Laporan Tugas Besar 1 IF2123

Aljabar Linier dan Geometri

Sistem Persamaan Linier dan Aplikasinya

oleh

Kelompok 25

Hana Fathiyah (13520047)

Bayu Samudra (13520128)

Febryola Kurnia Putri (13520140)

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2021

BAB I
Deskripsi Masalah

Pada tugas besar satu mata kuliah Aljabar Linier dan Geometri ini, kami diminta untuk membuat sebuah program menggunakan bahasa Java yang bisa digunakan untuk hal-hal sebagai berikut:

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).

Berikut spesifikasi dari program:

1. Program dapat menerima masukan (input) baik dari *keyboard* maupun membaca masukan dari file text. Untuk SPL, masukan dari *keyboard* adalah 3m, n, koefisien a_{ij}, dan b_i . Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

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

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

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

3. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah n, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

   8.0 2.0794
   9.0 2.1972
   9.5 2.2513

4. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah n (jumlah peubah x), semua nilai-nilai x_{1i}, x_{2i} , ..., x_{ni}, nilai y_i , dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran (*output*) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$.)

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

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

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

9. Bahasa program yang digunakan adalah Java.

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

11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:
    MENU
    1. Sistem Persamaaan Linier
    2. Determinan
    3. Matriks balikan
    4. Interpolasi Polinom
    5. Regresi linier berganda
    6. Keluar

    Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:
    1. Metode eliminasi Gauss
    2. Metode eliminasi Gauss-Jordan
    3. Metode matriks balikan
    4. Kaidah Cramer

    Begitu juga untuk pilihan menu nomor 2 dan 3.

BAB II
Teori Singkat

1. Metode Eliminasi Gauss

Dalam metode eliminasi Gauss, SPL dinyatakan dalam bentuk matriks *augmented.* Setelah itu, OBE diterapkan pada matriks *augmented* sampai terbentuk matriks eselon baris.

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

Persamaan yang berkorespondensi pada matriks eselon baris dipecahkan dengan teknik penyulihan mundur (*backward substitution).*

2. Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan merupakan pengembangan metode eliminasi Gauss. Dalam hal ini, operasi baris elementer (OBE) diterapkan pada matriks *augmented*, sehingga menghasilkan matriks eselon baris tereduksi.

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

Metode ini tidak memerlukan lagi substitusi secara mundur untuk memperoleh nilai-nilai variabel. Nilai variavel langsung diperoleh dari matriks *augmented* akhir. Metode eliminasi Gauss-Jordan terdiri dari dua fase, yaitu fase maju (forward phase) dan fase mundur (*backward phase).* Untuk fase maju (*forward phase)* atau fase eliminasi Gauss, dihasilkan nilai-nilai 0 di bawah 1 utama.

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \sim \overset{\text{OBE}}{\cdots} \sim \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Untuk fase mundur (*backward phase),* dihasilkan nilai-nilai 0 di atas satu utama.

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \overset{R1-(3/2)R2}{\sim} \begin{bmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \underset{R2-(1/2)R3}{\overset{R1+(5/4)R3}{\sim}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Matriks eselon baris tereduksi

Melalui matriks *augmented* terakhir, diperoleh nilai x1 = 1, x2 = 2, dan x3 = 3.

3. Determinan

Ada beberapa aturan pada determinan. Misalkan A adalah matriks n x n. Matriks B adalah matriks yang diperoleh dengan memanipulasi matriks A. Jika sebuah baris pada A dikalikan dengan k, nilai det(B) = k det(A). Jika dua baris pada A dipertukarkan, nilai det(B) = -det(A). Jika sebuah baris pada A ditambahkan dengan k kali baris yang lain, nilai det(B) = det(A).

Cara menghitung determinan ada dua: reduksi baris dan ekspansi kofaktor. Untuk cara reduksi baris, determinan matriks diperoleh dengan melakukan OBE pada matriks A sampai diperoleh matriks segitiga (segitiga bawah atau atas.

$$[A] \overset{\text{OBE}}{\sim} [\text{matriks segitiga bawah}]$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \overset{\text{OBE}}{\sim} \begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \cdots & a'_{3n} \\ 0 & 0 & 0 & a'_{nn} \end{bmatrix}$$

$$\text{maka } \det(A) = (-1)^p \, a'_{11} a'_{22} \ldots a'_{nn} \qquad {}^{*)}$$

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

Ada beberapa teorema tentang determinan. Pertama, jika matriks A mengandung sebuah baris nol atau kolom nol, nilai det(A) = 0. Kedua, jika matriks A di-*transpose*, nilai determinan matriks A setelah di-*transpose* sama dengan nilai matriks A sebelum di-*transpose.* Ketiga, jika A = BC, nilai det(A) = det(B) det(C). Keempat, sebuah matriks hanya mempunyai balikan jika dan hanya jika nilai det(A) tidak sama dengan nol. Kelima, nilai determinan matriks A invers sama dengan 1/det(A)..

4. Matriks Balikan

JIka A dan B matriks bujur sangkar sedemikian rupa sehingga A^{-1}B = B^{-1}A = I , maka B disebut balikan atau *invers* dari A dan dapat dituliskan B = A^{-1} (B sama dengan *invers* A). Jika tidak ditemukan matriks B, maka A dikatakan matriks tunggal (singular). Karena karakteristik ini, kita dapat mencari solusi x pada Ax = B dengan mengalikan invers A dengan ruas kiri dan invers A dengan ruas kanan, sehingga didapatkan Ix = A^{-1}B ⇔ x = A^{-1}B. Metode yang digunakan untuk mencari matriks balikan adalah metode OBE, dengan cara meng-augmentasi-kan matriks A dan matriks identitasnya, lalu lakukan OBE sehingga bagian kiri matriks augment tersebut berbentuk matriks identitas.

