## **SISTEM PERSAMAAN LINIER, DETERMINAN,**

## **DAN APLIKASINYA**

## 

## **LAPORAN TUGAS BESAR**

## Diajukan Untuk Memenuhi Tugas IF 2123 Aljabar Linier dan Geometri

## 

## oleh

## **ARJUNA MARCELINO 13519021**

## **GIANT ANDREAS TAMBUNAN 13519127**

## **KEVIN KATSURA D. SITANGGANG 13518216**

## 

## 

## **TEKNIK INFORMATIKA**

## **INSTITUT TEKNOLOGI BANDUNG**

## **BANDUNG**

## **2020**

## 

## **BAB I**

## **DESKRIPSI MASALAH**

## 

## Sistem persamaan linier (SPL) 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 R. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan metode Cramer. Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

## Sistem persamaan linier 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*1*n* = *y*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

## 

## *a*0 + 8.0*a*1 + 64.00*a*2 = 2.0794

## *a*0 + 9.0*a*1 + 81.00*a*2 = 2.1972

## *a*0 + 9.5*a*1 + 90.25*a*2 = 2.2513

## Penyelesaian sistem persamaandengan metode eliminasi Gauss menghasilkan *a*0 = 0.6762, *a*1 = 0.2266, dan *a*2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah *p*2(*x*) = 0.6762 + 0.2266*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.

## 

## **SPESIFIKASI TUGAS**

## Buatlah program dalam Bahasa Java untuk

## Menghitung solusi SPL dengan metode eliminasi metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan *n* pebuah dan *n* persamaan).

## Menyelesaikan persoalan interpolasi.

## Menghitung determinan matriks dengan berbagai cara yang disebutkan di atas, matriks kofaktor, dan matriks *adjoin* dari sebuah matriks n x n.

## Spesifikasi program adalah sebagai berikut:

## 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

## 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

-3 7 8.3

0.5 -10 -9

1. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah *n*, (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*, *yn*), dan nilai *x* yang akan ditaksir nilai fungsinya. Jika masukanya 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

1. 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 = 2*s* – *t*, *x*2 = *s*, dan *x*1 = *t*.)
2. Untuk persoalan determinan, matriks balikan, matriks kofator, dan adjoin, maka luarannya sesuai dengan persoalan masing-masing
3. Untuk persoalan polinom interpolasi, luarannya adalah persamaan polinom dan taksiran nilai fungsi pada *x* yang diberikan.
4. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam file**.
5. Bahasa program yang digunakan adalah Java.
6. Program **tidak harus** berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas *Eclipse* misalnya).
7. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan ditrancang masing-masing. Misalnya, menu:

MENU

1. Sistem Persamaaan Linier

2. Determinan

3. Matriks balikan

4. Matriks kofaktor

5. Adjoin

6. Interpolasi Polinom

7. 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.

## 

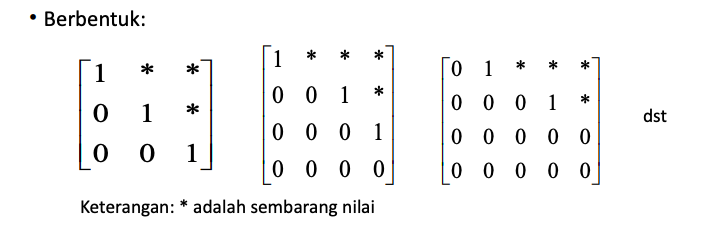
## **BAB II**

## **TEORI SINGKAT**

## 

## **2.1. MATRIKS ESELON**

Matriks eselon (atau bentuk eselon baris) adalah matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol.

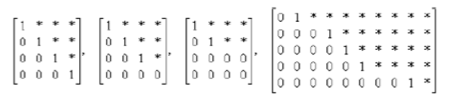
Berbentuk :

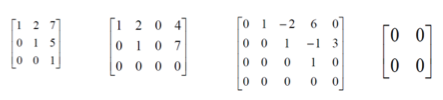
Sifat-sifat matriks eselon baris:

1. Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama)

2. Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.

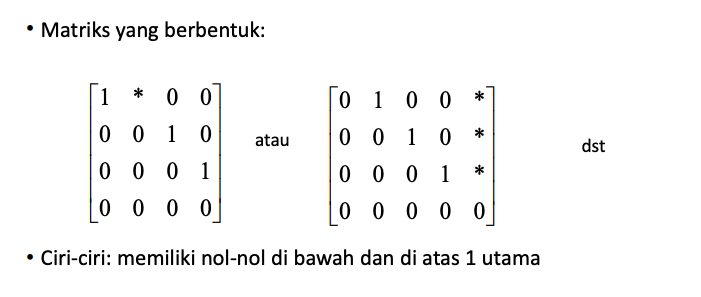
3. Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.

