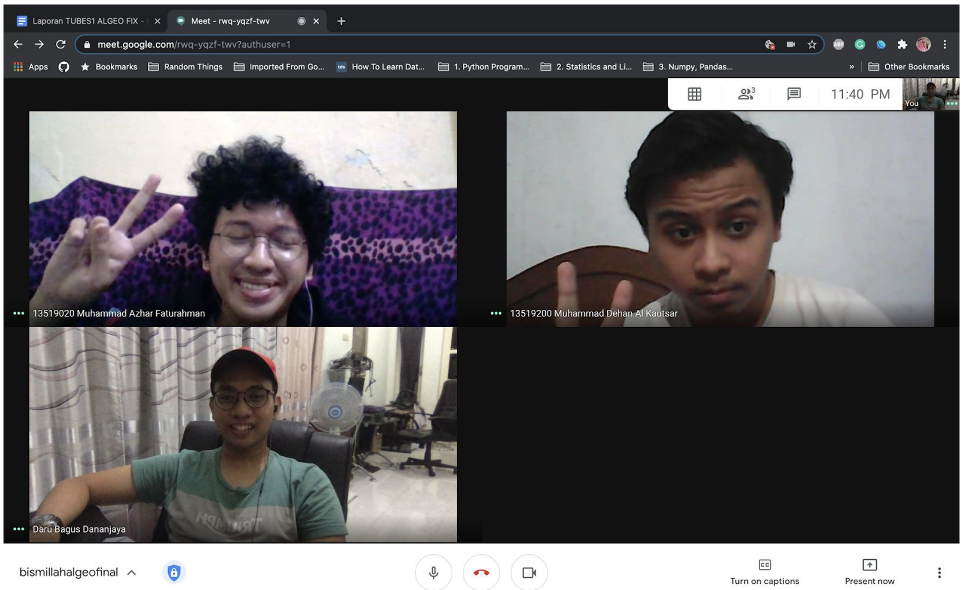
LAPORAN TUGAS BESAR I

IF2123 ALJABAR LINIER DAN GEOMETRI



Laporan ini dibuat untuk memenuhi tugas

Mata Kuliah IF 2123 Aljabar Linier dan Geometri

Disusun Oleh :

Kelompok 40

Muhammad Azhar Faturahman (13519020)

Daru Bagus Dananjaya (13519080)

Muhammad Dehan Al Kautsar (13519200)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER I TAHUN 2020/2021

DAFTAR ISI

BAB I

DESKRIPSI MASALAH

Buatlah program dalam Bahasa Java untuk :

1. Menghitung solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan).
2. Menyelesaikan persoalan interpolasi dan regresi linier.
3. Menghitung matriks balikan.
4. Menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program adalah sebagai berikut:

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

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

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

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

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah *n*, (*x0*, *y0*), (*x1*, *y1*), ..., (*xn*, *yn*), dan nilai *x* yang akan ditaksir nilai fungsinya. Jika masukannya dari file,

maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

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

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), semua nilai-nilai $x1i$, $x2i$, ..., xni, nilai yi, dan nilai-nilai xk yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. Untup persoalan SPL, luaran (output) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x4 = -2$, $x3 = 2s - t$, $x2 = s$, dan $x1 = t$.)
6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing
7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.
8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.
9. Bahasa program yang digunakan adalah Java.
10. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

```
MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar
```

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

```
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
```

BAB II

TEORI SINGKAT

A. Abstraksi

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

$$a11\ x1 + a12\ x2 + ... + a1n\ xn = b1$$

$$a21\ x1 + a22\ x2 + ... + a2n\ xn = b2$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$am1\ x1 + am2\ x2 + ... + amn\ xn = bm$$

yang dalam hal ini *xi* adalah peubah, *aij* dan *bi* adalah koefisien ∈ R. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan (*x* = *A-1b*), dan kaidah *Cramer* (khusus untuk SPL dengan *n* peubah dan *n* persamaan). Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

Sebuah matriks M berukuran $n \times n$:

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

determinannya adalah :

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

Determinan matriks M berukuran $n \times n$ dapat dihitung dengan beberapa cara: reduksi baris dan ekspansi kofaktor.

SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

B. Operasi Baris Elementer (OBE), Metode Eliminasi Gauss, dan Metode Eliminasi Gauss-Jordan

Operasi baris elementer (OBE) merupakan operasi aritmatika (penjumlahan dan perkalian) yang dikenakan pada setiap unsur dalam suatu baris pada sebuah matriks.

Operasi baris elementer meliputi :

1. Pertukaran Baris
2. Perkalian suatu baris dengan konstanta tak nol
3. Penjumlahan hasil perkalian suatu baris dengan konstanta tak nol (seperti butir 2) dengan baris yang lain.

Tujuan dilakukan operasi baris elementer pada suatu matriks adalah menghasilkan matriks yang memenuhi beberapa sifat berikut :

1. Pada baris tak nol maka unsur tak nol pertama adalah 1 (membuat satu utama).
2. Pada baris yang berurutan, baris yang lebih rendah memuat 1 utama yang lebih ke kanan.
3. Jika ada baris nol (baris yang semua unsurnya nol), maka ia diletakkan pada baris paling bawah.
4. Pada kolom yang memuat unsur 1 utama, maka unsur yang lainnya adalah nol.

Jika butir 1, 2, dan 3 dipenuhi, maka matriks hasil OBE dinamakan berbentuk eselon baris (prosesnya dinamakan eliminasi Gauss). Sementara itu, jika semua poin dipenuhi matriks dinamakan berbentuk eselon baris tereduksi (prosesnya dinamakan eliminasi Gauss-Jordan).

C. Determinan

Determinan merupakan pemetaan dengan domain berupa matriks bujur sangkar, sementara kodomain berupa suatu nilai skalar.

