**LAPORAN TUGAS BESAR 1**

**ALJABAR LINEAR DAN GEOMETRI**

**IF2123**

# **BAB I**

# **DESKRIPSI MASALAH**

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

*a*11 *x*1 + *a*12 *x*2 + + *a*1*n xn* = *b*1

*a*21 *x*1 + *a*22 *x*2 + .... + *a*2*n xn* = *b*2

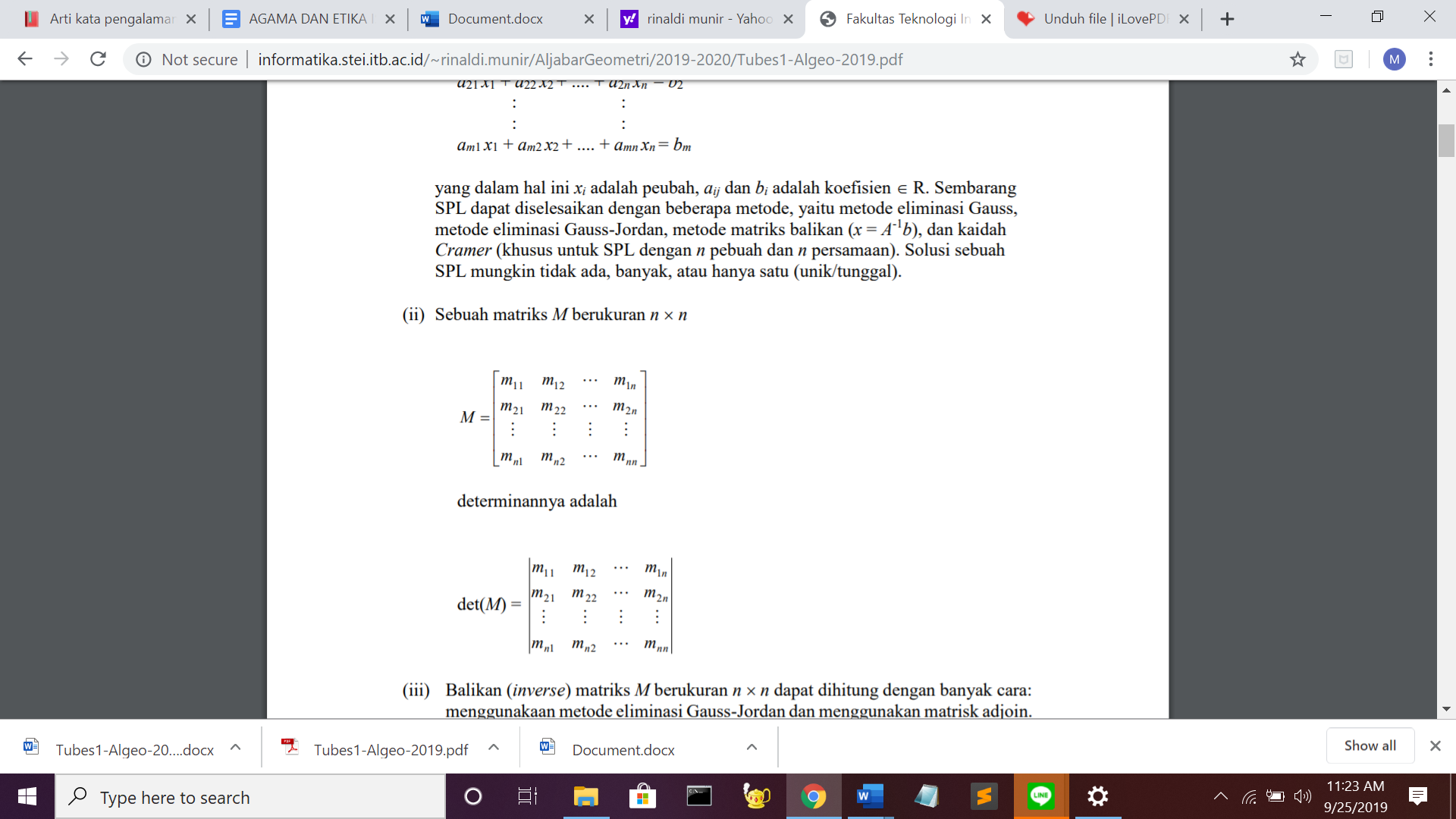
: :

: :

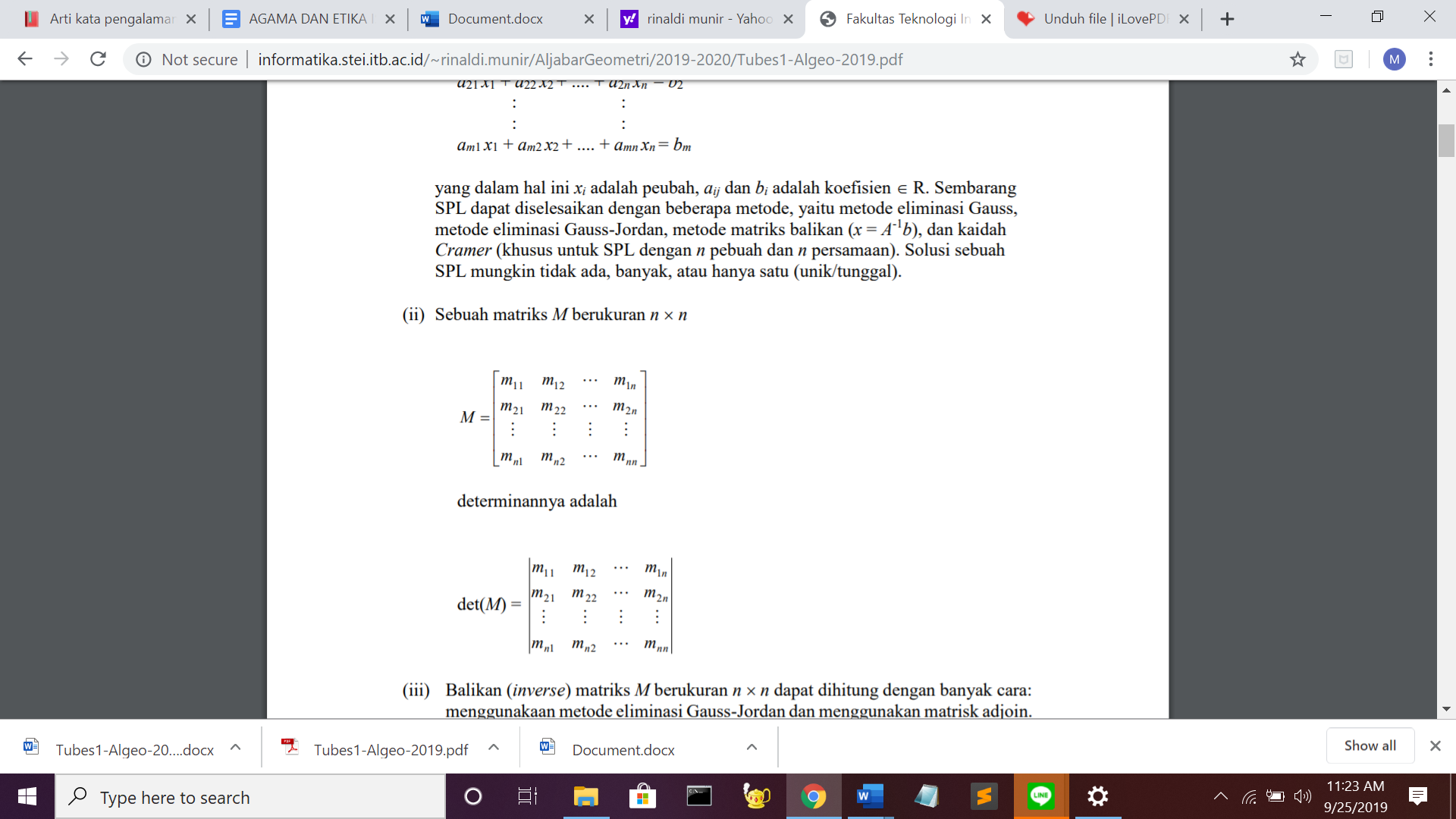
*am*1 *x*1 + *am*2 *x*2 + + *amn xn* = *bm*