5. Matriks Kofaktor

Matriks kofaktor adalah matriks yang dibentuk oleh elemen-elemen minor dari suatu matriks. Minor elemen pada baris ke-i dan kolom ke-j merupakan hasil penghitungan determinan submatriks dengan mengabaikan baris ke-i dan kolom ke-j pada matriks awal. Minor yang sudah dihitung dapat menentukan kofaktor dari elemennya. Jika elemen berada baris ke-i dan kolom ke-j maka kofaktor dari elemen itu = (-1)^$(i + j)$*minor elemen.

Dengan kata lain jika baris dan kolom sama-sama genap atau sama-sama ganjil maka kofaktornya sama dengan minornya. Sebaliknya, maka kofaktornya berlawanan tanda dengan minornya. Matriks kofaktor bisa digunakan untuk mencari determinan juga. Caranya dengan menjumlahkan perkalian elemen matriks dengan elemen kofaktornya yang bersesuaian dalam suatu baris atau kolom tertentu. Oleh karena itu ketika menghitung minor untuk mencari kofaktor dari suatu matriks, maka penghitungan kofaktor ini bersifat rekursif jika menghitung determinan dari submatriks-nya juga menggunakan metode kofaktor. Sifat rekursif ini memiliki basis ketika matriks/submatriks-nya berisi satu elemen dan berukuran 1x1.

Determinan dari matriks/submatriks berukuran 1x1 tersebut adalah elemen yang berada di dalamnya. Setelah bisa menentukan determinan dari suatu matriks menggunakan matriks kofaktornya, matriks kofaktor juga bisa digunakan untuk mencari invers dari matriks. Selanjutnya akan dijelaskan pada bagian matriks adjoin.

6. Matriks Adjoin

Matriks adjoin merupakan matriks yang dihasilkan dari matriks kofaktor yang telah ditranspose. Matriks ini dapat digunakan untuk mencari matriks invers. Caranya adalah dengan mengalikan matriks adjoin dengan 1/determinan matriks asli. Determinannya bisa dicari dengan menggunakan metode OBE atau metode matriks kofaktor juga.

7. Kaidah Cramer

Jika Ax = b adalah sistem persamaan linier yang terdiri dari n persamaan linier dengan n peubah (*variable)* sedemikian sehingga nantinya determinan matriks tidak sama dengan 0 karena determinan yang 0 akan menyebabkan hasil menjadi tidak terdefinisi. Solusi ini dinamakan dengan solusi unik, yaitu memiliki satu solusi yang pasti untuk setiap variabelnya. Kita dapat mencari solusi dari persamaan yaitu, x1,x2,x3,.....xn dengan cara berikut ini:

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

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

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

8. Interpolasi Polinom

Interpolasi polinom merupakan salah satu cara untuk memperkirakan suatu nilai. Jika terdapat n+1 titik, yaitu (x0,y0), (x1,y1), (x2,y2), … , (xn,yn), maka dapat ditentukan polinom pn(x) yang melewati semua titik tersebut sedemikian rupa sehingga yi= pn(xi) untuk setiap i = 0, 1, 2, … , n. Polinom yang melewati titik-titik tersebut berbentuk pn(x) = a0+ a1x + a2x2 + … + anxn. Misal jika hanya ada dua titik maka polinomnya adalah p1(x) = a0+ a1x yang berupa persamaan garis lurus. Lalu jika ada tiga titik maka polinomnya adalah p2(x) = a0 + a1x + a2x2 yang berupa persamaan kuadrat dan kurvanya berbentuk parabola, demikian seterusnya.

Dengan menyulihkan/mensubstitusi titik (xi,yi) ke dalam persamaan polinom pn(x) = a0 + a1x + a2x2 + … + anxn untuk i = 0,1,2…,n akan diperoleh n+1 buah SPL dalam a0, a1 , a2, …, an. Lalu dengan menggunakan metode eliminasi gauss, kita dapat menentukan polinom interpolasinya. Terakhir, dengan menggunakan polinom interpolasi tersebut, kita bisa memperkirakan nilai y di sembarang titik dengan x berada pada rentang [x0, xn].

9. Regresi Linear Berganda

Regresi Linear 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 β_i dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:

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

Sistem persamaan dapat diselesaikan dengan sistem eliminasi gauss. Sebagai contoh, jika terdapat matriks berukuran 3x3, yaitu

$$\begin{bmatrix} x_{11} & x_{21} & y_1 \\ x_{12} & x_{22} & y_2 \\ x_{13} & x_{23} & y_3 \end{bmatrix}$$

maka akan didapatkan matriks SPL sebagai berikut:

$$\begin{bmatrix} 3 & x_{11}+x_{12}+x_{13} & x_{21}+x_{22}+x_{23} & y_1+y_2+y_3 \\ x_{11}+x_{12}+x_{13} & x_{11}^2+x_{12}^2+x_{13}^2 & x_{11}x_{21}+x_{12}x_{22}+x_{13}x_{23} & x_{11}y_1+x_{12}y_2+x_{13}y_3 \\ x_{21}+x_{22}+x_{23} & x_{21}x_{11}+x_{22}x_{12}+x_{23}x_{13} & x_{21}^2+x_{22}^2+x_{23}^2 & x_{21}y_1+x_{22}y_2+x_{23}y_3 \end{bmatrix}$$

Untuk penyelesaian matriks ini dapat dilakukan dengan tahap sebagai berikut:

1. tambahkan 1 kolom di matriks paling kiri yang semua nilainya bernilai 1
2. hapus kolom paling kanan matriks setelah langkah 1 dilakukan menghasilkan matriks B
3. Lakukan transpos sebanyak 1 kali pada matriks tersebut yang menghasilkan matriks C
4. matriks SPL akhir didapatkan dari perkalian antara matriks B dengan matriks C yang menghasilkan matriks (k+1) x (k+2)
5. Selesaikan dengan menggunakan metode invers ataupun crammer yang akan menghasilkan (k+1) bilangan B0,...,Bk yang dapat diselesaikan pada fungsi regresi

BAB III
Implementasi Program

1. **Struktur Kelas**

   Program ini memiliki nama package tubes.algeo. Program ini memiliki 5 ADT yang didefinisikan pada paket tubes.algeo.lib.type. Berikut ini adalah ADT yang terdapat pada program ini:

   - Matriks

     Matriks merupakan ADT yang digunakan untuk menampung data berupa array 2 dimensi. Konstruktor dari ADT ini ada 4, yaitu konstruktor untuk matriks kosong, konstruktor untuk ADT persegi, konstruktor untuk adt persegi panjang, dan konstruktor penyalinan.

