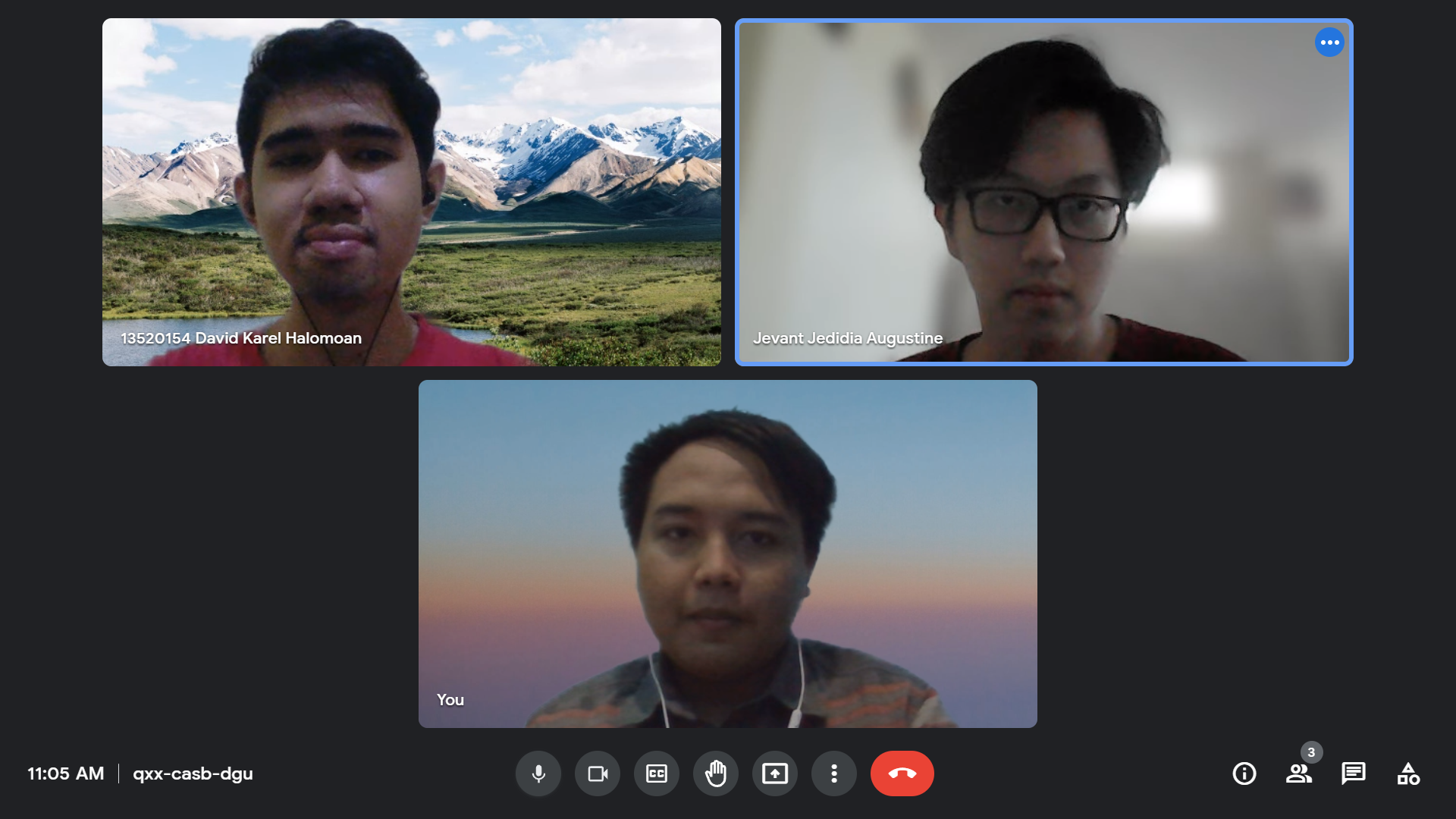
**Laporan Tugas Besar 1**

**IF 2123 Aljabar Linier dan Geometri**

**Sistem Persamaan Linier, Determinan, dan Aplikasinya**

**Semester I Tahun 2021/2022**

****

**Disusun Oleh :**

**Kelompok 51 13520133**

**Jevant Jedidia Augustine 13520133**

**David Karel Halomoan 13520154**

**Daffa Romyz Aufa 13520162**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

# DAFTAR ISI

[**DAFTAR ISI**](#_4wih21ur80b) **2**

[**BAB 1 Deskripsi Masalah**](#_fyhtitofmr39) **3**

[I. Interpolasi Polinom](#_xlyux53g8o5) 3

[II. Regresi Linier Berganda](#_mshejvhf9rf4) 4

[**BAB 2 Teori Singkat**](#_t9nfvsln55i7) **6**

[Metode Eliminasi Gauss](#_sv4qg3a40sc6) 6

[Metode Eliminasi Gauss-Jordan](#_xlq0vyps7pd) 6

[Determinan](#_rhzt7owzmbsc) 6

[Matriks Balikan](#_8zmrahd5ogry) 8

[Matriks Kofaktor](#_9anmo1xj74xb) 8

[Matriks Adjoin](#_t0r2bpsyv3u) 8

[Kaidah Cramer](#_qffrk0ihnl26) 8

[Interpolasi Polinom](#_65n3graycw6e) 9

[Regresi Linier Berganda](#_p9su517n63et) 10

[**BAB 3 Implementasi Pustaka dan Program dalam Java**](#_eskqi0bm8p1z) **11**

[TubesDriver.java](#_ce2uufo8nrqh) 11

[Matriks.java](#_5kadw8n5jj10) 11

[SPL.java](#_wkfmhkqy0fty) 13

[Determinan.java](#_f26xvhg3rqce) 14

[MatriksBalikan.java](#_ogrsh56xi71f) 14

[InterpolasiPolinom.java](#_myqesruz6fx7) 15

[Regresi.java](#_sv2xgndgblq8) 16

[**BAB 4 Eksperimen**](#_nl9sc1e5nqag) **17**

[**BAB 5 Kesimpulan**](#_wxzrwaqysojd) **26**

[**Daftar Referensi**](#_n4l10g73r94k) **27**

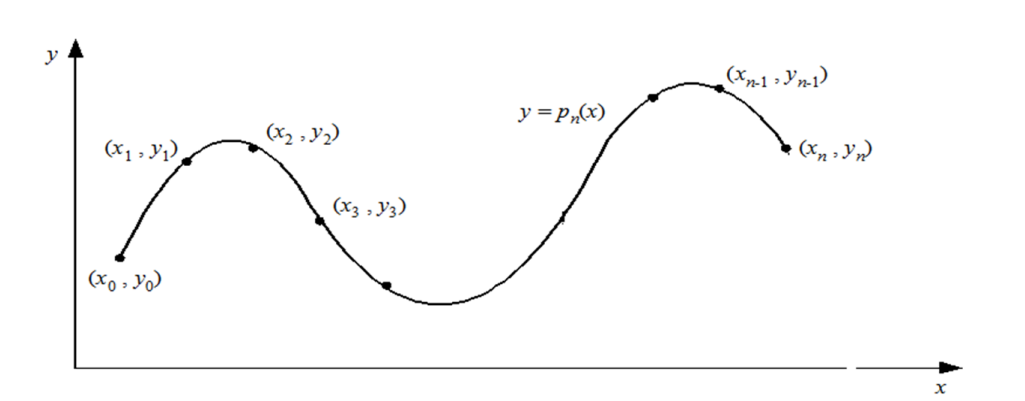