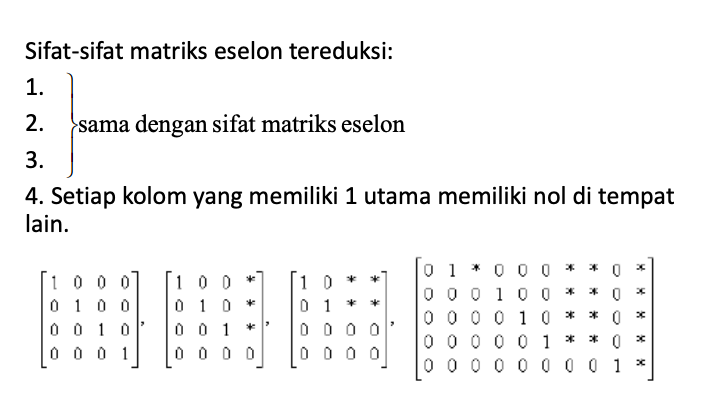




**2.2. Matriks Eselon Tereduksi**

Matriks eselon tereduksi adalah matriks yang di atas dan di bawah 1 utama nya bernilai nol. Matriks yang diperoleh setelah melakukan beberapa langkah proses eliminasi Gaussian dikatakan dalam bentuk eselon atau bentuk eselon baris.







Matriks dalam bentuk eselon memiliki sifat-sifat berikut.

• Semua baris lengkap dengan nol ada di bagian bawah

• Nilai bukan nol pertama di baris bukan nol bergeser ke kanan relatif terhadap suku bukan nol pertama di baris sebelumnya (lihat contoh)

• Setiap baris yang bukan nol dimulai dengan 1

Perbedaan antara Matriks Eselon dengan Matriks Eselon Tereduksi :

• Bentuk eselon baris adalah salah satu format matriks yang diperoleh dari proses eliminasi Gaussian.

• Dalam bentuk eselon baris, elemen bukan nol berada di sudut kanan atas, dan setiap baris bukan nol memiliki 1. Elemen bukan nol pertama di baris bukan nol bergeser ke kanan setelah setiap baris.

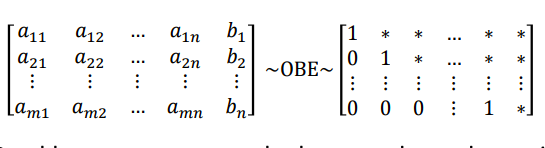
• Proses selanjutnya dari eliminasi Gaussian memberikan matriks yang lebih disederhanakan, di mana semua elemen lain dalam kolom yang berisi 1 adalah nol. Matriks dalam bentuk itu dikatakan dalam bentuk eselon baris tereduksi. Artinya, dalam bentuk eselon baris tereduksi, tidak ada kolom yang menyertakan 1 dan nilai selain nol.

**2.3. Eliminasi Gauss**

Eliminasi gauss ditemukan oleh **Carl Friedrich Gauss,** metode ini dapat dimanfaatkan untuk memecahkan sistem persamaan linear dengan merepresentasikan (mengubah) menjadi bentuk matriks, matriks tersebut lalu diubah ke bentuk **Eselon Baris** melalui [Operasi Baris Elementer.](https://www.profematika.com/pengenalan-operasi-baris-elementer/) Kemudian sistem diselesaikan dengan substitusi balik.

Langkah penerapan Eliminasi Gauss :

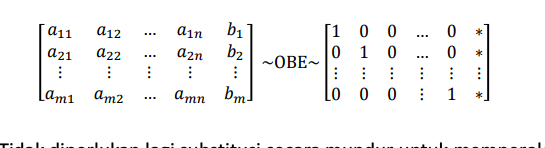
1. Nyatakan SPL dalam bentuk matriks Augmented
2. Terapkan OBE pada matriks augmented sampai berbentuk matriks eselon baris



1. Pecahkan persamaan yang berkoresponden dengan matriks eselon baris

**2.4 Eliminasi Gauss-Jordan**

Eliminasi Gauss-Jordan adalah prosedur pemecahan sistem persamaan linear dengan mengubahnya menjadi bentuk matriks eselon baris tereduksi dengan Operasi Baris Elementer. Metode Eliminasi Gauss-Jordan ini merupakan pengembangan dari Metode Eliminasi Gauss. Operasi baris elementer (OBE) diterapkan pada matriks augmented sehingga menghasilkan matriks eselon baris tereduksi

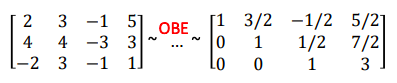


Tidak diperlukan lagi substitusi secara mundur untuk memperoleh nilainilai variabel. Nilai variabel langsung diperoleh dari matriks augmented akhir.