     Kelas ini memiliki beberapa selektor yaitu sebagai berikut:
     - int getNRows()
     - int getNCols()
     - public double[][] getMatrix()

`

     Kelas Ini memiliki sebuah predikat yaitu
     - public boolean isSquare()

     Kelas ini memiliki setter dan getter sebagai berikut
     - public void setElmt(int row, int col, double value)
     - public double getElmt(int row, int col)

     Kelas ini memiliki method yaitu sebagai berikut
     - public void copyMatrix(Matriks matriks)
     - public void transpose()
     - public Matriks plusMinusOBE(int rowTarget, int row, double factor)
     - public Matriks scaleRow(int rowTarget, double factor)
     - public void swapRow(int row1, int row2)
     - public Matriks cutOneRow(int row)
     - public Matriks cutOneCol(int col)
     - public void gantiOneColom(Matriks rep, int col)
     - public Matriks akhirDariAugmented()
     - public void addRowToRow(int addingRow, int addedRow, double multiplier)
     - public int toSegitigaAtas()
     - public double minorMatriks(int x, int y)
     - public Matriks kofaktor()
     - public Matriks plus(Matriks m) throws Exception
     - public Matriks minus(Matriks m) throws Exception
     - public Matriks product(Matriks m)
     - public double determinanByKofaktor()

- public double determinanByReduksi()
- public Matriks adjoinMatrix()
- public Matriks conMultiplyMatrix(double k)
- public Matriks augmentedKanan(Matriks right)
- public void eliminasiGauss()
- public void eliminasiGaussJordan()
- public Matriks inverseByKofaktor()
- public Matriks inverseByAugmented()
- private double methodCrammerHelper(Matriks sol, int col)
- public Matriks metodeCrammer() throws Exception
- public String getMatriksStr()

- Node

  Kelas ini merepresentasikan sebuah titik dalam ruang R^n. Kelas ini akan digunakan dalam menampung segala data yang berkaitan dengan Interpolasi dan regresi.

  Kelas ini memiliki sebuah konstruktor yang menerima x dan y. X dalam bentuk array. Hal ini bertujuan untuk menampung data yang memiliki banyak parameter. Nilai y merupakan hasil dari relasi terhadap parameter.

  Berikut ini adalah method yang berada pada kelas ini:
  - public int getXDimension()
  - public double[] getX()
  - public double getY()
  - public int compareTo(Node o)
  - public String toString()

- LinearFunction

  Linear function merupakan kelas yang merepresentasikan fungsi linear dengan parameter banyak. Fungsi ini digunakan untuk menampung hasil regresi dari data yang diinputkan.

  Kelas ini memiliki sebuah konstruktor yang menerima daftar koefisien sebuah fungsi linear peubah banyak. Nilai pertama pada array koefisien merupakan nilai konstanta dilanjutkan dengan koefisien peubah.

  Berikut ini beberapa method yang ada pada kelas ini:
  - public int getVariableNumber()
  - public double calculate(double[] value) throws Exception
  - public String getStrResult()
  - public double[] getCoefficients()

- Polynomial
  Kelas ini merepresentasikan fungsi polinom peubah tunggal. Kelas ini digunakan untuk menampung hasil interpolasi. Kelas ini memiliki satu konstruktor yang menerima array yang berisi nilai koefisien (dinamai constant). Koefisien diurutkan dari derajat terkecil dengan elemen pertama merupakan nilai konstanta dari polinomial.

  Berikut ini beberapa method yang ada pada kelas ini:
  - public int getDegree()
  - public double calculate(double x)
  - public String getPolynomialStr()

- SPLResult
  Kelas ini menampung hasil operasi pada proses pemecahan Sistem Persamaan Linear. Kelas ini memiliki sebuah konstruktor yang menerima matriks eselon tereduksi dan sebuah integer status. Status merupakan angka keadaan hasil dari perhitungan SPL. Berikut ini adalah angka keadaan dari hasil SPL
  - SPL tidak memiliki hasil memiliki angka keadaan -1
  - SPL memiliki hasil tunggal memiliki angka keadaan 0
  - SPL memiliki hasil banyak memiliki angka keadaan 1

  Berikut ini adalah method pada SPLResult
  - public boolean isNoSolution()
  - public boolean isOneSolution()
  - public boolean isManySolution()
  - public double[] getVariableValue() throws Exception
  - public String getStrResult()
  - public String getMatriksStr()

Selain tipe data diatas, program ini memiliki beberapa kelas eksepsi custom. Kelas ini didefinisikan pada package tubes.algeo.lib.error. Berikut ini adalah kelas-kelas yang dimaksud:

- InterpolasiException
  Eksepsi ini dilemparkan saat terjadi kesalahan saat menghitung hasil interpolasi.
- RegresiException
  Eksepsi ini dilemparkan saat terjadi kesalahan saat perhitungan hasil regresi.

Pada program ini terdapat dua buah kelas utilitas yang didefinisikan pada package tubes.algeo.lib.util. Berikut ini adalah utilitas yang ada:

- kelas floatingPoint
  Kelas ini digunakan untuk membandingkan antara dua buah floating point berbeda. Pada program ini, digunakan nilai toleransi perbedaan sebesar 1×10^{-9}.