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

1. Sebuah matriks *M* berukuran *n x* *n*

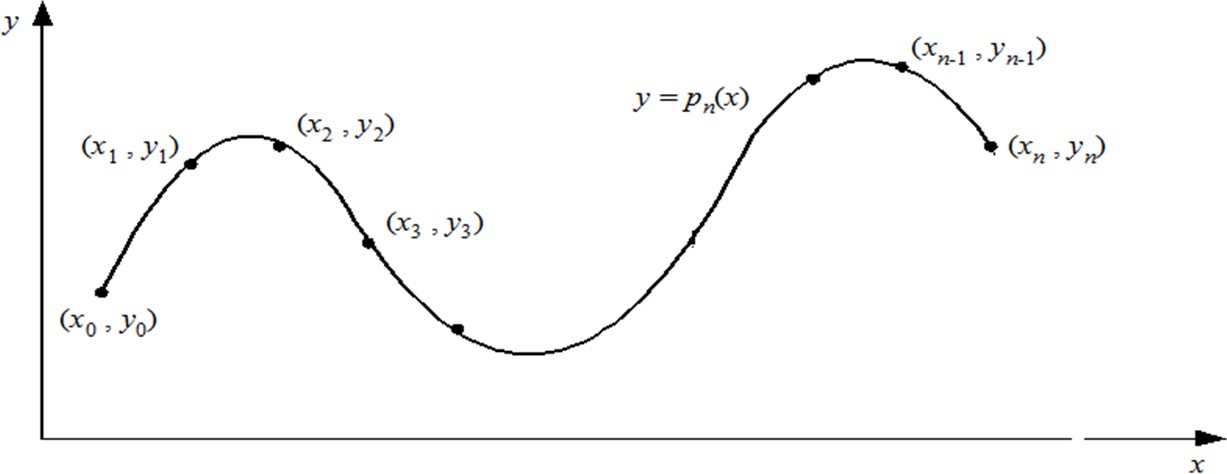


determinannya adalah



1. Balikan (*inverse*) matriks *M* berukuran *n* x *n* dapat dihitung dengan banyak cara: menggunakaan metode eliminasi Gauss-Jordan dan menggunakan matrisk adjoin.

Kembali ke sistem persamaan linier (SPL). SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, salah satunya adalah mengestimasi nilai fungsi dengan interpolasi polinom. Persoalan interpolasi polinom adalah sebagai berikut: Diberikan *n*+1 buah titik berbeda, (*x*0, *y*0), (*x*1, *y*1), ..., (*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 [*x*0, *xn*].

Polinom interpolasi derajat *n* yang menginterplolasi titik-titik (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*, *yn*). adalah berbentuk *pn*(*x*) = *a*0 + *a*1*x* + *a*2*x*2 + … + *anxn*. Jika hanya ada dua titik, (*x*0, *y*0) dan (*x*1, *y*1), maka polinom yang menginterpolasi kedua titik tersebut adalah *p*1(*x*) = *a*0 + *a*1*x* yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (*x*0, *y*0), (*x*1, *y*1), dan (*x*2, *y*2), maka polinom yang menginterpolasi ketiga titik tersebut adalah *p*2(*x*) = *a*0 + *a*1*x*

+ *a*2*x*2 atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (*x*0, *y*0), (*x*1, *y*1), (*x*2, *y*2), dan (*x*3, *y*3), polinom yang menginterpolasi keempat titik tersebut adalah *p*3(*x*) = *a*0 + *a*1*x* + *a*2*x*2 + *a*3*x*3, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat *n* untuk *n* yang lebih tinggi asalkan tersedia (*n*+1) buah titik data. Dengan menyulihkan (*xi*, *yi*) ke dalam persamaan polinom *pn*(*x*) = *a*0 + *a*1*x* + *a*2*x*2 + … + *anxn* untuk *i* = 0, 1, 2, …, *n*, akan diperoleh *n* buah sistem persamaan lanjar dalam *a*0, *a*1, *a2*, …, *an*,

*a*0 + *a*1*x*0 + *a*2*x*02 + ... + *an x*0*n* = *y*0

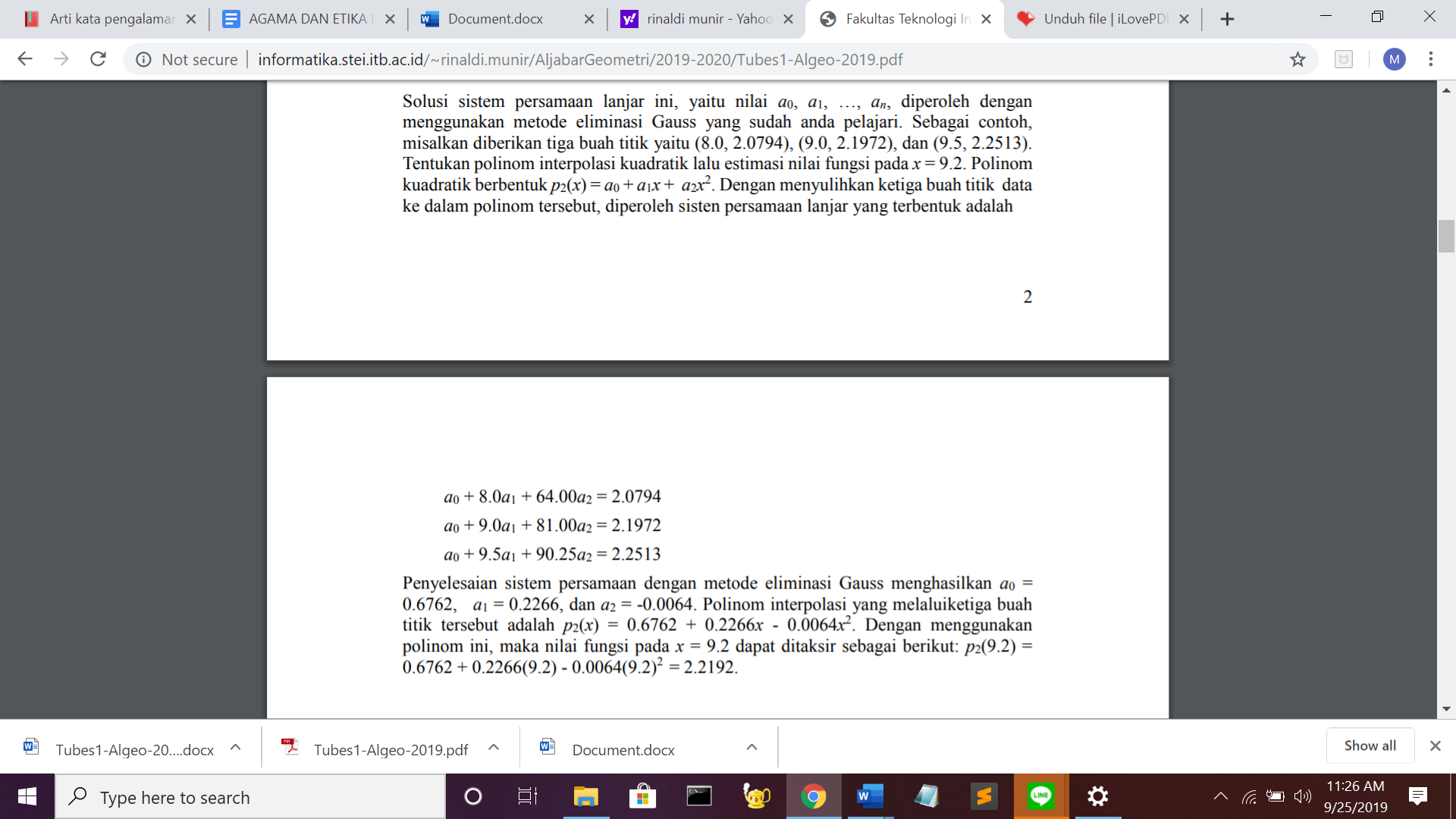
*a*0 + *a*1*x*1 + *a*2*x*12 + ... + *an x n* = *y*

1 1

... ...

*a*0 + *a*1*xn* + *a*2*xn*2 + ... + *an xnn* = *yn*

Solusi sistem persamaan lanjar ini, yaitu nilai *a*0, *a*1, …, *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 *p*2(*x*) = *a*0 + *a*1*x* + *a*2*x*2. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah



Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan *a*0 = 0.6762, *a*1 = 0.2266, dan *a*2 = -0.0064. Polinom interpolasi yang melaluiketiga buah titik tersebut adalah *p*2(*x*) = 0.6762 + 0.2266*x* - 0.0064*x*2. Dengan menggunakan polinom ini, maka nilai fungsi pada *x* = 9.2 dapat ditaksir sebagai berikut: *p*2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

# **BAB II**

# **TEORI SINGKAT**

1. Metode Eliminasi Gauss dan Gauss Jordan

Metode eliminasi Gauss adalah suatu cara mengoperasikan nilai-nilai di dalam matriks menjadi matriks yang lebih sederhana dan banyak digunakan dalam penyelesaian sistem persamaan linier. Prosedur penyelesaian dari metode ini adalah dengan melakukan operasi baris menjadi matriks eselon-baris. Metode ini mengubah persamaan linear tersebut ke dalam matriks augmentasi dan mengoperasikannya. Sistem persamaan linier merupakan salah satu sistem persamaan yang terdiri dari sejumlah persamaan dan variabel yang berhingga. Untuk dapat menyelesaikan suatu sistem persamaan linier adalah mencari nilai-nilai variabel-variabel persamaan tersebut.