# BAB 1 Deskripsi Masalah

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-1b), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

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

### I. 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 + a2x2 + … + anxn. 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 + a2x2 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 + a2x2 +a3x3, 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 dalam persamaan polinom pn(x) = a0 + a1x + a2x2 + … + anxn untuk *i* = 0, 1, 2, …, *n*, akan diperoleh n buah sistem persamaan lanjar dalam a0, a1, a2, …, an,

*a0 + a1x0 + a2x02 + … + anx0n = y0*

*a0 + a1x1 + a2x12+ … + anx1n = y1*

*... ...*

*a0 + a1xn + a2xn2 + … + anxnn = yn*

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 + a2x2. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah

a0 + 8.0a1 + 64.00a2 = 2.0794

a0 + 9.0a1 + 81.00a2 = 2.1972

a0 + 9.5a1 + 90.25a2 = 2.2513

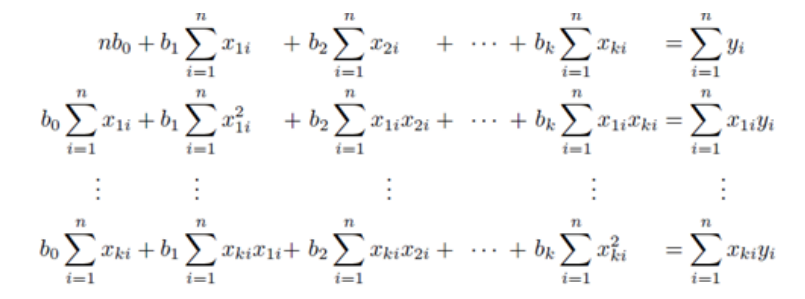
Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan a0 = 0.6762, a1 = 0.2266, dan a2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah p2(x) = 0.6762 + 0.2266x - 0.0064x2. Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: p2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

### II. 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.

yi = β0 + β1x1i + β2x2i + … + βkxki + 𝝐i

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



Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

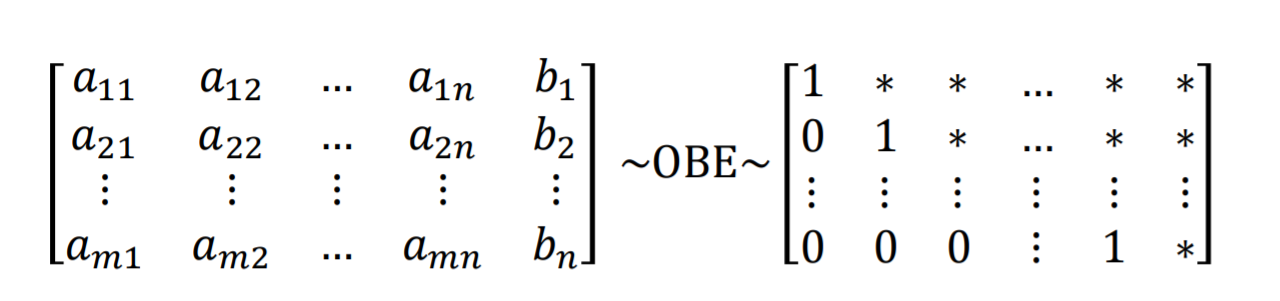
# BAB 2 Teori Singkat

#### **Metode Eliminasi Gauss**

Langkah-langkah pengerjaan metode eliminasi Gauss adalah:

1. Nyatakan SPL dalam bentuk matriks augmented

2. Terapkan OBE pada matriks augmented sampai terbentuk matriks eselon baris



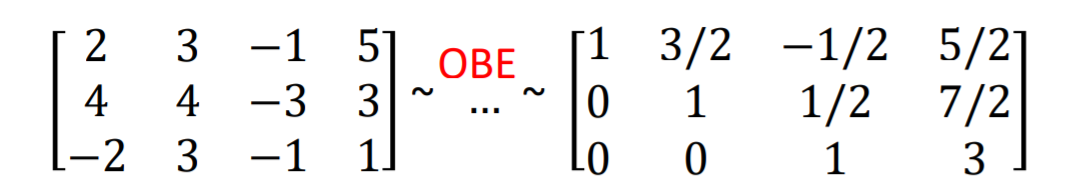
3. Pecahkan persamaan yang berkoresponden pada matriks eselon baris dengan teknik penyulihan mundur (backward substitution)

#### **Metode Eliminasi Gauss-Jordan**

Metode eliminasi Gauss-Jordan terdiri dari dua fase:

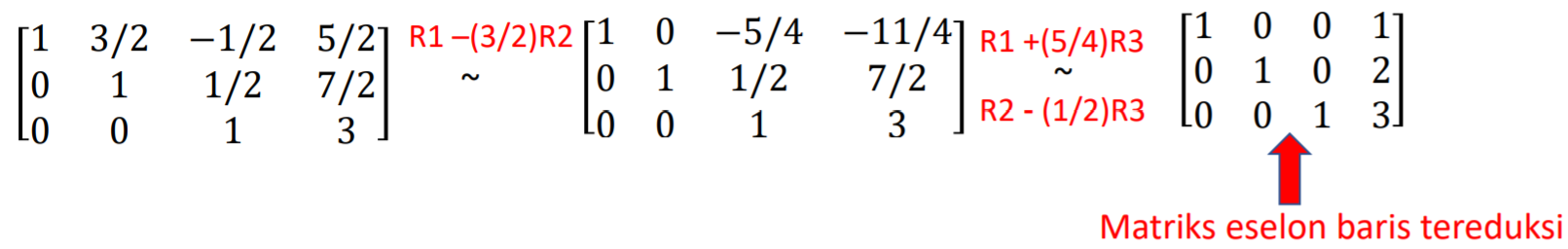
1. Fase maju (forward phase) atau fase eliminasi Gauss

- Menghasilkan nilai-nilai 0 di bawah 1 utama



2. Fase mundur (backward phase)

- Menghasilkan nilai-nilai 0 di atas satu utama



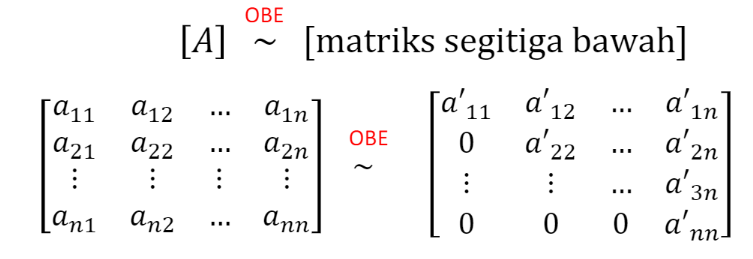
##### 

#### **Determinan**

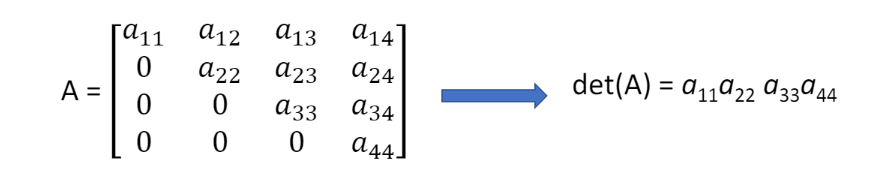
Determinan suatu matriks dapat dicari dengan 2 cara:

1. **Dengan reduksi baris/OBE**

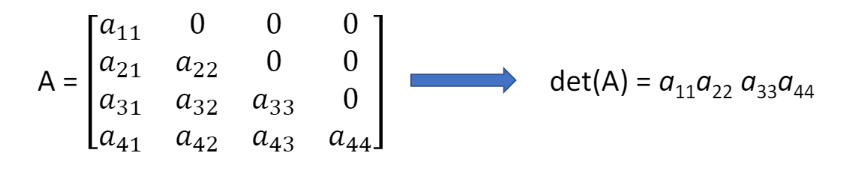
Sebuah matriks dilakukan OBE sampai diperoleh matriks segitiga atas atau matriks segitiga bawah.



Determinan dari sebuah matriks segitiga (atas atau bawah) dapat dicari dengan mengalikan setiap elemen pada diagonal utama.



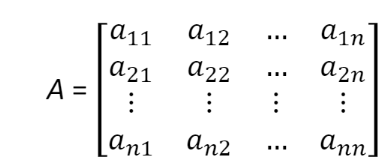
Matriks segitiga atas



Matriks segitiga bawah

1. **Dengan ekspansi kofaktor**

Pada matriks n x n:

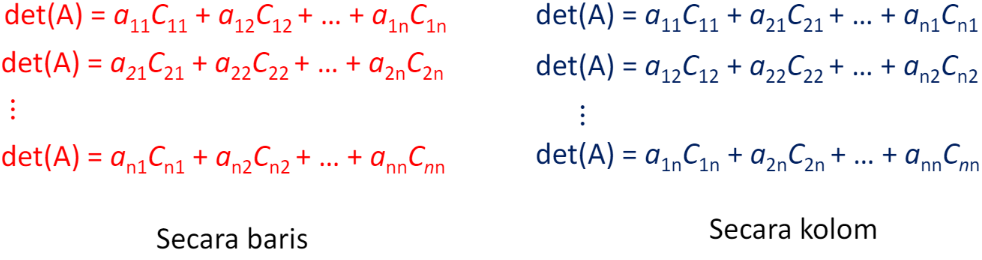


Didefinisikan:

Mij = minor entri aij, yaitu determinan submatriks yang elemen-elemennya tidak berada pada baris i dan kolom j.