- kelas terminalColor
  Kelas ini berisi konstanta-konstanta ajaib (*magic constant*) yang digunakan untuk mewarnai layar console.

Pada program ini terdiri beberapa sub-package diantaranya sebagai berikut:

- Package tubes.algeo.lib
  Paket ini berisi library-library dasar yang digunakan pada program ini. Package ini berisi beberapa kelas berikut
  - Kelas IO
    Kelas IO merupakan kelas statis yang berisikan beberapa prosedur yang digunakan untuk melakukan proses input/output pada program. Kelas ini terdiri dari berbagai metod statis yang dapat menangani proses pembacaan data dari file atau terminal ataupun penulisan output pada file atau terminal.

    Berikut ini adalah method statis yang ada:
    - public static Scanner readFile(String path) throws FileNotFoundException
    - public static Matriks readFileMatriks(String path) throws FileNotFoundException
    - public static Node[] readFileNodes(String path) throws FileNotFoundException
    - public static Node[] readFileNodes1(String path) throws FileNotFoundException
    - static FileWriter openWriteMode(String path) throws Exception
    - public static void writeTextFile(String path, String text) throws Exception
    - public static Matriks readMatrix()
    - public static Node[] readNodes()
    - public static Node[] readNodes1()
    - private static Node[] getNodes(Scanner stream, boolean showHint, boolean oneVariable)
    - private static Matriks getMatriks(Scanner stream, boolean showHints)

  - Kelas Interpolasi
    Kelas ini menangani proses perhitungan interpolasi. Kelas ini hanya memiliki sebuah method statis yaitu sebagai berikut:

    public static Polynomial getFunction(Node[] titik) throws InterpolasiException

  - Kelas Regresi
    Kelas ini menangani proses perhitungan regresi. Kelas ini hanya memiliki dua method statis yaitu sebagai berikut:
    - public static Matriks augmentedBuilder(Node[] nodes)

- public static LinearFunction getFunction(Node[] nodes) throws RegresiException

- Kelas SPL
  Kelas ini merupakan kelas yang menncari penyelesaian dari SIstem Persamaan linear. Kelas ini memiliki beberapa method statis diantaranya sebagai berikut:
    - public static int solutionChecker(Matriks augmented)
    - static Matriks backwardSubstitution(Matriks eselon)
    - static Matriks getCoefficientMatriks(Matriks augmented)
    - static Matriks getConstantaMatriks(Matriks augmented
    - public static SPLResult gaussElimination(Matriks augmented)
    - public static SPLResult gaussJordanElimination(Matriks augmented)
    - public static SPLResult cramer(Matriks augmented) throws Exception
    - public static SPLResult matriksInverseMethod(Matriks augmented) throws Exception

- Package tubes.algeo.menu
  Pada package ini berisi definisi method-method statis yang menangani tampilan pada output dan mengambilan data pada saat menginput data. Berikut ini adalah beberapa kelas yang ada pada package ini:
    - Kelas mainMenu
      Kelas ini berisi fungsi statis yang digunakan untuk menampilkan menu utama. Berikut ini adalah beberapa fungsi yang ada
        - public static void menu() throws Exception
        - private static void menuUtama() throws IOException

    - Kelas menuDeterminan
      Kelas ini yang menampilkan tampilan menu determinan. Berikut ini adalah beberapa fungsi statis yang ada pada fungsi ini:
        - public static void menu()
        - public static void menudeterminan()

    - Kelas menuInput
      Kelas ini bertanggung jawab atas pengambilan input dari user. Berikut ini adalah beberapa fungsi statis yang ada:
        - public static Matriks getMatriks(String operasi, String petunjuk)
        - private static int ShowTitle(String operasi, String petunjuk) throws InputMismatchException
        - public static Node[] getNodes1(String operasi, String petunjuk)
        - public static Node[] getNodes(String operasi, String petunjuk)
        - public static int pilihanOpsi(String[] pilihan, String petunjuk)

- Kelas menuInterpolasi
  Kelas ini digunakan untuk menampilkan menu interpolasi. Berikut ini adalah fungsi statis yang ada
  - public static void menu()
  - public static void menuinterpolasi()
  - static void hitungPrakiraanData(Polynomial pl)

- Kelas menuInverse
  Kelas ini digunakan untuk menampilkan menu untuk melakukan proses invers. Beriku tini adalah fungsi statis yang ada
  - public static void menu()
  - public static void menuinverse()

- Kelas menuOutput
  Kelas ini bertanggung jawab atas proses menyajikan data hasil perhitungan kepada user. Berikut ini adalah fungsi statis yang ada pada kelas ini:
  - static void showResult(String str)
  - static void showMatriksResult(Matriks mat)
  - static void showSPLDefinitonResult(SPLResult result)
  - static void showSPLMatriksResult(SPLResult result)
  - static void showLinearFuncDefinition(LinearFunction res)

- Kelas menuRegresi
  Kelas ini digunakan untuk menampilkan menu untuk melakukan regresi berganda. Berikut ini adalah fungsi statis yang ada:
  - public static void menu()
  - public static void menuregresi()

- Kelas menuSPL
  Kelas ini digunakan untuk menampilkan menu untuk melakukan pencarian solusi SPL. Berikut ini adalah fungsi statis yang ada:
  - public static void menu()
  - public static void menuspl()

Program ini memiliki main entry yang berada pada kelas App, Kelas App memiliki sebuah fungsi main entry, yaitu main. Fungsi ini akan memanggil fungsi mainMenu,menu() untuk menampilkan main menu.

2. **Garis Besar Program**
   Untuk menjalankan program ini, pengguna harus sudah mengistall java versi 17. Pengguna dapat menjalankan perintah berikut pada terminal:
   - Windows
     ./run.bat

- Linux/Mac
  chmod +x run.sh
  ./run.sh

- Makefile
  make run

Selain menggunakan perintah diatas, Pengguna juga dapat menjalankan file jar yang ada pada folder bin.

