LAPORAN TUGAS BESAR

IF2123

ALJABAR LINIER DAN GEOMETRI

SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

DIBUAT OLEH:

HIZKIA RADITYA PRATAMA ROOSADI / 13519087

ANDRIANATA PUTRA TJANDRA / 13519147

RICHARD RIVALDO / 13519185

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2020

BAB I

DESKRIPSI MASALAH

1.1 Abstraksi

Sistem persamaan linier (SPL) Ax = b dengan n peubah (variable) dan m persamaan adalah berbentuk

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

yang dalam hal ini xi adalah peubah, aij dan bi adalah koefisien ∈ R. 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, atau hanya satu (unik/tunggal).

Sebuah matriks M berukuran n × n

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

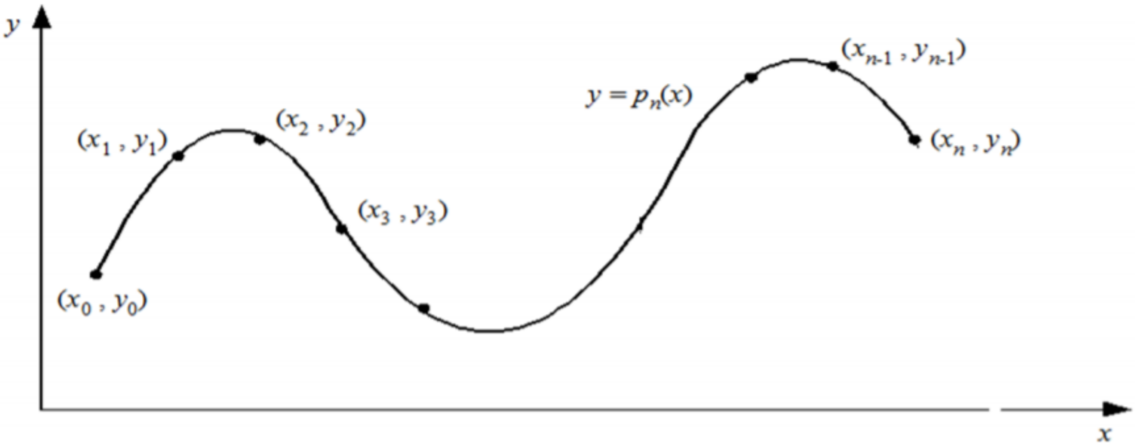
determinannya adalah

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

Determinan matriks M berukuran n × n dapat dihitung dengan beberapa cara: reduksi baris dan ekspansi kofaktor. SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

1.2 Interpolasi Polinom

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x0, y0), (x1, y1), ..., (xn, yn). Tentukan polinom pn(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga yi = pn(xi) untuk i = 0, 1, 2, …, n.



Setelah polinom interpolasi pn(x) ditemukan, pn(x) dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x0, xn].

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x0, y0),(x1, y1), ..., (xn, yn). adalah berbentuk pn(x) = a0 + a1x + a2x 2 + … + anx n . Jika hanya ada dua titik, (x0, y0) dan (x1, y1), maka polinom yang menginterpolasi kedua titik tersebut adalah p1(x) = a0 + a1x yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x0, y0), (x1, y1), dan (x2, y2), maka polinom yang menginterpolasi ketiga titik tersebut adalah p2(x) = a0 + a1x + a2x 2 atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x0, y0), (x1, y1), (x2, y2), dan (x3, y3), polinom yang menginterpolasi keempat titik tersebut adalah p3(x) = a0 + a1x + a2x 2 + a3x 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 (xi, yi) ke

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

dalam persamaan polinom pn(x) = a0 + a1x + a2x 2 + … + anx n untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a0, a1, a2, …, an, Solusi sistem persamaan lanjar ini, yaitu nilai a0, a1, …, an, 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 p2(x) = a0 + a1x + a2x 2 . Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan a0 = 0.6762, a1 = 0.2266, dan a2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah p2(x) = 0.6762 + 0.2266x - 0.0064x 2 . Dengan menggunakan polinom

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$
$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$
$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: p2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

1.3 Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu

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

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

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

1.4 Spesifikasi tugas

Buatlah program dalam **Bahasa Java** untuk

1. Menghitung 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).

2. Menyelesaikan persoalan interpolasi dan regresi linier.

3. Menghitung matriks balikan

4. Menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program adalah sebagai berikut:

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

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

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

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

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

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

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

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

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

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

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

9. Bahasa program yang digunakan adalah Java.

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

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

MENU

1. Sistem Persamaaan Linier

2. Determinan

3. Matriks balikan

4. Interpolasi Polinom

5. Regresi linier berganda

6. Keluar

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

1. Metode eliminasi Gauss

2. Metode eliminasi Gauss-Jordan

3. Metode matriks balikan

4. Kaidah Cramer

Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB II

TEORI SINGKAT

3.1 Eliminasi Gauss

Menurut Anton & Rorres (2013), Eliminasi Gauss merupakan metode penyelesaian sistem persamaan linier dengan cara mengeliminasi setiap elemen di matriks, sampai terbentuk suatu matriks eselon baris. Anton & Rorres (2013) juga menyebutkan bahwa matriks eselon baris merupakan matriks yang memiliki angka satu utama (*leading one*) dalam setiap barisnya, kecuali baris yang seluruh elemennya adalah 0. Jika ada baris dengan seluruh elemennya adalah 0, baris tersebut akan dipindahkan ke baris paling bawah dari matriks.

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

Keterangan: * adalah sembarang nilai

Gambar 1: Contoh Matriks Eselon Baris

Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2018-2019/Matriks-Eselon.pdf

3.2 Eliminasi Gauss-Jordan

Menurut Anton & Rorres (2013), Eliminasi Gauss-Jordan merupakan metode penyelesaian sistem persamaan linier dengan cara mengeliminasi setiap elemen di matriks, sampai terbentuk suatu matriks eselon baris tereduksi. Anton & Rorres (2013) juga menyebutkan bahwa matriks eselon baris tereduksi merupakan matriks yang memiliki angka satu utama (*leading one*) di setiap barisnya dan 0 dibawah dan diatas satu utamanya, kecuali untuk baris yang seluruh elemennya adalah 0. Jika ada baris dengan seluruh elemennya adalah 0, baris tersebut akan dipindahkan ke baris paling bawah dari matriks.

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

Gambar 2: Contoh Matriks Eselon Baris Tereduksi

Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2018-2019/Matriks-Eselon.pdf

3.3 Determinan

Menurut Anton & Rorres (2013), Kata determinan mulai digunakan oleh matematikawan Carl Friedrich Gauss pada tahun 1801 untuk menjelaskan suatu properti dari fungsi tertentu. Dalam konteks matriks, determinan digunakan untuk menjelaskan sebuah angka skalar yang merepresentasikan fungsi matriks itu sendiri. Seperti yang telah disebutkan dalam bab sebelumnya, determinan matriks n × n dapat dihitung dengan beberapa cara yaitu reduksi baris dan ekspansi kofaktor.

3.4 Matriks Balikan

Menurut Rao & Mitra (1972), Untuk matriks tidak singular A, terdapat suatu matriks A^{-1} sedemikian sehingga $AA^{-1} = I$. I yang dimaksud merupakan matriks identitas, yaitu matriks dengan ukuran n x n yang memiliki nilai 1 di seluruh elemen diagonalnya dan 0 di elemen lain. Untuk mencari matriks balikan, metode yang dapat digunakan adalah dengan eliminasi Gauss-Jordan atau ekspansi kofaktor.

3.5 Matriks Kofaktor

Menurut Anton & Rorres (2013), untuk matriks persegi A, kofaktor A_{ij} merupakan hasil determinan dari sub-matriks A setelah eliminasi baris ke i dan kolom ke j dari matriks A. Matriks kofaktor adalah suatu matriks yang memuat entri kofaktor dari matriks A sesuai dengan letak elemennya. Berikut contohnya

Let

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

The minor of entry a_{11} is

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

The cofactor of a_{11} is

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

Gambar 3: Contoh Cara Mencari Entri Kofaktor

Sumber: Anton, H., & Rorres, C. (2013). *Elementary linear algebra: applications version*. John Wiley & Sons.

3.6 Matriks Adjoin

Menurut Anton & Rorres (2013), matriks adjoin adalah hasil transpose dari matriks kofaktor. Menurut Anton & Rorres (2013), transpose adalah operasi pada matriks yang membalik elemen pada matriks berdasar diagonalnya untuk matriks persegi. Untuk matriks tidak persegi, transpose juga menukar ukuran kolom dan baris dari matriks tersebut.

3.7 Kaidah Cramer

Menurut Anton & Rorres (2013), kaidah Cramer adalah suatu aturan untuk menyelesaikan sistem persamaan linier yang dapat direpresentasikan dalam matriks Ax = b. Kaidah ini menyatakan bahwa untuk sistem persamaan linier yang dapat direpresentasikan koefisiennya kedalam suatu matriks A, terdapat solusi unik untuk setiap x yaitu $x_n = \det(A_n) / \det(A)$. Dalam konteks ini, n mengarah kepada kolom yang akan diubah seluruhnya dengan entri jawaban pada spl. Sebagai contoh:

Misal spl berikut

(i) $x + y = 5$
(ii) $2x + 4y = 18$

Maka matriks A= $\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}$ (matriks koefisien spl) dan matriks B= $\begin{bmatrix} 5 \\ 18 \end{bmatrix}$ (matriks entri hasil)

Dari kedua matriks diatas didapatkan matriks A_1= $\begin{bmatrix} 5 & 1 \\ 18 & 4 \end{bmatrix}$ dan A_2= $\begin{bmatrix} 1 & 5 \\ 2 & 18 \end{bmatrix}$ dengan menukar kolom ke n dengan matriks entri hasil

Maka, akan didapatkan x= $\frac{\det(A_1)}{\det(A)}$ dan y= $\frac{\det(A_2)}{\det(A)}$ sesuai ketentuan dari kaidah cramer.