Cij = (-1)i+j Mij = kofaktor entri aij

Maka dengan menggunakan kofaktor, determinan matriks dapat dihitung dengan menggunakan salah satu persamaan berikut:



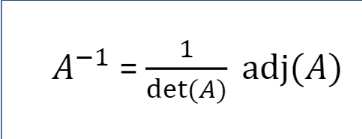
#### **Matriks Balikan**

Matriks balikan (inverse) dari sebuah matriks A adalah B sedemikian rupa sehingga:

AB = BA = I

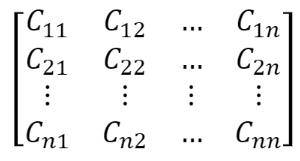
Dapat dikatakan bahwa A dan B merupakan balikan matriks satu sama dengan yang lain.

Balikan matriks A dapat dicari dengan menggunakan rumus:



#### **Matriks Kofaktor**

Matriks kofaktor dari A(matriks berukuran n x n) adalah:



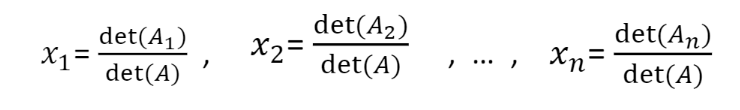
Dengan Cij adalah kofaktor entri aij

#### **Matriks Adjoin**

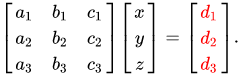
Adjoin dari sebuah matriks A adalah transpose dari matriks kofaktor A.

#### **Kaidah Cramer**

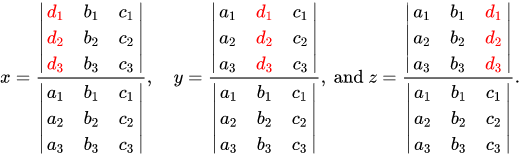
Jika Ax=b adalah SPL yang terdiri dari n persamaan linier dengan n peubah sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi yang unik yaitu:



Pada matriks:



Nilai x, y, dan z dapat dicari dengan melakukan:



#### **Interpolasi Polinom**

Bila diberikan n+1 buah titik berbeda, (x0, y0), (x1, y1), ..., (xn, yn), dapat ditentukan polinom *pn(x)* yang melalui semua titik-titik tersebut sehingga:

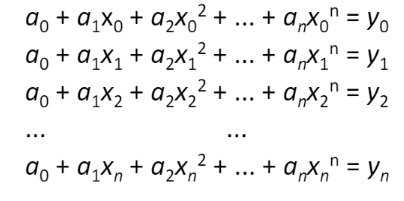
*yi = pn(x)* untuk i = 0, 1 , 2, …, n

Setelah *pn(x)* ditemukan, *pn(x)* dapat digunakan untuk menghitung perkiraan nilai y di x = a, yaitu *y = pn(a)*.

Polinom yang akan terbentuk adalah:

*pn(x) = a0 + a1x + a2x2 + … + anxn*

Dengan memasukkan nilai (x0, y0), (x1, y1), ..., (xn, yn), kedalam persamaan polinom, akan didapat persamaan sebagai berikut:



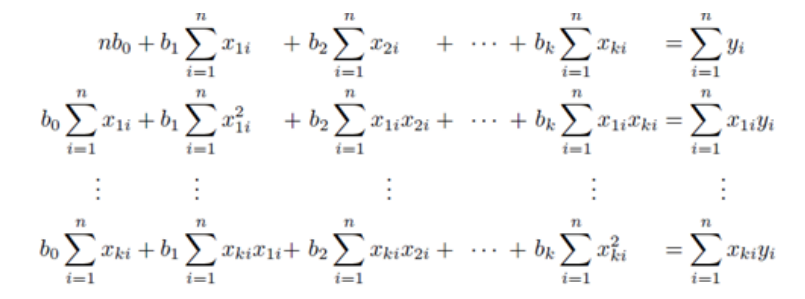
Persamaan tersebut dapat diselesaikan menggunakan metode eliminasi Gauss-Jordan untuk mendapatkan nilai dari a0, a1, a2, …, an.

#### **Regresi Linier Berganda**

Rumus yang dapat digunakan untuk menghitung regresi linear berganda adalah:

*yi = β0 + β1x1i + β2x2i + … + βkxki + 𝝐i*

Untuk mendapatkan nilai dari tiap β, digunakan *Normal Estimation Equation for Multiple Linear Regression.*



Yang kemudian akan diselesaikan dengan menggunakan metode Gauss-Jordan.

# 

# BAB 3 Implementasi Pustaka dan Program dalam Java

#### **TubesDriver.java**

Mencetak menu utama dan memanggil pustaka-pustaka berdasarkan input yang diterima.

#### **Matriks.java**

Mendefinisikan data tipe Matriks yang memiliki baris, kolom, dan isi.

**getBaris**

Mengembalikan baris dari matriks.

**getKolom**

Mengembalikan kolom dari matriks.

**getKonten**

Mengembalikan jumlah baris, kolom, dan isi dari sebuah matriks.

**setKonten**

Menetapkan jumlah baris, kolom, dan isi dari sebuah matriks.

**setElement**

Memasukkan/mengganti elemen matriks pada kolom dan baris tertentu.

**getElement**

Mengembalikan elemen dari matriks pada kolom dan baris tertentu.

**getIdxBarisTerakhir**

Mengembalikan indeks baris terakhir dari matriks.