Determinan sering digunakan untuk :

1. Memeriksa keberadaan invers matriks
2. Menentukan solusi sistem persamaan linier dengan metode cramer
3. Pemeriksaan basis suatu ruang vektor, dsb.

$$A = \begin{pmatrix} a_{11} & a_{12} \dots a_{1n} \\ a_{21} & a_{22} \dots a_{2n} \\ \vdots & \vdots \quad \vdots \\ a_{n1} & a_{n2} \dots a_{nn} \end{pmatrix}$$

Notasi determinan dari matriks A ditulis :

$$\textbf{det (A)} \text{ atau } \begin{vmatrix} a_{11} & a_{12} \dots a_{1n} \\ a_{21} & a_{22} \dots a_{2n} \\ \vdots & \vdots \quad \vdots \\ a_{n1} & a_{n2} \dots a_{nn} \end{vmatrix} \text{ atau } |A|.$$

D. Matriks Kofaktor dan Adjoin

Misalkan A adalah matriks *n x n* :

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

Didefinisikan :

Mij = minor entri *aij*

Mij = determinan upa-matriks (submatrix) yang elemen elemennya tidak berada pada baris i dan kolom j

Cij = (-1)^(i +j) Mij $\rightarrow$ kofaktor entry *aij*

Kofaktor *Cij* berkorespondensi dengan minor entri *Mij* , hanya berbeda tanda positif atau negatif , tergantung nilai i dan j.

Maka matriks kofaktor dari A adalah :

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

Adjoin dari A adalah transpose matriks kofaktor A :

adj(A) = transpose matriks kofaktor

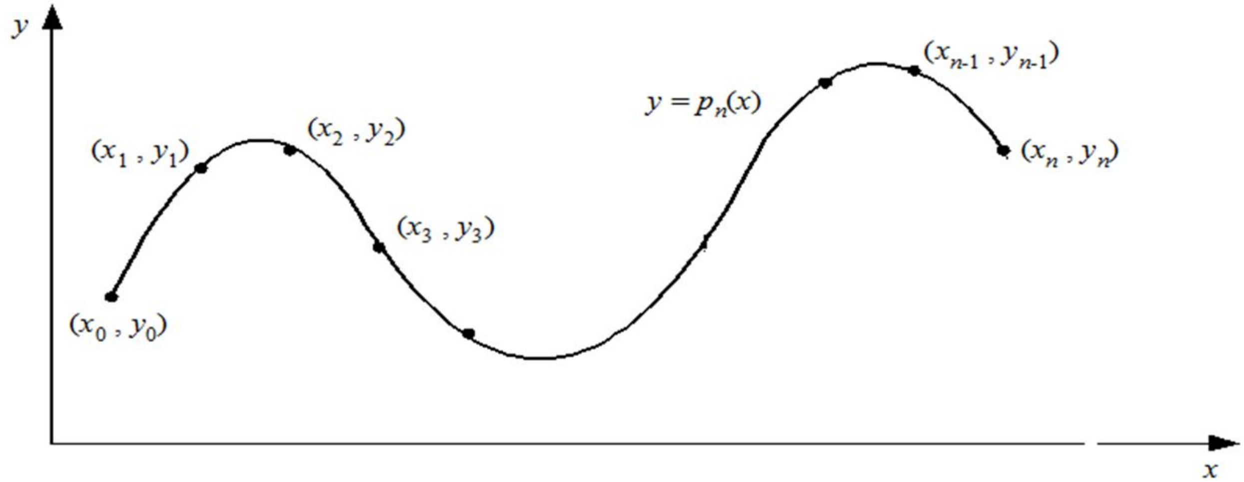
E. Kaidah Cramer

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

$x_1 = \frac{det(A1)}{det(A)}$, $x_2 = \frac{det(A2)}{det(A)}$, ... , $x_n = \frac{det(An)}{det(A)}$

F. 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 + ... + an x0n = y0

a0 + a1x1 + a2x12 + ... + an x1n = y1

...   ...

a0 + a1xn + a2xn2 + ... + an xnn = 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.

G. Regresi Linier Berganda

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

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

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

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB III
IMPLEMENTASI PROGRAM

Pada tugas besar ini, kami mendeklarasikan 5 buah class, yaitu :

1. Matriks.java

   Pada class ini, dideklarasikan atribut matriks berupa :

   - maxNBrsKol yang berfungsi sebagai indikator ukuran maksimum dari sebuah matriks
   - Matriks[][] yang berfungsi sebagai kontainer
   - NBrsEff yang berfungsi sebagai indikator ukuran baris matriks aktif
   - NKolEff yang berfungsi sebagai indikator ukuran kolom matriks aktif

Method	Deskripsi
Matriks(int NBrsEff, int NKolEff)	Konstruktor pembentuk matriks dinamis
void BacaMatriks()	Membaca matriks dari input user
void bacaFileMatriks()	Membaca matriks dari file .txt
void tulisMatriks()	Menampilkan matriks ke layar
void tulisFileMatriks(String namaFile)	Menyimpan matriks ke dalam suatu file
int NBElmt()	Menghasilkan nilai efektif sebuah matriks yang berupa (NBrsEff*NKolEff)
void doubleMatriks()	Mengubah ukuran maksimum baris dan kolom matriks menjadi 2x ukuran semula
void halfMatriks()	Mengubah ukuran maksimum baris dan kolom matriks menjadi 0.5x ukuran semula
float[][] copyArrayMatriks()	Duplikasi array matriks ke dalam array lain
Matriks copyMatriks	Duplikasi matriks ke dalam matriks lain
void transpose()	Melakukan transpose terhadap matriks

Matriks KaliMatriks()	Melakukan perkalian dua buah matriks dengan prekondisi Kolom(M1)=Baris(M2)
Float GetElmt(final int i, final int j)	Mengembalikan elemen matriks pada indeks baris ke-i dan kolom ke-j
void SetElmt(final int i, final int j, final float value)	Melakukan set elemen pada indeks ke-i dan kolom ke-j dengan nilai value
int GetFirstIdxBrs()	return indeks baris pertama, yaitu 0
int GetFirstIdxKol()	return indeks kolom pertama, yaitu 0
int GetLastIdxBrs()	return indeks baris terakhir, yaitu NBrsEff-1
int GetLastIdxKol()	return indeks kolom terakhir, yaitu NKolEff-1
float GetDiagonal(final int i)	Mengembalikan elemen matriks pada indeks baris ke-i dan kolom ke-i
void PlusRow(final int origin, final int target, final float koef)	Melakukan operasi penjumlahan antara 2 baris
void SwapRow(final int origin, final int target)	Melakukan pertukaran elemen antara 2 buah baris
void MakeOne(final int i, final float koef)	Membagi baris i dengan konstanta koef untuk membuat 1 utama
Void GaussElimination()	Membuat matriks augmented menjadi matriks eselon baris dengan segitiga atas
Void GJordanElimination()	Membuat matriks augmented menjadi Matriks eselon baris tereduksi
float DeterminanKofaktor()	Menghitung nilai determinan sebuah matriks dengan menggunakan ekspansi kofaktor
Float DeterminanGauss()	Menghitung nilai determinan sebuah matriks dengan menggunakan metode eliminasi Gauss
Matriks Kofaktor()	Membentuk matriks kofaktor dari sebuah

	matriks bujursangkar
Matriks Adjoin()	Melakukan transpose matriks kofaktor
Matriks InversKofaktor()	Membentuk invers matriks menggunakan metode Kofaktor
Matriks InverseGaussJordan()	Membentuk matriks invers menggunakan metode Gauss-Jordan