3.8 Interpolasi Polinom

Menurut Anton & Rorres (2013), untuk n banyak titik di bidang xy yang memiliki koordinat x tertentu, terdapat sebuah solusi unik berbentuk polinomial dengan derajat setidaknya n - 1 yang melewati semua titik tersebut. Perhitungan atau pencarian polinomial ini disebut dengan interpolasi polinom. Untuk melakukan interpolasi polinom, dengan asumsi titik-titik sudah diketahui nilainya, hal yang harus dilakukan adalah memasukan semua titik tersebut ke dalam persamaan polinom, yang dapat direpresentasikan dalam bentuk matriks, dan melakukan eliminasi Gauss-Jordan untuk mendapat hasil koefisien dari setiap polinom tersebut. Contoh kita mengetahui 3 titik, maka kita perlu memasukan ketiga titik tersebut ke persamaan dengan derajat n - 1 = 2 yaitu $a_0 + a_1x + a_2x^2 = y$ dan menyelesaikan seluruhnya dari situ.

3.9 Regresi Linier Berganda

Menurut Montgomery, Peck, & Vining (2012), regresi linier berganda adalah pendekatan pencarian suatu nilai tertentu yang menggunakan banyak regressor. Seperti yang kita tahu, regresi linier adalah pendekatan pencarian suatu nilai menggunakan garis linier. Garis linier tersebut disusun berdasar letak titik yang kita masukan untuk dianalisis. Regresi linier berganda menggunakan banyak garis sebagai parameternya. Garis tersebut, karena juga direpresentasikan dalam polinom, dapat dimasukan kedalam suatu matriks yang dapat dicari solusinya menggunakan eliminasi Gauss-Jordan.

BAB III

IMPLEMENTASI PROGRAM

3.1 Primitif Matriks

Dalam implementasi untuk program ini, suatu primitif matriks diperlukan. Matriks dikonstruksi dengan memanfaatkan tipe data *double* yang sudah ada pada java dan membentuk *array* dua dimensi dari tipe data tersebut. Sebagai penyeleksi, nama Matrix diberi kepada *array double* dua dimensi tersebut. Setelah itu, metode untuk mengkonstruksi matriks juga diperlukan. Metode ini dinamakan MakeMatrix. MakeMatrix menerima input banyak baris dan kolom dari pengguna, serta elemen apa saja yang akan berada di dalam matriks tersebut. Selain MakeMatrix, metode lain untuk konstruktor matriks adalah MakeSquareMatrix. Metode tersebut menerima input dimensi dari pengguna dan membuat matriks persegi dengan ukuran sesuai dengan dimensi tersebut, serta elemen apa saja yang ada didalamnya. Metode yang ditulis selanjutnya adalah TulisMatriks. Sesuai dengan namanya, metode ini hanya bertugas untuk menerima matriks dan menuliskannya ke layer menggunakan rekurens. Beberapa metode lain juga dibuat seperti IsBrsValid dan IsKolValid untuk memastikan bahwa jumlah baris dan kolom yang diinput oleh pengguna sesuai (yaitu pada range diatas 0). Metode seperti GetElmt dan SetElmt untuk memanggil elemen spesifik pada matriks atau mengatur elemen tertentu dengan input dari pengguna. Terakhir, beberapa metode yang dibuat dan akan digunakan untuk metode dari subprogram lainnya adalah Transpose dan KaliMatrix. Metode Transpose menerima matriks dan menukar susunan elemen berdasar diagonalnya, pada dasarnya elemen baris menjadi elemen satu kolom dan elemen kolom menjadi elemen satu baris. Selain itu, jika matriks bukan matriks persegi, Transpose juga akan menukar ukuran baris dan kolom dari matriks. Metode KaliMatriks menerima 2 matriks sebagai parameter dan mengoutputkan hasil perkalian dari dua matriks tersebut.

3.2 Eliminasi Gauss

Program akan mulai dengan definisi *class* yang bersifat *public*, untuk eliminasi gauss, digunakan method bernama 'gauss' yang menerima parameter berupa Matrix, yaitu M.

Tahap pertama dari eliminasi gauss ini adalah melakukan OBE sampai elemen diagonal bernilai 1 dan nilai-nilai di bawah garis diagonal bernilai 0. Proses ini dilakukan pada method doGauss.

Mulai dari proses ini, perlu diperiksa apakah baris terakhir memiliki masalah, yaitu jika matriks yang menampung variable memiliki determinan 0.

Kasus pertama jika determinan tidak sama dengan 0, setelah matriks OBE diperoleh, dilakukan *back substitution*, yaitu dimulai dari baris paling terakhir diperoleh solusi baris terakhir, lalu iterasi mundur sampai baris pertama.

Kasus kedua, jika determinan bernilai 0 dan setelah dilakukan OBE, baris paling terakhir pada kolom yang menampung konstanta bernilai tidak sama dengan 0, maka SPL ini tidak memiliki solusi.

Kasus ketiga, jika determinan bernilai 0 dan setelah dilakukan OBE, baris paling terakhir pada kolom yang menampung konstanta bernilai sama dengan 0. Dari sini, akan diperoleh banyak solusi sehingga akan digunakan parameter sebagai solusi.

Untuk program ini, jika matriks tidak memiliki solusi atau solusinya banyak, akan dibuat sebuah array of double *dummy* supaya keluaran tetap berupa array of double.

3.3 Eliminasi Gauss Jordan

Program ini memiliki proses yang hampir sama dengan metode Gauss, yaitu melakukan OBE, namun dengan perbedaan matriks yang dihasilkan adalah matriks identitas untuk matriks yang menampung variable.

Jika determinan bukan 0, maka secara berurutan, solusi dari x_1, x_2, x_3,…,x_N (N adalah jumlah variable) adalah nilai pada kolom terakhir pada baris ke 1, 2, 3,…, N.

3.4 Determinan

Untuk menghitung determinan, program dibagi menjadi dua method, yaitu hitungDeterminanEK untuk metode ekspansi kofaktor dan hitungDeterminanRB untuk metode reduksi baris.

Metode ekspansi kofaktor menggunakan sebuah method lain yaitu minor() untuk mencari matriks minor dari suatu elemen.

Pertama, diinisiasi double det untuk menampung determinannya dan sign untuk kofaktor elemen. Program akan melakukan iterasi dengan variable pada baris pertama dari kolom pertama hingga terakhir. Untuk tiap iterasi, akan diambil matriks minornya dan dihitung determinan dari matriks minor tersebut, dikali elemen ke-i, dan dikali kofaktornya (sign). Jika matriks minor tersebut berukuran lebih dari 2x2, maka untuk menghitung determinan matriks tersebut dipanggil hitungDeterminanEK(), yang berarti method hitungDeterminanEK() adalah method yang rekursif.

Jika iterasi satu baris sudah selesai, nilai tersebut dimasukkan ke dalam det dan sign akan dikali -1.

Hasil yang dikeluarkan adalah nilai dari det.

Untuk metode reduksi baris, digunakan hitungDeterminanRB() yang memanfaatkan method BarBar() untuk melakukan operasi satu baris dengan baris lainnya. Misalnya kita dapat mengurangi baris 1 dengan 2 kali baris 2.

Pertama diinisiasi sebuah double yaitu pivot, untuk menampung nilai elemen pada diagonal pada baris ke i (i adalah variable iterasi)

Matriks akan dilakukan sebuah iterasi dengan variable i. Setiap nilai i bertambah, pivot akan menangkap elemen ke (i, i) sebagai patokan. Lalu dilakukan iterasi variable j untuk iterasi per barisnya untuk mengurangi setiap baris yang bukan baris i dengan pivot dikali baris ke-i dibagi elemen (i, j). Operasi ini dilakukan tiap baris sehingga semua elemen dibawah diagonal matriks akan bernilai 0.

Determinan adalah perkalian dari semua elemen pada diagonal matriks.

3.5 Matriks Balikan Metode Gauss Jordan

Untuk balikan matriks, kedua metode yang digunakan dalam tugas ini adalah metode Gauss-Jordan dan metode adjoin. Kedua metode diletakkan di file .java yang sama yaitu Inverse.java. Metode Gauss-Jordan menggunakan prinsip yang sama seperti yang telah dijelaskan sebelumnya. Awalnya, sebuah matriks dibuat sesuai dengan input dari pengguna. Matriks yang digunakan disini adalah matriks persegi. Di saat yang sama, sebuah matriks identitas juga dibuat berukuran sama dengan matriks tersebut. Matriks identitas adalah matriks dengan angka 1 di elemen diagonalnya dan angka 0 di elemen lain. Setelah itu, metode Gauss-Jordan diterapkan kepada matriks input dari pengguna. Di bagian ini juga, setiap proses yang dilakukan akan dicatat ke sebuah array yang juga sudah di inisiasi sebelumnya. Setelah melakukan eliminasi Gauss-Jordan dan mendapatkan array yang berisi pencatatan proses yang dilakukan, array tersebut kemudian diimplementasikan ke matriks identitas. Cara ini dinamakan *backwards substitution*. Dengan mengimplementasikan proses Gauss-Jordan yang sudah di catat ke matriks identitas, matriks inverse/balikan akan didapat. Program kemudian memanfaatkan prosedur TulisMatriks untuk menulis matriks inverse ke layar. Pengguna dapat memilih untuk menyimpan inverse dari matriks ke suatu file lain atau tidak.