**getIdxKolomTerakhir**

Mengembalikan indeks kolom terakhir dari matriks.

**isSquare**

Memeriksa apakah matriks adalah square.

**getString**

Mengembalikan isi dari matriks sebagai string.

**hasDeterminan**

Memeriksa apakah matriks memiliki determinan.

**scalarMultiplication**

Mengalikan tiap elemen dalam matriks dengan sebuah konstanta.

**reduksiBaris**

Mengembalikan matriks tereduksi dari sebuah matriks.

**eselonTereduksi**

Mengembalikan matriks eselon tereduksi dari sebuah matriks.

**add**

Mengembailkan hasil tambah matriks A dan matriks B.

**Substract\_by\_**

Mengembailkan hasil kurang matriks A dan matriks B.

**Multiply**

Mengembalikan hasil kali matriks A dan matriks B.

**transpose**

Mengembalikan transpose dari sebuah matriks.

**getSubmatrix**

Mengembalikan submatrix dari sebuah matriks.

**getMatriksAdjoin**

Mengembalikan adjoin dari sebuah matriks.

**inverWithAdjoin**

Meng-invers matriks menggunakan matriks adjoin.

**getDeterminanKofaktor**

Mendapatkan determinan dari matriks menggunakan metode kofaktor.

**getDeterminanReduksiBaris**

Mendapatkan determinan dari matriks menggunakan metode OBE.

**adaMatriksBalikan**

Memeriksa apakah matriks memiliki matriks balikan.

**inversWithGaussJordan**

Meng-invers matriks menggunakan metode Gauss-Jordan.

#### **SPL.java**

**inputSPLFile**

Menerima input berupa file lalu mengubahnya menjadi bentuk augmented matriks.

**inputSPLManual**

Menerima input dari user lalu memasukkannya ke dalam augmented matriks.

**solveWithGauss**

Menyelesaikan persamaan SPL menggunakan metode Gauss.

**solveWithGaussJordan**

Menyelesaikan persamaan SPL menggunakan metode Gauss-Jordan.

**hasManySolution**

Memeriksa apakah SPL memiliki solusi yang banyak.

**hasNoSolution**

Memeriksa apakah SPL tidak memiliki solusi.

**hasSingleSolution**

Memeriksa apakah SPL hanya memiliki 1 solusi.

**singleEselonSolver**

Menyelesaikan SPL yang hanya memiliki 1 solusi dengan metode eselon tereduksi.

**manyEselonSolver**

Menyelesaikan SPL yang memiliki banyak solusi dengan metode eselon tereduksi.

**solveWithInversMatriks**

Menyelesaikan SPL dengan metode invers matriks.

**solveWithCramer**

Menyelesaikan SPL dengan metode Cramer.

**pilihanInput**

Mencetak menu pilihan dan menerima input dari user untuk memilih apakah data dari file atau secara manual.

**aksi**

Menerima SPL lalu menerima input dari user untuk menentukan metode penyelesaian dari SPL. SPL kemudian diselesaikan sehingga didapat hasilnya.

**konfirmasiInputFile**

Menerima input untuk memutuskan apakah hasil dari SPL akan dimasukkan ke dalam sebuah file .txt. Lalu diterima input nama file dimana hasil dari SPL akan disimpan.

**outputFile**

Menyimpan hasil SPL ke dalam file.

**exitProc**

Mengembalikan program ke menu utama.

#### **Determinan.java**

**inputDeterminanFile**

Menerima input berupa file lalu mengubahnya menjadi bentuk augmented matriks.

**inputDeterminanManual**

Menerima input dari user lalu memasukkannya ke dalam augmented matriks.

**pilihanInput**

Mencetak menu pilihan dan menerima input dari user untuk memilih apakah data dari file atau secara manual.

**aksi**

Menerima SPL lalu menerima input dari user untuk menentukan metode penyelesaian untuk mencari determinan. SPL kemudian diselesaikan sehingga didapat determinannya.

**konfirmasiInputFIle**

Menerima input untuk memutuskan apakah hasil dari pencarian determinan akan dimasukkan ke dalam sebuah file .txt. Lalu diterima input nama file dimana hasil dari pencarian determinan akan disimpan.

**outputFile**

Menyimpan hasil pencarian determinan ke dalam file.

**exitProc**

Mengembalikan program ke menu utama.

#### **MatriksBalikan.java**

**inputMatriksBalikanFile**

Menerima input berupa file lalu mengubahnya menjadi bentuk augmented matriks.

**pilihanInput**

Mencetak menu pilihan dan menerima input dari user untuk memilih apakah data dari file atau secara manual.

**inputMatriksBalikanManual**

Menerima input dari user lalu memasukkannya ke dalam augmented matriks.

**aksi**

Menerima SPL lalu menerima input dari user untuk menentukan metode penyelesaian untuk mencari matriks balikan. SPL kemudian diselesaikan sehingga didapat matriks balikan.

**konfirmasiInputFIle**

Menerima input untuk memutuskan apakah hasil dari pembalikan matriks akan dimasukkan ke dalam sebuah file .txt. Lalu diterima input nama file dimana hasil dari pembalikan matriks akan disimpan.

**outputFile**

Menyimpan hasil pembalikan matriks ke dalam file.

**exitProc**

Mengembalikan program ke menu utama.

#### **InterpolasiPolinom.java**

**inputIPFile**

Menerima input berupa file lalu mengubahnya menjadi bentuk tabel interpolasi polinom.

**inputIPManual**

Menerima input dari user lalu memasukkannya ke dalam tabel interpolasi polinom.

