan dengan menggunakan metode adjoin.
27. `InverseIdentity() → Matrix` berfungsi untuk menghitung matriks balikan dengan menggunakna metode eliminasi Gauss-Jordan

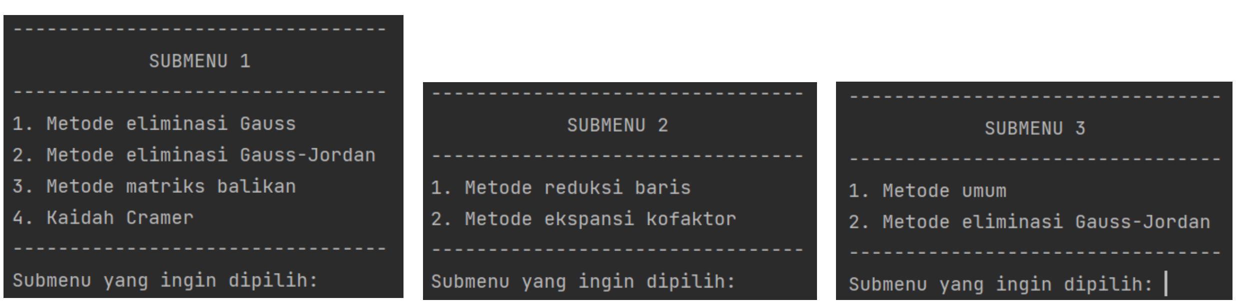
28. `InterpolasiPolinom(double x, PrintWriter output)` berfungsi untuk menghitung taksiran nilai interpolasi polinom
29. `displayRegSPL(PrintWriter output)` berfungsi menampilkan fungsi hasil dari regresi linier berganda.
30. `elmtReg(int row , int col) → double` berfungsi menentukan nilai dari tiap SPL yang dibentuk dari data yang didapat untuk keperluan regresi linier berganda.
31. `regLinierBerganda() → Matrix` berfungsi menentukan nilai konstanta serta koefisien-koefisien variabel bebas pada SPL yang dibentuk dari data yang tersedia dengan metode *Normal Estimation Equation* untuk regresi linier berganda.
32. `taksirReg() → double` berfungsi menghitung taksiran dari persamaan yang dihasilkan dari regresi linier berganda.
33. `col2p(int i) → String` berfungsi sebagai fungsi tambahan untuk membentuk huruf parameter pada method `displayMatrixSolution()`, dengan i=0 tidak menghasilkan parameter dan i=1..26 membentuk huruf a hingga z. Parameter lebih dari 26 buah mengakibatkan huruf parameter menjadi tidak terdefinisi.

3.2 Main.java

File ini berfungsi sebagai program utama yang dijalankan dengan menggunakan library class Matrix. Pada program utama akan muncul tampilan pertama berupa menu pilihan fungsi yang diinginkan. Kemudian menu 1, 2, dan 3 masing-masing akan merujuk ke submenu untuk pemilihan metode penyelesaiannya. Daftar menu dan submenu selengkapnya dapat dilihat di gambar 3.2.1 dan gambar 3.2.2.



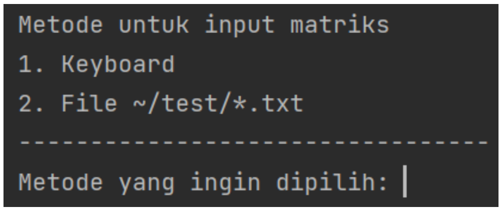
Gambar 3.2.1. Tampilan menu



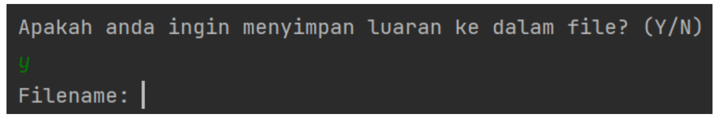
(a) (b) (c)

Gambar 3.2.2. Tampilan submenu 1, 2, dan 3

Masing-masing fungsi dari program utama menerima input dalam bentuk matriks baik melalui keyboard maupun file dalam folder test seperti pada Gambar 3.2.3(a). Luaran dari program akan secara otomatis muncul di layar, dengan opsi untuk menyimpan luaran seperti pada Gambar 3.2.3(b) ke dalam folder testOutput.



(a)



(b)

Gambar 3.2.3. Tampilan menu I/O ke file

Struktur dari class Main yang dibuat adalah dengan mengimplementasikan global variable atau field dan method print dan get, sehingga `main()` dari class Main lebih rapi dan terstruktur. Program utama juga dapat diulangi setelah suatu fungsi selesai digunakan dengan method `askRepeat()`.