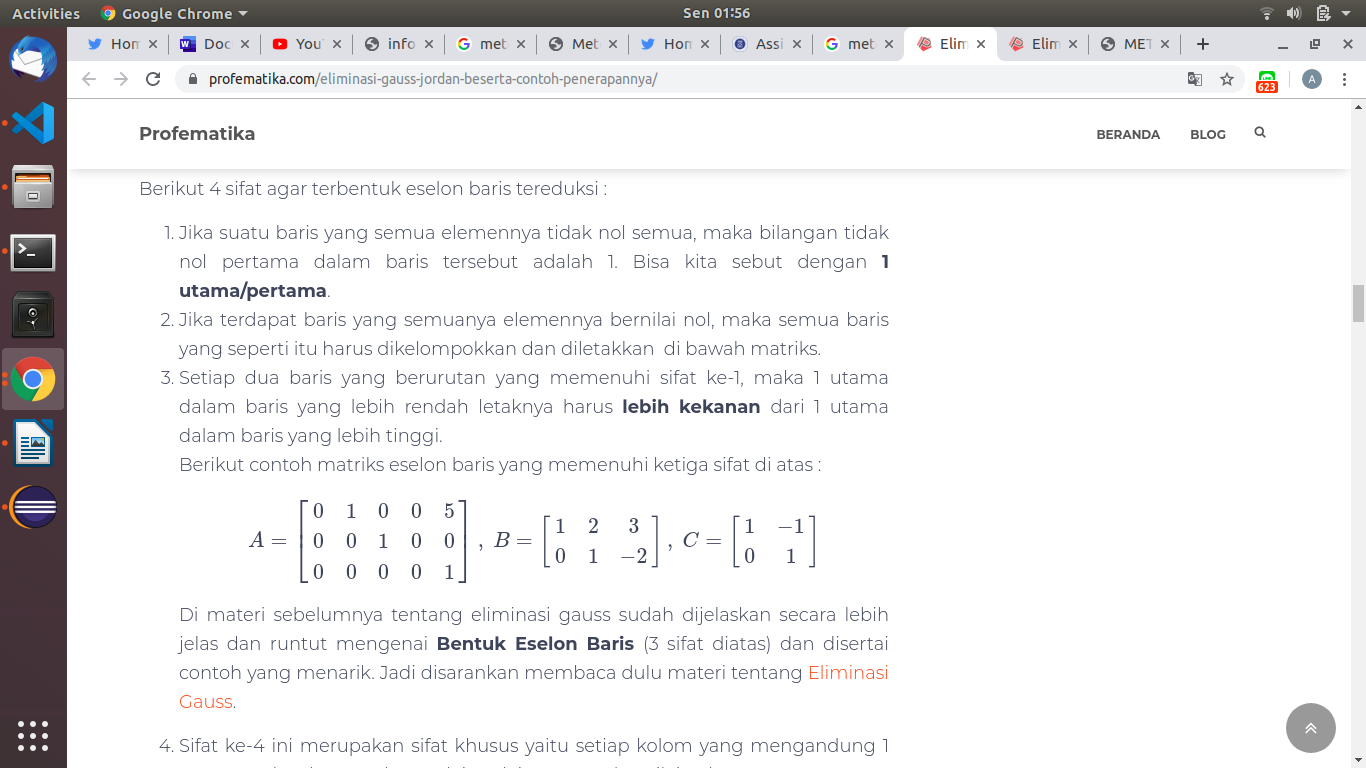
Untuk mendapatkan bentuk SPL segitiga dari SPL yang diketahui, metode eliminasi Gauss menggunakan sejumlah Operasi Baris Elementer (OBE):

1. Menukar posisi dua buah persamaan (dua baris matriks augmented)
2. Menambah sebuah persamaan (baris matriks augmented) dengan suatu kelipatan persamaan lain (baris lain)
3. Mengalikan sebuah persamaan (baris matriks augmented) dengan sebarang konstanta tak nol.

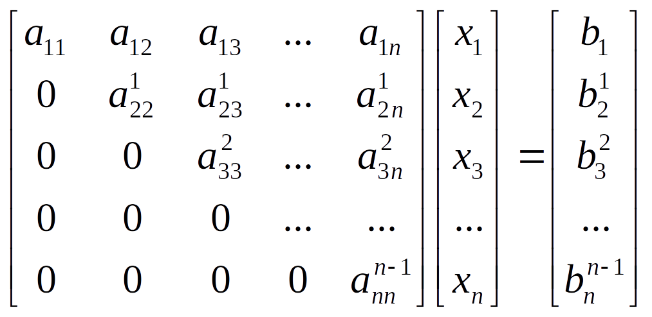
Pemakaian operasi-operasi baris elementer di atas pada sebuah SPL tidak akan mengubah penyelesaian SPL yang bersangkutan. Penyelesaian sebuah SPL tidak bergantung pada susunan penulisan persamaan, sehingga operasi baris no 1 dapat dilakukan. Dalam setiap persamaan, kedua ruas menyatakan nilai yang sama sehingga operasi baris no 2 dapat digunakan. Demikian pula operasi baris nomor 3 menghasilkan persamaan yang ekivalen. Pada metode Gauss, matriks augmented akan dioperasikan menjadi matriks eselon baris yang memiliki sifat:

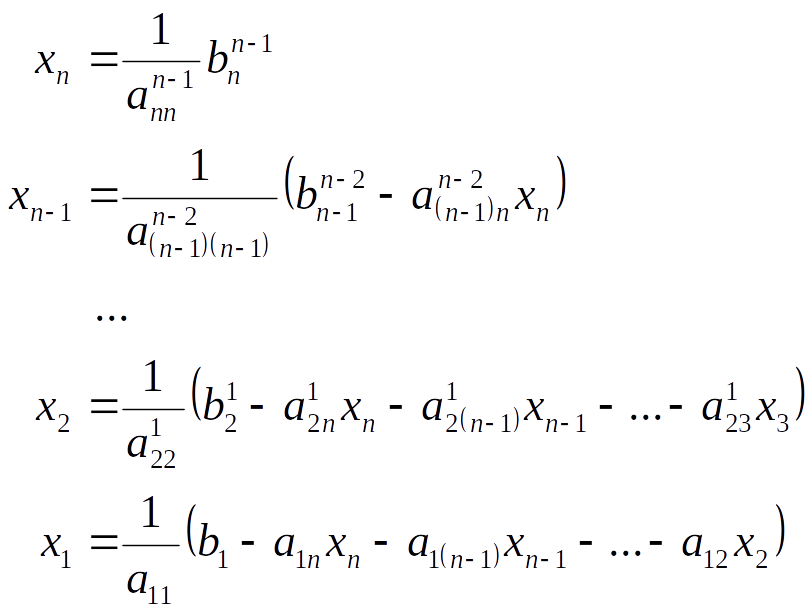
1. Jika suatu baris yang semua elemennya tidak nol semua, maka bilangan tidak nol pertama dalam baris tersebut adalah 1. Bisa kita sebut dengan **1 utama/pertama**.
2. Jika terdapat baris yang semuanya elemennya bernilai nol, maka semua baris yang seperti itu harus dikelompokkan dan diletakkan di bawah matriks.
3. Setiap dua baris yang berurutan yang memenuhi sifat ke-1, maka 1 utama dalam baris yang lebih rendah letaknya harus **lebih kekanan** dari 1 utama dalam baris yang lebih tinggi.

Berikut adalah matriks yang memenuhi syarat matriks eselon baris.