Setelah perintah dijalankan, Program akan menampilkan menu utama setelah perintah di atas dijalankan. Anda dapat memilih salah satu menu berikut:

1). Sistem Persamaan Linier
2). Determinan
3). Matriks balikan
4). Interpolasi Polinom
5). Regresi linier berganda
6). Keluar

Jawaban terhadap pilihan haruslah berupa bilangan bulat antara 1 dan 6 (inklusif).

Setelah memilih menu yang diinginkan, pengguna dihadapkan dengan pilihan input. INput dapat diambil dari file ataupun console. Jika user memilih console, muncul beberapa petunjuk yang dapat diikuti untuk mengisikan data. Jika user memilih untuk memasukan data menggunakan File, Pengguna perlu memperhatikan format sebagai berikut

- Format file untuk menu SPL, Determinan, dan Matriks Balikan

m n 11 12 13 14 ... 1n 21 22 23 24 ... 2n ... m1 m2 m3 m4 ... mn	Contoh: 2 3 1 2 3 4 5 6

  NIlai m menyatakan jumlah kolom dan n jumlah baris. Setelah itu diikuti oleh matriks yang ingin diinputkan.

- Format file untuk menu interpolasi

ndata x1 y1 x2 y2	Contoh: 3

... xn yn	1 2 2 3 0 4

Format file diawali dengan jumlah data yang diinginkan. Lalu dilanjutkan dengan pasangan x dan y yang diinginkan. Data boleh mengacak.

- Format file untuk menu regresi linear

nparams ndata x11 x12 x13 ... y1 x21 x22 x23 ... y2 ... xn1 xn2 xn3 ... yn	Contoh: 3 2 1 2 1 2 2 3 3 1

Format file diawali dengan jumlah parameter dari tiap data dan jumlah datanya. Setelah itu, dilanjutkan dengan data yang ingin diinpukan

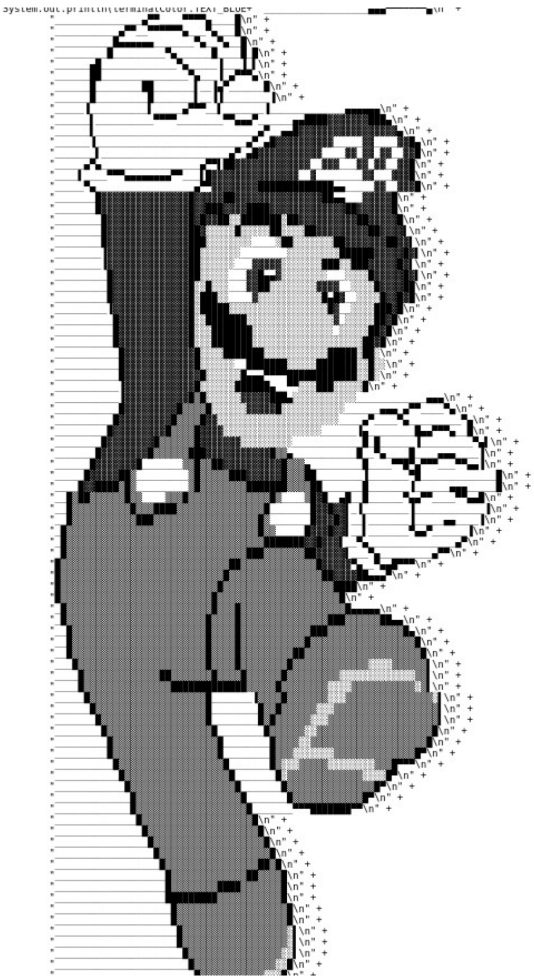
Setelah menginputkan data, bila user memilih menu 1, 2, dan 3 akan ditampilkan submenu untuk memilih algoritma yang ingin digunakan untuk menyelesaikan permasalahan.

Khusus untuk SPL, algoritma Matriks balikan dan Kaidah Cramer hanya menerima inputan dengan jumlah persamaan yang pas, yaitu jumlah persamaan sama dengan jumlah variabel. Selain itu, matriks harus tidak memiliki determinan yang 0. Bila kedua syarat ini tidak dipenuhi, akan diberikan error.

Bila proses perhitungan selesai, pengguna dapat memilih untuk menyimpan hasil perhitungan atau ditampilkan saja pada terminal. Bila sudah ditampilkan, user dapat menekan enter untuk kembali ke menu utama.

Untuk keluar dari program, user dapat memilih menu 6 pada menu utama.

BAB IV
Eksperimen



1. **Kasus 1 (Menemukan solusi dari SPL dalam bentuk Ax = b)**

```
Masukkan pilihan anda: 1
```

```
================================================================
****************************************************************
Operasi yang dipilih : Sistem Persamaan Linear

Petunjuk :
Masukkan matriks augmented yang diinginkan

Silahkan pilih mode membaca data yg diinginkan:
1. Baca dari File
2. Baca dari Konsol
****************************************************************
================================================================
Masukkan pilihan Anda: 1
```

```
Masukkan pilihan Anda: 1
Masukkan path file input:
```

```
================================================================
****************************************************************
Menu Input Sistem Persamaan Linear
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks balikan
4. Kaidah Cramer
****************************************************************
================================================================
Masukkan pilihan Anda:
```

```
Pilih mode menampilkan solusi:
1. Simpan pada File
2. Tampilkan pada layar

Masukkan pilihan Anda:
```

a. Masukan

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$

```
4 5
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6
```

Keluaran (Program)

Metode Eliminasi Gauss: Persamaan memiliki satu buah solusi

```
Solusi :
x_1 = 0.6666666666666667; x_2 = -2.666666666666667; x_3 = -1.0
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada nomor 1a memiliki satu buah solusi.

b. Masukan

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