Metode eliminasi Gauss-Jordan terdiri dari dua fase:

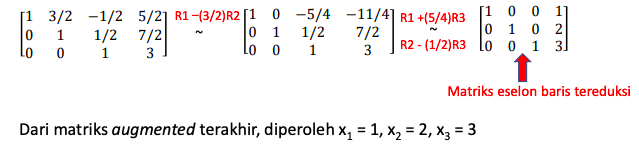
1. Fase maju (forward phase) atau fase eliminasi Gauss

- Menghasilkan nilai-nilai 0 di bawah 1 utama



2. Fase mundur (backward phase)

- Menghasilkan nilai-nilai 0 di atas satu utama

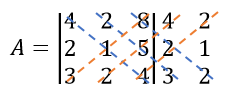


**2.5 Determinan**

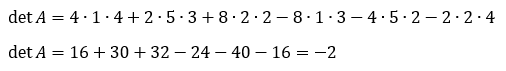
Determinan adalah suatu bilangan yang diperoleh dari suatu proses dengan aturan tertentu terhadap matriks bujur sangkar. Determinan dari sebuah matriks bujur sangkar A, dinotasikan dengan det(A), atau |A|.

Dalam mencari nilai determinan suatu matriks, dapat diperoleh dengan Metode Reduksi Baris dan Kofaktor. Khusus untuk matriks yang berorde 2x2 dan 3x3, pencarian nilai determinan dapat dilakukan dengan Metode Sarrus.

* Aturan *Sarrus*

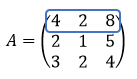


Tarik garis putus-putus seperti gambar di atas. Kalikan elemen-elemen yang terkena garis putus-putus tersebut. Hasil kali elemen yang terkena garis putus-putus berwarna biru diberi tanda positif (+), sedangkan hasil kali elemen yang terkena garis putus-putus berwarna oranye diberi tanda negatif (-). Ingat urutan penulisannya juga, ya!

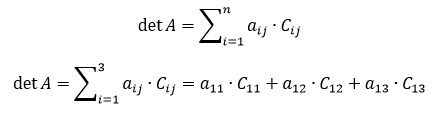


* Metode Minor-Kofaktor

Berdasarkan rumus minor-kofaktor di atas, determinan matriks A dapat dicari dengan menghitung jumlah seluruh hasil kali antara kofaktor matriks bagian dari matriks A dengan elemen-elemen pada salah satu baris atau kolom matriks A. Jadi, pertama, kita pilih salah satu baris atau kolom matriks A untuk mendapatkan nilai determinannya. Misalnya, kita pilih baris ke-1. Elemen-elemen matriks baris ke-1, yaitu a11, a12, dan a13.



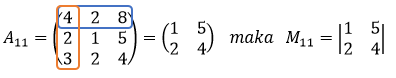
Selanjutnya, karena kita pilih elemen-elemen pada baris ke-1, rumus determinan matriks yang kita gunakan adalah sebagai berikut:



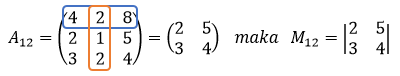
Langkah kedua, kita cari kofaktor matriks bagian dari matriks A (Cij). Cij = (-1)i+j Mij dan Mij = det Aij dengan Aij merupakan matriks bagian dari matriks A yang diperoleh dengan menghilangkan baris ke-i dan kolom ke-j.

Sebelumnya, kita telah memilih elemen-elemen pada baris ke-1, yaitu a11, a12, dan a13. Oleh karena itu, matriks bagian dari matriks A nya adalah A11, A12, dan A13.

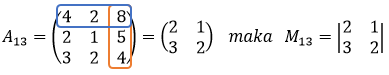
* A11 diperoleh dengan menghilangkan elemen-elemen pada baris ke-1 dan kolom ke-1.



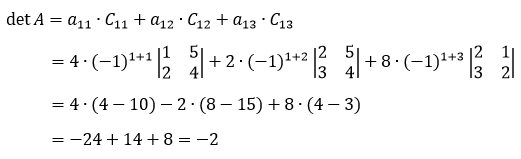
* A12 diperoleh dengan menghilangkan elemen-elemen pada baris ke-1 dan kolom ke-2.



* A13 diperoleh dengan menghilangkan elemen-elemen pada baris ke-1 dan kolom ke-3.

