15IF2123 ALJABAR LINEAR DAN GEOMETRI

IMPLEMENTASI SISTEM LINEAR, REGRESI, DAN INTERPOLASI DENGAN BAHASA PEMROGRAMAN JAVA

**Laporan Tugas Besar 1**

Disusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri pada Semester 1 (satu)

Tahun Akademik 2024/2025

**Oleh:**

**Nathanael Rachmat 13523142**

**Andrew Tedjapratama 13523148**

**Theo Kurniady 13523154**

**Kelompok Lastesest**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2024**

# DAFTAR ISI

[**DAFTAR ISI 2**](#_xwefbe6t0b1e)

[**BAB I 4**](#_x9finrbsl6h9)

[1.1 Latar Belakang Masalah 4](#_pgvknwi3v5y5)

[1.2 Spesifikasi Program 4](#_dpibjdcoyyed)

[1.2.1 Bahasa Pemrograman dan Lingkup Pengerjaan 4](#_ycvhm1r1hszj)

[1.2.2 Tujuan Pengembangan Program 5](#_5nwwnotlm42b)

[1.2.3 Input and Output 5](#_sd4nhcuc638z)

[**BAB II 6**](#_1h9b7pir0r6v)

[2.1 Sistem Persamaan Linear 6](#_4uhfvjmelbps)

[2.2 Determinan 7](#_5t39yzwlq683)

[2.3 Matriks Invers 8](#_83gb0awgss1y)

[2.3.1 Cara Menghitung Invers Matriks 9](#_ep78zj1wuop9)

[2.3.2 Aplikasi Matriks Invers 9](#_fuoc9x4nloms)

[2.4 Interpolasi Polinom 9](#_oytfotcgk7ai)

[2.5 Interpolasi Bicubic Spline 11](#_8g2d86ol03j3)

[2.6 Regresi Berganda 12](#_e41n18p2r07k)

[**BAB III 14**](#_as9gj9lqi81x)

[**3.1 Class Main 14**](#_z4ovxrebxhp4)

[3.1.1. Atribut 14](#_f9oym2sc0cwl)

[3.1.2. Metode 14](#_ferlv88awp3t)

[3.2 Class OBE 15](#_p9uup3i28ijm)

[3.2.1. Atribut 15](#_9orjsovb27tl)

[3.2.2. Metode 16](#_zcnheb53gwsn)

[3.3 Class Eselon 18](#_rbk4aqpw0pu0)

[3.3.1. Atribut 18](#_6y1n7pkogq64)

[3.3.2. Metode 18](#_l0b3y13ex471)

[3.4 Class SPL 20](#_g5zu7h2s7cew)

[3.4.1. Atribut 20](#_hntl7ats94x2)

[3.4.2. Metode 21](#_eny24pghlu1e)

[3.5 Class Determinan 22](#_v3bigw857c1x)

[3.5.1. Atribut 22](#_9nmihimeycbp)

[3.5.2. Metode 22](#_gelvyzk3z69r)

[3.6 Class Kofaktor 22](#_fwddnayfyd3b)

[3.6.1. Atribut 22](#_ja1xmxh5ksdd)

[3.6.2. Metode 23](#_p7d68d92163r)

[3.7 Class Inverse 23](#_vja016mtyiw)

[3.7.1. Atribut 23](#_v80uod64786x)

[3.7.2. Metode 24](#_g3mhnxduf3b8)

[3.8 Class InterpolasiPolinomial 25](#_joswzdm6hgsn)

[3.8.1. Atribut 25](#_moubskhwa91m)

[3.8.2. Metode 26](#_6p0g47ro6idf)

[3.9 Class RegresiBerganda 26](#_y6g1pxm0u3ak)

[3.9.1. Atribut 26](#_b96d30jpl705)

[3.9.2. Metode 27](#_hi659g8q7gos)

[3.10 Class Bicubic Spline 28](#_ot7hom1ycg3k)

[3.10.1. Atribut 28](#_u06zttvqca7v)

[3.10.2. Metode 29](#_5cporjebwjse)

[3.11 InputFile 30](#_wcqrloghf6ef)

[3.11.1. Atribut 30](#_bplx9yaxy0qa)

[3.11.2. Metode 30](#_9f3vhgmepywi)

[3.12 WriteFile 31](#_g86orcew0eo)

[3.12.1. Atribut 31](#_av7dcs36u8sq)

[3.12.2. Metode 31](#_orrk444orc6a)

[**BAB IV 32**](#_vfll34ekvn5n)

[**BAB V 46**](#_foe6uodlzl1v)

[**LAMPIRAN 47**](#_kenb0rm7ewp)

# BAB I

DESKRIPSI MASALAH

## 1.1 Latar Belakang Masalah

Dalam tugas besar mata kuliah Aljabar Linier dan Geometri, penulis merancang sebuah aplikasi aljabar linear menggunakan bahasa pemrograman Java. Aplikasi ini dirancang untuk menangani berbagai fungsi, seperti menyelesaikan sistem persamaan linear (SPL), determinan matriks, matriks invers, interpolasi polinomial, interpolasi bicubic spline, dan regresi linear. Untuk meningkatkan efisiensi dan akurasi dalam menyelesaikan permasalahan aljabar linear, penulis merancang sebuah alat bantu yaitu program komputer berbasis Java yang mampu mengimplementasikan berbagai metode penyelesaian SPL dan aplikasi terkait lainnya. Program ini diharapkan dapat mempermudah serta memberikan solusi numerik yang tepat dalam waktu yang singkat.