```
4 6
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 0 1 -1 5
-1 2 0 -2 -1 -1
```

Keluaran (Program)

Metode Eliminasi Gauss: Persamaan memiliki banyak solusi

```
Solusi :
x_1 = -4.0; x_2 = -7.0 + r_4; x_3 = 6.0 + 1.5*r_4; x_4 = -5.666666666666667 + 0.3333333333333333*r_4; x_5 = r_4
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada nomor 1b memiliki banyak solusi.

c. Masukan

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

```
3 7
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1
```

Keluaran (Program)

Metode Eliminasi Gauss-Jordan: Persamaan memiliki banyak solusi

```
Solusi :
x_1 = r_0; x_2 = 1.0 - r_5; x_3 = r_2; x_4 = -2.0 - r_5; x_5 = 1.0 + r_5; x_6 = r_5
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada nomor 1c memiliki banyak solusi.

d. Masukan

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

```
\Algeo01-20047\test\tc> python .\HilbertBuilder.py
```

untuk n = 6

```
6 7
1.0 0.5 0.3333333333333333 0.25 0.2 0.16666666666666666 1
0.5 0.3333333333333333 0.25 0.2 0.16666666666666666 0.14285714285714285 0
0.3333333333333333 0.25 0.2 0.16666666666666666 0.14285714285714285 0.125 0
0.25 0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0
0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0
0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0
```

untuk n = 10

```
10 11
1.0 0.5 0.3333333333333333 0.25 0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 1
0.5 0.3333333333333333 0.25 0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.090909090909
0.3333333333333333 0.25 0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.0909090909090909
0.25 0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.083333333333333
0.2 0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.
0.16666666666666666 0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.0769
0.14285714285714285 0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.0714
0.125 0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.07142857142857142 0.0666
0.1111111111111111 0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.07142857142857142 0.0666666666
0.1 0.09090909090909091 0.08333333333333333 0.07692307692307693 0.07142857142857142 0.06666666666666667 0.0625 0.05
```

Keluaran (Program)

untuk n = 6

Metode Eliminasi Gauss-Jordan

```
Solusi :
x_1 = 36.000000000981274; x_2 = -630.0000000294922; x_3 = 3360.0000002058987; x_4 = -7560.000000547128; x_5 = 7560.000000613376; x_6
= -2772.00000024457
```

untuk n = 10

Metode Eliminasi Gauss-Jordan

```
Solusi :
x_1 = 78.94734705139106; x_2 = -3074.2090397502466; x_3 = 38273.659437811315; x_4 = -221273.5144539274; x_5 = 684245.7394900108; x_6
= -1185393.719790316; x_7 = 1112689.683356879; x_8 = -460610.42294385814; x_9 = -13547.088273327332; x_10 = 48619.87221454781
**** Silahkan tekan enter untuk melanjutkan program ****
```

Berdasarkan program yang kami buat, kedua persamaan pada nomor 1d memiliki banyak solusi.

2. **Kasus 2 (Menemukan solusi dari SPL dalam bentuk matriks *augmented*)**

   a. Masukan

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

```
4 5
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
```

Keluaran (Program)

Metode Eliminasi Gauss: Persamaan memiliki banyak solusi

```
Solusi :
x_1 = -1.0 + r_3; x_2 = 2.0*r_2; x_3 = r_2; x_4 = r_3
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada nomor 2a memiliki banyak solusi.

b. Masukan

```
6 5
2 0 8 0 8
0 1 0 4 6
-4 0 6 0 6
0 -2 0 3 -1
2 0 -4 0 -4
0 1 0 -2 0
```

Keluaran (Program)

Metode Eliminasi Gauss: Persamaan memiliki satu buah solusi

```
Solusi :
x_1 = 0.0; x_2 = 2.0; x_3 = 1.0; x_4 = 1.0
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada nomor 2b memiliki satu buah solusi.

3. Kasus 3 (Menemukan solusi dari SPL dalam bentuk persamaan berikut)

a. Masukan

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

```
4 5
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
```

Keluaran (Program)

Metode matriks balikan: Persamaan memiliki satu buah solusi

```
Solusi :
x_1 = -0.22432432432432428; x_2 = 0.1824324324324324; x_3 = 0.7094594594594593; x_4 = -0.2581081081081081
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada nomor 3a memiliki satu buah solusi.

b. Masukan

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

Keluaran (Program)

Metode Gauss: Tidak memiliki solusi

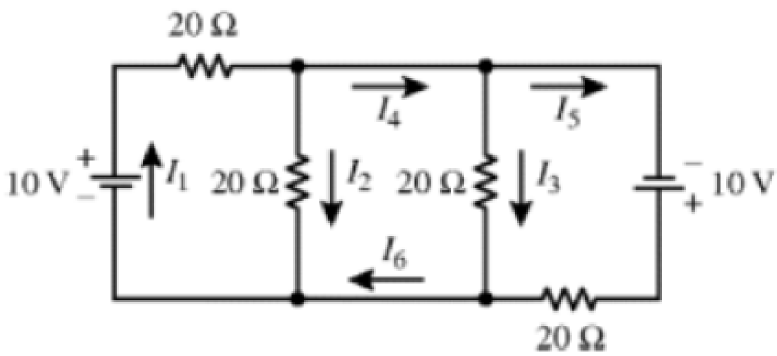
```
Solusi :
Persamaan tidak memiliki solusi
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, persamaan pada soal nomor 3b tidak memiliki solusi.

4. Kasus 4 (Menentukan nilai arus yang mengalir dengan hukum-hukum ohm)

Masukan