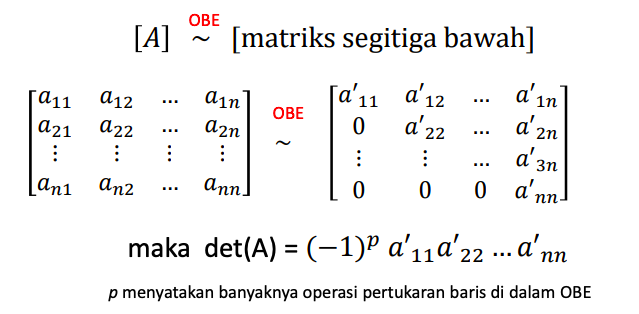


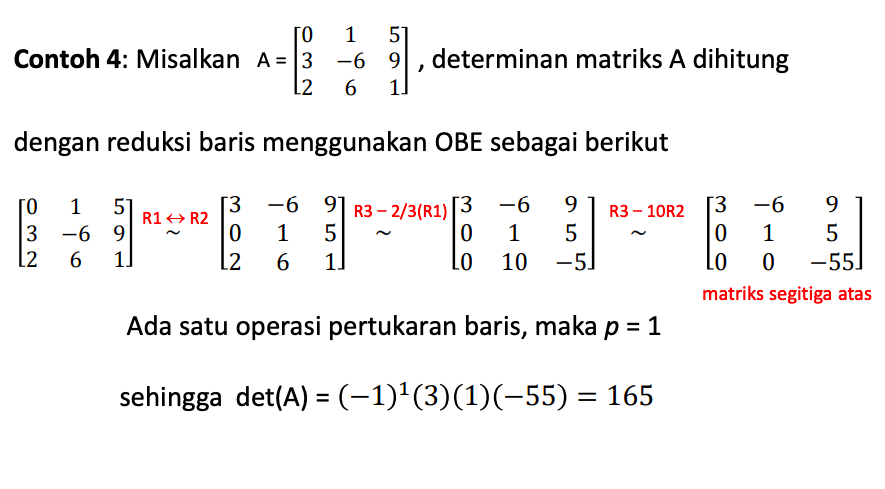
Sehingga,



* Metode Reduksi Baris

Determinan matriks A dapat diperoleh dengan melakukan OBE pada matriks A sampai diperoleh matriks segitiga (segitiga bawah atau atas)





**2.10 Matriks Balikan (Invers)**

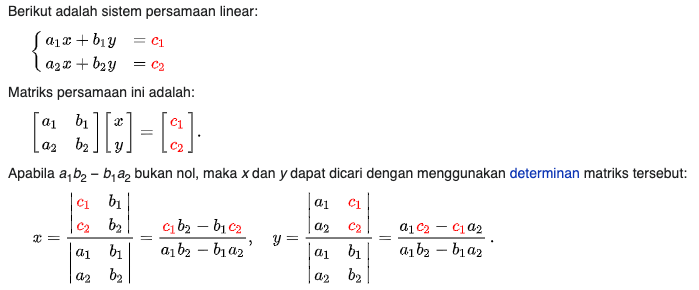
Invers atau Balikan dari suatu matriks dapat dihitung sebagai berikut.



**2.11 Kaidah Cramer**

**Kaidah Cramer** adalah rumus yang dapat digunakan untuk menyelesaikan [sistem persamaan linear](https://id.wikipedia.org/wiki/Sistem_persamaan_linear). Metode ini menggunakan [determinan](https://id.wikipedia.org/wiki/Determinan) suatu [matriks](https://id.wikipedia.org/wiki/Matriks) dan matriks lain yang diperoleh dengan mengganti salah satu kolom dengan vektor yang terdiri dari angka di sebelah kanan persamaannya. Metode ini dinamai dari matematikawan [Swiss](https://id.wikipedia.org/wiki/Swiss) [Gabriel Cramer](https://id.wikipedia.org/w/index.php?title=Gabriel_Cramer&action=edit&redlink=1) (1704–1752)

Contoh penyelesaian Sistem Persamaan Linear menggunakan Kaidah Cramer adalah :





**2.12 Interpolasi Polinomial**

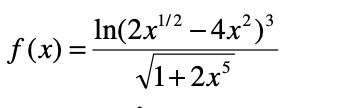
Interpolasi polinomial digunakan untuk mencari titik-titik antara dari n buah titik, P1(x1,y1), P2(x2,y2), P3(x3,y3), …, PN(xN,yN) dengan menggunakan pendekatan fungsi polinomial pangkat n-1:



Besar n didapat dari jumlah titik yang tersedia. Kemudian masukkan titik titik tersebut kedalam persamaan sehingga didapat persamaan dengan variabel a. Setelah itu, eliminasi persamaan sehingga didapat besar variabel a0 sampai a(n-1).

**Aplikasi interpolasi polinom:**

1. Menghampiri fungsi rumit menjadi lebih sederhana Contoh:

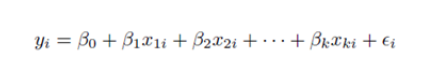


Hitung: f’(x) dan ∫f(x) dx Perhitungan men jadi lebih mudah jika f(x) dihampiri dengan 5 2/1 2 3 1 2 ln( 2 4 ) ( ) x x x f x + − = Perhitungan men jadi lebih mudah jika f(x) dihampiri dengan polinom p(x). Polinom p(x) diperoleh dengan menginterpolasi beberapa titik diskrit dari f(x)

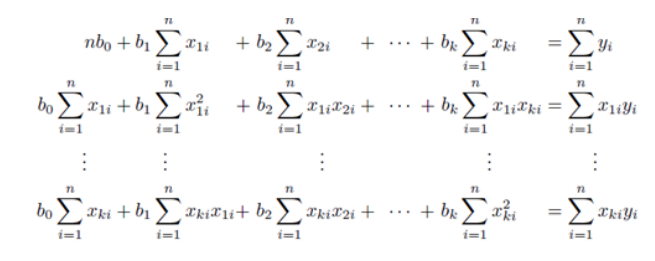
2. Menggambar kurva (jika hanya diketahui titik-titik diskrit saja)

**2.13. Regresi Linear**

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



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



Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss

## 

## **BAB III**

## **IMPLEMENTASI PROGRAM DALAM JAVA**

## **MATRIKS.java**

class MATRIKS ini berisi *procedure* dan *function* untuk memroses masukan menjadi sebuah matriks augmented yang siap digunakan untuk operasi tertentu.