BAB IV

EKSPERIMEN

Untuk memvalidasi kebenaran program yang dibuat, dilakukan ujian terhadap tujuh buah studi kasus yang disediakan serta empat buah matriks persegi. Data uji yang digunakan semuanya tersimpan dalam folder test, dan semua luaran yang dihasilkan akan disimpan dalam folder testOutput.

4.1 Penyelesaian SPL Berbentuk Ax=b

Dengan mengubah bentuk SPL A**x**=**b** menjadi matriks augmented (A|**b**), program dapat menghitung solusi dari SPL yaitu **x** = (x_1, x_2, x_3, ..., x_n). Solusi dari SPL pada data uji dapat dilihat pada gambar 4.1.

SPL	Penyelesaian
$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$	`SPL tidak dapat diselesaikan dengan metode ini.`
(a.1)	(a.2)

$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$	`x1 = 3.00 + a`
	`x2 = 2.00a`
	`x3 = b`
	`x4 = -1.00 + a`
	`x5 = a`

(b.1)	(b.2)

$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$	`x1 = c`
	`x2 = 1.00 - 1.00a`
	`x3 = b`
	`x4 = -2.00 - 1.00a`
	`x5 = 1.00 + a`
	`x6 = a`

(c.1)	(c.2)

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

dengan n = 6.

(d.1)

```
x1 = 31.66
x2 = -462.96
x3 = 2101.72
x4 = -4111.77
x5 = 3641.33
x6 = -1200.39
```

(d.2)

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

dengan n = 10.

(e.1)

```
x1 = 26.64
x2 = -334.58
x3 = 1156.02
x4 = -1227.91
x5 = -83.29
x6 = 237.78
x7 = 433.44
x8 = -573.57
x9 = 1283.29
x10 = -924.91
```

(e.2)

Gambar 4.1. Penyelesaian SPL dengan menggunakan (a) metode matriks balikan, (b) metode eliminasi Gauss, (c) metode eliminasi Gauss-Jordan, dan (d,e) kaidah Cramer.

4.2 Penyelesaian SPL Berbentuk Matriks Augmented

Misalkan solusi dari SPL yang diberikan adalah x_1, x_2, x_3, ..., x_n, maka solusi dari SPL tersebut dapat dilihat pada gambar 4.2.

SPL

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

(a.1)

Penyelesaian

```
SPL tidak dapat diselesaikan dengan metode ini.
```

(a.2)

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

(b.1)

```
x1 = 0
x2 = 2.00
x3 = 1.00
x4 = 1.00
```

(b.2)

Gambar 4.2. Penyelesaian SPL dengan menggunakan (a) metode matriks balikan dan (b) metode eliminasi Gauss.

4.3 Penyelesaian SPL Berbentuk Persamaan

Dengan mengubah bentuk SPL menjadi matriks augmented, dapat dihitung solusi SPL dengan program utama.

SPL

Penyelesaian

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

(a.1)

```
SPL tidak dapat diselesaikan dengan metode ini.
```

(a.2)

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

(b.1)

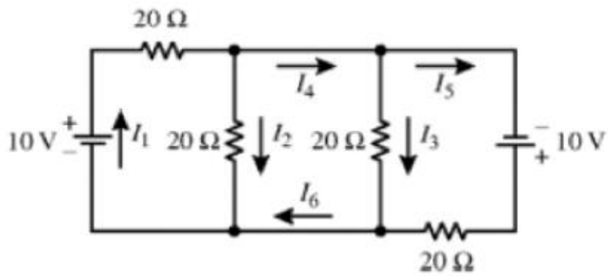
```
SPL tidak memiliki solusi.
```

(b.2)

Gambar 4.3. Penyelesaian SPL dengan menggunakan (a) kaidah Cramer dan (b) metode eliminasi Gauss-Jordan.

4.4 Perhitungan Arus pada Rangkaian Listrik Sederhana

Diberikan suatu rangkaian listrik sederhana yang mengandung sel baterai dan resistor sebagai berikut.