3.6 Matriks Balikan Metode Adjoin

Metode adjoin merupakan metode alternatif untuk menemukan hasil balikan matriks. Metode ini memanfaatkan determinan dari suatu matriks. Jika determinan matriks yang ingin di balik dihitung sebagai 0, akan di*output*kan pesan error. Jika determinan tidak nol, matriks kofaktor dari matriks tersebut akan dicari. Matriks kofaktor dicari menggunakan metode determinan juga, serta metode minor matriks, lebih tepatnya kita mencari determinan dari setiap matriks minor dan meletakkanya di suatu matriks yang berukuran sama dengan matriks awal. Setelah matriks kofaktor didapatkan, langkah selanjutnya adalah memanfaatkan metode transpose untuk mendapatkan adjoin dari matriks. Setelah itu, semua elemen pada matriks akan dikalikan dengan 1/determinan untuk mendapatkan matriks inverse. Terakhir, matriks akan di*output* ke monitor. Pengguna dapat memilih untuk menyimpan inverse dari matriks ke suatu file lain atau tidak.

3.7 Interpolasi

Pada bagian awal, program akan dilakukan *import library* yang dibutuhkan dalam program tersebut. Setelah itu, didefinisikan sebuah *class* bernama Interpolasi yang bersifat *public* sesuai dengan nama file yang dibuat. Di dalam *class* ini atributnya hanya berupa *instances variables* yang terdapat di dalam *methods* tanpa ada *class variables* apapun. Secara umum, variabel-variabel tersebut merupakan variabel yang digunakan untuk menyimpan data input dalam bentuk matriks serta hasil operasi matriks terhadap data yang dimiliki. Kelas ini terdiri dari dua buah *method* dan sebuah *main*.

Bagian *main* di sini akan berperan untuk mengambil input jumlah titik yang akan digunakan untuk *method* interpolasi yang meneriima data dari *keyboard* pengguna. Program akan terus meminta pengguna untuk memasukkan input titik yang bernilai valid. Setelah itu, program akan memanggil *method* tersebut dan *method* tersebut akan berjalan sesuai dengan ketentuan berikut.

Method pertama bernama PolinomInterpolasi yang bersifat *public static* dan bertipe *void*. Bagian ini merupakan bagian yang telah disebutkan sebelumnya, yaitu melakukan operasi interpolasi dengan menggunakan input yang berasal dari *keyboard*. *Method* ini akan menerima sebuah parameter berupa integer yang berisikan jumlah titik yang akan diinput. Setelah itu, program akan membuat sebuah matriks *augmented* berdasarkan besar yang sesuai dengan jumlah titik yang telah diinput sebelumnya dengan menggunakan primitif matriks yang ada pada file Matrix.java. Isi dari matriks tersebut adalah titik-titik yang secara manual dimasukkan oleh pengguna, terurut dari absis dan kemudian ordinat.

Kemudian, program akan meminta pengguna untuk menginput sebuah titik yang ingin diestimasi. Titik ini akan diestimasi dengan menggunakan matriks yang dibuat yang berisi hasil interpolasi yang diselesaikan dengan menggunakan operasi SPL pada matriks *augmented* yang telah dibuat.

Setelah didapatkan sebuah hasil interpolasi, maka akan dicek apakah hasil tersebut bernilai NaN atau tidak. Jika bernilai NaN, maka program akan mengeluarkan pesan kesalahan dan polinom interpolasi tidak terdefinisi. Sebaliknya, program akan menampilkan persamaan polinom interpolasi dalam bentuk $y = a + bx + (-cx^2)$ dan seterusnya sesuai dengan hasil yang didapatkan. Selain itu, juga akan ditampilkan hasil estimasi yang didapat oleh program dengan menggunakan titik yang telah dimasukkan tadi.

Setelah menampilkan output, program akan mengeluarkan *prompt* bagi pengguna untuk menyimpan output yang telah didapat ke dalam file atau tidak. Jika iya, maka file akan disimpan sesuai dengan input nama dan format yang diinput pengguna ke direktori program tersebut. Sebaliknya, jika tidak disimpan, program akan lanjut dan kembali ke bagian *main menu* dari program tersebut. Program akan melemparkan sebuah *exception* jika ada kesalahan saat melakukan hal tersebut.

Method kedua sendiri bernama FileInterpolasi, juga bersifat *public static* dan bertipe *void*. Bagian ini merupakan bagian program yang melaksanakan operasi interpolasi dengan input yang dibaca dari file yang dimasukkan oleh pengguna. Di sini, *method* ini tidak membutuhkan parameter apapun untuk bekerja. Pengguna akan diminta untuk memasukkan nama file beserta dengan formatnya, jumlah kolom dan jumlah baris dari matriks yang ada di dalam tersebut, serta titik yang ingin diestimasi oleh pengguna.

Kemudian, program akan mencoba untuk membaca isi dari file tersebut ke dalam bentuk matriiks dengan menggunakan fungsi BacaFileToMatrix yang ada di main_menu.java. Jika berhasil, maka data tersebut akan disimpan ke dalam variabel matriks dan jika gagal, maka program akan mengeluarkan sebuah pesan *exception*.

Segmen-segmen program selanjutnya sama dengan segmen program yang ada di *method* PolinomInterpolasi, yaitu melakukan kalkulasi hasil, pengecekan isNaN, penampilan output sesuai hasil pengecekkan tersebut, dan bagian program untuk menyimpan output perhitungan tersebut ke dalam sebuah file baru.

Catatan:

Matriks yang ada di dalam input file yang diberikan berbentuk matriks *augmented* yang terdiri dari titik-titik x dan y yang akan dikalkulasi.

3.8 Regresi Linier Ganda

Sama dengan program Interpolasi.java sebelumnya, program RegresiGanda.java ini diawali dengan melakukan *import library* yang digunakan dalam pembuatan program tersebut. Setelah itu, maka akan didefinisikan sebuah kelas dengan sifat *public* yang bernama sama dengan nama file program, yaitu RegresiGanda. Dalam kelas ini juga tidak ada *class variables*, dan *instance variables* hanya berupa variabel untuk menampung input serta menyimpan hasil operasi matriks yang dilakukan. Sama seperti *class* Interpolasi, kelas ini juga mengandung dua buah *method* dan satu buah *main*.

Main dalam program ini akan meminta pengguna untuk memasukkan dua buah integer, yaitu jumlah data dan jumlah variabel independen dari data yang akan dimasukkan dan dikalkulasi. Jika salah satu atau kedua data yang dimasukkan tidak valid, maka akan dikeluarkan pesan yang bersesuaian dengannya. Jika tidak, maka program akan memanggil *method* untuk mencari persamaan regresi selanjutnya.

Method pertama memiliki sifat *public static* dan tidak mengembalikan nilai apapun, diberi nama MultiRegresi. *Method* ini akan melakukan operasi untuk mencari persamaan regresi linier dari input data yang dimasukkan pengguna melalui *keyboard* yang telah disebutkan di atas. Kedua input yang dimasukkan di program *main* itu juga akan digunakan sebagai parameter dari *method* ini.

Bagian pertama dari *method* ini adalah meminta input dari pengguna untuk memasukkan data berupa titik-titik absis dan titik-titik ordinat sebagai data variabel independen dan variabel dependen. Kedua input ini akan disimpan ke dalam variabel bertipe matriks yang sebelumnya telah dideklarasikan. Setelah itu, program akan masuk ke dalam bagian kalkulasi persamaan regresi yang didapat dari data tersebut.

Bagian kalkulasi ini sendiri tidak menggunakan SPL, melainkan menggunakan beberapa matriks perantara, yaitu matriks transpose, matriks invers, dan matriks hasil perkalian dari kedua komponen sebelumnya. Dengan mendeklarasikan sebuah matriks hasil untuk menyimpan hasil dari kalkulasi tersebut, maka di sini program menggunakan operasi $(X^TX)^{-1}(X^TY)$ untuk mendapatkan hasil tersebut.

Pengguna akan kemudian diminta untuk melakukan input data regresi satu persatu untuk tiap variabel independen yang akan diestimasi dengan persamaan regresi yang nantinya didapat. Input ini juga akan dimasukkan ke dalam matriks dengan ukuran yang disesuaikan dengan faktor-faktor sebelumnya, terutama jumlah variabel independen. Setelah input tersebut didapatkan, dengan menggunakan perkalian matriks kita bisa mendapatkan hasil regresi data tersebut dengan mengalikan matriks data regresi dengan matriks hasil yang sebelumnya telah didapatkan.

Setelah melakukan pengecekkan isNaN terhadap matriks yang dihasilkan oleh kalkulasi tersebut, program kemudian akan mengeluarkan output yang bersesuaian dengan hasil pengecekkan tersebut. Jika valid atau bukan NaN, maka output akan berupa persamaan regresi linier yang telah dikalkulasi dalam bentuk y = a + bx1 + (-cx2) dan seterusnya. Selain itu, program juga akan mengeluarkan output nilai regresi yang didapat dari perhitungan data regresi dan persamaan regresi tersebut. Jika tidak, maka akan diberikan pesan kesalahan dan data regresi tidak akan bisa dihitung.

