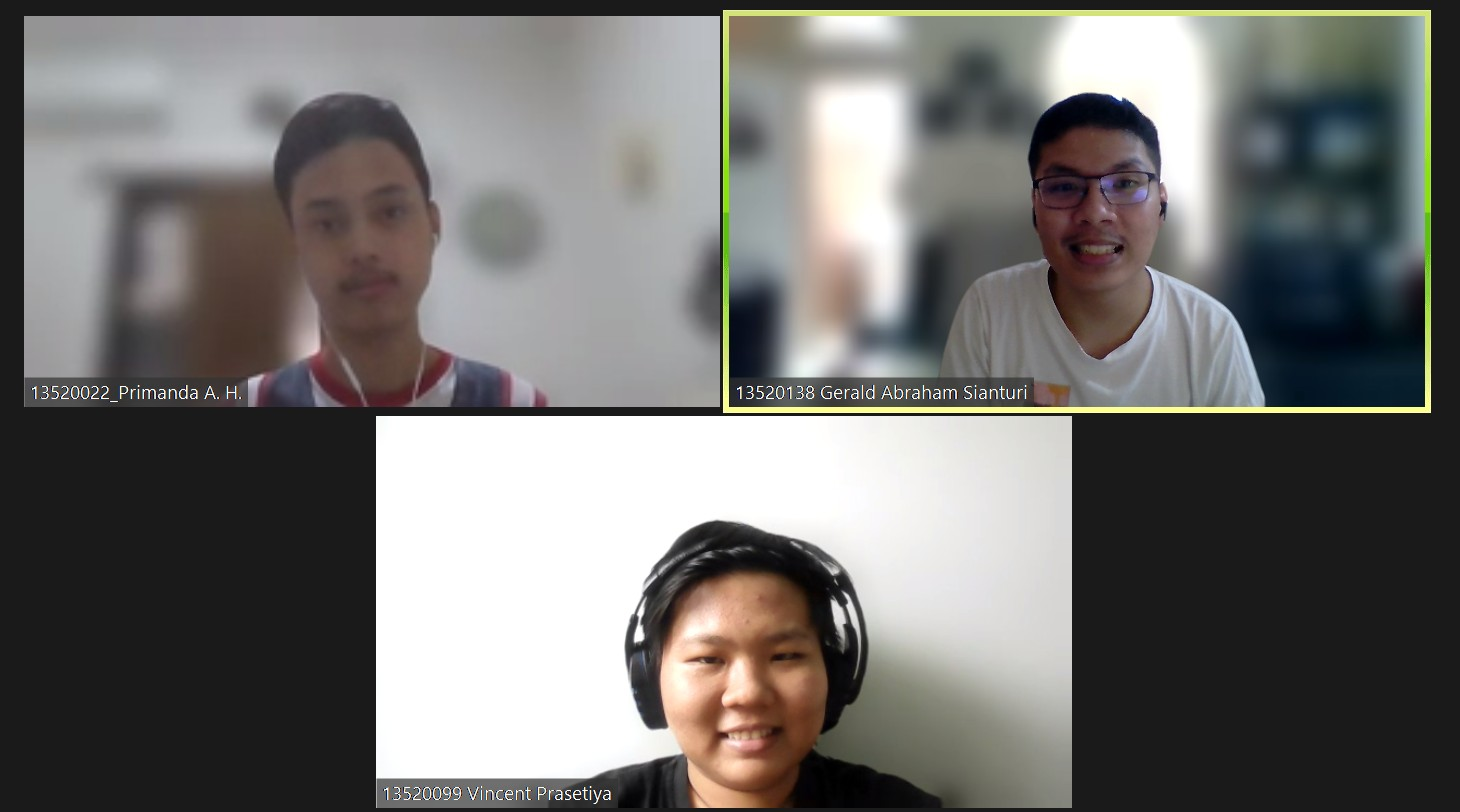
LAPORAN TUGAS BESAR ALJABAR LINIER DAN GEOMETRI

Disusun oleh kelompok **3Masketir:**

|  |  |
| --- | --- |
| 1. Primanda Adyatma Hafiz | (13520022) |
| 1. Vincent Prasetiya Atmadja | (13520099) |
| 1. Gerald Abraham Sianturi | (13520138) |



Makalah yang dibuat untuk memenuhi tugas mata kuliah Aljabar Linier dan Geometri (IF2123)

TAHUN PELAJARAN 2021/2022

# DAFTAR ISI

[DAFTAR ISI ii](#_TOC_250006)

[BAB I DESKRIPSI MASALAH 3](#_TOC_250005)

* 1. Masalah pertama 3
  2. Masalah kedua 5

[BAB II LANDASAN TEORI 6](#_TOC_250004)

* 1. Metode Eliminasi Gauss 6
  2. Metode Eliminasi Gauss-Jordan 6
  3. Determinan Matriks 6
  4. Matriks Balikan 7
  5. Matriks Kofaktor 7
  6. Matriks Adjoin 8
  7. Kaidah Kramer 8
  8. Interpolasi Polinom 9

[BAB III IMPLEMENTASI PROGRAM 10](#_TOC_250003)

* 1. Deskripsi Singkat 10
  2. Metode DeterminanMatriks 10
  3. Metode DriverMatriks 10
  4. Metode InversMatriks 10
  5. Metode RegresiLinear 10
  6. Metode Gauss 11
  7. Metode Gauss Jordan 12
  8. mainFile 12
  9. Garis Besar Program 13

[BAB IV EKSPERIMEN 15](#_TOC_250002)

* 1. *testcase* 1 16
  2. *testcase* 2 21
  3. *testcase* 3 22
  4. *testcase* 4 24
  5. *testcase* 5 24
  6. *testcase* 6 24
  7. *testcase* 7 27

[BAB V KESIMPULAN 28](#_TOC_250001)

* 1. Kesimpulan 28
  2. Refleksi 28
  3. Saran 28

[DAFTAR REFERENSI 32](#_TOC_250000)

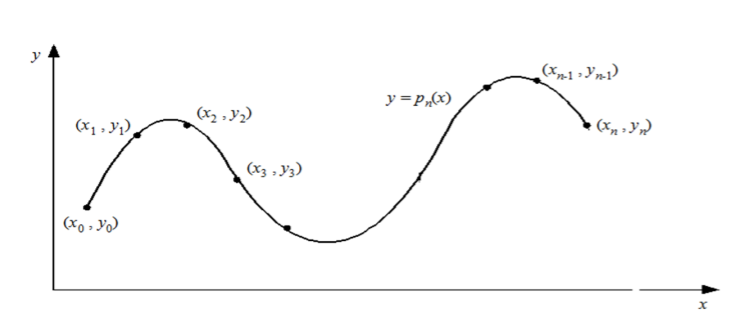
# BAB I DESKRIPSI MASALAH

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 , 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 eliminasii Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung deteriminan, 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.