Didefinisikan sebuat tipe bentukan MATRIKS <Tab : array of float [array of float [100] [100]], int NBrsEff, int nKolEff>

public MATRIKS ()

//I. S. Terdapat sebuah MATRIKS kosong berukuran 100x100

//F.S. MATRIKS dengan NBrsEff dan NKolEff bernilai 0

public void KeyboardSPL (integer m,n)

//I. S. Terdefinisi sebuah MATRIKS kosong dan ukuran MATRIKS m x (n+1)

// Masukan dari keyboard adalah koefisien aij , dan bi

//F.S. MATRIKS augmented yang siap dioperasikan

public void KeyboardDetBalikan (int n)

//I. S. Terdefinisi sebuah MATRIKS kosong dan ukuran MATRIKS n x n

// Masukan dari keyboard adalah koefisien aij

//F.S. MATRIKS augmented yang siap dioperasikan

public void KeyboardInterpolasi (int n)

//I. S. Terdefinisi sebuah MATRIKS kosong dan ukuran MATRIKS m x (n+1)

// Masukan dari keyboard adalah (x0, y0), (x1, y1), ..., (xn, yn)

//F.S. MATRIKS augmented yang siap dioperasikan

public void BacaFileMatriks (String namafile)

//I. S. Terdapat sebuah file.txt yang berisikan matriks augmented

//F.S. MATRIKS augmented yang siap dioperasikan

public void BacaFileTitikInterpolasi (String namafile)

//I. S. Terdapat sebuah file.txt yang berisikan titik-titik yang dinyatakan pada setiap baris tanpa //koma dan tanda kurung

//F.S. MATRIKS augmented yang siap dioperasikan

public float pangkat (float a, int b)

//Mengembalikan hasil dari a pangkat b

**operator.java**

class MATRIKS ini berisi *procedure* dan *function* untuk memroses MATRIKS sesuai operasi tertentu yang tersedia.

public void tukarBaris (MATRIKS M, int i, int j)

//I.S. Terdefinisi sebuah MATRIKS dan 2 buah integer yang merupakan baris yang ingin //ditukar

//F.S. MATRIKS mengalami pertukaran pada elemen baris i dengan elemen abris j

public void SwapByIndeksAwal (int i , int j , MATRIKS M)

//I.S. Terdefinisi sebuah MATRIKS dan 2 buah integer yang merupakan baris yang ingin //ditukar

//F.S. MATRIKS mengalami pertukaran baris berdasarkan indeks awal

public void TulisMatriks (MATRIKS M)

//I.S. Terdefinisi sebuah MATRIKS M

//F.S. Menampilkan ke layar seluruh elemen MATRIKS M

public int IndeksAwalBaris (int i, MATRIKS M)

//Mengembalikan nilai indeks paling awal bukan 0 pada baris tertentu

public boolean IsLastRowZero (int i, MATRIKS M)

//Mengembalikan true apabila semua elemen di baris terakhir bernilai 0 dan false jika ada yang tidak bernilai 0

public boolean IsNoSolution (MATRIKS M)

//Mengembalikan true apabila MATRIKS tidak memiliki solusi dan false apabila ada solusi

public boolean IsAllDiagonalOne (MATRIKS M)

//Mengembalikan true apabila semua elemen pada diagonal utama bernilai 1 dan false jika ada yang tidak bernilai 1

public MATRIKS KaliMatriks (MATRIKS matriks1, MATRIKS matriks2)

//Terdefinisi sebuah MATRIKS M1 berukuran m x n dan M2 berukuran n x p

//Mengembalikan MATRIKS M yang merupakan hasil kali matriks M1 dengan M2 dengan ukuran m x p

public MATRIKS Transpose (MATRIKS M)

//Mengembalikan MATRIKS M "di-transpose", yaitu setiap elemen M(i,j) ditukar nilainya dengan elemen M(j,i)

public MATRIKS Adjoint (MATRIKS M)

//Mengembalikan Adjoin dari MATRIKS , yaitu setiap transpose dari M yang elemen-elemennya merupakan kofaktor dari elemen-elemen M.

public MATRIKS MatriksInvers (MATRIKS M)

//Mengembalikan matriks invers dari MATRIKS , yaitu jika dikalikan dikalikan dengan M akan menghasilkan matriks identitas.

public MATRIKS RemoveZeroRow (MATRIKS M)

//Mengembalikan M yang sudah dihapus baris yang semua elemennya bernilai 0

public MATRIKS CopyMatriksDenganJumlahBaris (int i, MATRIKS matriks)

//Mengembalikan hasil duplikasi M dengan jumlah baris sebanyak i

public void MATRIKS SPLGauss (MATRIKS M)

//I.S. Terdefinisi sebuah MATRIKS dengan ukuran tertentu dan akan dilakukan operasi baris //elementer pada elemen-elemennya.