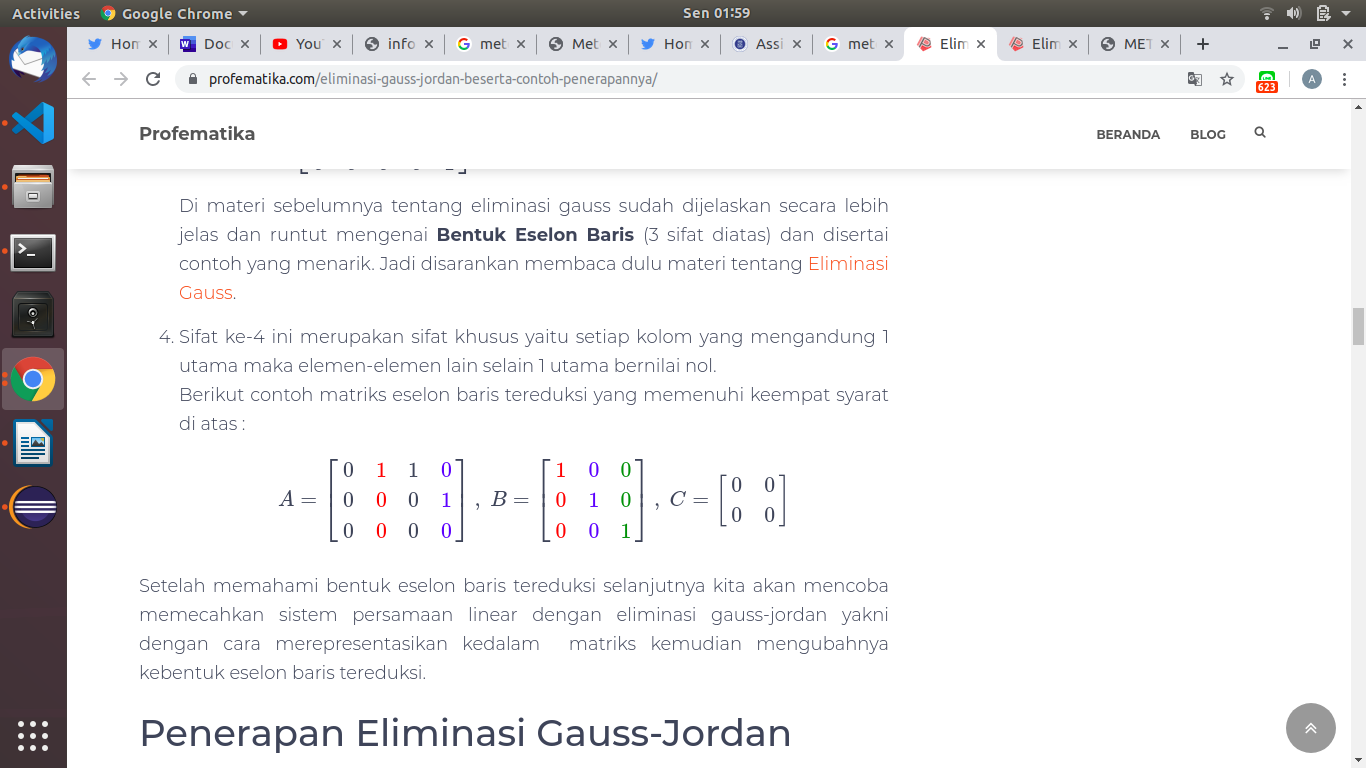
Setelah dilakukan OBE, untuk mendapatkan solusi dari sebuah SPL dilakukan substitusi mundur:





Pada metode eliminasi Gauss-Jordan, matriks augmented akan dioperasikan sehingga menjadi matriks eselon baris tereduksi. Matriks Eselon Baris Tereduksi adalah sebuah bentuk matriks eselon baris yang lebih disederhanakan yang bertujuan agar lebih mudah dalam pencarian pemecahan (solusi) dari suatu sistem persamaan .

Agar mencapai bentuk eselon baris tereduksi diperlukan 4 sifat yang terdiri 3 sifat bentuk eselon baris dan 1 sifat tambahan yaitu setiap kolom yang mengandung 1 utama maka elemen-elemen lain selain 1 utama bernilai nol. Berikut adalah contoh matriks eselon baris tereduksi.



# **BAB III**

# **IMPLEMENTASI PROGRAM**

## Class BacaFile

Class BacaFile adalah class yang berisi fungsi-fungsi khusus untuk menangani pembacaan file disimpan dalam file bernama BacaFile.java.

Atribut dari objek BacaFile ada 2 yaitu:

* + - 1. filename

private String filename;

filename adalah atribut bertipe string yang digunakan untuk menyimpan nama file yang akan dibaca.

* + - 1. opsiBaca

private int opsiBaca;

opsiBaca adalah atribut bertipe int yang menyimpan opsi pembacaan file. Opsi baca yang valid ada 3 yaitu:

/\* Opsi Baca \*/

/\*

1. baca SPL (Matriks nxn+1)

2. baca determinan (matriks nxn) (sama juga buat matriks balikan

dan kofaktor, adjoin juga)

3. baca interpolasi (kumpulan titik" x y)

\*/

Untuk *instantiate* sebuah objek BacaFile digunakan konstruktor:

public BacaFile (String filename, int opsiBaca);

Konstruktor BacaFile digunakan untuk membentuk objek BacaFile dari komponen-komponennya yaitu filename dan opsiBaca.

Adapun fungsi/prosedur beserta spesifikasinya yang berada di dalam class ini adalah:

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static int[] jumlahData(String filename); | /\* Mengembalikan panjang data dengan dimensi(bentuk data) \*/  /\* Nilai pertama mengembalikan nilai panjang file \*/  /\* Nilai kedua mengembalikan nilai lebar file \*/ |
| public Matrix inputMatrix(); | /\* Dilakukan pembacaan file dari atribut filename sesuai opsiBaca \*/  /\* Mengembalikan Matrix yang berasal dari file tersebut \*/ |
| public Matrix inputPersamaanMatrix(); | /\* Dilakukan pembacaan file dari atribut filename sesuai opsiBaca \*/  /\* Dilakukan pembacaan Matrix sistem persamaan linear \*/ |
| public double[][] inputTitik(); | /\* Dilakukan pembacaan file dari atribut filename sesuai opsiBaca \*/  /\* Dilakukan pembacaan titik disimpan dalam matriks / array multidimensional \*/ |

## Class TulisFile

Class TulisFile adalah class yang berisi fungsi-fungsi khusus untuk menangani penulisan output ke dalam sebuah file. Class ini disimpan dalam file bernama TulisFile.java.

Atribut dari objek TulisFile ada 2 yaitu:

* + - 1. filename

private String filename;

filename adalah atribut bertipe string yang digunakan untuk menyimpan nama file yang akan dibaca.

* + - 1. opsiTulis

private int opsiTulis;

opsiTulis adalah atribut bertipe int yang menyimpan opsi penulisan file. opsiTulis yang valid ada 3 yaitu:

/\* Opsi Tulis \*/

/\*

1. keluaran SPL ( Y x ^ 4 bla blabla)

2. baca determinan (matriks nxn) (sama juga buat matriks balikan

dan kofaktor, adjoin juga) ( keluaran bentuk matriks)

1. baca interpolasi (kumpulan titik" x y) ( keluaran : polinom

dan titik)

\*/

Adapun fungsi/prosedur beserta spesifikasinya yang berada di dalam class ini adalah:

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static void tulisHasilMatriks(String filename, Matrix matrix); | /\* I. S. filename sembarang, matrix terdefinisi yang akan disimpan \*/  /\* F. S. filename berisi matrix hasil \*/  /\* Mengeluarkan pesan error apabila filename tidak ada/tidak valid \*/  /\* Mengeluarkan pesan error apabila gagal menyimpan ke filename \*/ |
| public static void tulisHasilInterpolasi(String filename,String polinom,double X,double Y); | /\* I. S. filename sembarang, polinom hasil interpolasi terdefinisi yang akan disimpan \*/  /\* F. S. filename berisi polinom dan nilai hasil interpolasi \*/  /\* Mengeluarkan pesan error apabila filename tidak ada/tidak valid \*/  /\* Mengeluarkan pesan error apabila gagal menyimpan ke filename \*/ |

## Class Matrix

Class Matrix adalah class yang berisi atribut, fungsi, dan prosedur yang menangani objek matriks. Class ini disimpan dalam file Matrix.java.

Atribut dari objek Matrix ada 3 yaitu:

* + - 1. matrix

private double matrix[][]; // matrix container

matrix adalah atribut bertipe double yang menyimpan elemen elemen matriks dalam array multidimensional.

* + - 1. M

private int M; // rows size

M adalah atribut bertipe integer yang menyimpan banyaknya baris dalam suatu objek Matrix.

* + - 1. N

private int N; // columns size

N adalah atribut bertipe integer yang menyimpan banyaknya kolom dalam suatu objek Matrix.

Untuk *instantiate* sebuah objek Matrix digunakan 3 buah konstruktor:

1. Konstruktor Matrix kosong 1x1

public Matrix ()

Konstruktor tersebut digunakan untuk membentuk objek Matrix yang terdiri dari satu baris dan satu kolom. Matrix yang terbentuk siap diisi elemennya.

1. Konstruktor Matrix kosong mxn

public Matrix (int m, int n);

Konstruktor tersebut digunakan untuk membentuk objek Matrix yang terdiri dari m baris dan n kolom. Matrix yang terbentuk siap diisi elemennya.

1. Konstruktor Matrix dari sebuah array multidimensional

public Matrix (double[][] data);

Konstruktor tersebut digunakan untuk membentuk objek Matrix yang atribut matrixnya adalah data, M nya adalah jumlah baris dari data, N nya adalah jumlah kolom dari data.