1. Interpolasi Polinom

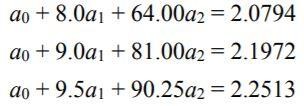
Persoalan interpolasi polinom adalah sebagai berikut : Diberikan buah titik berbeda, . tentukan polinom yang menginterpolasi (melewati) semua titik-titik dtersebut sedemikian rupa sehingga untuk .



Setelah polinom interpolasi ditemukan, dapat digunakan untuk meghitung perkiraan nilai di sembarang titik di dalam selang .

Polinom interpolasi derajat yang menginterpolasi titik-titik . adalah berbetuk . Jika hanya ada dua titik , maka polinom yang menginterpolasi kedua titik tersebut adalah yaitu berupa persamaan garis lurus. Jika tersedia tiga titik , maka polinom yang menginterpolasi ketiga titik tersebut adalah atau persamaan kuadrat dan kurvanya berbentuk parabola. Jika tersedia empat titik , polinom yang menginterpolasi keempat titik tersebut adalah , demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat untuk yang lebih tinggi asalkan tersedia buah titik data. Dengan menyulihkan ke dalam persamaan polinom untuk , akan diperoleh buah sistem persamaan lanjar dalam

Solusi sistem persamaan lanjar ini, yaitu nilai 𝑎*0*, 𝑎*1*, …, 𝑎𝑛, diperoleh dengan menggunakan metode eliminasi Gauss yang telah dipelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk 𝑝*2*(x) = 𝑎*0* + 𝑎*1*𝑥 + 𝑎*2*𝑥*2*. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah



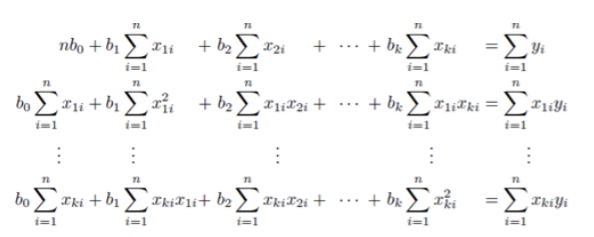
Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan

𝑎*0* = 0.6762, 𝑎*1* = 0.2266, dan 𝑎*2* = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah 𝑝*2*(x) = 0.6762 + 0.2266𝑥 - 0.0064𝑥*2* . Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: 𝑝*2*(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(*9*.*2*)*2* = 2.2192.

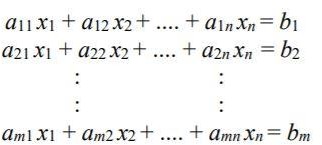
1. Regresi Linier Berganda

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.

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



Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.



# BAB II LANDASAN TEORI

* 1. Metode Eliminasi Gauss

Dalam ilmu matematika, Metode Eliminasi Gauss atau dikenal sebagai reduksi baris adalah sebuah algoritma untuk menyelesaikan sistem persamaan linear. Prosesnya terdiri dari urutan operasi yang dilakukan pada matriks koefisien yang sesuai. Metode ini dapat digunakan untuk menghitung rank dari matriks, determinan dari matriks persegi dan invers dari matriks yang dapat diinverskan. Metode ini dinamai berdasarkan Carl Friedrich Gauss (1777-1855) meskipun telah diketahui oleh ahli matematika China sejak 179 CE.

Untuk melaksanakan Eliminasi Gauss ada matriks diperlukan beberapa langkah operasi baris elementer untuk mengubah matriks sampai bagian kiri bawah dari matriks mengandung angka nol sebanyak mungkin. Terdapat 3 jenis operasi baris elementer yaitu menukar dua buah baris, mengalikan sebuah baris dengan bilangan bukan nol, dan menambahkan atau mengurangkan kelipatan dari sebuah baris ke baris lainnya.

Dengan menggunakan kumpulan operasi tersebut selalu dapat dibentuk matriks eselon baris yakni ketika satu utama telah terbentuk dan seluruh elemen di bawah satu utama bernilai nol. Nantinya hasil akhir matriks eselon baris yang didapatkan akan bernilai sama meskipun rangkaian operasi yang dilakukan berbeda. Setelah didapatkan matriks eselon baris dilakukan proses substitusi dari baris matriks terbawah untuk menyelesaikan sistem persamaan linear

* 1. Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss Jordan adalah sebuah algoritma yang dapat digunakan untuk menyelesaikan sistem persamaan linear sama halnya Metode Eliminasi Gauss. Dalam metode ini digunakan pula operasi baris elementer akan tetapi perbedaannya dengan Metode Eliminasi Gauss yakni metode ini akan mengubah persamaan ke bentuk matriks eselon baris tereduksi. Matriks eselon baris tereduksi mirip dengan matriks eselon baris akan tetapi elemen di atas satu utama selalu bernilai nol.

* 1. Determinan Matriks

Dalam bidang aljabar linear, determinan adalah nilai yag dapat dihitung dari unsur suatu matriks persegi. Determinan matriks A ditulis dengan tanda det(A), det A, atau |A|. Determinan dapat dianggap sebagai faktor penskalaan transformasi yang digambarkan oleh matriks. Secara geomteris determinan memberikan volume dari parallelepiped berimensi n atau dengan kata lain . Dalam sistem persamaan linear, determinan digunakan untuk menentukan apakah suatu matriks merupakan matriks singular atau matriks yang nilai determinannya nol. Karena matriks singular memiliki determinan nol maka matriks tersebut tidak memiliki invers dan solusi dari sistem persamaan linearnya tidak unik.