Sama dengan file sebelumnya, pada *method* ini pengguna juga akan diberikan *prompt* jika ingin menyimpan output tersebut ke dalam file. Jika ya, pengguna akan dimintai nama file beserta formatnya dan mengeluarkan pesan berhasil jika output telah berhasil disalin ke dalam file tersebut. Jika tidak, maka program akan lanjut ke bagian *main menu*. Jika terjadi kesalahan dalam proses ini, program akan memberikan sebuah *exception* dan pesan kesalahan yang bersesuaian.

Method kedua juga bersifat *public static* dan bertipe *void*, yaitu FileRegresi. Tidak seperti *method* sebelumnya, *method* ini tidak menerima parameter apapun agar bisa berjalan. Awalnya, pengguna akan diminta untuk memasukkan nama file, jumlah baris dan kolom dari matriks augmented yang ada di dalam file tersebut, serta data yang ingin diregresikan secara satu persatu untuk tiap variabel independen.

Setelah itu, bagian program yang selanjutnya juga terdiri dari segmen-segmen yang sama dengan bagian program dari *method* pertama ketika telah mendapatkan input data regresi. Segmen tersebut berupa kalkulasi persamaan regresi linier serta pembentukan matriks-matriksnya, pengecekkan isNaN untuk hasil tersebut, serta pengeluaran output sesuai dengan hasil pengecekkan yang didapat. Pengguna juga akan diberi *prompt* untuk memilih menyimpan output ke file baru atau tidak.

Catatan:

Matriks yang ada di dalam input file berupa matriks augmented yang berisi nilai variabel-variabel independen dan variabel dependennya.

BAB IV

EKSPERIMEN

4.1 Solusi SPL

1. Studi Kasus 1A

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 1
Sub Menu 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
5. Kembali
Silakan input angka 1-5 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 2
```

```
Masukkan pilihan (1-2): 2
Masukan jumlah baris: 4
Masukan jumlah kolom: 5
Masukan elemen ke [0][0]: 1
Masukan elemen ke [0][1]: 1
Masukan elemen ke [0][2]: -1
Masukan elemen ke [0][3]: -1
Masukan elemen ke [0][4]: 1
Masukan elemen ke [1][0]: 2
Masukan elemen ke [1][1]: 5
Masukan elemen ke [1][2]: -7
Masukan elemen ke [1][3]: -5
Masukan elemen ke [1][4]: -2
Masukan elemen ke [2][0]: 2
Masukan elemen ke [2][1]: -1
Masukan elemen ke [2][2]: 1
Masukan elemen ke [2][3]: 3
Masukan elemen ke [2][4]: 4
Masukan elemen ke [3][0]: 5
Masukan elemen ke [3][1]: 2
Masukan elemen ke [3][2]: -4
Masukan elemen ke [3][3]: 2
Masukan elemen ke [3][4]: 6
```

```
Berikut matrix anda:
1.0  1.0  -1.0  -1.0  1.0

2.0  5.0  -7.0  -5.0  -2.0

2.0  -1.0  1.0  3.0  4.0

5.0  2.0  -4.0  2.0  6.0

Matriks tidak memiliki solusi atau multi solusi
Nilai x1 adalah: 0.0
Apakah anda ingin save jawaban? (Y/N):
```

Program dimulai dengan main menu. Pada main menu, dipilih menu 1 untuk menyelesaikan SPL. Untuk eksperimen ini, digunakan metode Kaidah Cramer dan input diambil dari hasil ketik pengguna. Input yang dimasukkan berbentuk matriks augmented sehingga jumlah kolom 1 lebih dari jumlah baris.

Dari hasil perhitungan didapat bahwa SPL ini tidak memiliki solusi unik yang berarti tidak memiliki solusi atau banyak solusi. Sebagai *placeholder*, dikeluarkan sebuah array double yang bernilai 0.

2. **Studi Kasus 1B**

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 1
Sub Menu 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
5. Kembali
Silakan input angka 1-5 sesuai pilihan: 1
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 2
Masukan jumlah baris: 5
Masukan jumlah kolom: 6
Masukan elemen ke [0][0]: 1
Masukan elemen ke [0][1]: -1
Masukan elemen ke [0][2]: 0
Masukan elemen ke [0][3]: 0
Masukan elemen ke [0][4]: 1
Masukan elemen ke [0][5]: 3
Masukan elemen ke [1][0]: 1
Masukan elemen ke [1][1]: 1
Masukan elemen ke [1][2]: 0
```

```
Masukan elemen ke [1][1]: 1
Masukan elemen ke [1][2]: 0
Masukan elemen ke [1][3]: -3
Masukan elemen ke [1][4]: 0
Masukan elemen ke [1][5]: 6
Masukan elemen ke [2][0]: 2
Masukan elemen ke [2][1]: -1
Masukan elemen ke [2][2]: 0
Masukan elemen ke [2][3]: 1
Masukan elemen ke [2][4]: -1
Masukan elemen ke [2][5]: 5
Masukan elemen ke [3][0]: -1
Masukan elemen ke [3][1]: 2
Masukan elemen ke [3][2]: 0
Masukan elemen ke [3][3]: -2
Masukan elemen ke [3][4]: -1
Masukan elemen ke [3][5]: -1
Masukan elemen ke [4][0]: 0
Masukan elemen ke [4][1]: 0
Masukan elemen ke [4][2]: 0
Masukan elemen ke [4][3]: 0
Masukan elemen ke [4][4]: 0
Masukan elemen ke [4][5]: 0
Berikut matrix anda:
1.0  -1.0  0.0  0.0  1.0  3.0

1.0  1.0  0.0  -3.0  0.0  6.0

2.0  -1.0  0.0  1.0  -1.0  5.0

-1.0  2.0  0.0  -2.0  -1.0  -1.0

0.0  0.0  0.0  0.0  0.0  0.0

Matrix memiliki banyak solusi
Nilai x1 adalah: 0.0
Apakah anda ingin save jawaban? (Y/N):
```

Program dimulai dengan menjalankan main_menu.java. Dari main menu, dipilih nomor 1 untuk menyelesaikan SPL lalu menggunakan metode Gauss. Input dilakukan dari keyboard per elemennya sebagai sebuah matriks augmented. Karena matriks yang menampung variable tidak persegi, maka dianggap ada sebuah persamaan baru lagi yang setiap konstantanya bernilai 0, yaitu yang diinpt pada baris terakhir.

Dari hasil dikeluarkan, matriks memiliki banyak solusi dan tidak memiliki solusi tunggal/unik.

3. **Studi Kasus 1C**

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 1
Sub Menu 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
5. Kembali
Silakan input angka 1-5 sesuai pilihan: 2
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 2
Masukan jumlah baris: 6
Masukan jumlah kolom: 7
Masukan elemen ke [0][0]: 0
Masukan elemen ke [0][1]: 1
Masukan elemen ke [0][2]: 0
Masukan elemen ke [0][3]: 0
Masukan elemen ke [0][4]: 1
```

```
Masukan elemen ke [5][6]: 0
Berikut matrix anda:
0.0  1.0  0.0  0.0  1.0  0.0  2.0

0.0  0.0  0.0  1.0  1.0  0.0  -1.0

0.0  1.0  0.0  0.0  0.0  1.0  1.0

0.0  0.0  0.0  0.0  0.0  0.0  0.0

0.0  0.0  0.0  0.0  0.0  0.0  0.0

0.0  0.0  0.0  0.0  0.0  0.0  0.0

Matriks multi solusi
Nilai x1 adalah: 0.0
Apakah anda ingin save jawaban? (Y/N):
```

Seperti studi kasus sebelumnya, pada studi kasus ini digunakan juga menu 1, yaitu penyelesaian SPL namun dengan metode Gauss Jordan.

Karena matriks yang menampung variable bukan matriks persegi, maka ditambahkan beberapa persamaan yang setiap konstantanya bernilai 0. Hasil yang didapat adalah matriks memiliki banyak solusi.

4. **Studi Kasus 1D**

Untuk matriks Hilbert, digunakan sebuah method khusus untuk menentukan matriks Hilbert dengan ukuran N seperti di bawah ini. Khusus untuk kasus ini, eksperimen dilakukan pada file Hilbert.java.

```
public static Matrix Hilbert(int a){
    Matrix M = new Matrix(a,a+1);
    int counter = 0;
    Matrix.SetElmt(M, 0, 0, 1);
    for (int i = 0; i < a; i++) {
        for (int j = 0; j < a+1; j++) {
            double diva = j+1+counter;
            double d = 1/diva;
            if (j == a){
                Matrix.SetElmt(M, i, j, 0);
            }
            else{
                Matrix.SetElmt(M, i, j, d);
            }

        }
        counter++;
    }
    Matrix.SetElmt(M, 0, a, 1);
    return M;
}
```

Matriks Hilbert tersebut kemudian di selesaikan dengan metode Gauss Jordan sehingga didapatkan hasil pada gambar. Solusi yang didapat adalah unik.