//F.S. MATRIKS.Tab sudah berbentuk matriks eselon

public void MATRIKS SPLGaussJordan (MATRIKS M)

//I.S. Terdefinisi sebuah MATRIKS dengan ukuran tertentu dan akan dilakukan operasi baris //elementer pada elemen-elemennya dan juga substitusi mundur

//F.S. MATRIKS.Tab sudah berbentuk matriks eselon baris

public void SPLMatriksInvers (MATRIKS M)

//I.S. Terdefinisi sebuah MATRIKS yang ingin dicari solusi SPL nya menggunakan metode //matriks invers

//F.S. Mengeluarkan solusi SPL yang didapat

public void Cramer (MATRIKS M)

//I.S. Terdefinisi sebuah MATRIKS yang ingin dicari solusi SPL nya menggunakan kaidah //cramer

//F.S. Mengeluarkan solusi SPL yang didapat

public void TulisSolusiBanyakSPL (MATRIKS M)

//I.S. M terdefinisi dan memiliki solusi banyak SPL

//F.S. Menampilkan ke layar solusi banyak dari SPL

public void MenulisSolusiSPL (MATRIKS M)

//I.S. M terdefinisi lalu dicek apakah memiliki penyelesaian atau tidak

//F.S. Menampilkan ke layar solusi SPL jika ada atau pesan jika tidak memiliki solusi

public void MenulisSolusiInterpolasi (MATRIKS M)

//I.S. M terdefinisi lalu dicek apakah memiliki penyelesaian atau tidak

//F.S. Menampilkan ke layar solusi SPL jika ada atau pesan jika tidak memiliki solusi

public float DeterminanKofaktor DeterminanKofaktor(MATRIKS M)

//Mengembalikan nilai determinan MATRIKS M yang dihitung determinan matriksnya dengan //cara ekspansi kofaktor

public float DeterminanReduksiBaris DeterminanKofaktor(MATRIKS M)

//Mengembalikan nilai determinan MATRIKS M yang dihitung determinan matriksnya dengan //cara reduksi baris

public float[] SubsitusiMundur (MATRIKS M)

//Mengembalikan sebuah array yang berisikan solusi dari MATRIKS M

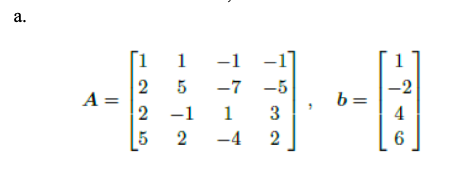
public float pangkat (float a, int b)