Fungsi dan prosedur dari Class Matrix terbagi menjadi empat bagian:

1. Getter

Getter terdiri atas 3 fungsi yang masing-masing mengembalikan nilai atribut dari Matrix.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public int getRow(); | /\* Mengembalikan jumlah baris Matrix \*/ |
| public int getColumn(); | /\* Mengembalikan jumlah kolom Matrix \*/ |
| public double[][] getMatrix(); | /\* Mengembalikan komponen matrix \*/ |

1. Setter

Setter terdiri atas 4 prosedur yang masing-masing mengubah nilai atribut/bagian Matrix dengan nilai parameternya.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public void setRow(int m); | /\* Untuk mengubah elemen M dari Matrix menjadi m \*/ |
| public void setColumn(int n); | /\* Untuk mengubah elemen N dari Matrix menjadi n \*/ |
| public void setMatrix (double [][] arr); | /\* Untuk mengubah elemen matrix dari Matrix menjadi arr \*/ |
| public void setElmtMatrix (double x, int i, int j); | /\* Untuk mengubah elemen matrix baris i kolom j dengan nilai x \*/ |

1. Operasi pada Matrix

Operasi pada Matrix terdiri atas 6 fungsi/prosedur yang berkaitan dengan operasi dasar matriks.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public void swap\_row(int i, int j); | /\* I. S. Matrix terdefinisi \*/  /\* F. S. Matriks baris i telah bertukar tempat dengan matriks baris j \*/ |
| public static Matrix delBrsMatrix (Matrix M, int idx\_brs); | /\* Matrix M terdefinisi, idx\_brs <= baris M\*/  /\* Mengembalikan Matrix yang menghilangkan baris ke idx\_brs \*/ |
| public static Matrix delKolMatrix (Matrix M, int idx\_kol); | /\* Matrix M terdefinisi, idx\_kol <= kolom M\*/  /\* Mengembalikan Matrix yang menghilangkan kolom ke idx\_kol \*/ |
| public static Matrix multiplication (Matrix M, Matrix N); | /\* Matrix M dan N terdefinisi dan jumlah kolom M = jumlah baris N \*/  /\* Mengembalikan matriks hasil perkalian MxN \*/ |
| public static Matrix minor (Matrix M, int idx\_brs, int idx\_kol); | /\* Matrix M terdefinisi dan harus nxn, idx\_brs <= baris M, idx\_kol <= kolom M \*/  /\* Mengembalikan sub matriks/minor Matrix yang menghilangkan baris ke idx\_brs dan kolom ke idx\_kol \*/ |
| public static Matrix transpose (Matrix M); | /\* Matrix M terdefinisi, tidak harus nxn \*/  /\* Mengembalikan Matrix transpose baris dan kolomnya \*/  /\* Apabila M adalah matrix 2x3, maka keluarannya adalah matrix 3x2 \*/ |

1. Fungsi/Prosedur Lain

Fungsi/prosedur lain pada Class Matrix meliputi 5 fungsi/prosedur yaitu:

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public Boolean isSquare(); | /\* Mengembalikan true apabila ukuran Matrix nxn \*/ |
| public void show(); | /\* I. S. Matrix terdefinisi \*/  /\* F. S. Matrix ditampilkan pada layar \*/ |
| public static void input(Matrix M); | /\* I.S. Matrix M sembarang \*/  /\* F.S. terbentuk matriks M terdefinisi dengan jumlah baris dan kolom sesuai input \*/ |
| public static void inputSPL(Matrix M); | /\* I.S. Matrix M sembarang \*/  /\* F.S. terbentuk Matrix M terdefinisi \*/  /\* Matrix M berupa Matrix augmented \*/ |
| public static void inputNxN(Matrix M); | /\* I.S. Matrix M sembarang \*/  /\* F.S. terbentuk Matrix M terdefinisi \*/  /\* Matrix M berupa Matrix berukuran nxn \*/ |

## Class Dependencies

Class Dependencies adalah class yang berisi fungsi atau prosedur tambahan untuk operasi Matrix. Class ini disimpan dalam file bernama Dependencies.java. Class ini tidak memiliki atribut dan hanya berisi *static function/procedure*.

Fungsi dan prosedur di dalam Class Dependencies dibagi menjadi 3 bagian:

1. Fungsi untuk menghitung determinan

Dalam menghitung determinan, dapat digunakan 3 cara yang masing-masing diimplementasikan dalam fungsi berikut.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static double kofaktorDet (Matrix M); | /\* Matrix M terdefinisi berukuran nxn \*/  /\* Mengembalikan determinan matriks menggunakan metode kofaktor \*/ |
| public static double sarrusDet (Matrix M); | /\* M harus berukuran 3x3 \*/  /\* Mengembalikan determinan matriks M menggunakan metode sarrus \*/ |
| public static double detOBE(Matrix M); | /\* Matrix M terdefinisi berukuran nxn \*/  /\* Mengembalikan determinan matriks M menggunakan metode operasi baris elementer \*/ |

1. Fungsi untuk menghitung invers

Dalam menghitung invers, dapat digunakan 2 cara yang masing-masing diimplementasikan dalam fungsi pada tabel di bawah. Pada bagian ini terdapat pula fungsi antara yang dibuat untuk memudahkan pembuatan fungsi utama.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static Matrix inversAdj (Matrix M); | /\* Matrix M terdefinisi dan merupakan matrix nxn \*/  /\* Mengembalikan matriks balikan dari matrix M dengan metode adjoin \*/ |
| public static Matrix createInversMatrix (Matrix M); | /\* Fungsi untuk membentuk augmented matrix yang digunakan dalam proses invers Matriks\*/  /\* Membentuk matriks dengan ukuran dari nxn menjadi nx2n\*/  /\* Matriks yang baru dibentuk berupa matriks ones\*/ |
| public static Matrix inversGauss (Matrix M); | /\* Matrix M terdefinisi dan merupakan matrix nxn \*/  /\* Mengembalikan matriks balikan dari matrix M dengan metode gauss - jordan \*/ |

1. Fungsi lain

Fungsi lain dalam Class Dependencies berisi fungsi tambahan operasi matrix sebagai berikut.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static Matrix hitungKofaktor(Matrix M); | /\* Matrix M terdefinisi dan harus nxn \*/  /\* Mengembalikan matriks kofaktor dari M \*/ |
| public static Matrix hitungAdjoin (Matrix M); | /\* Matriks M terdefinisi dan harus nxn \*/  /\* Mengembalikan adjoin dari matriks M \*/ |

## Class Interpolasi

Class Interpolasi adalah class yang berisi fungsi/prosedur khusus untuk menangani persoalan interpolasi. Class ini disimpan dalam file bernama Interpolasi.java.

Atribut dari objek Interpolasi ada 6 yaitu:

1. matrix

Matrix matrix;

matrix adalah atribut bertipe Matrix yang menyimpan augmented matrix dari titik-titik yang diinput.

1. N

int N;

N adalah atribut bertipe int yang menyimpan jumlah derajat polinom.

1. polinom

double[] polinom;

polinom adalah atribut array bertipe double yang menyimpan polinom hasil interpolasi.

1. arr\_X

double[] arr\_X;

arr\_X adalah atribut array bertipe double yang menyimpan nilai titik X.

1. arr\_Y

double[] arr\_Y;

arr\_Y adalah atribut array bertipe double yang menyimpan nilai titik Y.

1. allDefined

boolean allDefined = false;

allDefined adalah atribut bertipe boolean yang bernilai true jika semua atribut dari Interpolasi sudah didefinisikan. Pada saat *instantiate* objek, nilai dari allDefined adalah false.

Pada class ini, didefinisikan pula scanner utama yang memanfaatkan java.util.Scanner:

Scanner scanner = new Scanner(System.in);

Adapun fungsi/prosedur beserta spesifikasinya yang berada di dalam class ini adalah:

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public void inputTitik(); | /\* I.S. : Titik belum dimasukkan \*/  /\* F.S. : Titik telah terdefinisi \*/  /\* Kondisi Valid : Jumlah titik yang dimasukkan minimal 2, agar tidak menimbulkan keambiguan \*/ |
| public void inputTitikFile(String filename); | /\* I.S. : Titik belum dimasukkan, file terdefinisi \*/  /\* F.S. : Titik telah terdefinisi \*/  /\* Kondisi Valid : Jumlah titik yang dimasukkan minimal 2, agar tidak menimbulkan keambiguan \*/ |
| public String getOutputPolinom (); | /\* Mengembalikan string polinom hasil interpolasi \*/ |
| public static double [] getResultGaussian (Matrix matrix); | /\* Mengembalikan array hasil dari perhitungan Gauss-Jordan \*/ |
| public static double[] reverse(double a[], int n); | /\* Mengembalikan array yang elemennya sama dengan a tetapi dibalik urutannya \*/ |
| public static double getInterpolasi (boolean allDefined, double x, int N, double[] polinom); | /\* Mengembalikan nilai y hasil polinom \*/  /\* apabila ada parameter yang tidak terpenuhi mengembalikan -999 \*/ |
| public static double bacaTitikBaru(); | /\* Mengembalikan nilai titik yang dibaca lewat scanner \*/ |
| public void ubahPersLinear(); | /\* I. S. : Titik telah diinput kedalam objek\*/  /\* F. S. : Terdapat augmented matrix dari persamaan garis nilai titik yang telah diinput \*/  /\* Fungsi mengubah titik" menjadi sistem persamaan linear (dalam bentuk augmented matrix) \*/ |
| public void polinomInterpolasi(); | /\*I. S. Fungsi Polinom hasil interpolasi belum terdefinisi \*/  /\*F. S. Fungsi Polinom hasil interpolasi telah terdefinisi\*/ |
| public void prosedurInterpolasi(); | /\* Prosedur untuk menjalankan semua fungsi interpolasi \*/ |

## Class SPL

Class SPL adalah class yang berisi fungsi atau prosedur untuk mencari solusi sistem persamaan linier. Class ini disimpan dalam file bernama SPL.java. Class ini tidak memiliki atribut dan hanya berisi *static function/procedure*.