Perhitungan determinan dapat dengan menggunakan metode ekspansi kofaktor dan metode reduksi baris. Determinan dalam ekspansi kofaktor adalah total hasil perkalian dari sebuahh baris atau kolom terhdapat matriks kofaktornya. Sedangkan determinan dalam metode reduksi baris adalah hasil perkalian dari diagonal utama matriks eselon baris.

* 1. Matriks Balikan

Dalam aljabar linear, suatu matriks yang merupakan matriks persegi dan determinannya bukan bernilai nol disebut invertible atau memiliki matriks invers. Invers matriks adalah kebalikan dari sebuah matriks yang apabila matriks tersebut dikalikan dengan inversnya, akan menjadi matriks identitas. Invers dari matriks A dilambangkan dengan dan memenuhi persamaan:

Dalam menghitung matriks invers dapat digunakan dua metode yakni metode matriks adjoin dan metode matriks reduksi. Untuk metode matriks adjoin maka pertama-tama perlu dihitung transpose dari matriks kofaktor lalu dibagi dengan determinan matriks. Sedangkan untuk menghitung invers matriks dengan metode matriks reduksi perlu dibentuk matriks [A|I] lalu ditransformasi menjadi matriks [I|] dengan operasi baris elementer.

* 1. Matriks Kofaktor

Matriks kofaktor adalah matriks yang dihilangkan baris ke-i dan kolom ke-j dengan elemen-elemennya merupakan hasil perkalian minor dengan suatu angka yang besarnya menuruti aturan dimana adalah baris dan adalah kolom. Kofaktor dari suatu elemen baris ke-i dan kolom ke-j dari matriks A dilambangkan dengan .

* 1. Matriks Adjoin

Matriks adjoin dari sebuah matriks A adalah transpose dari matriks kofaktor A. Matriks adjoin digunakan untuk mencari matriks balikan menggunakan matriks kofaktor yang ditranspose dengan matriks balikan (*inverse of* A atau 𝐴−*1*) dinyatakan sebagai :

* 1. Kaidah Kramer

Kaidah Cramer adalah rumus yang dapat digunakan untuk menyelesaikan sistem persamaan linear dengan banyak persamaan sama dengan banyak variabel, dan berlaku ketika sistem tersebut memiliki solusi yang tunggal. Rumus ini menyatakan solusi dengan menggunakan determinan matriks koefisien dan determinan matriks lain yang diperoleh dengan mengganti salah satu kolom matriks koefisien dengan nilai yang berada pada sebelah kanan persamaan. Metode ini dinamai dari matematikawan Swiss Gabriel Cramer yang pada tahun 1750 menerbitkan kaidah ini untuk sebarang banyaknya variabel.

Akan tetapi komputasi dengan menggunakan kaidah cramer kurang efektif karena perlu menghitung nilai determinan dibandingkan dengan eliminasi gauss yang hanya perlu menghitung satu buah determinan.

* 1. Interpolasi Polinom

Interpolasi polinom merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Teknik interpolasi polinom nantinya akan menerima input titik sebanyak n buah lalu dari titik-titik tersebut akan dihasilkan persamaan polinomial berderajat n. Persamaan polinomial yang diperoleh selanjutnya digunakan untuk melakukan memprediksi nilai dari suatu data yang berada di luar rentang data yang diketahui.

# BAB III IMPLEMENTASI PROGRAM

* 1. Deskripsi Singkat

Terdapat 6 kelas, yakni Determinan OBE, Gauss, GaussJordan, inversMatriks, regresiLinear, dan SPLCrammer. Setiap kelas kemudian di *export* ke dalam format .jar.

* 1. Metode Determinan Matriks
     1. Fungsi public static double DetbyKofaktor(double[][] inputan)

Metode DetbyKofaktor mengembalikan nilai determinan dari matriks inputan dengan metode kofaktor

* + 1. Fungsi double calcDeterminanOBE (double[][] inputMat, int nrow, int ncol)

Metode calcDeterminanOBE mengembalikan nilai determinan dari matriks inputan dengan metode OBE

* 1. Metode DriverMatriks
     1. Prosedur public static void main()

Sebagai program utama.

* 1. Metode InversMatriks
     1. Fungsi public static double[][] minor(double[][] inputMatriks)

Mengembalikan matriks minor dari matriks inputMatriks

* + 1. Fungsi public static double[][] kofaktor(double[][] inputnya)

Mengembalikan matriks kofaktor dari matriks inputnya

* + 1. Fungsi public static double[][] adjoin(double[][] inputnya)

Mengembalikan matriks adjoin dari matriks inputnya

* + 1. Fungsi public static double[][] transposelocal(double[][] inputnya)

Mengembalikan matriks transpose dari matriks inputnya

* + 1. Fungsi public static double[][] inversebyKofaktor(double[][] inputan)

Mengembalikan matriks invers dari matriks inputan dengan metode ekspansi kofaktor

* + 1. Fungsi public static double[][] inversebyGaussJordan(double[][] inputan)

Mengembalikan matriks invers dari matriks inputan dengan metode Gauss Jordan

* 1. Metode RegresiLinear
     1. Fungsi static double[] multiReg(double[][] varDep, double[] varIkat)

Mengembalikan array berisi koefisien-koefisien korelasi dari regresi

* + 1. Fungsi static double[] multiplication(double[][] kiri, double[] kanan)

Mengembalikan array berisi hasil kali matriks m x n dengan array dengan panjang n

* 1. Metode Gauss
     1. Prosedur GaussMatrix

Konstruktor matriks yang mengatur variabel input dan memasukkan hasil matrix ke variabel global

* + 1. Prosedur transformToGauss()

Melakukan transformasi matrik input ke matriks gauss

* + 1. Prosedur sortMatrixRow()

Mengurutkan matriks berdasarkan lokasi satu utama

* + 1. Fungsi int getSwapCount()