```
Masukkan nilai n untuk Hilbert: 6
Berikut matrix anda:
1.0  0.5  0.3333333333333333  0.25  0.2  0.16666666666666666  1.0

0.5  0.3333333333333333  0.25  0.2  0.16666666666666666  0.14285714285714285  0.0

0.3333333333333333  0.25  0.2  0.16666666666666666  0.14285714285714285  0.125  0.0

0.25  0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.0

0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.0

0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.0

Nilai x1 adalah: 36.000000000980656
Nilai x2 adalah: -630.0000000292673
Nilai x3 adalah: 3360.000000203484
Nilai x4 adalah: -7560.000000539232
Nilai x5 adalah: 7560.000000603351
Nilai x6 adalah: -2772.000000240222
PS C:\Users\Asus\Algeo01-19087>
```

```
Masukkan nilai n untuk Hilbert: 10
Berikut matrix anda:
1.0  0.5  0.3333333333333333  0.25  0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  1.0

0.5  0.3333333333333333  0.25  0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.0

0.3333333333333333  0.25  0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.0

0.25  0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.0

0.2  0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.07142857142857142  0.0

0.16666666666666666  0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.07142857142857142  0.06666666666666667  0.


0.14285714285714285  0.125  0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.07142857142857142  0.06666666666666667  0.0625  0.0

0.125  0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.07142857142857142  0.06666666666666667  0.0625  0.058823529411764705  0.0

0.1111111111111111  0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.07142857142857142  0.06666666666666667  0.0625  0.058823529411764705  0.05555555555555555
.0

0.1  0.09090909090909091  0.08333333333333333  0.07692307692307693  0.07142857142857142  0.06666666666666667  0.0625  0.058823529411764705  0.05555555555555555  0.05263157894736842
0.0

Nilai x1 adalah: 99.99634656426257
Nilai x2 adalah: -4949.682659994306
Nilai x3 adalah: 79193.21483186251
Nilai x4 adalah: -600538.1369470842
Nilai x5 adalah: 2522224.25868042
```

```
Nilai x1 adalah: 99.99634656426257
Nilai x2 adalah: -4949.682659994306
Nilai x3 adalah: 79193.21483186251
Nilai x4 adalah: -600538.1369470842
Nilai x5 adalah: 2522224.25868042
Nilai x6 adalah: -6305485.547186624
Nilai x7 adalah: 9608261.78322886
Nilai x8 adalah: -8750305.214203862
Nilai x9 adalah: 4375119.565502384
Nilai x10 adalah: -923630.2349573893
```

5. Studi Kasus 2A

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 1
Sub Menu 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
5. Kembali
Silakan input angka 1-5 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 2
Masukan jumlah baris: 4
Masukan jumlah kolom: 5
Masukan elemen ke [0][0]: 1
Masukan elemen ke [0][1]: -1
Masukan elemen ke [0][2]: 2
Masukan elemen ke [0][3]: -1
Masukan elemen ke [0][4]: -1
Masukan elemen ke [1][0]: 2
```

```
Berikut matrix anda:
1.0  -1.0  2.0  -1.0  -1.0

2.0  1.0  -2.0  -2.0  -2.0

-1.0  2.0  -4.0  1.0  1.0

3.0  0.0  0.0  -3.0  -3.0

Matriks tidak ada solusi atau multi solusi
Nilai x1 adalah: 0.0
Apakah anda ingin save jawaban? (Y/N):
```

Untuk studi kasus 2A, diberikan matriks dalam bentuk augmented berbentuk 4x5 dengan elemen yang terdapat pada gambar kedua. Metode yang digunakan adalah kaidah cramer dan hasil yang diperoleh dari SPL tersebut adalah ada banyak solusi yang bisa didapatkan dari SPL tersebut.

6. Studi Kasus 2B

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 1
Sub Menu 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
5. Kembali
Silakan input angka 1-5 sesuai pilihan: 2
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 2
Masukan jumlah baris: 6
Masukan jumlah kolom: 5
Masukan elemen ke [0][0]: 2
Masukan elemen ke [0][1]: 0
Masukan elemen ke [0][2]: 8
Masukan elemen ke [0][3]: 0
Masukan elemen ke [0][4]: 8
Masukan elemen ke [1][0]: 0
Masukan elemen ke [1][1]: 1
Masukan elemen ke [1][2]: 0
```

```
Masukan elemen ke [5][0]: 0
Masukan elemen ke [5][1]: 1
Masukan elemen ke [5][2]: 0
Masukan elemen ke [5][3]: -2
Masukan elemen ke [5][4]: 0
Berikut matrix anda:
2.0  0.0  8.0  0.0  8.0

0.0  1.0  0.0  4.0  6.0

-4.0  0.0  6.0  0.0  6.0

0.0  -2.0  0.0  3.0  -1.0

2.0  0.0  -4.0  0.0  -4.0

0.0  1.0  0.0  -2.0  0.0

Matriks tidak ada solusi atau multi solusi
Nilai x1 adalah: 0.0
Apakah anda ingin save jawaban? (Y/N):
```

Untuk studi kasus ini, digunakan matriks augmented dengan ukuran 6x5 seperti yang ditunjukkan oleh gambar kedua. Dengan metode Gauss-Jordan, didapat bahwa matriks tidak memiliki solusi atau memiliki banyak solusi.

Hal ini dikarenakan jumlah baris yang lebih banyak 2 atau lebih daripada kolom sehingga solusi tidak mungkin unik.

7. Studi Kasus 3A

Berikut studi kasus 3 soal A.

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

Berikut hasil pengujian dari program

```
Berikut matrix anda:
8.0  1.0  3.0  2.0  0.0

2.0  9.0  -1.0  -2.0  1.0

1.0  3.0  2.0  -1.0  2.0

1.0  0.0  6.0  4.0  3.0

Nilai x1 adalah: -0.2243243243243243
Nilai x2 adalah: 0.18243243243243246
Nilai x3 adalah: 0.7094594594594594
Nilai x4 adalah: -0.2581081081081797
Apakah anda ingin save jawaban? (Y/N):
```

Berikut hasil uji dari kalkulator online sebagai pembanding

solve

$$8a + x + 3y + 2z = 0$$
$$2a + 9x - y - 2z = 1$$
$$a + 3x + 2y - z = 2$$
$$a + 6y + 4z = 3$$

Result: App Enlarge Cust

$$x = \frac{27}{148} \text{ and } y = \frac{105}{148} \text{ and } z = -\frac{191}{740} \text{ and } a = -\frac{83}{370}$$

catatan: a=x1,x=x2,y=x3,z=x4

setelah perhitungan ke bentuk desimal, didapatkan

a=x1=-0.2243243243

x=x2=0.1824324324

y=x3=0.7094594595

z=x4=-0.2581081081

analisis:

seperti yang dapat dilihat dari kedua data diatas, data hasil percobaan menggunakan program dan menggunakan kalkulator online tidak jauh berbeda dan sangat mendekati. Dari percobaan ini dapat disimpulkan program berjalan sesuai dengan input pengguna.

Metode yang digunakan diatas adalah metode SPL dengan eliminasi Gauss. Perlu diketahui juga bahwa untuk metode laiinya juga menghasilkan data yang serupa seperti yang ditunjukkan sebagai berikut.

Metode eliminasi Gauss-Jordan

```
Berikut matrix anda:
8.0  1.0  3.0  2.0  0.0

2.0  9.0  -1.0  -2.0  1.0

1.0  3.0  2.0  -1.0  2.0

1.0  0.0  6.0  4.0  3.0

Nilai x1 adalah: -0.22432432432432436
Nilai x2 adalah: 0.18243243243243246
Nilai x3 adalah: 0.7094594594594594
Nilai x4 adalah: -0.25810810810810797
Apakah anda ingin save jawaban? (Y/N):
```

Metode cramer

```
8.0  1.0  3.0  2.0  0.0

2.0  9.0  -1.0  -2.0  1.0

1.0  3.0  2.0  -1.0  2.0

1.0  0.0  6.0  4.0  3.0

Nilai x1 adalah: -0.22432432432432434
Nilai x2 adalah: 0.1824324324324324
Nilai x3 adalah: 0.7094594594594594
Nilai x4 adalah: -0.258108108108108
Apakah anda ingin save jawaban? (Y/N):
```

Metode matriks balikan

```
Berikut matrix anda:
8.0  1.0  3.0  2.0  0.0

2.0  9.0  -1.0  -2.0  1.0

1.0  3.0  2.0  -1.0  2.0

1.0  0.0  6.0  4.0  3.0

Nilai x1 adalah: -0.22432432432432434
Nilai x2 adalah: 0.18243243243243243
Nilai x3 adalah: 0.7094594594594594
Nilai x4 adalah: -0.2581081081081081
Apakah anda ingin save jawaban? (Y/N):
```

8. Studi Kasus 3B

Berikut studi kasus 3 soal B

b.

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

Berikut adalah hasil pengujian dari program dengan metode eliminasi Gauss-Jordan