**IPSolver**

Menyelesaikan permasalahan interpolasi polinom hingga didapatkan sebuah fungsi.

**pilihanInput**

Mencetak menu pilihan dan menerima input dari user untuk memilih apakah data dari file atau secara manual.

**konfirmasiInputFIle**

Menerima input untuk memutuskan apakah hasil dari interpolasi polinom akan dimasukkan ke dalam sebuah file .txt. Lalu diterima input nama file dimana hasil dari interpolasi polinom akan disimpan.

**outputFile**

Menyimpan hasil interpolasi polinom ke dalam file.

**exitProc**

Mengembalikan program ke menu utama.

#### **Regresi.java**

**inputMatriksDataFile**

Menerima input berupa file dan mengubah isi file menjadi matriks yang berisikan sampel dari pengubah x.

|  | x1 | x2 | xn | y |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

Contoh matriks data

**inputMatriksDataManual**

Menerima input dari user yaitu jumlah peubah dan jumlah sampel dari dan memasukkannya ke dalam Matriks data.

**pilihanInput**

Mencetak menu pilihan dan menerima input dari user untuk memilih apakah data dari file atau secara manual.

**aksi**

Menerima matriks data dan mengubahnya menjadi matriks regresi, lalu menyelesaikan matriks regresi dengan metode Gauss-Jordan.

**konfirmasiInputFIle**

Menerima input untuk memutuskan apakah hasil dari regresi akan dimasukkan ke dalam sebuah file .txt. Lalu diterima input nama file dimana hasil dari regresi akan disimpan.

**outputFile**

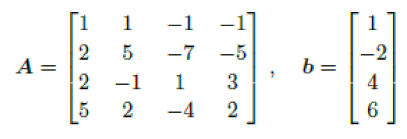
Menyimpan hasil regresi ke dalam file.

**exitProc**

Mengembalikan program ke menu utama.

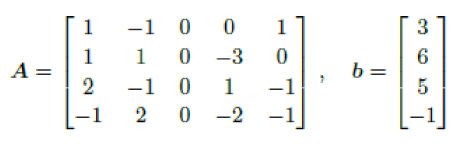
# BAB 4 Eksperimen

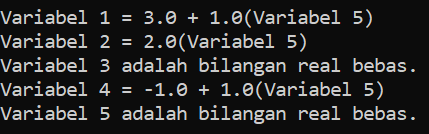
**1.a**

****

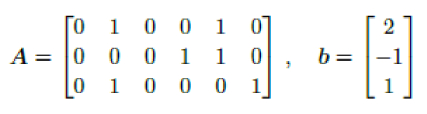
****

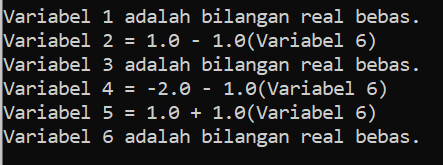
**1.b**

****

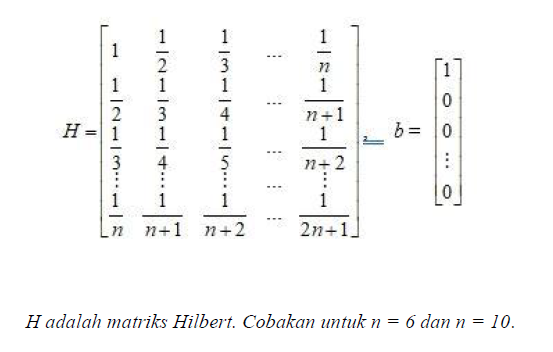
****

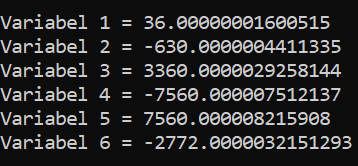
**1.c**

****

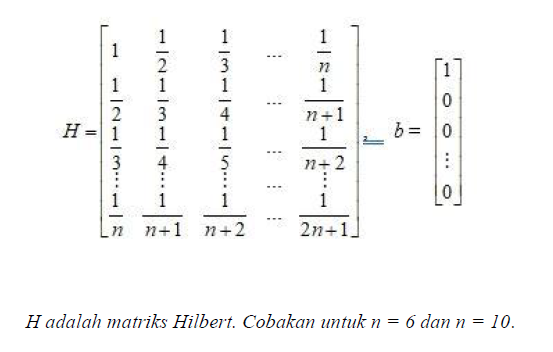
****

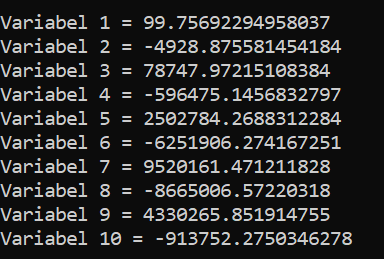
**1.d(n=6)**

****

****

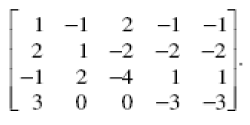
**1.d(n=10)**

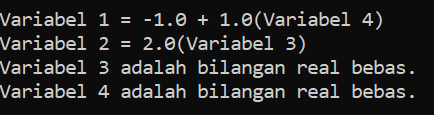
****

****

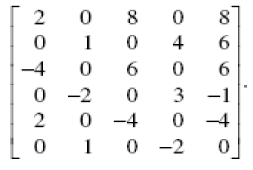
Hasil yang didapat bila memasukkan desimal dan pecahan berbeda akibat pembulatan.