Mengembalikan jumlah penukaran baris pada operasi baris elementer

* + 1. Fungsi int getDivVal()

Mengembalikan total operasi pembagian pada operasi elementer

* + 1. Fungsi double[][] getGaussMatrix()

Mengembalikan matriks hasil eliminasi Gauss

* + 1. Prosedur formattingZero()

Mengkonversi bilangan x dimana menjadi nol

* + 1. Prosedur displayMat()

Menampilkan matriks hasil eliminasi Gauss ke layar

* + 1. Prosedur rowOperation(int rowfirst, int rowsecond, double multiplier, Boolean plus)

Melakukan operasi aritmetika antar baris matriks

* + 1. Prosedur multiplyRow(int row, double multiplier)

Mengalikan elemen baris pada matriks dengan suatu konstanta

* + 1. Fungsi double getDeterminan()

Mengembalikan determinan matriks hasil eliminasi Gauss

* + 1. Fungsi double[][] getResult(double[][] inputMat, int nrow, int ncol)

Mengembalikan matriks hasil eliminasi Gauss yang telah berbentuk parameter

* + 1. Fungsi boolean isNoSolution(double[][] inputRes)

Mengembalikan true jika SPL tidak memiliki solusi

* + 1. Fungsi int numParams()

Mengembalikan jumlah parameter dari SPL

* 1. Metode Gauss Jordan
     1. Prosedur GaussMatrix()

Konstruktor matriks yang mengatur variabel input dan memasukkan hasil matriks ke variabel global

* + 1. Prosedur transformToGaussJordan()

Melakukan transformasi matriks input ke matriks gauss

* + 1. Fungsi int getSwapCount()

Mengembalikan jumlah penukaran baris pada operasi baris elementer

* + 1. Fungsi int getDivVal()

Mengembalikan total operasi pembagian pada operasi elementer

* + 1. Fungsi double[][] getGaussJordanMatrix()

Mengembalikan matriks hasil eliminasi Gauss Jordan

* + 1. Prosedur formattingZero()

Mengkonversi bilangan x dimana menjadi nol

* + 1. Prosedur rowOperation(int rowfirst, int rowsecond, double multiplier, Boolean plus)

Melakukan operasi aritmetika antar baris matriks

* + 1. Fungsi double[][] getResult(double[][] inputMat, int nrow, int ncol)

Mengembalikan matriks hasil eliminasi Gauss Jordan yang telah berbentuk parameter

* 1. mainFile
     1. Fungsi int CheckInteger(int min, int max, String message)

Fungsi ini akan mengembalikan integer input apabila memenuhi kondisi

* + 1. Fungsi int whatMethod(int numb)

Fungsi ini akan mengembalikan integer yang menunjukkan metode penyelesaian apa yang dipakai

* + 1. Fungsi int whatInput()

Fungsi ini akan mengembalikan integer yang menunjukkan metode input apa yang dipakai

* + 1. Fungsi int whatOutput()

Fungsi ini akan mengembalikan integer yang menunjukkan metode output apa yang dipakai

* + 1. Prosedur void inputFile(File input, int service)

Prosedur ini akan melakukan pembacaan file dari file input sesuai dengan servicenya

* + 1. Fungsi String LinearEq(int getmethod)

Fungsi yang akan mengembalikan string yang berupa solusi dari persamaan linier sesuai methode yang diinginkan

* + 1. Fungsi String DeterminanMat(int getmethod)

Fungsi yang akan mengembalikan string yang berupa solusi dari determinan sesuai methode yang diinginkan

* + 1. Prosedur InverseMat(int getmethod)

Prosedur yang akan menginvers matriks sesuai methode yang diinginkan

* + 1. Fungsi String Interpolasi()

Fungsi yang akan mengembalikan string yang berupa solusi dari interpolasi

* + 1. Fungsi String RegresiSolver()

Fungsi yang akan mengembalikan string yang berupa solusi dari regresi Linear Berganda

* + 1. Prosedur outputFinalString(int getoutput, String hasil, String filename)

Prosedur yang akan mengeluarkan String hasil ke layer atau menuliskan ke file output

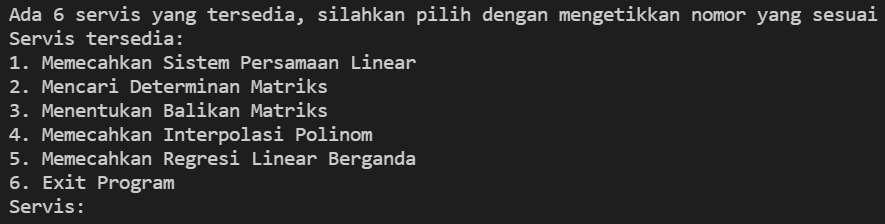
* + 1. Prosedur outputFinalMatrix(int getoutput, String filename)

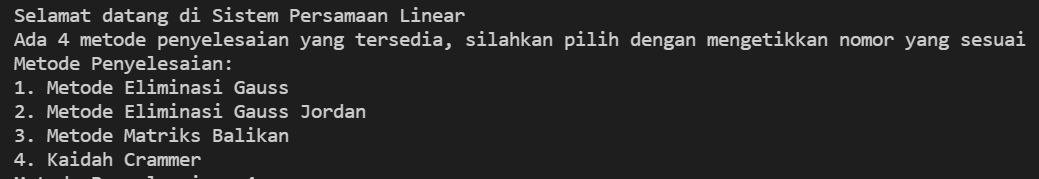
Prosedur yang akan mencetak Matriks hasil akhir ke layer atau menuliskan ke file output

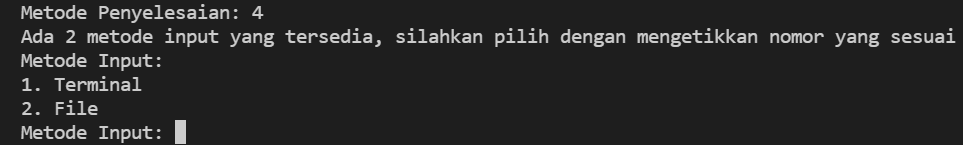
* 1. Garis Besar Program

Garis besar program utama dapat dijelaskan sebagai berikut:

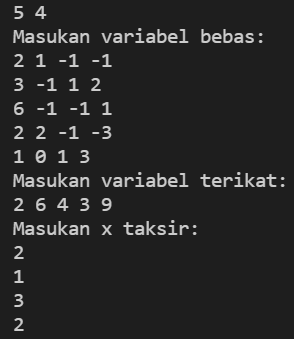
* + 1. Program dapat dijalankan melalui *command prompt* dengan memastikan *directory* tepat. Kemudian input **java -jar MainProgram.jar**
    2. Pertama pengguna akan diberikan menu utama seperti berikut:



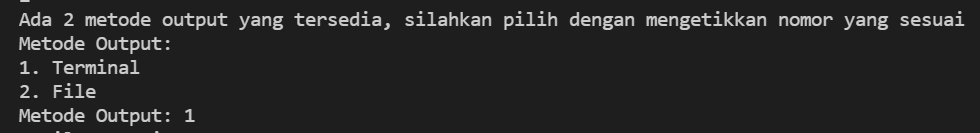
* + 1. Pengguna bisa memilih SPL, determinan, matriks balikan, interpolasi polinom, regresi linear berganda, atau keluar dari program.
    2. Jika pengguna memilih SPL misalnya, pengguna akan kembali diprompt untuk memlih metode yang ingin dipakai. Jika pengguna memilih SPL misalnya, pengguna akan kembali diprompt untuk memlih metode yang ingin dipakai
    3. Program kemudian menampilkan dua opsi input, dapat berupa file atau masukan dari keyboard



* + 1. Program akan memprompting user untuk memasukan baris, kolom, dan matriks dengan format tergantung dengan apa yang dicari dan dengan metode apa mencarinya. Misal pada kasus regresi, inputan seperti berikut



* + 1. User akan diberi 2 pilihan untuk menerima output dari terminal atau ke file

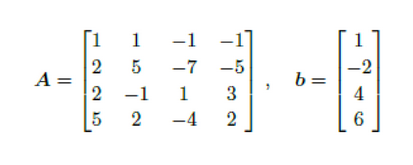


* + 1. Setelah selesai pada aksi sekuensial tertentu, pengguna akan kembali menu utama kembali hingga akhirnya pengguna keluar dari program dengan memasukan nomor 6 pada servis

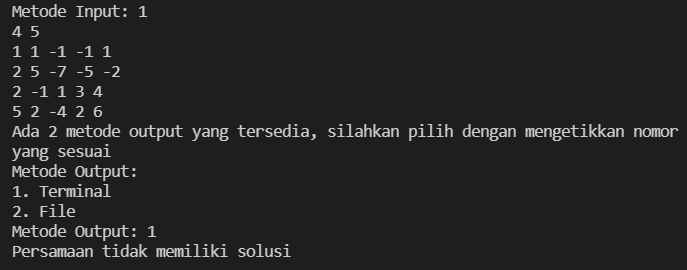
# BAB IV EKSPERIMEN

Dilakuan *test case* sesuai yang ada pada pdf spesifikasi tubes Algeo 1:

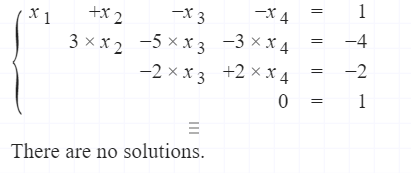
* 1. Temukan solusi SPL Ax = b, berikut:
     + SPL 1



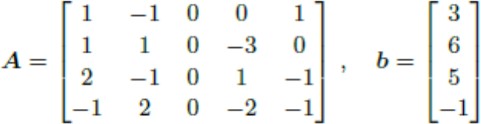
Menurut program dengan metode Gauss, SPL tidak memiliki solusi.



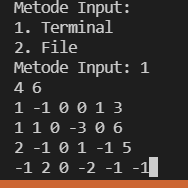
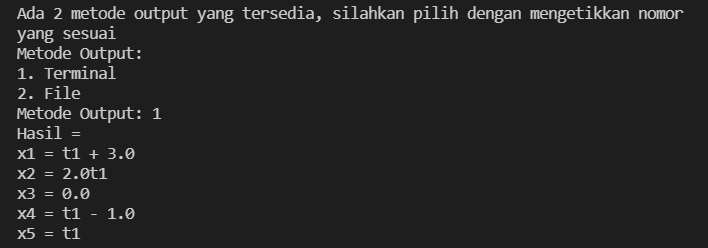
Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.



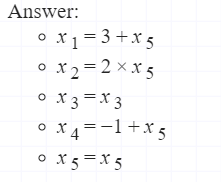
* + - SPL 2



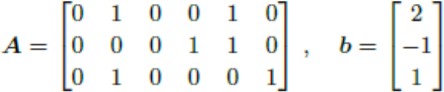
Menurut program dengan metode Gauss-Jordan, SPL memiliki solusi sebagai berikut:

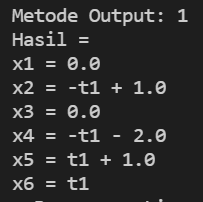
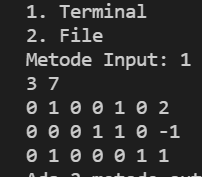
Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.



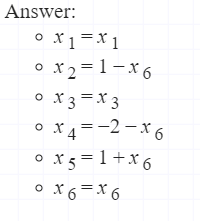
* + - SPL 3



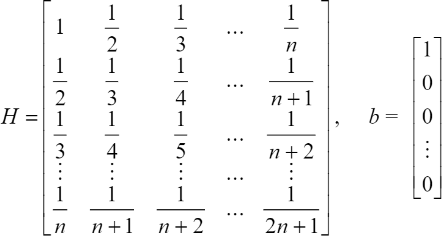
Menurut program dengan metode Gauss, SPL memiliki solusi sebagai berikut:



Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.