2. Regresi.java

   Pada class regresi ini, dideklarasikan atribut yaitu:

   - Parameter[] yang berfungsi sebagai kontainer dari parameter yang akan dicari solusinya

Method	Deskripsi
Regresi(int NBrsEff, int NKolEff)	Konstruktor pembentuk matriks augmented SPL Regresi
void bacaRegresi()	Procedure untuk membaca data regresi dari input user
void bacaFileRegresi(string namaFile)	Procedure untuk membaca data regresi dari input file
void bacaParameter()	Procedure membaca koefisien parameter matriks
void tulisRegresi()	Procedure menuliskan Persamaan dan hasil regresi di terminal
void tulisFileRegresi(string namaFile)	Procedure menuliskan Persamaan dan hasil regresi di file
int nPeubah()	Selektor yang mengembalikan banyaknya peubah
int nData()	Selektor yang mengembalikan banyaknya data
Regresi normalEstimation()	Fungsi melakukan Normal Estimation terhadap matriks data dan mengembalikan

	Matriks Augmented SPL Regresi.
float hasilRegresi()	Fungsi mengembalikan hasil regresi

3. SPL.java

   Pada class regresi ini, dideklarasikan atribut yaitu:

   - Solusi[] berfungsi sebagai kontainer dari solusi eksak SPL
   - Persamaan[] berfungsi sebagai kontainer koefisien persamaan dalam bentuk string
   - Status[] berfungsi sebagai kontainer jenis variabel (eksak, parametrik, atau dapat disubstitusi)

Method	Deskripsi
SPL(int NBrsEff, int NKolEff)	Konstruktor pembentuk matriks augmented SPL
void bacaSPL()	Prosedur membaca Matriks Augmented SPL dari terminal
void bacaFileSPL(String namaFile)	Prosedur membaca Matriks Augmented SPL dari file .txt
void tulisSPL()	Prosedur menulis hasil SPL pada terminal
void tulisFileSPL(String namaFile)	Prosedur menulis hasil SPL pada file .txt bernama namaFile
int jenisSolusi()	Fungsi untuk menentukan jenis solusi (unik, banyak, tidak ada) dari Matriks Augmented yang telah dilakukan penyelesaian menggunakan metode Gauss atau metode Gauss-Jordan
boolean isAllDiagonalOne()	Fungsi untuk mengecek apakah setiap elemen pada diagonal merupakan angka 1
boolean isRowZeroEx(int i)	Fungsi untuk mengecek apakah setiap elemen pada baris i (kecuali kolom terakhir) merupakan angka 0

boolean isLastElemtRowZero(int i)	Fungsi untuk mengecek apakah elemen pada baris i dan kolom terakhir merupakan angka 0
void metodeGauss()	Prosedur untuk menyelesaikan SPL dengan metode Gauss
void metodeGaussJordan()	Prosedur untuk menyelesaikan SPL dengan metode Gauss Jordan
void solusiGauss()	Prosedur untuk menentukan solusi dari SPL yang telah dilakukan fungsi metode Gauss dan metode Gauss Jordan
void metodeInvers()	Prosedur untuk menyelesaikan SPL dengan metode Matriks Invers
void metodeCramer()	Prosedur untuk menyelesaikan SPL dengan metode Kaidah Cramer

4. UI.java

   Class UI digunakan sebagai main driver dari seluruh program tugas besar ini. Tidak ada atribut yang dideklarasikan di sini.

Method	Deskripsi
Public static void main(String[] args)	Main program
Public static void MainMenu()	Display Main Menu
Public static void MenuSPL()	Display menu SPL
Public static void MenuDeterminan()	Display menu Determinan
Public static void MenuInvers()	Display menu Invers
Public static void MenuInterpolasi()	Display menu Interpolasi
Public static void MenuRegresi()	Display menu Regresi
Public static void getAllDataFiles()	Prosedur untuk menampilkan list berupa nama file yang valid
Public static void clearScreen()	Prosedur untuk mengosongkan layar terminal

Public static void tekanEnter()	Prosedur untuk menampilkan “Tekan ENTER untuk melanjutkan”

5. Interpolasi.java

   Pada class interpolasi, tidak dilakukan deklarasi atribut apapun karena atribut yang digunakan adalah atribut matriks

Method	Deskripsi
Interpolasi(int NBrsEff, int NKolEff)	Konstruktor pembentuk matriks augmented dengan nama class Interpolasi.
Float bacaInterpolasi()	Membaca titik-titik yang akan menjadi acuan taksiran dari terminal.
Float bacaFileInterpolasi(String namaFile)	Membaca titik-titik yang akan menjadi acuan taksiran dari file berekstensi .txt.
void tulisInterpolasi (float x, float y, Interpolasi z)	Menuliskan taksiran nilai yang ingin dicari di terminal.
void tulisFileInterpolasi (float x, float y, interpolasi z, String namaFile)	Menuliskan taksiran nilai yang ingin dicari ke file ekstensi .txt.
Float InterpolasiPolinom(float x)	Melakukan proses Interpolasi sehingga menghasilkan nilai taksiran yang diinginkan.
Interpolasi ConvertToMatrixAug()	Mengubah titik-titik yang sudah diketahui sebelumnya menjadi *matrix augmented* sehingga nilai variabel dapat dicari (dalam kasus ini menggunakan metode gauss).