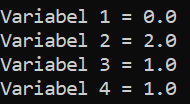
**2.a**

****

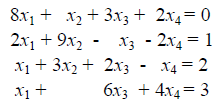
****

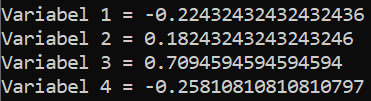
**2.b**

****

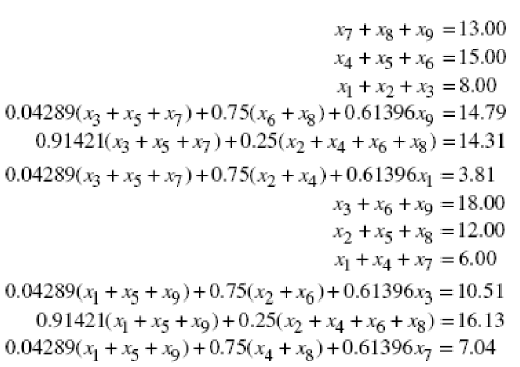
****

**3.a**

****

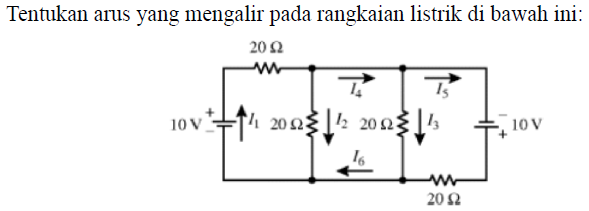
****

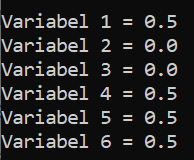
**3.b**

****

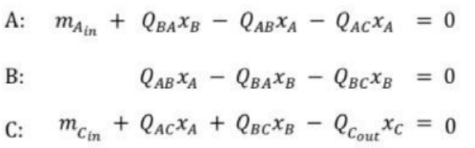
****

**4**

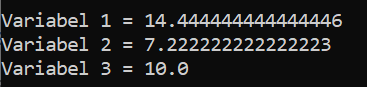
****

****

**5**

****

Dimana QAB = 40, QAC = 80, QBA = 60, QBC = 20 dan QCout = 150 m3 /s dan mAin = 1300 dan mCin = 200 mg/s

****

**6.a**

Program menerima titik-titik berikut :

(0.1, 0.003)

(0.3, 0.067)

(0.5, 0.148)

(0.7, 0.248)

(0.9, 0.370)

(1.1, 0.518)

(1.3, 0.697)

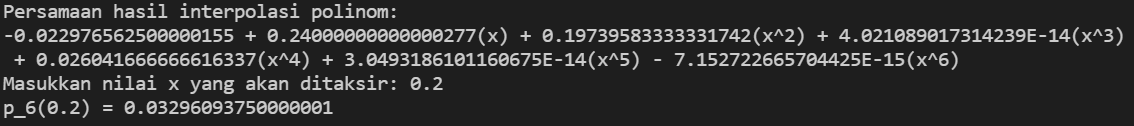
Program menguji nilai-nilai berikut :

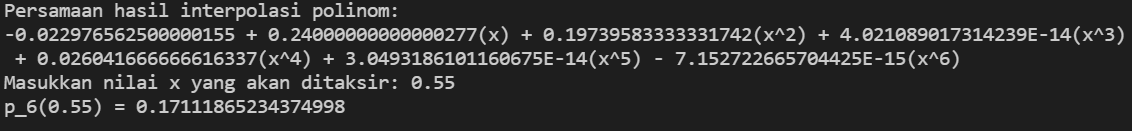
X = 0.2

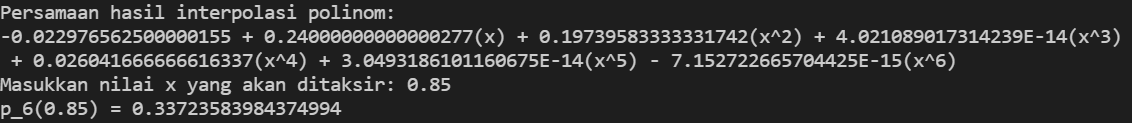
X = 0.55

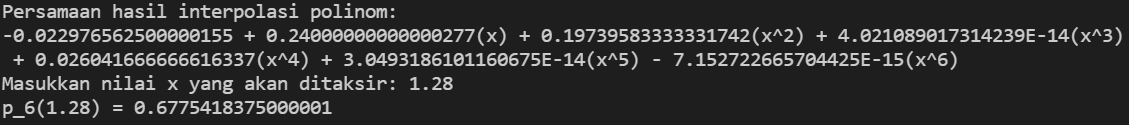
X = 0.85

X = 1.28

****

****

****

****

**6.b**

Program menerima titik-titik berikut :

(6.567, 12624)

(7.000, 21807)

(7.258, 38391)

(7.451, 54517)

(7.548, 51952)

(7.839, 28228)

(8.161, 35764)

(8.464, 20813)

(8.709, 12408)

(9.000, 10534)

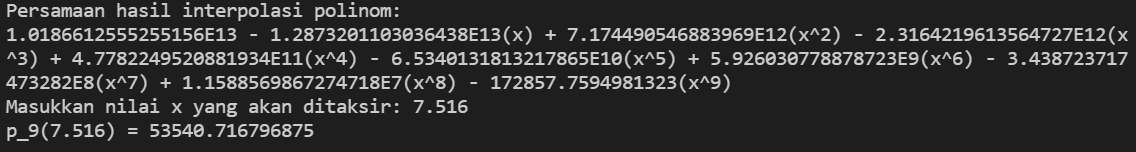
Program menguji nilai-nilai berikut :