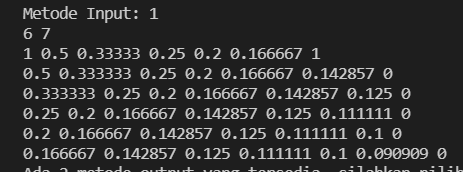
* + - SPL 4



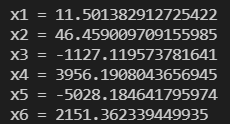
Matriks Hilbert n=6

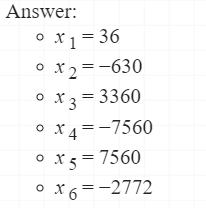
Menurut program dengan metode Kramer, SPL memiliki solusi sebagai berikut:

* + - * Inputan

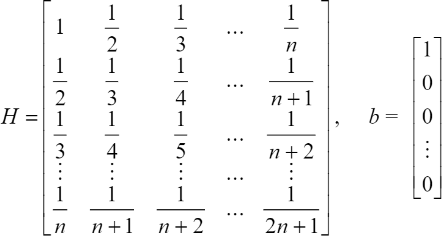


* + - * Outputan



Jika dibandingkan dengan penghitung online, berbeda

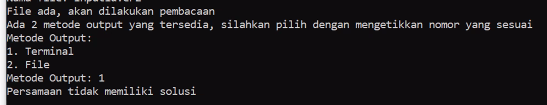
* + - SPL 5



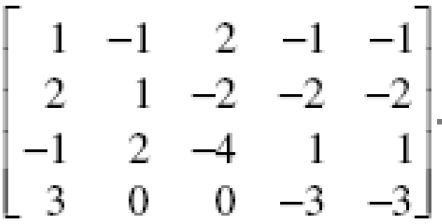
Matriks Hilbert n=10

Menurut program dengan metode Inverse, SPL memiliki solusi sebagai berikut:

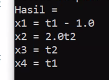
* + - * Output:



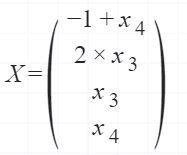
* 1. SPL Ax=B
     + SPL 1 matriks augmented



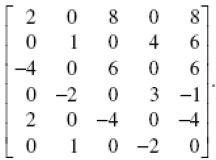
menggunakan metode Gauss-Jordan, didapatkan:



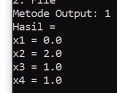
Dari kalkulator *online*, didapat hasil yang mirip (berbeda parameter) :



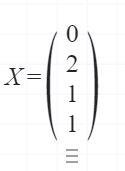
* + - SPL 2 matriks augmented



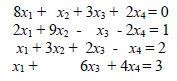
Menggunakan metode Gauss-Jordan, didapatkan:



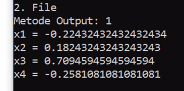
Saat dicoba di kalkulator *online*, didapat hasil yang sama:



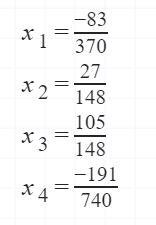
* 1. SPL Augmented
     + SPL 1 persamaan



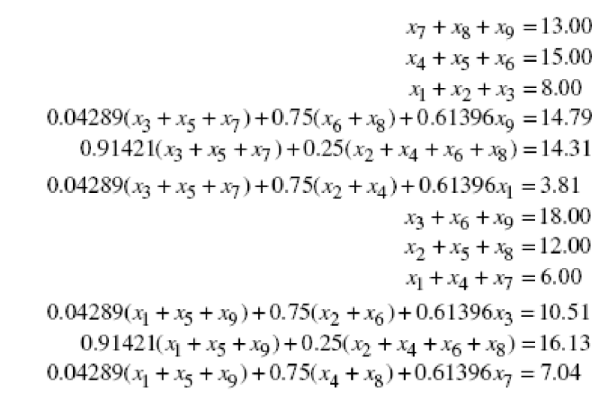
Dan dengan menggunakan metode Kramer, didapatkan solusinya:

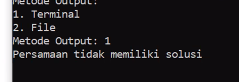


yang sama dengan solusi yang di kalkulator *online*:

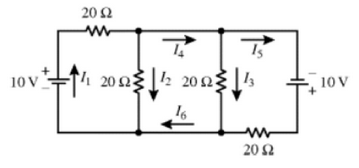


* + - SPL 2 persamaan

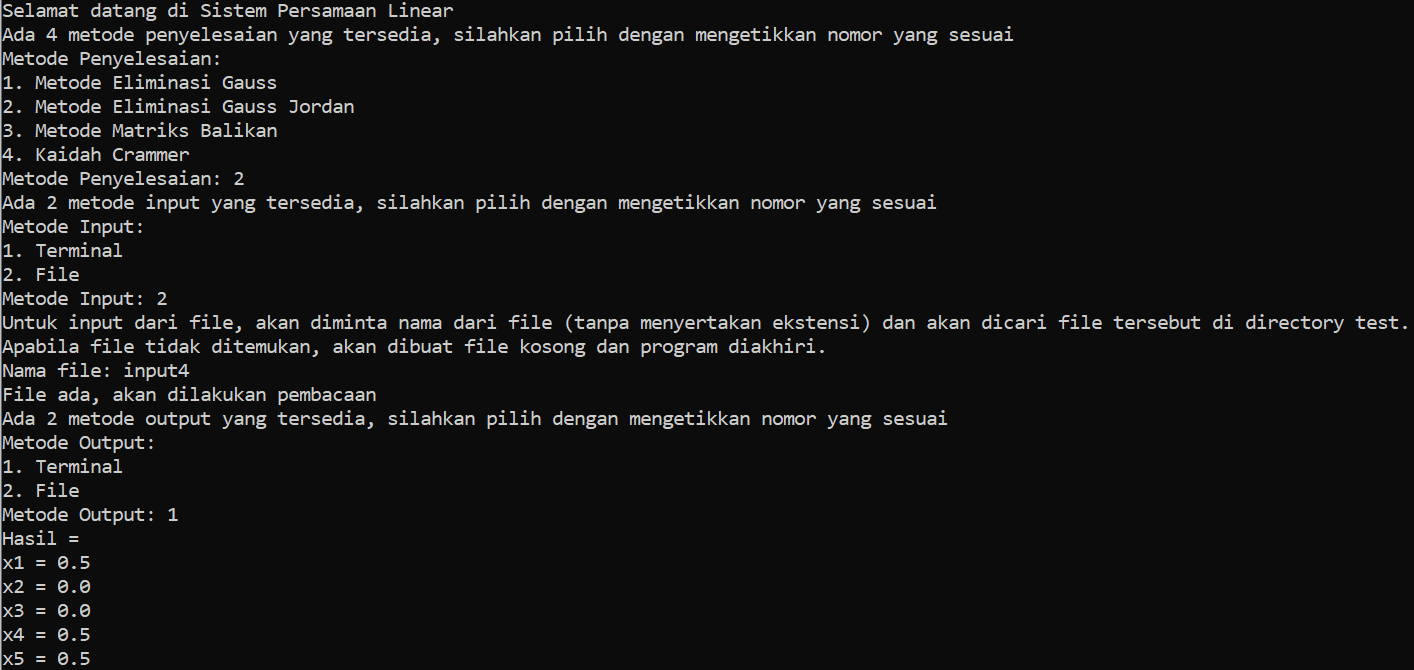




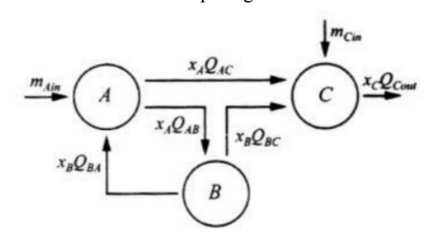
* 1. Berlaku hukum-hukum arus Kirchoff menyatakan bahwa jumlah aljabar dari semua arus yang memasuki suatu simpul haruslah nol. Diberikan sebuah rangkaian:



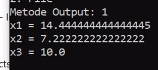
Setelah persamaan dibuat, dilakukan pemrosesan SPLnya, diperoleh, sebagai berikut



* 1. Reaktor

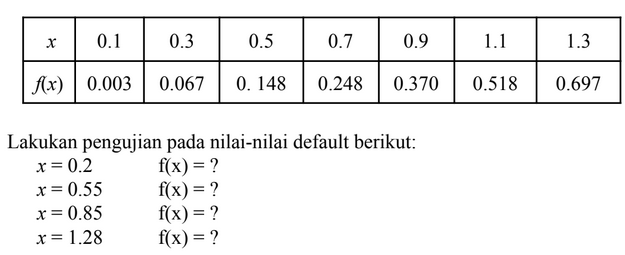


Setelah persamaan dibuat, dilakukan pemrosesan SPLnya, diperoleh, sebagai berikut

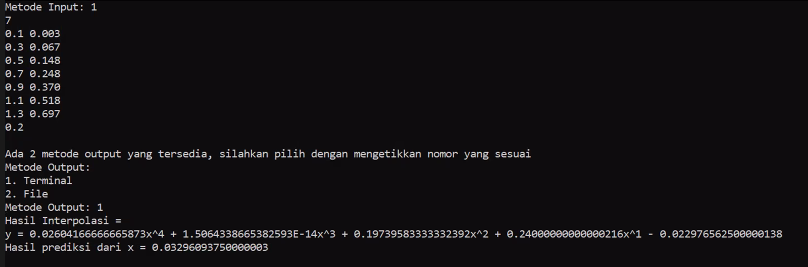


* 1. Untuk persamaan interpolasi

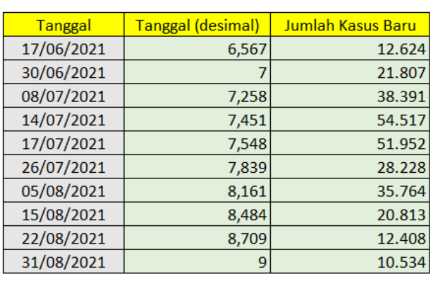
4.6.1 *testcase* (a)

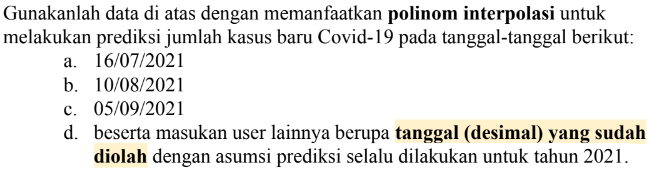


Dilakukan interpolasi, input dari terminal, dan output juga dalam terminal, diperoleh sebagai berikut



4.6.2 *testcase* (b)