BAB IV

EKSPERIMEN

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 1:

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_1a.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 1
-----------------------------------
Metode Eliminasi Gauss
-----------------------------------
Solusi SPL tidak ada
-----------------------------------
```

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_1b.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 2
-----------------------------------
Metode Eliminasi Gauss Jordan
-----------------------------------
X_1 = 3.000000 + A
X_2 = + 2.0A
X_3 = B
X_4 = -1.000000 + A
X_5 = A
Dengan A,B merupakan bilangan real
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_1c.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 1
-----------------------------------
Metode Eliminasi Gauss
-----------------------------------
X_1 = B
X_2 = 2.000000 - (1.000000 + A )
X_3 = C
X_4 = -1.000000 - (1.000000 + A )
X_5 = 1.000000 + A
X_6 = A
Dengan A,B,C merupakan bilangan real
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_1d_1.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 2
-----------------------------------
Metode Eliminasi Gauss Jordan
-----------------------------------
X_1 = 35.254673
X_2 = -608.556213
X_3 = 3214.003174
X_4 = -7178.365723
X_5 = 7137.078613
X_6 = -2604.813477
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_1d_2.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 1
-----------------------------------
Metode Eliminasi Gauss
-----------------------------------
X_1 = 31.117798
X_2 = -402.880371
X_3 = 862.685547
X_4 = 2362.687500
X_5 = -4512.609375
X_6 = -16487.250000
X_7 = 38416.710938
X_8 = -7804.710938
X_9 = -29144.445313
X_10 = 16692.439453
-----------------------------------
Simpan Hasil? (y/n) :
```

Hasil analisis:

1a) Dengan menggunakan metode eliminasi gauss, didapatkan bahwa SPL tersebut tidak memiliki solusi karena pada saat dilakukan OBE di baris terakhir berisi angka 0 dan di akhir kolom terdapat suatu nilai sehingga syarat SPL tidak memiliki solusi terpenuhi. Sehingga SPL tersebut tidak memiliki solusi.

1b) Dengan menggunakan metode eliminasi gauss jordan, didapatkan bahwa setelah dilakukan metode tersebut baris terakhir dari SPL tersebut menjadi bernilai 0 semua. Dengan melakukan *reverse substitution* didapatkan bahwa solusi SPL berikut ini merupakan solusi parametrik dengan nilai X = 3 + A, X2 = 2A, X3 = B, X4 = -1 + A, dan X5 = A, dengan A dan B merupakan elemen bilangan real.

1c) Dengan menggunakan metode eliminasi gauss, didapatkan bahwa bentuk *matrix augmented* mirip dengan saat 1b, sehingga solusi SPL berikut juga merupakan solusi parametrik dengan X1

= B, X2 = 1 - A, X3 = C, X4 = -2 - A, dan X5 = 1 + A, serta X6 = A dengan A,B,C merupakan elemen bilangan real.

1d) Matriks tersebut adalah matriks Hilbert. Dengan menggunakan metode eliminasi Gauss Jordan untuk n = 6, dan metode eliminasi Gauss untuk n = 10, didapatkan bahwa matriks Hilbert tersebut memiliki solusi unik.

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 2:

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_2_a.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 1
-----------------------------------
Metode Eliminasi Gauss
-----------------------------------
X_1 = -1.000000 + (+ 2.0A ) - 2.0A + B
X_2 = + 2.0A
X_3 = A
X_4 = B
Dengan A,B merupakan bilangan real
-----------------------------------
Simpan Hasil? (y/n) : 
```

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_2_b.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 1
-----------------------------------
Metode Eliminasi Gauss
-----------------------------------
X_1 = 0.0
X_2 = 2.0
X_3 = 1.0
X_4 = 1.0
-----------------------------------
Simpan Hasil? (y/n) : y
-----------------------------------
```

Hasil analisis:

2a) Dengan menggunakan metode eliminasi Gauss, didapatkan bahwa setelah dilakukan metode tersebut baris terakhir dari SPL tersebut menjadi bernilai 0 semua. Dengan melakukan *reverse substitution* didapatkan bahwa solusi SPL berikut ini merupakan solusi parametrik dengan X1 = -1 + B , X2 = 2A, X3 = A, dan X4 = B dengan A dan B merupakan elemen real.

2b) Dengan menggunakan metode eliminasi Gauss, didapatkan SPL yang memenuhi kriteria untuk menjadi solusi-solusi unik. Dengan melakukan *reverse substitution* didapatkan bahwa solusi dari SPL 2b merupakan solusi unik dengan X1 = 0 , X2 = 2, X3 = 1, dan X4 = 1

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 3:

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_3a.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 3
-----------------------------------
Metode Matriks Balikan
-----------------------------------
X_1 = -0.22432432
X_2 = 0.18243243
X_3 = 0.7094594
X_4 = -0.25810808
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_3b.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 4
-----------------------------------
Metode Kaidah Cramer
Nilai variabel tidak dapat ditentukan karena nilai determinan awal adalah 0.
-----------------------------------
Solusi SPL tidak ada
-----------------------------------
Simpan Hasil? (y/n) :
```

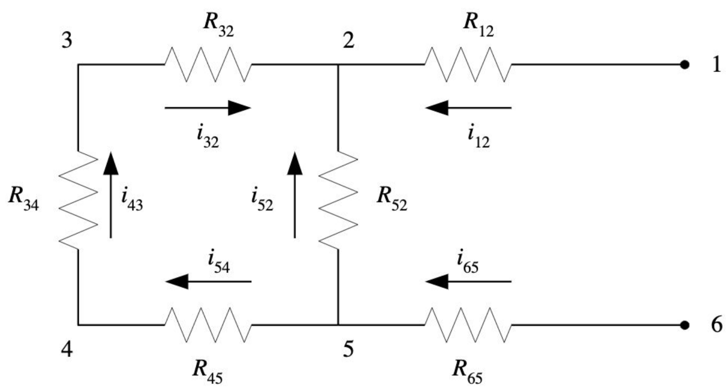
Hasil analisis:

3a) Sama seperti studi kasus nomor 1 dan 2, untuk mencari solusi suatu SPL kita hanya tinggal menggunakan menu SPL, dengan menggunakan metode balikan didapatkan solusi unik dengan X1 = -0.22432432, X2 = 0.18243243, X3 = 0.7094594, dan X4 = -0.2510808.

3b) Untuk studi kasus nomor 3b, dengan menggunakan kaidah cramer, ditemukan bahwa sistem persamaan linear tersebut tidak memiliki solusi karena determinan matriks A dari Ax = B adalah 0. Oleh karena itu, solusi SPL tidak dapat dicari menggunakan metode cramer.

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 4:

```
-----------------------------------
Masukkan nama file data matriks SPL : testcase_4.txt
-----------------------------------
Data matriks SPL Berhasil Terbaca
-----------------------------------
           PILIH METODE
-----------------------------------
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss Jordan
3. Metode Matriks Balikan
4. Metode Kaidah Cramer
5. Kembali
-----------------------------------
Masukkan pilihan : 3
-----------------------------------
Metode Matriks Balikan
-----------------------------------
X_1 = 6.1538467
X_2 = -4.615384
X_3 = -1.5384625
X_4 = -6.1538467
X_5 = -1.5384625
X_6 = -1.5384625
X_7 = 169.23077
X_8 = 153.84616
X_9 = 146.15385
X_10 = 123.07693
-----------------------------------
Simpan Hasil? (y/n) :
```



Hasil analisis:

Studi kasus ke-4 adalah mencari nilai dari arus dan tegangan dari node yang tidak diketahui nilainya. Berdasarkan gambar rangkaian tersebut, kami akan menentukan nilai i_{12}, i_{52}, i_{32}, i_{65}, i_{54}, i_{43}, V_2, V_3, V_4, V_5. Dengan menyusun rangkaian menjadi persamaan linier seperti di bawah:

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

Dengan mengubah persamaan linear menjadi sebuah matriks augmented kemudian menyelesaikannya dengan metode matriks balikan dan berbagai metode lainnya. Didapatkan hasil i_{12} = X_1 = 6.153846, i_{52} = X_2 = -4.615382, i_{32} = X_3 = -1.5384636, i_{65} = X_4 = -6.153846, i_{54} = X_5 = -1.5384636, i_{43} = X_6 = -1.5384636, V_2 = X_7 = 169.2308 , V_3 = X_8 = 153.84616, V_4 = X_9 = 146.15384 , V_5 = X_10 = 123.07691.

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 5:

```
-----------------------------------
Masukkan nama file data regresi : testcase_5.txt
Masukkan nilai x yang akan ditaksir : 0.2
-----------------------------------
Data Interpolasi Berhasil Terbaca
-----------------------------------
Persamaan Polinom:
a + b*0.1 + c*0.010000001 + d*0.001 + e*1.00000005E-4 + f*1.0000001E-5 + g*1.0000001E-6 = 0.003
a + b*0.3 + c*0.09 + d*0.027000003 + e*0.008100001 + f*0.0024300006 + g*7.290002E-4 = 0.067
a + b*0.5 + c*0.25 + d*0.125 + e*0.0625 + f*0.03125 + g*0.015625 = 0.148
a + b*0.7 + c*0.48999998 + d*0.343 + e*0.24009998 + f*0.16806999 + g*0.11764899 = 0.248
a + b*0.9 + c*0.80999994 + d*0.7289999 + e*0.6560999 + f*0.5904899 + g*0.5314409 = 0.37
a + b*1.1 + c*1.21 + d*1.3310001 + e*1.4641001 + f*1.6105102 + g*1.7715613 = 0.518
a + b*1.3 + c*1.6899998 + d*2.1969998 + e*2.8560996 + f*3.7129292 + g*4.826808 = 0.697

Hasil Interpolasi dari 0.2 adalah: 0.03296092
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data regresi : testcase_5.txt
Masukkan nilai x yang akan ditaksir : 0.55
-----------------------------------
Data Interpolasi Berhasil Terbaca
-----------------------------------
Persamaan Polinom:
a + b*0.1 + c*0.010000001 + d*0.001 + e*1.00000005E-4 + f*1.0000001E-5 + g*1.0000001E-6 = 0.003
a + b*0.3 + c*0.09 + d*0.027000003 + e*0.008100001 + f*0.0024300006 + g*7.290002E-4 = 0.067
a + b*0.5 + c*0.25 + d*0.125 + e*0.0625 + f*0.03125 + g*0.015625 = 0.148
a + b*0.7 + c*0.48999998 + d*0.343 + e*0.24009998 + f*0.16806999 + g*0.11764899 = 0.248
a + b*0.9 + c*0.80999994 + d*0.7289999 + e*0.6560999 + f*0.5904899 + g*0.5314409 = 0.37
a + b*1.1 + c*1.21 + d*1.3310001 + e*1.4641001 + f*1.6105102 + g*1.7715613 = 0.518
a + b*1.3 + c*1.6899998 + d*2.1969998 + e*2.8560996 + f*3.7129292 + g*4.826808 = 0.697

Hasil Interpolasi dari 0.55 adalah: 0.17111865
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data regresi : testcase_5.txt
Masukkan nilai x yang akan ditaksir : 0.85
-----------------------------------
Data Interpolasi Berhasil Terbaca
-----------------------------------
Persamaan Polinom:
a + b*0.1 + c*0.010000001 + d*0.001 + e*1.00000005E-4 + f*1.0000001E-5 + g*1.0000001E-6 = 0.003
a + b*0.3 + c*0.09 + d*0.027000003 + e*0.008100001 + f*0.0024300006 + g*7.290002E-4 = 0.067
a + b*0.5 + c*0.25 + d*0.125 + e*0.0625 + f*0.03125 + g*0.015625 = 0.148
a + b*0.7 + c*0.48999998 + d*0.343 + e*0.24009998 + f*0.16806999 + g*0.11764899 = 0.248
a + b*0.9 + c*0.80999994 + d*0.7289999 + e*0.6560999 + f*0.5904899 + g*0.5314409 = 0.37
a + b*1.1 + c*1.21 + d*1.3310001 + e*1.4641001 + f*1.6105102 + g*1.7715613 = 0.518
a + b*1.3 + c*1.6899998 + d*2.1969998 + e*2.8560996 + f*3.7129292 + g*4.826808 = 0.697

Hasil Interpolasi dari 0.85 adalah: 0.33723593
-----------------------------------
Simpan Hasil? (y/n) :
```

```
------------------------------------
Masukkan nama file data regresi : testcase_5.txt
Masukkan nilai x yang akan ditaksir : 1.28
------------------------------------
Data Interpolasi Berhasil Terbaca
------------------------------------
Persamaan Polinom:
a + b*0.1 + c*0.010000001 + d*0.001 + e*1.00000005E-4 + f*1.0000001E-5 + g*1.0000001E-6 = 0.003
a + b*0.3 + c*0.09 + d*0.027000003 + e*0.008100001 + f*0.0024300006 + g*7.290002E-4 = 0.067
a + b*0.5 + c*0.25 + d*0.125 + e*0.0625 + f*0.03125 + g*0.015625 = 0.148
a + b*0.7 + c*0.48999998 + d*0.343 + e*0.24009998 + f*0.16806999 + g*0.11764899 = 0.248
a + b*0.9 + c*0.80999994 + d*0.7289999 + e*0.6560999 + f*0.5904899 + g*0.5314409 = 0.37
a + b*1.1 + c*1.21 + d*1.3310001 + e*1.4641001 + f*1.6105102 + g*1.7715613 = 0.518
a + b*1.3 + c*1.6899998 + d*2.1969998 + e*2.8560996 + f*3.7129292 + g*4.826808 = 0.697