Fungsi dan prosedur di dalam Class SPL dibagi menjadi 3 bagian berdasarkan metode penyelesaian SPL:

1. Metode Invers

Sistem persamaan linier dapat diselesaikan dengan metode invers hanya jika determinan matriks koefisiennya tidak sama dengan nol. Implementasi dari penyelesaian soal tersebut terdapat pada fungsi/prosedur berikut.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static double[] solveSPLInvers(Matrix M); | /\* Mengeluarkan array solusi SPL dari Matriks augmented M berukuran nxn+1 \*/  /\* Matrix koefisien M (tanpa kolom terakhir) harus memiliki determinan != 0 \*/ |
| public static void showResultInv(Matrix M); | /\* I. S. Matrix M adalah Matrix SPL augmented \*/  /\* Menampilkan solusi SPL ke layar \*/  /\* Mengeluarkan pesan error apabila SPL tidak dapat diselesaikan dengan metode invers \*/ |

1. Metode Crammer

Sistem persamaan linier dapat diselesaikan dengan metode crammer hanya jika determinan matriks koefisiennya tidak sama dengan nol. Implementasi dari penyelesaian soal tersebut terdapat pada fungsi/prosedur berikut.

|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static double[] solveSPLCrammer(Matrix M); | /\* Mengeluarkan array solusi SPL dari Matriks augmented M berukuran nxn+1 \*/  /\* Determinan koefisien M (tanpa kolom terakhir) tidak boleh 0 \*/ |
| public static void showResultCrammer(Matrix M); | /\* I. S. Matrix M adalah Matrix spl augmented \*/  /\* Mengeluarkan pesan error apabila SPL tidak dapat diselesaikan dengan metode crammer \*/ |

1. Metode Gauss dan Gauss-Jordan

Sistem persamaan linier dapat diselesaikan dengan metode Gauss dan Gauss-Jordan tanpa prekondisi apapun. Pada program ini, terdapat 3 kemungkinan solusi yaitu solusi tunggal, solusi banyak, dan tidak ada solusi. Implementasi dari penyelesaian soal tersebut terdapat pada fungsi/prosedur berikut.

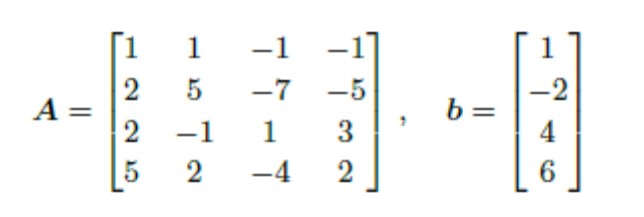
|  |  |
| --- | --- |
| Nama Fungsi/Prosedur | Spesifikasi |
| public static void makeUrutMatriks(double[][] M\_in, int n\_brs, int n\_kol); | /\* I. S. M terdefinisi \*/  /\* F. S. Membuat M menjadi matriks segitiga bawah (kumpulan 0 di kiri bawah) \*/ |
| public static double makeUrutMatriksDet(Matrix M); | /\* M terdefinisi, n adalah baris dan kolom M, count = 1 \*/  /\* Mengembalikan hasil perkalian -1 tiap di swap \*/ |
| public static int getIdxFirstNonZero(Matrix M, int i); | /\* Prekondisi matriks ukuran n\*(n+1) \*/  /\* Mengembalikan index bukan nol pertama dalam matrix pada baris i \*/ |
| public static void solveGaussDet(Matrix M); | /\* I. S. Matrix M terdefinisi \*/  /\* F. S. Matrix M menjadi bentuk eselon \*/ |
| public static void solveGauss(Matrix M); | /\* I. S. Matrix M terdefinisi \*/  /\* F. S. Matrix M menjadi bentuk eselon, diagonalnya semua 1 \*/ |
| public static void solveGaussJordan(Matrix M); | /\* I. S. Matrix M terdefinisi dan adalah matriks eselon \*/  /\* F. S. Matrix M menjadi bentuk eselon tereduksi \*/ |
| public static int countBarisKosong(Matrix M); | /\* Mengembalikan jumlah baris yang kosong pada suatu matriks eselon \*/ |
| public static boolean cekNoSolution (Matrix M); | /\* Mengembalikan true jika M tidak mempunyai solusi \*/ |
| public static void generateMultiSolutionGaussJordan (Matrix M); | /\* I. S. Matrix M terdefinisi \*/  /\* Membentuk array koefisien untuk menampung variable bebas \*/  /\* Menampilkan solusi bebas ke layar dengan metode gauss-jordan\*/ |
| public static void generateMultiSolutionGaussJordanFile (Matrix M, String filename); | /\* I. S. Matrix M terdefinisi, dibaca dari filename \*/  /\* Membentuk array koefisien untuk menampung variable bebas \*/  /\* Menampilkan solusi bebas ke layar dengan metode gauss-jordan \*/ |
| public static void generateMultiSolutionGauss (Matrix M); | /\* I. S. Matrix M terdefinisi \*/  /\* Membentuk array koefisien untuk menampung variable bebas \*/  /\* Menampilkan solusi bebas ke layar dengan metode gauss \*/ |
| public static void showResultGauss(Matrix M); | /\* I. S. Matrix M terdefinisi \*/  /\* Menampilkan solusi tergantung kasus (solusi unik, tidak ada solusi, solusi banyak) ke layar \*/  /\* Menggunakan metode gauss \*/ |
| public static void showResultGaussFile(Matrix M, String filename); | /\* I. S. Matrix M terdefinisi dibaca dari filename \*/  /\* Menampilkan solusi tergantung kasus (solusi unik, tidak ada solusi, solusi banyak) ke layar \*/  /\* Menggunakan metode gauss \*/ |
| public static void showResultGaussJordan(Matrix M); | /\* I. S. Matrix M terdefinisi \*/  /\* Menampilkan solusi tergantung kasus (solusi unik, tidak ada solusi, solusi banyak) ke layar \*/  /\* Menggunakan metode gauss-jordan \*/ |
| public static void showResultGaussJordanFile(Matrix M, String filename); | /\* I. S. Matrix M terdefinisi dibaca dari filename \*/  /\* Menampilkan solusi sesuai kondsi (solusi unik, tidak ada solusi, solusi banyak) ke layar \*/  /\* Menggunakan metode gauss-jordan \*/ |

## Class App

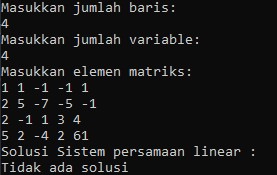
# **BAB IV**

# **EKSPERIMEN**

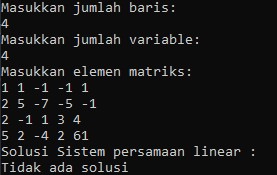
1. SPL
2. Testcase 1



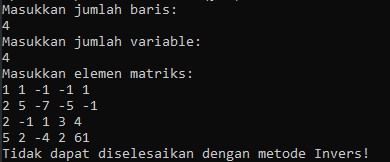
Solusi menggunakan metode gauss :



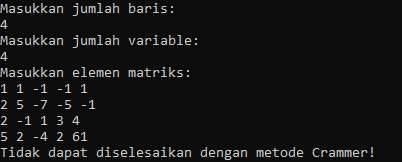
Solusi menggunakan metode gauss Jordan :



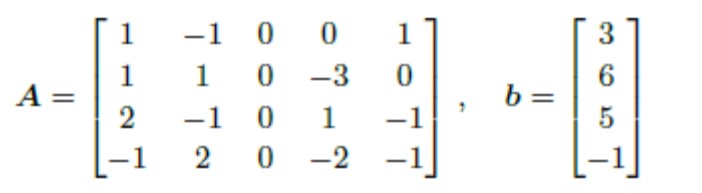
Solusi menggunakan metode Invers :



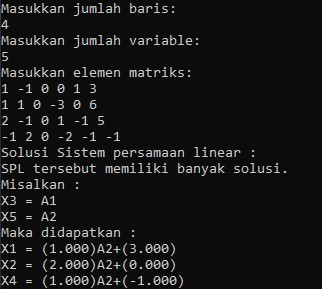
Solusi menggunakan metode Crammer :