```
0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  13.0

0.0  0.0  0.0  1.0  1.0  1.0  0.0  0.0  0.0  15.0

0.0  0.0  0.04289  0.0  0.04289  0.75  0.04289  0.75  0.61396  14.79

0.0  0.25  0.91421  0.25  0.91421  0.25  0.91421  0.25  0.0  14.31

0.61396  0.75  0.04289  0.75  0.04289  0.0  0.04289  0.0  0.0  3.81

0.0  0.0  1.0  0.0  0.0  0.0  1.0  0.0  1.0  18.0

0.0  1.0  0.0  0.0  1.0  0.0  0.0  1.0  0.0  12.0

1.0  0.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0  6.0

0.04289  0.75  0.61396  0.0  0.04289  0.75  0.0  0.0  0.04289  10.51

0.91421  0.25  0.0  0.25  0.91421  0.25  0.0  0.25  0.91421  16.13

0.04289  0.0  0.0  0.75  0.04289  0.0  0.61396  0.75  0.04289  7.04

Matriks tidak ada solusi atau multi solusi
Nilai x1 adalah: 0.0
```

Berikut adalah hasil dari kalkulator online sebagai pembanding

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	b
1	1	0	0	0	0	0	0	0	0	7.76110843529691822
2	0	1	0	0	0	0	0	0	0	9.443483581128714590 4
3	0	0	1	0	0	0	0	0	0	-9.34605463367848615
4	0	0	0	1	0	0	0	0	0	-11.718385921846581806
5	0	0	0	0	1	0	0	0	0	16.90257105254977155 8
6	0	0	0	0	0	1	0	0	0	9.815814869296810249 7
7	0	0	0	0	0	0	1	0	0	9.957277486549663587 3
8	0	0	0	0	0	0	0	1	0	-14.346054633678486149
9	0	0	0	0	0	0	0	0	1	17.38877714712882256 2
10	0	0	0	0	0	0	0	0	0	-20.613490853986077358
11	0	0	0	0	0	0	0	0	0	18.67133046592766978 7

Hide solution Recalculate

Solution set:

The system is inconsistent

Namun, setelah percobaan lebih lanjut dengan metode lain, hasil program adalah sebagai berikut

```
0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  13.0

0.0  0.0  0.0  1.0  1.0  1.0  0.0  0.0  0.0  15.0

0.0  0.0  0.04289  0.0  0.04289  0.75  0.04289  0.75  0.61396  14.79

0.0  0.25  0.91421  0.25  0.91421  0.25  0.91421  0.25  0.0  14.31

0.61396  0.75  0.04289  0.75  0.04289  0.0  0.04289  0.0  0.0  3.81

0.0  0.0  1.0  0.0  0.0  0.0  1.0  0.0  1.0  18.0

0.0  1.0  0.0  0.0  1.0  0.0  0.0  1.0  0.0  12.0

1.0  0.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0  6.0

0.04289  0.75  0.61396  0.0  0.04289  0.75  0.0  0.0  0.04289  10.51

0.91421  0.25  0.0  0.25  0.91421  0.25  0.0  0.25  0.91421  16.13

0.04289  0.0  0.0  0.75  0.04289  0.0  0.61396  0.75  0.04289  7.04

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 10
        at Matrix.GetElmt(Matrix.java:107)
        at SPL.IsOK(SPL.java:83)
        at SPL.IsNotOK(SPL.java:90)
        at SPL.gauss(SPL.java:96)
        at main_menu.main(main_menu.java:105)
```

Analisis:

Berdasar hasil dari program yang dijalankan, terutama pada metode eliminasi Gauss-Jordan, terlihat bahwa untuk kasus ini tidak memiliki solusi. Hal ini sesuai dengan hasil dari kalkulator online. Perlu dicatat bahwa untuk metode selain eliminasi Gauss-jordan, program akan menghasilkan error out of bounds yang disebabkan oleh tidak adanya solusi dalam matriks ini. Hal ini disebabkan indeks yang seharusnya di outputkan oleh program adalah x1 sampai x9 dan pada kasus ini memang tidak terbentuk hasil tersebut.

Catatan:
Dengan adanya error ini, program masih disimpulkan belum sempurna dan memerlukan perkembangan lebih lanjut agar dapat beroperasi dengan lebih baik dan optimal.

9. Studi Kasus 4

Berikut adalah soal studi kasus 4

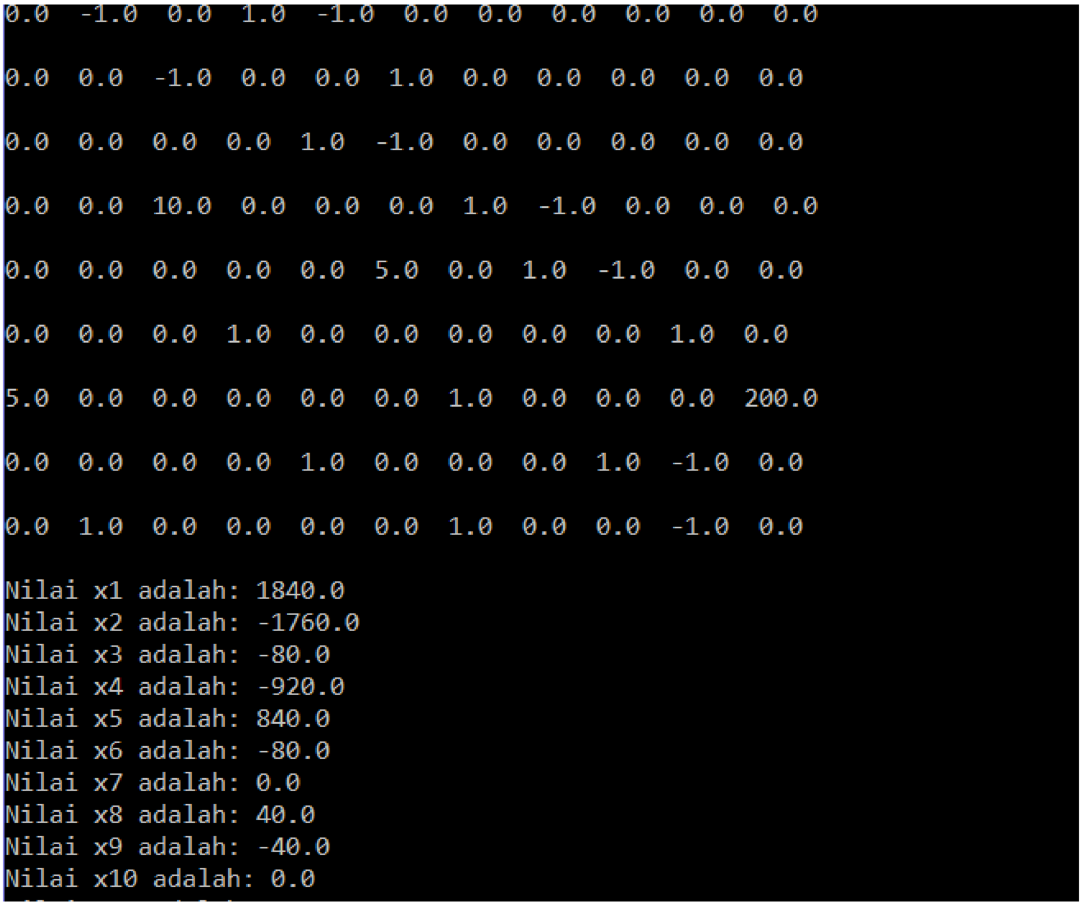
Dengan menyusun kesepuluh persamaan diatas didapatkan sistem persamaan linier sebagai berikut :

$$
\begin{array}{llllllllll}
i_{12} & + i_{52} & + i_{32} & & & & & & & = 0 \\
& - i_{52} & & + i_{65} & - i_{54} & & & & & = 0 \\
& & - i_{32} & & & + i_{43} & & & & = 0 \\
& & & & i_{54} & - i_{43} & & & & = 0 \\
& & i_{32} R_{32} & & & & V_2 & - V_3 & & = 0 \\
& & & & & i_{43} R_{43} & & + V_3 & - V_4 & = 0 \\
& & & i_{65} R_{65} & & & & & + V_5 & = V_6 \\
i_{12} R_{12} & & & & & & + V_2 & & & = V_1 \\
& & & & i_{54} R_{54} & & & & + V_4 \quad - V_5 & = 0 \\
& i_{52} R_{52} & & & & & + V_2 & & - V_5 & = 0
\end{array}
$$

Tentukan nilai dari:

$$i_{12}\ ,\quad i_{52}\ ,\quad i_{32}\ ,\quad i_{65}\ ,\quad i_{54}\ ,\quad i_{13}\ ,\quad V_2\ ,\quad V_3\ ,\quad V_4\ ,\quad V_5$$

Berikut hasil percobaan dari program



```
0.0  -1.0  0.0  1.0  -1.0  0.0  0.0  0.0  0.0  0.0  0.0

0.0  0.0  -1.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0

0.0  0.0  0.0  0.0  1.0  -1.0  0.0  0.0  0.0  0.0  0.0

0.0  0.0  10.0  0.0  0.0  0.0  1.0  -1.0  0.0  0.0  0.0

0.0  0.0  0.0  0.0  0.0  5.0  0.0  1.0  -1.0  0.0  0.0

0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0

5.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  200.0

0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0  -1.0  0.0

0.0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  -1.0  0.0

Nilai x1 adalah: 1840.0
Nilai x2 adalah: -1760.0
Nilai x3 adalah: -80.0
Nilai x4 adalah: -920.0
Nilai x5 adalah: 840.0
Nilai x6 adalah: -80.0
Nilai x7 adalah: 0.0
Nilai x8 adalah: 40.0
Nilai x9 adalah: -40.0
Nilai x10 adalah: 0.0
```

Analisis:

Perlu diketahui, berdasar perhitungan program, x1=i12=1840 A, x2=i52=-1760 A,x3=i32=-80 A, x4=i65=-920 A, x5=i54=840 A, x6=i13=-80 A, x7=V2=0 volt, x8= V3=40 V, x9=V4=-40 V,x10=V5=0 V. Perhitungan ini belum tentu 100% benar dan masih perlu perhitungan lebih lanjut. Namun, dalam sisi program, terlihat bahwa program berjalan cukup hingga dapat mengeluarkan output solusi dari SPL. Untuk kedepannya, masih perlu dikembangkan lagi dan dicek mengenai kasus seperti ini.

4.2 Interpolasi

1. Studi Kasus 5