Hasil Interpolasi dari 1.28 adalah: 0.67754185
------------------------------------
Simpan Hasil? (y/n) :
```

Hasil Analisis:

Pada studi kasus nomor 5, digunakan tabel yang tersedia di spesifikasi tugas besar Algeo 1 yang berisi titik-titik untuk ditemukan persamaan polinom nya. Kemudian dilakukan pengujian pada beberapa nilai-nilai yang akan ditaksir berapa nilai fungsi dari f(x) tersebut. Dengan menggunakan menu interpolasi polinom, didapatkan bahwa:

Di saat x = 0.2, f(0.2) = 0.03296092

Di saat x = 0.55, f(0.55) = 0.17111865

Di saat x = 0.85, f(0.85) = 0.33723593

Di saat x = 1.28, f(1.28) = 0.67754185

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 6:

```
------------------------------------
Masukkan nama file data regresi : testcase_6.txt
Masukkan nilai x yang akan ditaksir : 5.806
------------------------------------
Data Interpolasi Berhasil Terbaca
------------------------------------
Persamaan Polinom:
a + b*4.8 + c*23.04 + d*110.59201 + e*530.8417 + f*2548.0403 + g*12230.594 + h*58706.85 + i*281792.9 + j*1352606.0 = 8211.0
a + b*5.0 + c*25.0 + d*125.0 + e*625.0 + f*3125.0 + g*15625.0 + h*78125.0 + i*390625.0 + j*1953125.0 = 10118.0
a + b*5.516 + c*30.426254 + d*167.8312 + e*925.7569 + f*5106.475 + g*28167.314 + h*155370.9 + i*857025.9 + j*4727354.5 = 17025.0
a + b*5.71 + c*32.6041 + d*186.16942 + e*1063.0273 + f*6069.886 + g*34659.05 + h*197903.19 + i*1130027.1 + j*6452455.0 = 20796.0
a + b*6.5 + c*42.25 + d*274.625 + e*1785.0625 + f*11602.906 + g*75418.89 + h*490222.78 + i*3186448.2 + j*2.0711912E7 = 39294.0
a + b*7.194 + c*51.75363 + d*372.3156 + e*2678.4385 + f*19268.686 + g*138618.92 + h*997224.5 + i*7174033.0 + j*5.1609988E7 = 64958.0
a + b*8.097 + c*65.56141 + d*530.85077 + e*4298.299 + f*34803.324 + g*281802.53 + h*2281755.0 + i*1.847537E7 + j*1.49595088E8 = 113134.0
a + b*8.258 + c*68.19457 + d*563.15076 + e*4650.4995 + f*38403.824 + g*317138.8 + h*2618932.5 + i*2.1627144E7 + j*1.78596976E8 = 123503.0
a + b*9.033 + c*81.595085 + d*737.04846 + e*6657.7583 + f*60139.53 + g*543240.4 + h*4907090.5 + i*4.4325748E7 + j*4.00394496E8 = 177571.0
a + b*9.333 + c*87.10489 + d*812.94995 + e*7587.262 + f*70811.92 + g*660887.7 + h*6168064.5 + i*5.7566548E7 + j*5.3726861E8 = 145510.0

Hasil Interpolasi dari 5.806 adalah: 19361.535
------------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data regresi : testcase_6.txt
Masukkan nilai x yang akan ditaksir : 8.967
-----------------------------------
Data Interpolasi Berhasil Terbaca
-----------------------------------
Persamaan Polinom:
a + b*4.8 + c*23.04 + d*110.59201 + e*530.8417 + f*2548.0403 + g*12230.594 + h*58706.85 + i*281792.9 + j*1352606.0 = 8211.0
a + b*5.0 + c*25.0 + d*125.0 + e*625.0 + f*3125.0 + g*15625.0 + h*78125.0 + i*390625.0 + j*1953125.0 = 10118.0
a + b*5.516 + c*30.426254 + d*167.8312 + e*925.7569 + f*5106.475 + g*28167.314 + h*155370.9 + i*857025.9 + j*4727354.5 = 17025.0
a + b*5.71 + c*32.6041 + d*186.16942 + e*1063.0273 + f*6069.886 + g*34659.05 + h*197903.19 + i*1130027.1 + j*6452455.0 = 20796.0
a + b*6.5 + c*42.25 + d*274.625 + e*1785.0625 + f*11602.906 + g*75418.89 + h*490222.78 + i*3186448.2 + j*2.0711912E7 = 39294.0
a + b*7.194 + c*51.75363 + d*372.3156 + e*2678.4385 + f*19268.686 + g*138618.92 + h*997224.5 + i*7174033.0 + j*5.1609988E7 = 64958.0
a + b*8.097 + c*65.56141 + d*530.85077 + e*4298.299 + f*34803.324 + g*281802.53 + h*2281755.0 + i*1.847537E7 + j*1.49595088E8 = 113134.0
a + b*8.258 + c*68.19457 + d*563.15076 + e*4650.4995 + f*38403.824 + g*317138.8 + h*2618932.5 + i*2.1627144E7 + j*1.78596976E8 = 123503.0
a + b*9.033 + c*81.595085 + d*737.04846 + e*6657.7583 + f*60139.53 + g*543240.4 + h*4907090.5 + i*4.4325748E7 + j*4.00394496E8 = 177571.0
a + b*9.333 + c*87.10489 + d*812.94995 + e*7587.262 + f*70811.92 + g*660887.7 + h*6168064.5 + i*5.7566548E7 + j*5.3726861E8 = 145510.0