Gambar 4.4.1. Rangkaian listrik

Dengan menggunakan Kirchhoff’s Voltage Law dan Kirchhoff’s Current Law, didapat sistem persamaan sebagai berikut.

$$\begin{aligned} 20I_1 + 20I_2 &= 10 \\ -20I_2 + 20I_3 &= 0 \\ -20I_3 + 20I_5 &= 10 \\ I_1 - I_2 - I_4 &= 0 \\ -I_3 + I_4 - I_5 &= 0 \\ I_3 + I_5 - I_6 &= 0 \end{aligned}$$

Sistem persamaan linier di atas dapat diselesaikan dengan menggunakan kaidah Cramer. Program yang dibuat menunjukkan hasil sebagai berikut, dengan x_1, x_2, ..., x_6 masing-masing bersesuaian dengan I_1, I_2, ..., I_6.

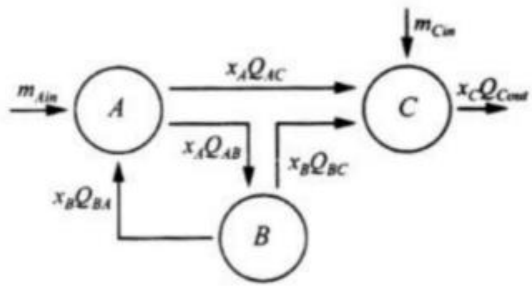
```
x1 = 0.50
x2 = 0
x3 = 0
x4 = 0.50
x5 = 0.50
x6 = 0.50
```

Gambar 4.4.2. Solusi SPL yang terbentuk

Jadi, solusi dari problema rangkaian listrik sederhana tersebut adalah $I_1 = 0.5A, I_2 = 0, I_3 = 0, I_4 = 0.5A, I_5 = 0.5A, I_6 = 0.5A$ dengan A adalah satuan arus listrik ampere.

4.5 Perhitungan Konsentrasi pada Sistem Reaktor

Diberikan suatu diagram sistem reaktor sebagai berikut.



Gambar 4.5.1. Diagram aliran sistem reaktor

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:

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

Gambar 4.5.2. Sistem persamaan linier berdasarkan diagram aliran sistem reaktor

Substitusi $Q_{AB} = 40$, $Q_{AC} = 80$, $Q_{BA} = 60$, $Q_{BC} = 20$ dan $Q_{Cout} = 150$ m^3 /s dan m_{Ain} = 1300 dan m_{Cin} = 200 mg/s pada gambar 4.5.2 menghasilkan SPL sebagai berikut.

$$\begin{gathered} -120x_A + 60x_B = -1300 \\ 40x_A - 80x_B = 0 \\ 80x_A + 20x_B - 150x_C = -200 \end{gathered}$$

Dengan melihat x_1 sebagai x_A, x_2 sebagai x_B, dan x_3 sebagai x_C, didapat penyelesaian sebagai berikut.

```
x1 = 14.44
x2 = 7.22
x3 = 10.00
```

Gambar 4.5.3. Penyelesaian SPL problema sistem reaktor

4.6 Perhitungan Interpolasi

4.6.1 Taksiran Nilai Interpolasi f(x)

Diberikan sekumpulan titik-titik dengan nilai fungsinya sebagai berikut.

x	0.1	0.3	0.5	0.7	0.8	1.1	1.3
f(x)	0.003	0.067	0.148	0.248	0.370	0.518	0.697

Taksiran nilai pada titik-titik x tertentu adalah sebagai berikut.

$$x = 0.2 \rightarrow f(x) = 0.0330$$

```
P(0.200000) = 0.0330
```

$$x = 0.55 \rightarrow f(x) = 0.1711$$

```
P(0.550000) = 0.1711
```

$$x = 0.85 \rightarrow f(x) = 0.3372$$

```
P(0.850000) = 0.3372
```

$$x = 1.28 \rightarrow f(x) = 0.6775$$

```
P(1.280000) = 0.6775
```

4.6.2 Prediksi Jumlah Kasus Baru Covid-19

Diberikan data jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2021 hingga 31 Agustus 2021 sebagai berikut.

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

Gambar 4.6.2.1. Data jumlah kasus baru Covid-19

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)

Dengan menggunakan interpolasi polinom, dapat diprediksi jumlah kasus pada tanggal-tanggal sebagai berikut.

a. Tanggal 16/07/2021
   Tanggal desimal dari 16/07/2021 adalah 7+16/31 = 7.5161, sehingga menurut prediksi, pada tanggal 16/07/2021 ada sebanyak 53562 kasus.

```
P(7.516100) = 53562.1875
```

b. Tanggal 10/08/2021
   Tanggal desimal dari 10/08/2021 adalah 8+10/31 = 8.3226 sehingga menurut prediksi, pada tanggal 10/08/2021 ada sebanyak 36352 kasus.

```
P(8.322600) = 36351.6641
```

c. Tanggal 05/09/2021
   Tanggal desimal dari 05/09/2021adalah 9+5/30 = 9.1667, sehingga menurut prediksi, pada tanggal 05/09/2021 ada sebanyak –665034 kasus atau 0 kasus di kehidupan nyata.

```
P(9.166700) = -665034.2734
```

d. Tanggal 30/06/2021
   Tanggal desimal dari 25/06/2021 adalah 6+30/30 = 7, sehingga menurut prediksi, pada tanggal 30/06/2021ada sebanyak 67725 kasus.

```
P(7.000000) = 21831.2402
```