```
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text4.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 7
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 0.2
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P6(x) = -0.0230 + 0.2400x + 0.1974x^2 + 0.0000x^3 + 0.0260x^4 + 0.0000x^5 + (-0.0000x^6)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P6(0.2000) = 0.0330
Apakah anda ingin save jawaban? (Y/N): n
```

```
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text4.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 7
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 0.55
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P6(x) = -0.0230 + 0.2400x + 0.1974x^2 + 0.0000x^3 + 0.0260x^4 + 0.0000x^5 + (-0.0000x^6)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P6(0.5500) = 0.1711
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text4.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 7
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 0.85
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P6(x) = -0.0230 + 0.2400x + 0.1974x^2 + 0.0000x^3 + 0.0260x^4 + 0.0000x^5 + (-0.0000x^6)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P6(0.8500) = 0.3372
```

```
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text4.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 7
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 1.28
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P6(x) = -0.0230 + 0.2400x + 0.1974x^2 + 0.0000x^3 + 0.0260x^4 + 0.0000x^5 + (-0.0000x^6)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P6(1.2800) = 0.6775
Apakah anda ingin save jawaban? (Y/N): y
Masukan nama file: jawabaninterpolasi
Data berhasil disimpan!
```

```
jawabaninterpolasi - Notepad
File  Edit  Format  View  Help
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P6(x) = -0.0230 + 0.2400x + 0.1974x^2 + 0.0000x^3 + 0.0260x^4 + 0.0000x^5 + (-0.0000x^6)
P6(1.2800) = 0.6775
```

Eksperimen ini merupakan percobaan memasukkan data studi kasus interpolasi dalam bentuk matriks yang terdapat di dalam file *text4.txt*. Input data yang perlu diberikan oleh pengguna adalah nama file beserta dengan formatnya, jumlah baris dan kolom dari matriks yang ada di dalam file tersebut, serta langsung memasukkan titik yang ingin diestimasi nilai interpolasinya. Setelah memasukkan data-data yang dibutuhkan untuk melakukan kalkulasi tersebut, maka program akan menampilkan persamaan polinom interpolasi yang dihasilkan dari kalkulasi data-data yang ada di dalam matriks. Selain itu, program juga akan mengeluarkan data hasil estimasi dari titik yang dimasukkan sebelumnya.

Pada akhir program, program akan menanyakan pengguna untuk menyimpan output jawaban yang telah didapat pada suatu file ataupun tidak. Dari gambar 1-3, pengguna menjawab ‘N’ atau ‘n’ untuk menolak menyimpan output tersebut. Pada gambar keempat, pengguna memilih untuk menyimpan luaran tersebut ke dalam file bernama *jawabaninterpolasi*. Di sini, pengguna tidak menuliskan format dari file yang akan disimpan tersebut. Namun demikian, program masih bisa dibuka dalam bentuk *.txt* melalui Notepad ataupun *browser*. Bisa terlihat bahwa file tersebut akan berisi persamaan dan nilai hasil estimasi titik dengan polinom interpolasi yang telah didapatkan sebelumnya. Data yang ditampilkan dalam matriks tersebut mengindikasikan bahwa semakin besar data absisnya, maka semakin besar pula data pada ordinat atau hasilnya, meskipun pada kedua data terakhir peningkatan nilainya paling kecil dibandingkan peningkatan nilai data-data sebelumnya.

2. Studi Kasus 6

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text5.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 10
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 5.8065
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P9(x) = 227096397.6108 + (-415842670.2998x^1) + 318150082.9874x^2 + (-136003290.7297x^3) + 36176040.5222x^4 + (-6249554.7070x^5) + 704212.3017x^6 + (-50061.9561x^7) + 2041.9198x^8 + (-36.4710x^9)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P9(5.8065) = 22805.2688
Apakah anda ingin save jawaban? (Y/N): n
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text5.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 10
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 8.9677
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P9(x) = 227096397.6108 + (-415842670.2998x^1) + 318150082.9874x^2 + (-136003290.7297x^3) + 36176040.5222x^4 + (-6249554.7070x^5) + 704212.3017x^6 + (-50061.9561x^7) + 2041.9198x^8 + (-36.4710x^9)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P9(8.9677) = 175757.6575
Apakah anda ingin save jawaban? (Y/N): n
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text5.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 10
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 9.5000
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P9(x) = 227096397.6108 + (-415842670.2998x^1) + 318150082.9874x^2 + (-136003290.7297x^3) + 36176040.5222x^4 + (-6249554.7070x^5) + 704212.3017x^6 + (-50061.9561x^7) + 2041.9198x^8 + (-36.4710x^9)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P9(9.5000) = 68216.4285
Apakah anda ingin save jawaban? (Y/N): n
```

```
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text5.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 10
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 10.3226
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P9(x) = 227096397.6108 + (-415842670.2998x^1) + 318150082.9874x^2 + (-136003290.7297x^3) + 36176040.5222x^4 + (-6249554.7070x^5) + 704212.3017x^6 + (-50061.9561x^7) + 2041.9198x^8 + (-36.4710x^9)
Hasil estimasi dengan polinom interpolasi di atas adalah:
P9(10.3226) = -3009294.5401
Apakah anda ingin save jawaban? (Y/N): n
```

Eksperimen ini menggunakan data pada Studi Kasus 6 sebagai input matriks yang digunakan dalam kalkulasi interpolasi tersebut. Di sini, semua data dimasukkan ke dalam file *text5.txt* yang berisikan matriks augmented data x dan y dari tiap baris informasi. Seperti biasa, pengguna akan memasukkan nama file beserta formatnya terlebih dahulu, dan menuliskan jumlah baris serta kolom dari matriks yang ada di dalam file tersebut. Kemudian, pengguna akan menginput bentuk data uji dalam **tanggal(desimal)**, yaitu dari data pertama sampai ketiga adalah tanggal 25/05/20, 30/08/20, dan 15/09/20. Adapun pada pengujian terakhir, data tersebut didapat dari tanggal 10/10/20. Pada pengujian terakhir, data hasil interpolasi bernilai negatif dan bersifat eksponensial. Padahal, pada gambar ketiga dengan data 9.5000, hasil interpolasi tersebut masih bernilai positif. Hal ini diakibatkan adanya perubahan pada data matriks yang dimasukkan, yaitu pada dua data terakhir, hasil nilainya menurun dengan cukup besar. Padahal, sebelumnya data tersebut juga meningkat secara drastis. Hal ini menandakan bahwa pada data tersebut ada sebuah titik balik yang membuat hasil interpolasi menjadi bernilai negatif.

3. **Studi Kasus 7**

```
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text7.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 6
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 5
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P5(x) = 0.0000 + 2.0353x + (-3.5527x^2) + 3.2371x^3 + (-1.4213x^4) + 0.2363x^5
Hasil estimasi dengan polinom interpolasi di atas adalah:
P5(5.0000) = 176.0081
Apakah anda ingin save jawaban? (Y/N): n
```

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 4
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan data titik beserta formatnya: text7.txt
Masukkan jumlah baris dari data titik yang ada di dalam file tersebut: 6
Masukkan jumlah kolom dari data titik yang ada di dalam file tersebut: 2
Masukkan titik yang ingin diestimasi: 1.5
Polinom interpolasi yang dihasilkan dari titik-titik tersebut:
P5(x) = 0.0000 + 2.0353x + (-3.5527x^2) + 3.2371x^3 + (-1.4213x^4) + 0.2363x^5
Hasil estimasi dengan polinom interpolasi di atas adalah:
P5(1.5000) = 0.5835
Apakah anda ingin save jawaban? (Y/N): n
```

Pada studi kasus ini, diambil data input matriks dari file dengan nama *text7.txt*. File ini berisi data x dan y dari fungsi yang ada dalam studi kasus dengan n = 5, sehingga terbentuk titik-titik dari [0, 2] dengan interval 0,4. Pengguna akan tetap diharuskan memasukkan jumlah baris dan kolom dari matriks augmented yang ada di file tersebut sesuai dengan fungsi BacaFileToMatrix yang diimplementasikan. Kemudian, dari data yang dimiliki tersebut akan diperoleh sebuah persamaan interpolasi polinom yang memiliki derajat interpolasi sebesar 5, sesuai dengan n sebelumnya. Jika kita memasukkan sebuah data uji yang berada di luar *range* awal tadi, maka hasil yang akan didapatkan dari polinom interpolasi tersebut akan bernilai sangat jauh dari nilai-nilai yang didapat jika diujikan ke titik-titik yang berada di dalam *range*. Seperti yang terlihat di atas, pada gambar pertama diujikan titik X = 5.0000 dan pada gambar kedua diujikan X = 1.5000. Pada gambar pertama titik tersebut menjadi berorde 2, padahal jika kita lihat lagi fungsi yang diberikan, tidak mungkin terjadi hal demikian. Pengujian tersebut tidak valid dan pengujian yang valid bisa dilihat pada gambar kedua. Hal ini disebabkan karena kita tidak memberikan data apapun terkait dengan data-data yang dimiliki di luar *range* tersebut. Artinya, polinom interpolasi yang didapatkan hanya bisa digunakan di dalam *range* tersebut untuk mendapatkan hasil akurat yang sesuai dengan nilai yang seharusnya didapat jika kita memasukkannya ke dalam fungsi awal.

Catatan untuk Interpolasi:

Jika pengguna memasukkan matriks yang tidak memiliki penyelesaian, maka polinom interpolasi tidak akan terdefinisi. Matriks tidak akan terdefinisi apabila pengguna salah memasukkan jumlah baris dan kolom dari matriks yang ada di dalam file tersebut. Hasil interpolasi yang didapat akan sama apabila input dimasukkan melalui *keyboard*. Hasil interpolasi, baik polinom maupun hasil interpolasi tersebut sudah dicek dengan kalkulator interpolasi lain dan memang bernilai benar.

4.3 Regresi Linier Ganda

1. Studi Kasus 8