//Mengembalikan hasil dari a pangkat b

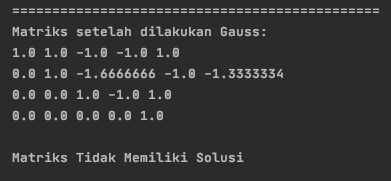
## **BAB IV**

## **EKSPERIMEN**

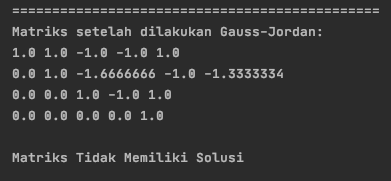
1. **SOLUSI SPL**



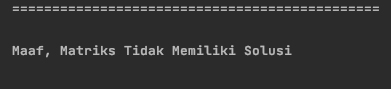
1.a. Eliminasi Gauss



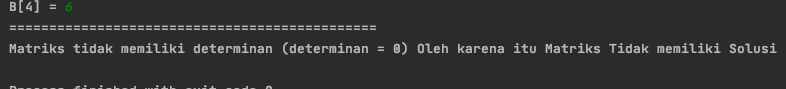
1.a. Eliminasi Gauss-Jordan

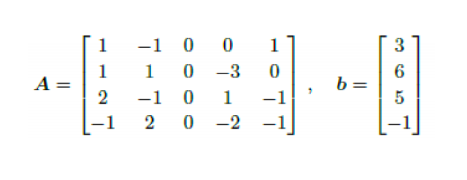


1.a. Metode Matriks Balikan

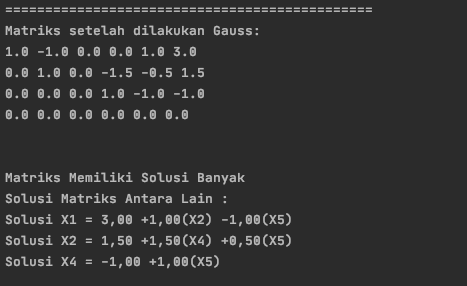


1.a. Kaidah Cramer

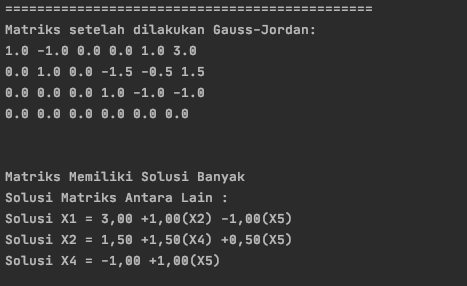




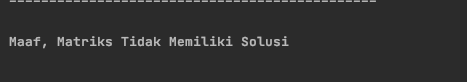
1.b. Metode Gauss



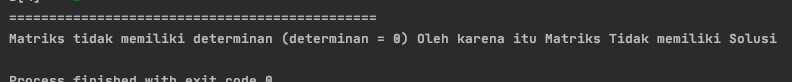
1.b. Metode Gauss-Jordann

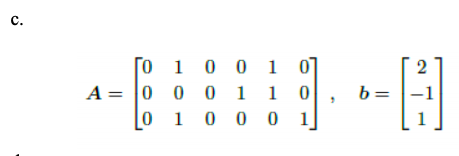


1.b. Metode Matriks Balikan

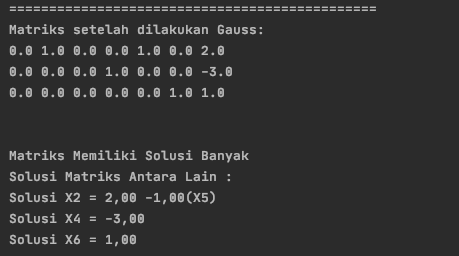


1.b. Kaidah Cramer

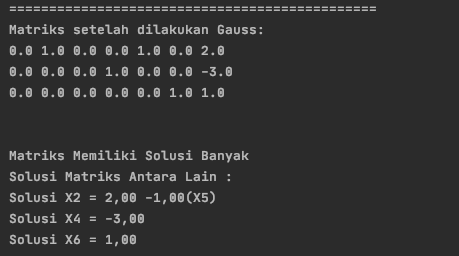




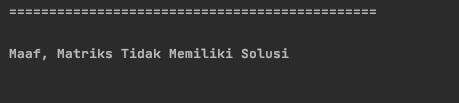
1.c. Metode Gauss



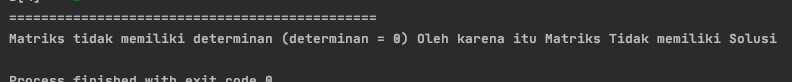
1.c. Metode Gauss-Jordann



1.c. Metode Matriks Balikan



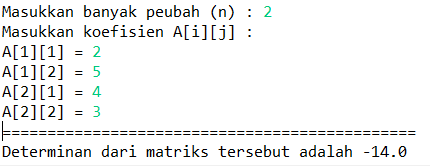
1.c. Kaidah Cramer



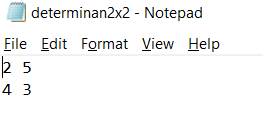
## **DETERMINAN**

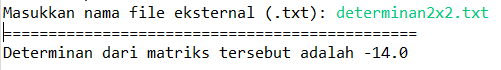
## Determinan matriks 2x2 :

* Metode reduksi baris masukan keyboard



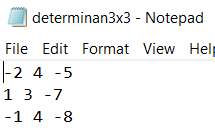
* Metode ekspansi kofaktor masukan dari file (determinan2x2.txt)

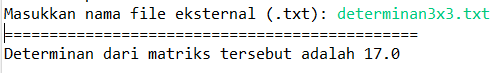




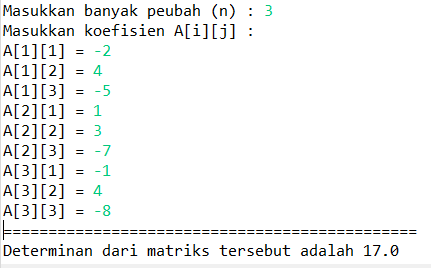
1. Determinan matriks 3x3 :

* Metode reduksi baris masukan dari file (determinan3x3.txt)



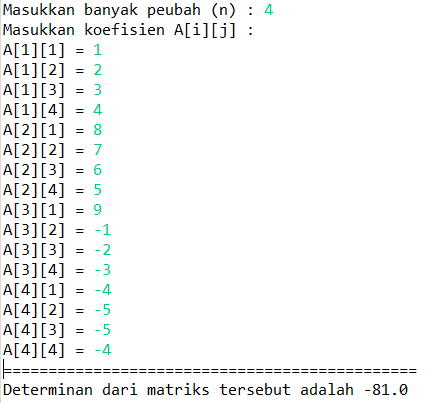


* Metode ekspansi kofaktor masukan keyboard

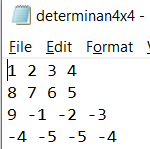


1. Determinan matriks 4x4 :

* Metode reduski baris masukan keyboard



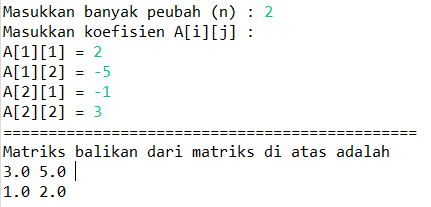
* Metode ekspansi kofaktor masukan dari file (determinan4x4.txt)