```
6 7
20 20 0 0 0 0 10
0 20 -20 0 0 0 0
0 0 -20 0 20 0 10
1 -1 0 -1 0 0 0
0 0 -1 1 -1 0 0
0 0 0 1 0 -1 0
```

Keluaran (Program)

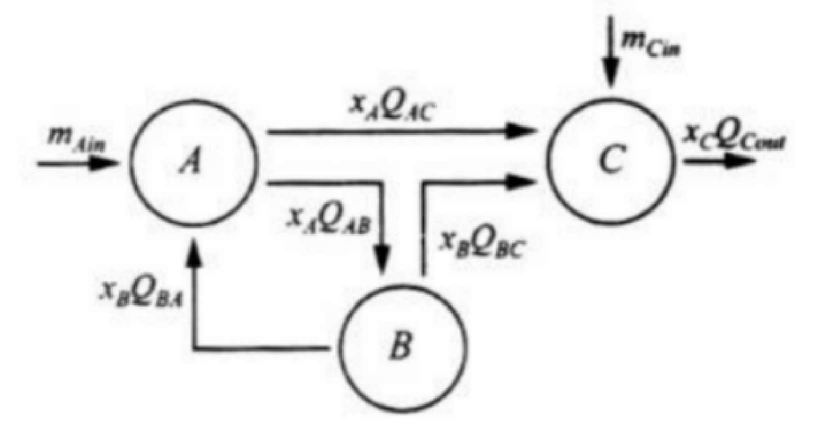
Metode Cramer: Memiliki satu buah solusi

```
Solusi :
x_1 = 0.5; x_2 = 0.0; x_3 = 0.0; x_4 = 0.5000000000000001; x_5 = 0.5; x_6 = 0.5
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis

Berdasarkan program yang kami buat, diperoleh nilai I1 = 0.5A, I2 = 0A, I3 = 0A, I4 = 0.5A, I5 = 0.5A, dan I6 = 0.5A

5. **Kasus 5 (Menemukan solusi dari SPL pada permasalahan reaktor)**



Dengan laju volume Q dalam m^3/s dan input massa min dalam mg/s. Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

$$\text{A:}\quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$$

$$\text{B:}\quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0$$

$$\text{C:}\quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0$$

Tentukan solusi xA, xB, xC dengan menggunakan parameter berikut : QAB = 40, QAC = 80, QBA = 60, QBC = 20 dan QCout = 150 m3 /s dan mAin = 1300 dan mCin = 200 mg/s.

```
3 4
-120 60 0 -1300
40 -80 0 0
80 20 -150 -200
```

Keluaran:

Metode Cramer: Persamaan memiliki satu buah solusi

```
Solusi :
x_1 = 14.444444444444445; x_2 = 7.222222222222222; x_3 = 9.999999999999998
**** Silahkan tekan enter untuk melanjutkan program ****
```

Analisis: Berdasarkan program yang kami buat, nilai xA = 14.44, xB = 7.22, dan xC = 10

6. Kasus 6 (Studi kasus interpolasi)

```
INTERPOLASI

=================================================================
*****************************************************************
Operasi yang dipilih : Interpolasi

Petunjuk :
Masukan data yang akan diinterpolasikan

Silahkan pilih mode membaca data yg diinginkan:
1. Baca dari File
2. Baca dari Konsol
*****************************************************************
=================================================================
```

```
=================================================================
*****************************************************************
Masukan aksi yang ingin Anda lakukan

1. Menampilkan fungsi hasil interpolasi
2. Menghitung prakiraan data
3. Kembali ke menu utama
```

a.

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

```
7
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
```

```
Solusi :
p(x) = -0.022976562500000675 + 0.24000000000001212*x + 0.197395833333264*x^2 + 1.7853625840461486E-13*x^3 + 0.026041666666437653*x^4
+ 1.4261776077240953E-13*x^5 - 3.4333068795385814E-14*x^6
```

untuk x = 0.2, f(x) = 0.033

```
Solusi :
0.032960937500000009
```

untuk x = 0.55, f(x) = 0.171

```
Solusi :
0.17111865234375
```

untuk x = 0.85, f(x) = 0.337

```
Solusi :
0.3372358398437499
```

untuk x = 1.28, f(x) = 0.678

```
Solusi :
0.6775418375
```

b. Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2021 hingga 31 Agustus 2021

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2021	6,567	12.624
30/06/2021	7	21.807
08/07/2021	7,258	38.391
14/07/2021	7,451	54.517
17/07/2021	7,548	51.952
26/07/2021	7,839	28.228
05/08/2021	8,161	35.764
15/08/2021	8,484	20.813
22/08/2021	8,709	12.408
31/08/2021	9	10.534

```
10
6.567 12.624
7 21.807
7.258 38.391
7.451 54.517
7.548 51.952
7.839 28.228
8.161 35.764
8.484 20.813
8.709 12.408
9 10.534
```

```
Solusi :
p(x) = 7.200451993714008E9 - 9.36254938586638E9*x + 5.342227792955563E9*x^2 - 1.7592218844233153E9*x^3 + 3.690161604088837E8*x^4 - 5.119166377063442E7*x^5 + 4700920.749605452*x^6 - 275755.4467855891*x^7 + 9381.838173084954*x^8 - 141.12139609648747*x^9
```

a. 16/07/2021

```
Masukkan tanggal desimal: 16/07/2021
tanggal(desimal) = 7.516
```

```
Solusi :
53.53946876525879
```

Prediksi jumlah kasus = 53.539

b. 10/08/2021

```
Masukkan tanggal desimal: 10/08/2021
tanggal(desimal) = 8.323
```

```
Solusi :
36.29771041870117
```

c. 05/09/2021

```
Masukkan tanggal desimal: 05/09/2021
tanggal(desimal) = 9.167
```

```
Solusi :
-667.8649978637695
```

d. 01/10/2021

```
Masukkan tanggal desimal: 01/10/2021
tanggal(desimal) = 10.032
```

```
Solusi :
-250609.33847045898
```

c.