```
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 5
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan matriks augmented beserta formatnya: text6.txt
Masukkan jumlah baris dari matriks augmented yang ada di dalam file tersebut: 20
Masukkan jumlah kolom dari matriks augmented yang ada di dalam file tersebut: 4
Masukkan data yang ingin diregresi:
Masukkan data 1: 50.0
Masukkan data 2: 76.0
Masukkan data 3: 29.30
Persamaan regresi linier yang dihasilkan:
y = -3.5078 + (-0.0026x1) + 0.0008x2 + 0.1542x3
Nilai tersebut dari persamaan regresi di atas adalah: 0.9384
Apakah anda ingin save jawaban? (Y/N): n
```

Studi kasus 8 ini menggunakan data input titik-titiknya telah dimasukkan ke dalam matriks augmented dan disimpan di file *text6.txt*. Pengguna akan memasukkan nama file beserta format dari file tersebut, jumlah baris serta jumlah kolom dari matriks yang ada di dalam file tersebut. Pengguna kemudian memasukkan data regresi yang ingin dicari sesuai dengan studi kasus yang diberikan, yaitu 50% *humidity*, suhu 76^0F, dan tekanan udara sebesar 29.30 (masing-masing data yang dimasukkan ke dalam matriks dianggap telah memiliki satuan yang sama). Persamaan regresi yang didapat bisa dilihat pada gambar tersebut, dengan besar estimasi nilai Nitrous Oxide dalam kondisi tersebut adalah 0.9384. Hasil tersebut didapat dengan melakukan pembulatan output ke 4 desimal di belakang koma, sehingga dimungkinkan adanya hasil estimasi yang lebih presisi di dalam matriks hasil yang outputnya tidak ditampilkan dalam program ini.

2. Studi Kasus Tambahan: Regresi Linier Ganda

```
Selamat Datang di Tugas Besar Algeo!
Tubes ini adalah hasil kolaborasi Hizkia (13519087), Rian (13519147), & Richard (13519185)
Main Menu
1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar
Silakan input angka 1-6 sesuai pilihan: 5
Silakan pilih stream input
1. Baca File
2. Keyboard
Masukkan pilihan (1-2): 1
Masukkan nama file berisikan matriks augmented beserta formatnya: text3.txt
Masukkan jumlah baris dari matriks augmented yang ada di dalam file tersebut: 5
Masukkan jumlah kolom dari matriks augmented yang ada di dalam file tersebut: 3
Masukkan data yang ingin diregresi:
Masukkan data 1: 2.5
Masukkan data 2: 2.5
Persamaan regresi linier yang dihasilkan:
y = 20.0000 + 0.5000x1 + 0.5000x2
Nilai tersebut dari persamaan regresi di atas adalah: 22.5000
Apakah anda ingin save jawaban? (Y/N): n
```

Studi kasus tambahan ini akan menggunakan data yang didapat dari Stat Trek. Data ini merupakan data hubungan IQ, tingkat jam belajar, dan nilai yang didapat dari masing-masing siswa. Informasi ini dimasukkan ke dalam file *text3.txt* yang akan dimasukkan sebagai input pembacaan file, dengan jumlah baris dan jumlah kolom masing-masing. Setelah itu, akan dicoba sebuah input sembarang, berupa pasangan variabel independen yang akan diuji dan diestimasi nilainya dengan persamaan regresi linier yang didapatkan. Persamaan regresi yang didapat dapat dilihat dalam gambar tersebut, dan untuk siswa yang memiliki IQ 2.5 dan belajar hanya 2.5 jam (sembarang input) maka sesuai dengan persamaan regresi tersebut didapatkan nilai ujian sebesar 22.5.

Catatan untuk Regresi Ganda:

Jika pengguna memasukkan matriks yang tidak memiliki penyelesaian, maka persamaan regresi tidak akan terdefinisi. Sama seperti sebelumnya pula, jika matriks yang dimasukkan salah dalam jumlah baris dan kolom, maka matriks tidak akan terdefinisi. Input melalui *keyboard* maupun file akan menghasilkan hasil yang sama. Semua hasil yang didapat, baik persamaan maupun hasil estimasi regresi dengan persamaan tersebut telah dicek dengan menggunakan kalkulator regresi lain, meskipun memang beberapa data yang diinput mungkin kurang masuk akal, seperti pada bagian IQ dalam Studi Kasus Tambahan. Akan tetapi, hal tersebut bukan merupakan hal yang keliru karena di sini memang persamaan regresi tersebut bersifat linier.

BAB V

SIMPULAN, SARAN, DAN REFLEKSI

5.1 Simpulan

Matriks dalam aljabar linier dan geometri merupakan salah satu bentuk sistem aljabar yang cukup unik, yaitu hanya nilai dari sistem aljabar linier tersebut yang direpresentasikan di dalamnya. Pada suatu matriks bisa dilakukan banyak sekali operasi aritmatika yang kemudian bisa digunakan sebagai fundamental dalam melakukan operasi dua buah matriks, misalnya determinan, transpose, adjoin, kofaktor, dan invers matriks. Dalam hal ini, operasi dua buah matriks, misalnya SPL dan perkalian matriks, juga memiliki banyak metode dan kaidah yang harus dipenuhi untuk melakukan operasi tersebut. Namun demikian, tentunya hasil operasi tersebut akan bernilai sama, apapun metode yang dipakai. Matriks ini juga memiliki peran yang besar dalam melakukan perhitungan mengenai interpolasi dan regresi linier ganda.

Secara keseluruhan, program berhasil dalam menerima input dari *keyboard* dan dari file sesuai keinginan pengguna. Untuk beberapa studi kasus, jawaban yang diberikan oleh program juga sudah mencukupi. Perlu diperhatikan, program masih memiliki kendala atau *bug* yang masih dapat diperbaiki lebih lanjut, seperti kasus SPL yang tidak memiliki solusi pada studi kasus 3B dan perlu di cek lagi kebenarannya, seperti pada studi kasus 4. Program yang telah dibuat masih dapat dikembangkan kedepannya jika diberi tambahan waktu.

5.2 Saran

Perlu kembali diingat, program memang didesain sedemikian rupa dan bergantung ke input dari pengguna untuk menghasilkan output yang sesuai. Sejauh ini, dari beberapa studi kasus dapat dilihat beberapa input dari pengguna yang sesuai, namun program masih menghasilkan eror, atau input yang sudah sesuai dari pengguna (studi kasus 4) namun perlu di cek ulang kebenarannya oleh pengguna. Dalam beberapa percobaan, yang dijadikan pembanding benar tidaknya luaran program adalah kalkulator online. Untuk kedepannya, ada baiknya program dicocokkan lagi dengan metode yang lebih efisien.

Studi kasus tentang aplikasi operasi matriks mengenai Hukum Kirchoff dalam kelistrikan kurang cocok untuk diselesaikan dengan interpolasi, melainkan akan didapatkan penyelesaian yang lebih akurat dengan menggunakan SPL. Adapun studi kasus mengenai regresi linier ganda masih sedikit dan mungkin bisa ditambahkan sebagai pembanding program.

5.3 Refleksi

Pada tugas besar dengan menggunakan bahasa pemrograman Java ini, digunakan paradigma pemrograman berorientasi objek yang seharusnya lebih berorientasi pada prinsip penggunaan kembali *code* yang telah dibuat sebelumnya untuk meminimalisir program serta kesalahan yang mungkin terjadi. Hal ini masih cukup sulit dilakukan oleh penulis karena terbiasa dengan paradigma pemrograman prosedural yang mengakibatkan program sejak awal didesain dengan menggunakan paradigma tersebut. Akibatnya, beberapa bagian program menjadi rumit dan mungkin sulit dibaca.

Untuk refleksi kinerja tim, tim pengembang program sudah cukup baik dalam melaksanakan pekerjaannya. Komunikasi sudah cukup lancar antar setiap anggota tim. Setiap anggota tim mengambil bobot kerja yang kurang lebih sama. Secara keseluruhan, kinerja tim yang produktif sangat membantu dalam terselesaikannya tugas ini.

REFERENSI

Anton, H., & Rorres, C. (2013). *Elementary Linear Algebra: Applications Version*. John Wiley & Sons.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (Vol. 821). John Wiley & Sons.

Rao, C. R., & Mitra, S. K. (1972). *Generalized Inverse of A Matrix and Its Applications. In Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*. The Regents of the University of California.

Regression Coefficients oleh Stat Trek dalam *https://stattrek.com/multiple-regression/regression-coefficients.aspx?tutorial=reg.*