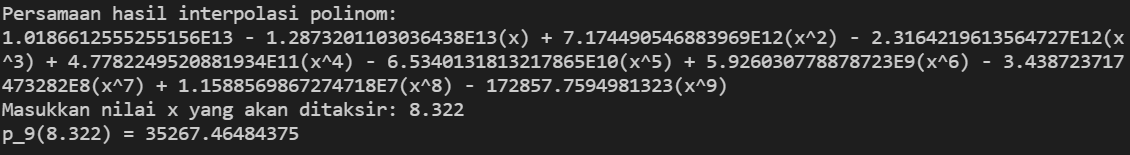
16/07/2021 ⇒ x = 7.516

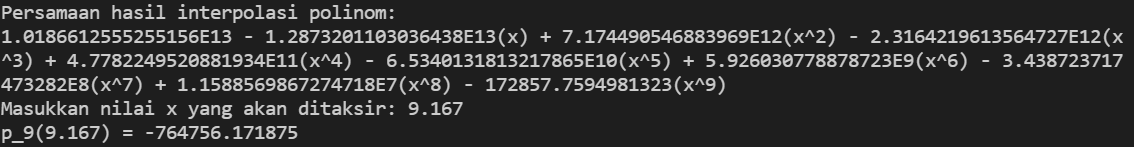
10/08/2021 ⇒ x = 8.322

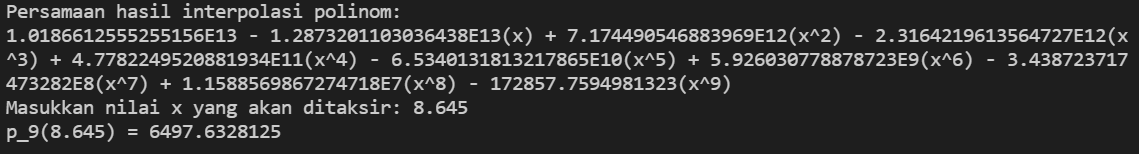
05/09/2021 ⇒ x = 9.167

20/08/2021 ⇒ x = 8.645

****

****

****

****

**6.c**

Program menerima titik-titik berikut :

(0, 0)

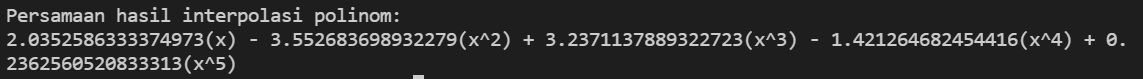
(0.4, 0.4188842301)

(0.8, 0.5071579685)

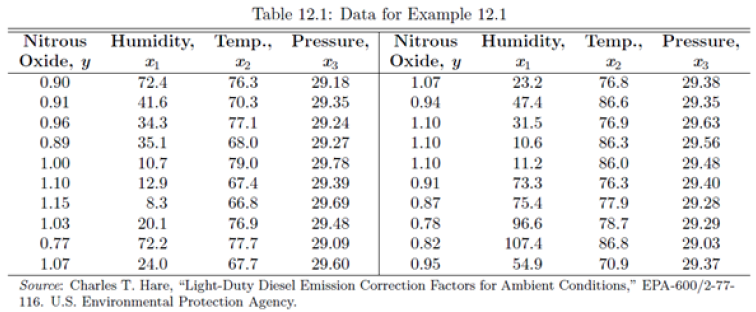
(1.2, 0.5609246748)

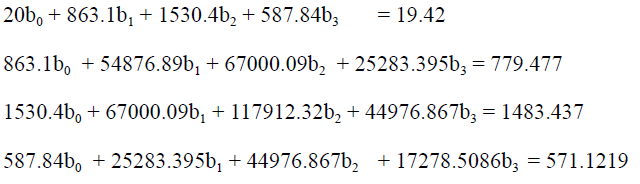
(1.6, 0.5836856613)

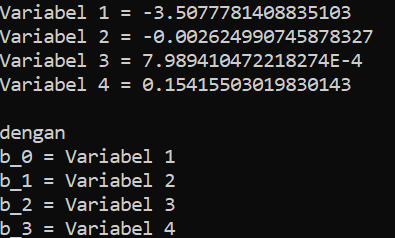
(2, 0.5766515298)

****

**7**

****

****

****

# BAB 5 Kesimpulan

Dari pembuatan pustaka dan program java ini, didapatkan sebuah program yang memanfaatkan metode eliminasi Gauss, metode eliminasi Gauss-Jordan, pencarian determinan, pencarian matriks balikan, pencarian matriks kofaktor, pencarian matriks adjoin, dan kaidah Cramer untuk dapat mengolah sebuah matriks sedemikian rupa sehingga sebuah SPL, interpolasi polinom, dan regresi linier berganda dapat diselesaikan. Walaupun program dapat menyelesaikan permasalahan dengan sangat cepat, bila nilai yang dimasukkan adalah nilai desimal yang dibulatkan, maka hasil yang didapat akan berbeda dengan hasil yang sebenarnya, tergantung dengan jumlah nilai yang dimasukkan. Saran pengembangan untuk tugas ini adalah membuat fungsionalitas baru agar dapat menerima bilangan dalam bentuk pecahan sebab masukan bilangan pecahan dan desimal bisa memiliki hasil yang berbeda (seperti pada file 1\_d(n=10).txt). Refleksi kami terhadap tugas ini adalah kami mendapat pengalaman *coding* baru. Kami juga belajar cara mengimplementasikan algoritma-algoritma rumit, seperti algoritma reduksi baris matriks, dan lain-lain, yang tentunya menambah kemampuan logika berpikir kami.

# Daftar Referensi

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/algeo.htm>

<https://www.youtube.com/watch?v=xk4_1vDrzzo>

<https://www.geeksforgeeks.org/different-ways-reading-text-file-java/>