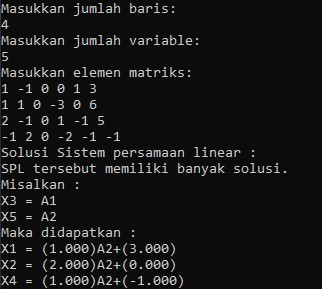
1. Testcase 2



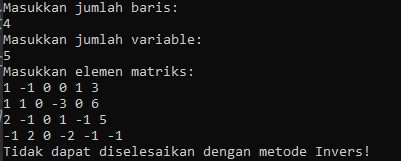
Solusi menggunakan metode Gauss :



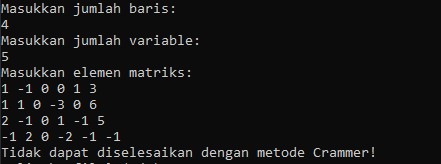
Solusi menggunakan metode Gauss Jordan :



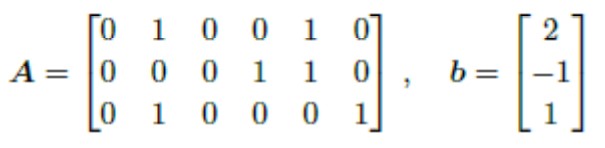
Solusi menggunakan metode Invers :



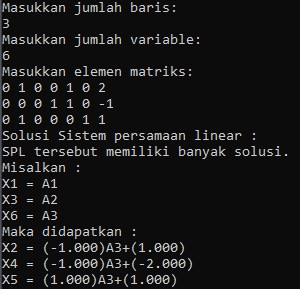
Solusi menggunakan metode Crammer :



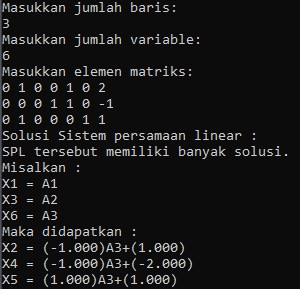
1. Testcase 3



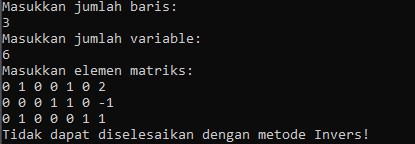
Solusi menggunakan metode Gauss :



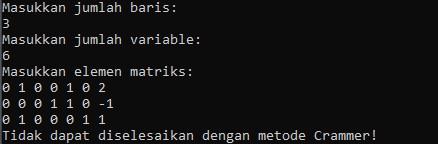
Solusi menggunakan metode Gauss Jordan :



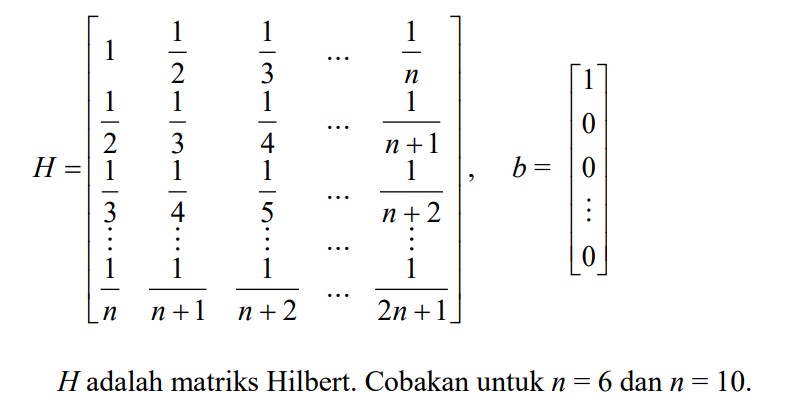
Solusi menggunakan metode Invers :



Solusi menggunakan metode Crammer :



1. Testcase 4

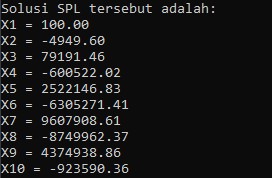


Solusi menggunakan metode Gauss :

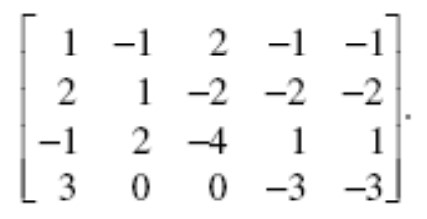
Solusi menggunakan metode Gauss Jordan :

Solusi menggunakan metode Invers :

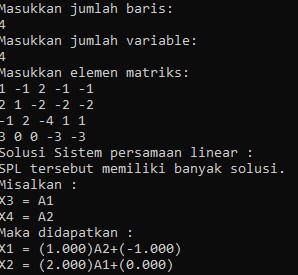
Solusi menggunakan metode Crammer :



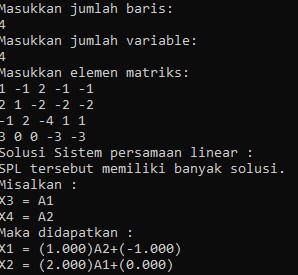
1. Testcase 5



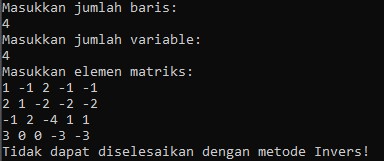
Solusi menggunakan metode Gauss :



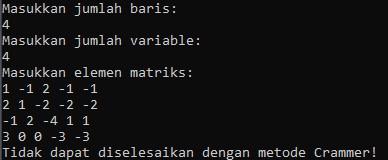
Solusi menggunakan metode Gauss Jordan :



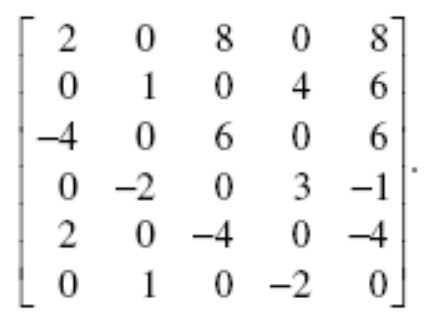
Solusi menggunakan metode Invers :



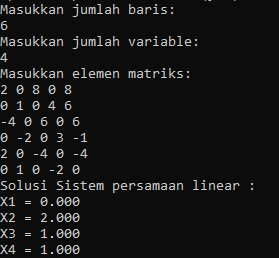
Solusi menggunakan metode Crammer :



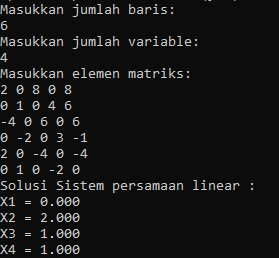
1. Testcase 6



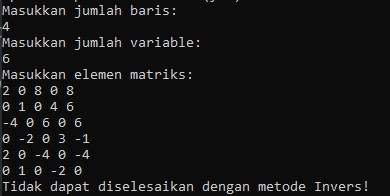
Solusi menggunakan metode Gauss :



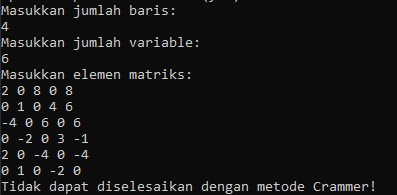
Solusi menggunakan metode Gauss Jordan :



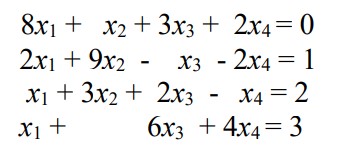
Solusi menggunakan metode Invers :



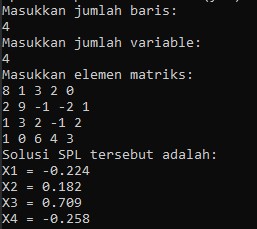
Solusi menggunakan metode Crammer :



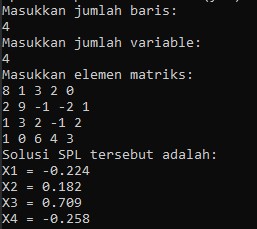
1. Testcase 7



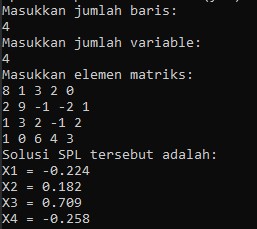
Solusi menggunakan metode Gauss :



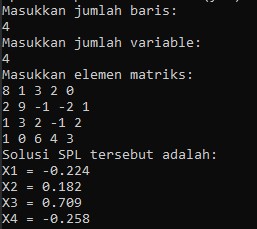
Solusi menggunakan metode Gauss Jordan :