4.6.3 Penyederhanaan f(x) pada Interval [a,b]

Diberikan sebuah fungsi sebagai berikut.

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

Fungsi $f(x)$ dapat disederhanakan pada interval [0,2] dengan mengambil $(n + 1)$ buah titik pada interval tersebut dengan jarak yang sama dan dilakukan interpolasi polinom berderajat n.

a. Untuk $n = 2$, maka interpolasi polinom berderajat dilakukan dilakukan pada 3 titik sampel, yaitu pada $x \in \{0, 1, 2\}$. Hasil penyederhanaan $f(x)$ adalah sebagai berikut.

```
P(x)= 0.0000+0.7873x-0.2495x^2
```

b. Untuk n=5, maka interpolasi polinom berderajat dilakukan dilakukan pada 6 titik sampel, yaitu pada $x \in \{0, 0.4, 0.8, 1.2, 1.6, 2.0\}$. Hasil penyederhanaan $f(x)$ adalah sebagai berikut.

```
P(x)= 0.0000+2.0350x-3.5528x^2+3.2382x^3-1.4222x^4+0.2365x^5
```

b. Untuk $n = 10$, maka interpolasi polinom berderajat dilakukan dilakukan pada 11 titik sampel, yaitu pada $x \in \{0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2.0\}$. Hasil penyederhanaan $f(x)$ adalah sebagai berikut.

```
P(x)= 0.0000+3.8745x-18.7155x^2+56.7508x^3-109.7103x^4+139.9269x^5-119.2412x^6+67.1667x^7-23.9949x^8+4.9221x^9-0.4413x^10
```

4.7 Perhitungan Regresi Linier Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

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

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

Gambar 4.7.1. Data percobaan

Dengan menggunakan metode *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, didapat persamaan fungsi sebagai berikut.

```
y = -3.5078 - 0.0026x1 + 0.0008x2 + 0.1542x3
```

Gambar 4.7.2. Persamaan regresi linier berganda

Selanjutnya, dengan estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76° F, dan tekanan udara sebesar 29.30, didapat taksiran dari sistem persamaan linier pada gambar 4.7.2 yaitu sebesar 0.938483.

```
y = -3.5078 - 0.0026x1 + 0.0008x2 + 0.1542x3
Masukkan nilai x1 : 50
Masukkan nilai x2 : 76
Masukkan nilai x3 : 29.30
taksiran y = 0.938434
```

Gambar 4.7.3. Taksiran yang didapat dari persamaan.

4.8 Determinan Matriks Persegi

Pengujian fungsionalitas determinan dilakukan pada dua buah macam matriks persegi, masing-masing berukuran 4×4 dan 5×5. Dalam hal ini, satu matriks akan diuji dengan satu buah metode, yaitu salah satu dari metode ekspansi kofaktor dan metode reduksi baris.

Matriks Persegi

$$\begin{pmatrix} 1 & 4 & 5 & 6 \\ 0 & -3 & 8 & 9 \\ 1 & 1 & 8 & -2 \\ 0 & 9 & -9 & 2 \end{pmatrix}$$

(a.1)

Determinan

```
Nilai determinan matriks adalah -330.0000
```

(a.2)

$$\begin{pmatrix} 2.2 & -1.3 & 8 & 9 & 1 \\ -2.1 & 1.1 & -2 & 0 & 3.4 \\ 1 & -1 & 7 & 0.4 & 4.4 \\ -2 & -3.2 & 5.6 & 7 & 1 \\ 7.1 & 3.4 & -6.2 & 2.3 & 9.7 \end{pmatrix}$$

(b.1)

```
Nilai determinan matriks adalah 9031.7250
```

(b.2)

Gambar 4.8. Perhitungan determinan dengan menggunakan (a) metode ekspansi kofaktor dan (b) metode reduksi baris.