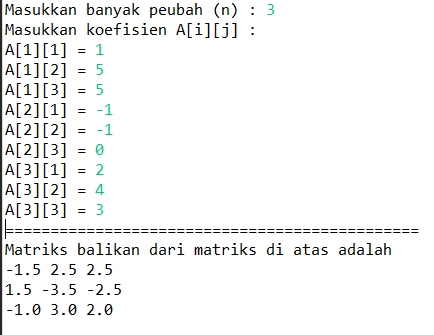


## **MATRIKS BALIKAN**

1. Matriks ukuran 2x2



1. Matriks ukuran 3x3



1. **INTERPOLASI POLINOM**
2. **REGRESI LINEAR BERGANDA**

(program tidak selesai)

## 

## **BAB V**

## **PENUTUP**

## 

## **5.1 Kesimpulan**

## Dari tugas besar berjudul “Sistem Persamaan Linear, Determinan, dan Aplikasinya”, maka program yang kami buat main.java, MATRIKS.java , dan operator.java dapat memenuhi beberapa case untuk program yang diselesaikan.

## **5.2 Saran**

## Berikut ini adalah saran-saran yang dapat kami berikan untuk tugas besar ini.

## 1. Dalam pembuatan program, sebaiknya membuat function dan procedure kecil yang lebih kecil sehingga dapat dimanfaatkan kapan saja pada kondisi lain. Selain dapat dimanfaatkan kapan saja, penggunaan function dan procedure ini dapat memperindah dan memperjelas struktur kode.

## 2. Setiap fungsi atau prosedur nya sebaiknya dibuat dalam masing-masing file java sehingga dapat memudahkan dalam pencarian fungsi dalam pengecekan ketersediaan fungsi.

## 

## **5.3 Refleksi**

1. Dalam pengerjaan tugas sebaiknya tidak dikerjakan mendekati deadline pengumpulan tugas.
2. Dalam pengerjaan tugas sebaiknya dilakukan secara mencicil sehingga tidak membebani aktivitas perkuliahaan lainnya.
3. Jika mengerjakan tugas dengan waktu sisa yang terbatas sebaiknya dapat belajar lebih cepat dan efisien

## 

## **REFERENSI**

## Blog.ruangguru.com. (2019, 7 Desember). Cara Mencari Determinan dan Invers Matriks. Diakses pada 2 Oktober 2020, dari [https://blog.ruangguru.com/cara-mencari-determinan- dan-invers-matriks](https://blog.ruangguru.com/cara-mencari-determinan-dan-invers-matriks)

## Id.bccrwp.org. (2020, 20 Februari). Perbedaan Antara Bentuk Eselon dan Bentuk Eselon Berkurang. Diakses pada 1 Oktober 2020, dari [https://id.bccrwp.org/compare/difference- between-echelon-form-and-reduced-echelon-form/](https://id.bccrwp.org/compare/difference-between-echelon-form-and-reduced-echelon-form/)

## Informatika.stei.itb.ac.id. (2020, 31 Agustus). Matriks Eselon. Diakses pada 2 Oktober 2020, dari http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-02-Matriks-Eselon.pdf

## Informatika.stei.itb.ac.id. (2020, 2 September). Sistem Persamaan Linier 2. Diakses pada 2 Oktober 2020, dari [http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/ Algeo-05-Sistem-Persamaan-Linier-2.pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdfhttp://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf)

## Informatika.stei.itb.ac.id. (2010). Interpolasi Polinom (Bagian 1). Diakses pada 2 Oktober 2020, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/MetNum/2010-2011/Interpolasi%20Polinom. pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/MetNum/2010-2011/Interpolasi%20Polinom.pdf)

## Mathcyber1997.com. (2020, 11 Mei). Materi, Soal, dan Pembahasan - Aturan Cramer. Diakses pada 2 Oktober 2020, dari [https://mathcyber1997.com/materi-soal-dan-pembahasan-aturan- cramer/#:~:text=Dalam%20bidang%20aljabar%20linear%2C%20Aturan,masing%20persamaan%20di%20sistem%20tersebut](https://mathcyber1997.com/materi-soal-dan-pembahasan-aturan-cramer/#:~:text=Dalam%20bidang%20aljabar%20linear%2C%20Aturan,masing%20persamaan%20di%20sistem%20tersebut).

## Statistikian.com. (2012, 10 Agustus). Regresi Linear Sederhana dengan SPSS. Diakses pada 2 Oktober 2020, dari [https://www.statistikian.com/2012/08/regresi-linear-sederhana-dengan-spss. html](https://www.statistikian.com/2012/08/regresi-linear-sederhana-dengan-spss.html)

## 