Hasil Interpolasi dari 8.967 adalah: 139564.47
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data regresi : testcase_6.txt
Masukkan nilai x yang akan ditaksir : 9.500
-----------------------------------
Data Interpolasi Berhasil Terbaca
-----------------------------------
Persamaan Polinom:
a + b*4.8 + c*23.04 + d*110.59201 + e*530.8417 + f*2548.0403 + g*12230.594 + h*58706.85 + i*281792.9 + j*1352606.0 = 8211.0
a + b*5.0 + c*25.0 + d*125.0 + e*625.0 + f*3125.0 + g*15625.0 + h*78125.0 + i*390625.0 + j*1953125.0 = 10118.0
a + b*5.516 + c*30.426254 + d*167.8312 + e*925.7569 + f*5106.475 + g*28167.314 + h*155370.9 + i*857025.9 + j*4727354.5 = 17025.0
a + b*5.71 + c*32.6041 + d*186.16942 + e*1063.0273 + f*6069.886 + g*34659.05 + h*197903.19 + i*1130027.1 + j*6452455.0 = 20796.0
a + b*6.5 + c*42.25 + d*274.625 + e*1785.0625 + f*11602.906 + g*75418.89 + h*490222.78 + i*3186448.2 + j*2.0711912E7 = 39294.0
a + b*7.194 + c*51.75363 + d*372.3156 + e*2678.4385 + f*19268.686 + g*138618.92 + h*997224.5 + i*7174033.0 + j*5.1609988E7 = 64958.0
a + b*8.097 + c*65.56141 + d*530.85077 + e*4298.299 + f*34803.324 + g*281802.53 + h*2281755.0 + i*1.847537E7 + j*1.49595088E8 = 113134.0
a + b*8.258 + c*68.19457 + d*563.15076 + e*4650.4995 + f*38403.824 + g*317138.8 + h*2618932.5 + i*2.1627144E7 + j*1.78596976E8 = 123503.0
a + b*9.033 + c*81.595085 + d*737.04846 + e*6657.7583 + f*60139.53 + g*543240.4 + h*4907090.5 + i*4.4325748E7 + j*4.00394496E8 = 177571.0
a + b*9.333 + c*87.10489 + d*812.94995 + e*7587.262 + f*70811.92 + g*660887.7 + h*6168064.5 + i*5.7566548E7 + j*5.3726861E8 = 145510.0

Hasil Interpolasi dari 9.5 adalah: 314283.62
-----------------------------------
Simpan Hasil? (y/n) :
```

```
-----------------------------------
Masukkan nama file data regresi : testcase_6.txt
Masukkan nilai x yang akan ditaksir : 4.967
-----------------------------------
Data Interpolasi Berhasil Terbaca
-----------------------------------
Persamaan Polinom:
a + b*4.8 + c*23.04 + d*110.59201 + e*530.8417 + f*2548.0403 + g*12230.594 + h*58706.85 + i*281792.9 + j*1352606.0 = 8211.0
a + b*5.0 + c*25.0 + d*125.0 + e*625.0 + f*3125.0 + g*15625.0 + h*78125.0 + i*390625.0 + j*1953125.0 = 10118.0
a + b*5.516 + c*30.426254 + d*167.8312 + e*925.7569 + f*5106.475 + g*28167.314 + h*155370.9 + i*857025.9 + j*4727354.5 = 17025.0
a + b*5.71 + c*32.6041 + d*186.16942 + e*1063.0273 + f*6069.886 + g*34659.05 + h*197903.19 + i*1130027.1 + j*6452455.0 = 20796.0
a + b*6.5 + c*42.25 + d*274.625 + e*1785.0625 + f*11602.906 + g*75418.89 + h*490222.78 + i*3186448.2 + j*2.0711912E7 = 39294.0
a + b*7.194 + c*51.75363 + d*372.3156 + e*2678.4385 + f*19268.686 + g*138618.92 + h*997224.5 + i*7174033.0 + j*5.1609988E7 = 64958.0
a + b*8.097 + c*65.56141 + d*530.85077 + e*4298.299 + f*34803.324 + g*281802.53 + h*2281755.0 + i*1.847537E7 + j*1.49595088E8 = 113134.0
a + b*8.258 + c*68.19457 + d*563.15076 + e*4650.4995 + f*38403.824 + g*317138.8 + h*2618932.5 + i*2.1627144E7 + j*1.78596976E8 = 123503.0
a + b*9.033 + c*81.595085 + d*737.04846 + e*6657.7583 + f*60139.53 + g*543240.4 + h*4907090.5 + i*4.4325748E7 + j*4.00394496E8 = 177571.0
a + b*9.333 + c*87.10489 + d*812.94995 + e*7587.262 + f*70811.92 + g*660887.7 + h*6168064.5 + i*5.7566548E7 + j*5.3726861E8 = 145510.0

Hasil Interpolasi dari 4.967 adalah: 10328.952
-----------------------------------
Simpan Hasil? (y/n) :
```

Hasil Analisis:

Pada studi kasus nomor 6, digunakan tabel yang menunjukkan tanggal dalam desimal dan jumlah kasus covid-19 di Indonesia. Tanggal dalam desimal dapat dikonversi menggunakan rumus

Tanggal (desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut)

Kemudian, data yang terdapat dalam spesifikasi tugas besar Algeo 1 digunakan untuk melakukan prediksi jumlah kasus Covid-19 pada tanggal-tanggal berikut:

a. 25/05/20. Berdasarkan hasil dari program kelompok kami diprediksi total kasus berjumlah 19.361 jiwa.
b. 30/08/20. Berdasarkan hasil dari program kelompok kami diprediksi total kasus berjumlah 139.564 jiwa.
c. 15/09/20. Berdasarkan hasil dari program kelompok kami diprediksi total kasus berjumlah 314.283 jiwa.
d. 29/04/20 (Masukan user lainnya). Berdasarkan hasil dari program kelompok kami diprediksi total kasus berjumlah 10.328 jiwa.

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 7:

Hasil Analisis:

Pada studi kasus nomor 7, kami melakukan penyederhanaan fungsi $f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$ menjadi sebuah polinom interpolasi berderajat 20 di dalam selang [0, 2] dengan jarak antar titik sebesar 0.1. Dengan melakukan proses interpolasi polinom, kemudian dengan memasukkan angka yang akan ditaksir, didapatkan angka yang merepresentasikan f(x). Dalam hal ini kami menggunakan angka 0.69 sebagai angka yang akan ditaksir dan didapatkan angka 0.48689017 yang sesuai ketika kami menggunakan kalkulator untuk menghitung nilai tersebut ke dalam fungsi.

Di bawah ini adalah hasil eksekusi program terhadap contoh studi kasus nomor 8:

```
-----------------------------------
Masukkan nama file data regresi : testcase_8.txt
Masukkan Nilai Parameter x1: 50
Masukkan Nilai Parameter x2: 76
Masukkan Nilai Parameter x3: 29.30
-----------------------------------
Data Regresi Berhasil Terbaca
-----------------------------------
Persamaan Regresi :
y = -3.5113497  - 0.0026244328x1 + 7.9925323E-4x2 + 0.1542749x3
Hasil Regresi : 0.93842626
-----------------------------------
Simpan Hasil? (y/n) :
```

Hasil Analisis:

Pada studi kasus 8, kami diminta untuk melakukan *Multiple Linear Regression*, dari data tabel yang berisi sebanyak 20 buah dengan kolom y = Nitrous Oxide, x1 = Humidity, x2 = Temperatur, dan x3 = Tekanan udara. Kemudian data tersebut kami ubah menjadi matriks augmented menggunakan teknik *Normal Estimation Equation for Multiple Linear Regression*, dan diperoleh sistem persamaan linear yang dapat kami selesaikan dengan metode Gauss. Dari sana didapat persamaan regresi berupa:

$$y = -3.5113497 - 0.0026244328 * X_1 + 7.9925323 * 10^{-4} * X_2 + 0.1542749 * X_3$$

Kemudian kami akan mengestimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30. **Didapat hasil nilai Nitrous Oxide sebesar 0.93842626.**

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

Kesimpulan

1. Kami telah membuat sebuah program yang mampu 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. Kami telah membuat sebuah program yang mampu menyelesaikan persoalan interpolasi polinom dan regresi linier
3. Kami telah membuat sebuah program yang mampu menghitung matriks balikan
4. Kami telah membuat sebuah program yang mampu menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

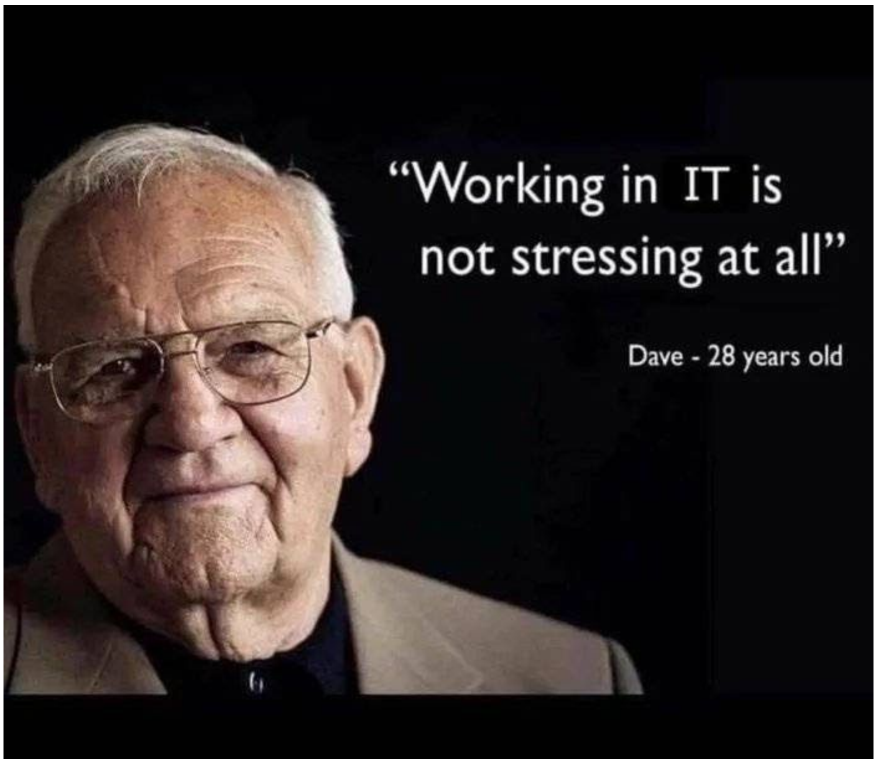
Saran

Kedepannya program dapat dilakukan dengan lebih serius apabila diberikan waktu pengerjaan yang lebih lama lagi. Yang dimaksud dengan lebih serius adalah dengan membuat *user interface* dengan menggunakan GUI, sehingga user yang menggunakan aplikasi ini seakan-akan berinteraksi langsung dengan program. Kemudian dalam membuat aplikasi kalkulasi seperti tugas besar ini hendaknya menggunakan tipe dasar yang paling presisi. Dalam hal ini adalah tipe Double. Karena dalam mengkalkulasi sebuah aljabar linier harus menggunakan ketelitian dan presisi yang cukup tinggi untuk menghindari terjadinya ketidakpastian nilai yang cukup besar.

Refleksi

Dalam mengerjakan tugas besar ini, terdapat beberapa kendala. Yang pertama, kurangnya pemahaman kami terhadap pemrograman berorientasi objek yang digunakan pada bahasa java, sehingga dibutuhkan waktu yang cukup lama untuk memahami apa itu pemrograman berorientasi objek. Kedua, kami membuat sebuah kesalahan pemilihan tipe data yang digunakan untuk menampung elemen, kami menggunakan tipe data float, yang kurang presisi sehingga terdapat deviasi yang cukup besar pada hasil akhir operasi.

Kesalahan tersebut murni datang dari kekurangan dan kemampuan kami, oleh karena itu kami akan berusaha hingga yang menjadi hambatan di saat ini tidak lagi menjadi hambatan di masa yang akan datang.



BAB VI

REFERENSI

http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/algeo20-21.htm

https://www.w3schools.com/java/default.asp

https://stackoverflow.com/questions/tagged/java

https://www.youtube.com/playlist?list=PLZS-MHyEIRo6V4_vk1s1NcM2HoW5KFG7i

https://bit.ly/KitabSuciDingDong

BAB VII

PEMBAGIAN TUGAS

NO	BAGIAN PROGRAM	SUBBAGIAN PROGRAM	PJ	STATUS
1	Sistem Persamaan Linier	Metode Eliminasi Gauss	Danan, Azhar	CLEAR (28/09)
		Metode Eliminasi Gauss-Jordan	Danan, Azhar	CLEAR (28/09)
		Metode Matriks Balikan	Danan	CLEAR (27/09)
		Kaidah Cramer	Dehan	CLEAR (26/09)
2	Determinan	Metode Eliminasi Gauss	Danan	CLEAR (26/09)
		Metode Ekspansi Kofaktor	Dehan	CLEAR (26/09)
3	Matriks Balikan	Metode Eliminasi Gauss-Jordan	Danan, Azhar	CLEAR (29/09)
		Metode Ekspansi Kofaktor dan Adjoin	Dehan	CLEAR (26/09)
4	Interpolasi Polinom	Aplikasi SPL	Dehan	CLEAR (29/09)
5	Regresi Linier Berganda	Aplikasi SPL	Azhar	CLEAR (29/09)