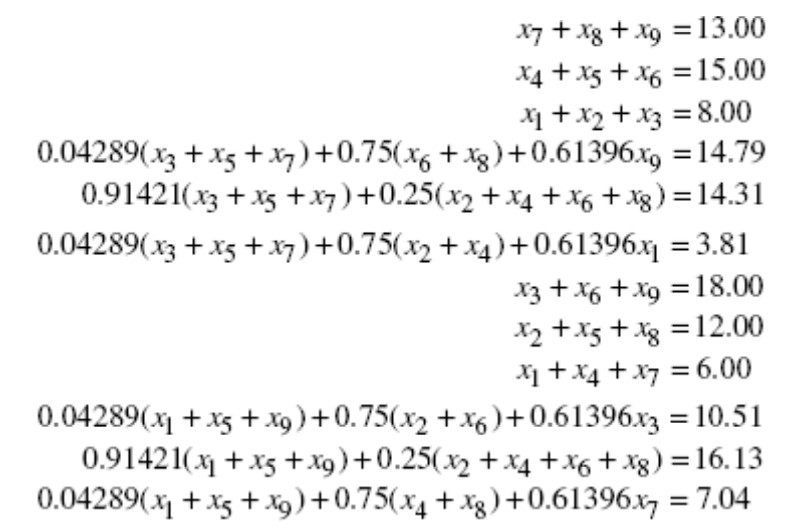
Solusi menggunakan metode Invers :



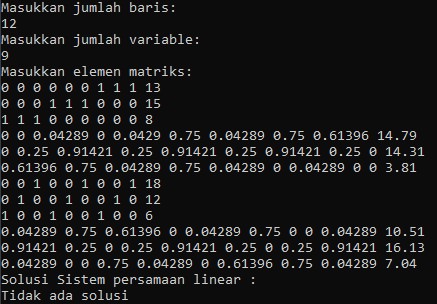
Solusi menggunakan metode Crammer :



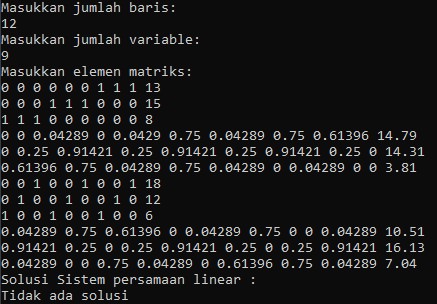
1. Testcase 8



Solusi menggunakan metode Gauss :



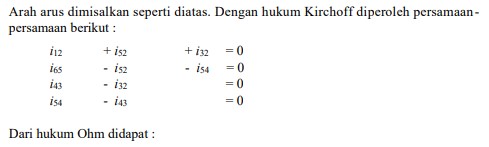
Solusi menggunakan metode Gauss Jordan :

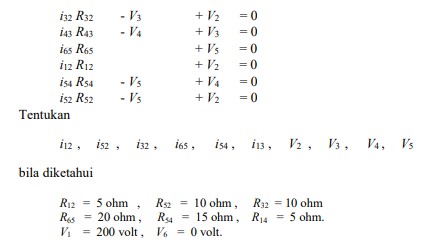


Solusi menggunakan metode Invers :

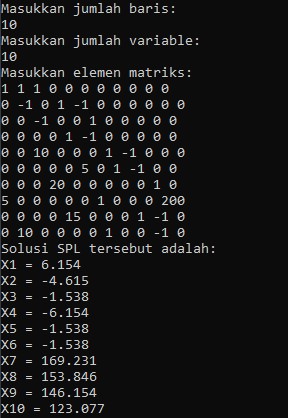
Solusi menggunakan metode Crammer :

1. Testcase 9

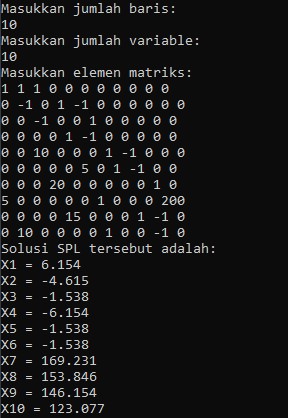




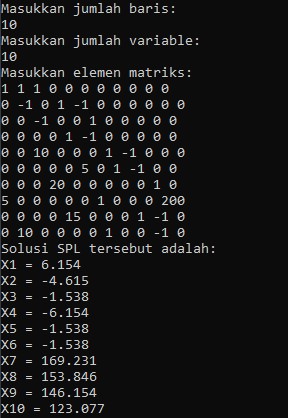
Solusi menggunakan metode Gauss :



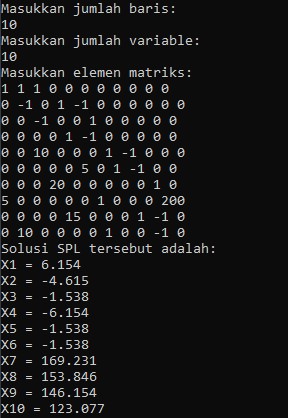
Solusi menggunakan metode Gauss Jordan :



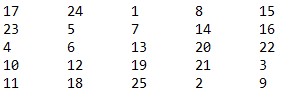
Solusi menggunakan metode Invers :



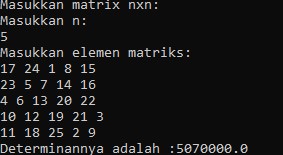
Solusi menggunakan metode Crammer :



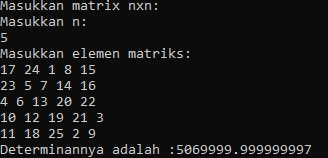
1. Determinan, Matriks balikan, Matriks Kofaktor, Matriks Adjoin
2. Testcase Matriks 5 x 5



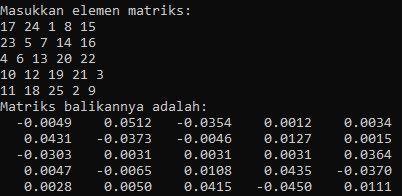
Determinan metode kofaktor :



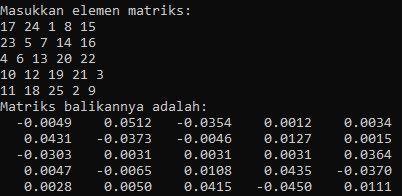
Determinan metode OBE :



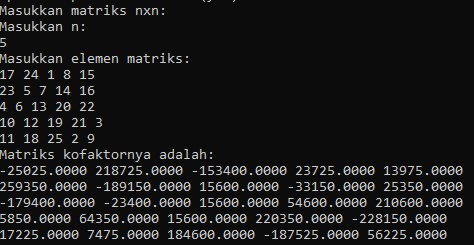
Matriks balikan metode Gauss-Jordan:



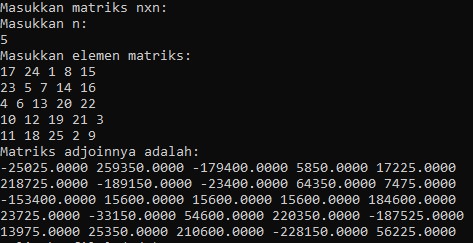
Matriks balikan metode adjoin:



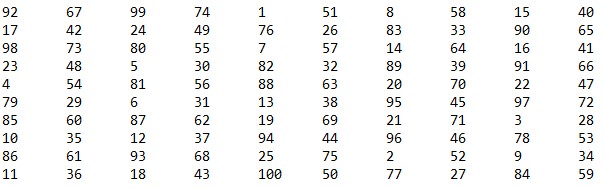
Matriks Kofaktor :



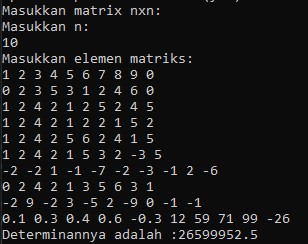
Matriks Adjoin :



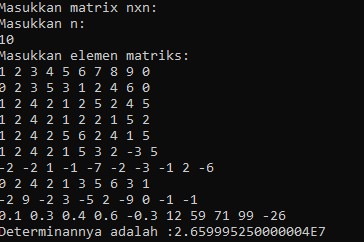
1. Testcase Matriks 10 x 10



Determinan Kofaktor :



Determinan OBE:



Matriks balikan metode gauss Jordan :

Matriks balikan metode OBE :

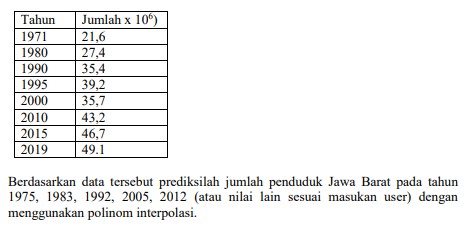
Matriks kofaktor :

Matriks Adjoin :

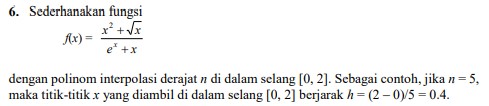
1. Interpolasi

Dfs fsdf

1. Testcase 1



1. Tesecase 2



# **BAB V**

# **KESIMPULAN DAN SARAN**