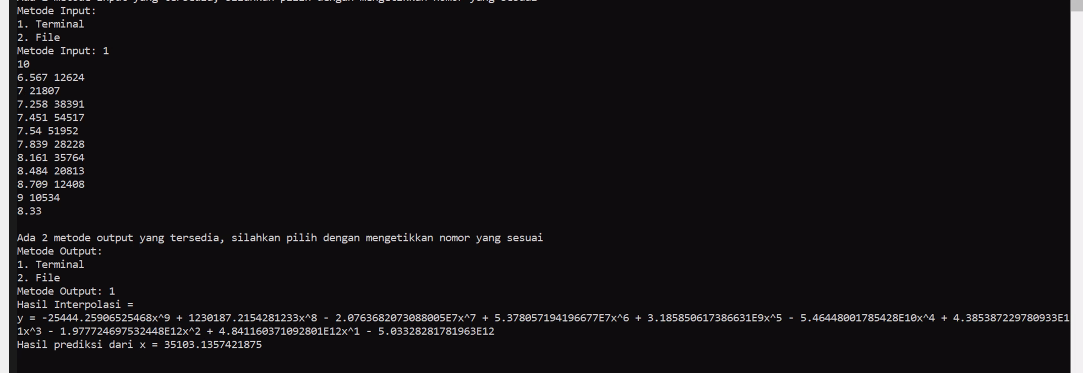




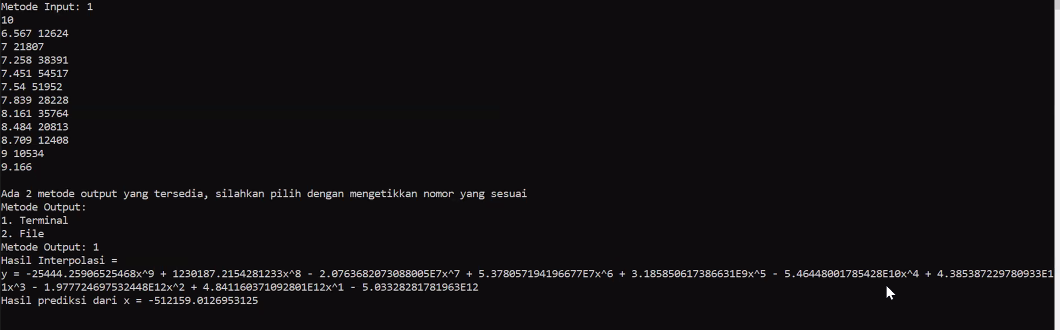
Dilakukan interpolasi, input dari terminal, dan output juga dalam terminal, diperoleh sebagai berikut, prediksi (a) juga ditampilkan di bawah ini



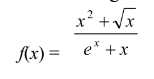
Dilakukan interpolasi, input dari terminal, dan output juga dalam terminal, diperoleh sebagai berikut, prediksi (b) juga ditampilkan di bawah ini



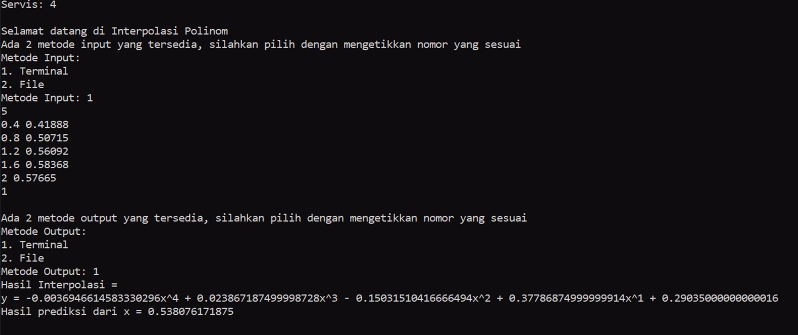
Dilakukan interpolasi, input dari terminal, dan output juga dalam terminal, diperoleh sebagai berikut, prediksi (c) juga ditampilkan di bawah ini



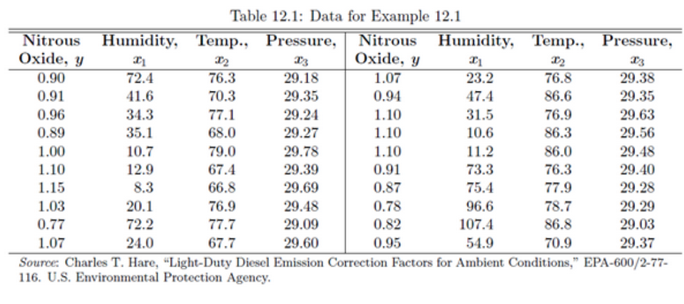
4.6.1 *testcase* (c)



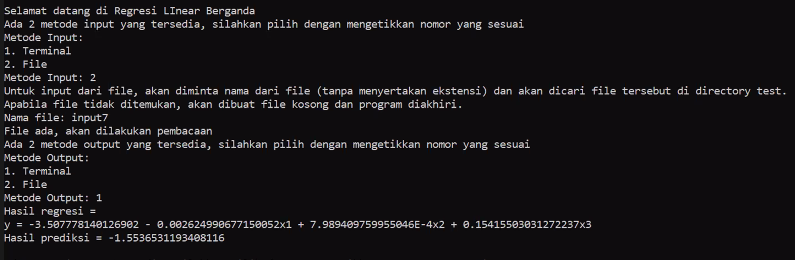
Dilakukan interpolasi, input dari terminal, dan output juga dalam terminal, diperoleh sebagai berikut, prediksi juga ditampilkan di bawah ini untuk x = 1



* 1. Untuk kasus regresi



Dilakukan regresi, input dari file, dan output dalam terminal, diperoleh sebagai berikut, prediksi juga ditampilkan di bawah ini



# BAB V KESIMPULAN

* 1. Kesimpulan

Banyak permasalahan matematis yang dapat diselesaikan dengan program ini, seperti memecahkan sistem persamaan linier (SPL) dari persamaan yg disusun dari suatu rangkaian listrik, atau interpolasi dari sekumpulan data yang bisa diproses hingga memperoleh fungsi yang lebih dekat dengan data, yang dapat diselesaikan dengan metode matriks.

* 1. Refleksi

Melalui tugas besar ini, kami memperoleh lebih banyak ilmu, mulai dari segi alur berpikir menyelesaikan permasalahan matematis, hingga lebih terbiasa dalam menggunakan bahasa Java. Kami juga semakin terbiasa untuk bekerja sama dalam tim, dengan menyamakan persepsi terhadap suatu permasalahan yang ada.

* 1. Saran

Pada pengerjaan program, ada beberapa hal yang tidak dalam perhatian kami, misalnya akan lebih baik jika dibuat ADT khusus matriks untuk simplifikasi dalam melakukan pemrograman. Akan lebih baik juga jika penggunaan *method* yang sudah dibuat dioptimasi agar tidak perlu dibuat *method*  lain yang sebenarnya sudah ada atau bisa dipakai untuk penyelesaian kasus tertentu.

# DAFTAR REFERENSI

<https://id.wikipedia.org/wiki/Matriks_(matematika)> (diakses dan diterjemahkan pada 30 September 2021)

<https://en.wikipedia.org/wiki/Gaussian_elimination> (diakses dan diterjemahkan pada 30 September 2021)

<https://id.wikipedia.org/wiki/Eliminasi_Gauss-Jordan> (diakses dan diterjemahkan pada 30 September 2021)

<https://id.wikipedia.org/wiki/Kaidah_Cramer> (diakses dan diterjemahkan pada 30 September 2021)

<https://www.madematika.net/2017/08/pengertian-minor-kofaktor-matriks.html> (diakses dan diterjemahkan pada 30 September 2021)