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

dengan polinom interpolasi derajat n di dalam selang [0, 2]. Sebagai contoh, jika n = 5, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak h = (2 – 0)/4 = 0.5.

```
5
0 0.0
0.5 0.44543086822735684
1.0 0.5378828427399902
1.5 0.580896939063937
2.0 0.576651529751722
```

```
Solusi :
p(x) = 0.0 + 1.5968826286735516*x - 1.8655379972662944*x^2 + 1.0074466399817417*x^3 - 0.2009084286490085*x^4
```

f(0) = 0

```
Solusi :
0.0
```

f(0.4) = 0.3996

```
Solusi :
0.3996003010922303
```

f(0,8) = 0.517

```
Solusi :
0.5170823719844306
```

f(1.2) = 0.554

```
Solusi :
0.5541485145866636
```

f(1.6) = 0.589

```
Solusi :
0.58906289224704
```

f(2.0) = 0.577

```
Solusi :
0.576651529751723
```

7. Kasus 7 (Studi kasus regresi linear)

```
Masukkan pilihan anda: 5
REGRESI

==============================================================
**************************************************************
Operasi yang dipilih : Regresi

Petunjuk :
Masukan data yang akan dihitung regresinya

Silahkan pilih mode membaca data yg diinginkan:
1. Baca dari File
2. Baca dari Konsol
**************************************************************
==============================================================
Masukkan pilihan Anda:
```

```
==============================================================
**************************************************************
Masukan aksi yang ingin anda lakukan

1. Menampilkan persamaan regresi
2. Menghitung prakiraan data
3. Kembali ke menu utama
Pilihan : 1
```

```
Pilih mode menampilkan solusi:
1. Simpan pada File
2. Tampilkan pada layar

Masukkan pilihan Anda: 2
```

Table 12.1: Data for Example 12.1

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

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

Persamaan regresi berdasarkan tabel:

```
Solusi :
f = -3.5077211462089144 - 0.0026242866445300907*x_1 + 7.980337380556002E-4*x_2 + 0.15415486581895818*x_3
```

Estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30

```
*** Menghitung nilai dari hasil regresi ***
Masukkan nilai variabel:
50 76 29.30
```

```
Solusi :
0.9384526541522815
```

Berdasarkan program yang kami buat, nilai Nitrous Oxide adalah sebesar 0.938

BAB V
Kesimpulan

1. Kesimpulan

Program matriks telah selesai dan berhasil diimplementasikan semua fungsinya yang sesuai dengan definisi dan materi matriks pada umumnya sebagaimana yang telah dipelajari pada mata kuliah Aljabar Linear dan Geometri. Hal-hal yang diimplementasikan mengenai matriks pada program ini adalah sebagai berikut:

a. Sifat-Sifat Matriks

Bagian ini berisikan implementasi dari fungsi matriks persegi, matriks identitas, dan matriks eselon pada program.

b. Operasi Elementer

Bagian ini berisikan fungsi-fungsu mengenai pertukaran kolom pertukaran baris, perkalian baris dengan konstanta, serta penambahan dan pemotongan baris/kolom.

c. Operasi Kompleks

Bagian ini berisikan materi-materi yang kompleks yaitu reduksi eselon(gauss dan gauss-jordan), invers matriks, interpolasi, dan regresi.

d. Operasi Lainnya

Bagian ini terdiri atas baris, kolom, determinan, kofaktor, adjoin, dan lain sebagainya.

Semua implementasi yang digunakan pada program ini kemudian berhasil digunakan untuk menyelesaikan metode yang ada dalam spesifikasi, diantaranya penyelesaian Sistem Persamaan Linear dengan metode eliminasi Gauss, Gauss Jordan, matriks balikan, dan cramer. Selanjutnya metode Sistem Persamaan Linear ini digunakan sebagai dasar dalam mencari koefisien polinom dan fungsi linear yang tepat berturut-turut dalam metode Interpolasi dan Regresi Linear Berganda.

2. Saran

Implementasi matriks pada program memerlukan banyak primitif-primitif lain yang membangunnya, mulai dari sifat, karakteristik, serta fungsi-fungsinya. Oleh karena itu, implementasi matriks akan lebih sesuai dalam struktur yang berorientasi objek karena akan menghasilkan struktur data yang lebih rapi dan berstruktur. Selain itu, dalam pengimplementasian setiap primitif-primitif yang diperlukan sangat direkomendasikan untuk melakukan test case untuk setiap kasus berkait, agar setiap fungsi dalam matriks dapat berjalan dan sesuai ekspektasi karena pada program ini, setiap fungsi atau prosedur yang dilakukan memiliki keterkaitan dengan fungsi ataupun prosedur yang lainnya. Oleh karena itu, jika satu fungsi memiliki masalah maka fungsi lainnya akan ikut bermasalah karena keduanya akan saling terkait satu sama lain.

3. Refleksi

Belajar mengimplementasikan matriks dalam suatu program merupakan hal yang menarik karena matriks memiliki banyak struktur data dan primitif-primitif di dalamnya. Setelah mengerjakan tugas ini, timbul apresiasi dari kami untuk semua yang telah terlibat dalam pembagian materi matriks ataupun tugas besar ini karena melalui tugas ini kami mempelajari banyak hal baru, seperti penggunaan bahasa baru, yaitu bahasa java yang belum kami kuasai sebelumnya. Selain itu, dalam mengerjakan tugas ini, kami menyadari banyak kegunaan dan manfaat dari matriks dalam kehidupan sehari-hari yang dapat kami terapkan. Hal yang paling penting adalah dalam pengerjaan tugas ini, kami merasa dapat memperbaiki kinerja kami dalam berkelompok mulai dari pembagian tugas ataupun *timeline* untuk mengejar *deadline*. Dengan *timeline* dan pembagian tugas yang jelas ini kami dapat berkontribusi bersama dengan baik dan mengumpulkan tugas dengan baik dan tepat waktu.

DAFTAR REFERENSI

Slide kuliah IF2123 Aljabar Linear dan Geometri tahun ajaran 2021/2022.