4.9 Matriks Balikan dari Matriks Persegi

Pengujian fungsionalitas matriks balikan dilakukan pada dua buah macam matriks persegi, masing-masing berukuran 4×4 dan 5×5. Dalam hal ini, satu matriks akan diuji dengan satu buah metode, yaitu salah satu dari metode adjoin matriks dan metode eliminasi Gauss-Jordan.

Matriks Persegi

$$\begin{pmatrix} 1 & 4 & 5 & 6 \\ 0 & -3 & 8 & 9 \\ 1 & 1 & 8 & -2 \\ 0 & 9 & -9 & 2 \end{pmatrix}$$

(a.1)

Matriks Balikan

```
2.68 -1.80 -1.68 -1.60
-0.29 0.20 0.29 0.28
-0.26 0.20 0.26 0.15
0.14 -0.00 -0.14 -0.05
```

(a.2)

$$\begin{pmatrix} 2.2 & -1.3 & 8 & 9 & 1 \\ -2.1 & 1.1 & -2 & 0 & 3.4 \\ 1 & -1 & 7 & 0.4 & 4.4 \\ -2 & -3.2 & 5.6 & 7 & 1 \\ 7.1 & 3.4 & -6.2 & 2.3 & 9.7 \end{pmatrix}$$

(b.1)

```
-0.02 -0.23 0.02 -0.00 0.07
0.35 0.39 -0.05 -0.42 -0.11
0.09 0.05 0.09 -0.11 -0.06
0.09 0.06 -0.10 0.03 0.01
-0.07 0.05 0.08 0.07 0.05
```

(b.2)

Gambar 4.9. Perhitungan matriks balikan dengan menggunakan (a) metode adjoin matriks dan (b) metode eliminasi Gauss-Jordan.

BAB V

SIMPULAN, SARAN, DAN REFLEKSI

5.1 Simpulan

Melalui tugas besar ini, kami berhasil merumuskan algoritma-algoritma fungsionalitas matriks yang menjadi spesifikasi dari tugas besar ini. Fungsionalitas yang berhasil kami rumuskan diantaranya :

- Penyelesaian SPL menggunakan metode Gauss , Gauss-Jordan , Invers, dan Cramer
- Determinan matriks menggunakan metode reduksi baris dan metode ekspansi kofaktor
- Membentuk matriks balikan (Invers)
- Aplikasi penyelesaian SPL menggunakan matriks *augmented* , seperti Interpolasi Linier dan regresi linier berganda (regresi linier berganda menggunakan metode *Normal Estimation Equation*)
- Fungsi untuk mengambil serta menyimpan hasil dari perhitungan algoritma matriks ke dalam file txt

5.2 Saran

Pada penelitian selanjutnya, fungsionalitas matriks dapat dikembangkan dengan memperhatikan efisiensi algoritma serta ADT (*Abstract Data Type*) matriks bersifat dinamis agar tidak menghabiskan alokasi memori yang banyak. Fungsi-fungsi modular dapat dioptimalkan agar mempermudah fungsi-fungsi utama yang akan digunakan. Pemahaman dalam bahasa pemrograman Java dapat didalami agar dapat merumuskan algoritma seefektif mungkin sehingga memudahkan untuk dibaca oleh manusia.

5.3 Refleksi

Melalui tugas besar ini, kami belajar banyak mengenai bahasa pemrograman Java , yang mana merupakan pertama kalinya kami mengoding dengan bahasa pemrograman ini. Ditambah dengan spesifikasi dari tugas besar ini, kami dapat merumuskan beberapa aplikasi matriks , yaitu seperti Interpolasi Linier dan Regresi Linier Berganda. Ditambah dengan waktu yang terbatas, kami berhasil melatih diri untuk mengerjakan tugas seefisien serta seefektif mungkin sehingga terkejar dengan deadline yang diberikan. Dari tugas besar ini, kami juga dilatih untuk bekerja sama dan berkoordinasi sebagai satu tim sehingga tugas besar ini bisa diselesaikan tepat waktu.

REFERENSI

1. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/
2. https://id.wikipedia.org/wiki/Matriks_(matematika)
3. https://id.wikipedia.org/wiki/Determinan
4. https://stackoverflow.com/questions/22411958/how-to-save-a-file-in-java
5. https://stackoverflow.com/questions/9494327/writing-to-console-with-system-out-and-printwriter
6. https://www.statistikian.com/2018/01/penjelasan-tutorial-regresi-linear-berganda.html
7. https://www.investopedia.com/terms/m/mlr.asp
8. https://www.baeldung.com/java-comparing-doubles
9. https://stackoverflow.com/questions/460364/how-to-use-classes-from-jar-files