## **1.2 Spesifikasi Program**

Program ini dirancang dan dikembangkan dengan antarmuka berbasis teks, dimana pengguna dapat memasukkan data secara interaktif melalui teks. Program ini memiliki fungsi-fungsi yang diimplementasikan menggunakan algoritma yang efisien, dirancang untuk menangani berbagai ukuran matriks dan tingkat kompleksitas sistem persamaan. Dengan antarmuka teks ini, program tetap mampu memberikan solusi numerik yang akurat dan dapat membantu analisis data dengan pendekatan matematika yang beragam.

### 1.2.1 Bahasa Pemrograman dan Lingkup Pengerjaan

Bahasa pemrograman Java dipilih untuk pengembangan tugas besar ini karena kehandalannya dalam memproses data yang kompleks serta kemampuannya mendukung pengorganisasian kode pemrograman berorientasi objek yang modular. Java memiliki keunggulan dalam hal portabilitas, yang memungkinkan aplikasi berjalan di berbagai sistem operasi tanpa memerlukan perubahan besar pada kode.

Proyek ini akan fokus pada pengembangan program berbasis terminal tanpa antarmuka grafis (GUI) yang mencakup penyelesaian sistem persamaan linear, perhitungan determinan dan matriks invers, interpolasi polinomial, interpolasi bicubic spline, serta regresi linear dan kuadratik berganda.

### 1.2.2 Tujuan Pengembangan Program

Tujuan dari pengembangan program ini antara lain:

* Mengimplementasikan berbagai metode penyelesaian sistem persamaan linear (SPL) secara efisien dan akurat.
* Menyediakan fitur perhitungan determinan dan invers matriks
* Mengaplikasikan teknik interpolasi polinomial dan bicubic spline untuk analisis data dan menghasilkan persamaan polinomial untuk menciptakan permukaan yang halus pada gambar serta mengimplementasikan fitur *image stretching* dan *image resizing*.
* Mengembangkan fungsi regresi linear dan kuadratik berganda untuk prediksi dan analisis data.

### 1.2.3 Input and Output

Program dapat menerima input atau masukan dari pengguna melalui terminal dan *file upload*. Masukan untuk sistem persamaan linear (SPL) menggunakan format matriks augmented. Output dari program ditampilkan melalui terminal maupun lewat file teks (.txt) yang memungkinkan pengguna untuk mengunduh file tersebut.

### 

# BAB II

TEORI SINGKAT

## **2.1 Sistem Persamaan Linear**

Sistem Persamaan Linear (SPL) adalah sekumpulan persamaan linear yang memiliki variabel yang sama dan solusinya ditentukan dengan menyelesaikan persamaan-persamaan tersebut secara bersamaan.. SPL banyak digunakan dalam berbagai bidang seperti sains, teknik, ekonomi, dan lainnya untuk memodelkan serta memecahkan berbagai masalah di dunia nyata. SPL dapat memiliki tiga kemungkinan solusi:

1. Mempunyai banyak solusi (tidak berhingga banyaknya)
2. Tidak memiliki solusi sama sekali, jika persamaan tidak konsisten atau kontradiktif.
3. Mempunyai solusi yang unik (tunggal), jika persamaan memiliki solusi tunggal yang memenuhi semua persamaan.

**2.1.1 Metode Penyelesaian SPL**

Beberapa metode yang sering digunakan untuk menyelesaikan SPL adalah:

1. Metode Eliminasi Gauss

Metode eliminasi Gauss adalah metode dasar untuk menyelesaikan SPL. Metode ini bekerja dengan mengubah matriks koefisien persamaan linear menjadi bentuk eselon baris. Bentuk ini memudahkan penyelesaian sistem melalui substitusi mundur untuk mendapatkan nilai dari variabel yang dicari.

1. Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan merupakan perbaikan dari metode eliminasi Gauss. Metode ini mengubah matriks koefisien menjadi bentuk eselon baris tereduksi, di mana solusi dapat diperoleh langsung tanpa perlu substitusi mundur. Bentuk ini memudahkan penghitungan secara langsung untuk mendapatkan solusi unik.

1. Metode Matriks Balikan

Metode ini digunakan untuk menyelesaikan SPL dalam bentuk matriks, yaitu , dengan A sebagai matriks koefisien, sebagai vektor solusi, dan b sebagai vektor konstanta. Jika matriks A memiliki balikan, maka solusi dapat diperoleh dengan rumus:

Metode ini membutuhkan perhitungan invers matriks, yang hanya dapat dilakukan jika matriks bersifat invertible.

1. Kaidah Cramer digunakan untuk menyelesaikan SPL dengan n variabel dan n persamaan. Kaidah ini menggunakan determinan untuk menghitung solusi masing-masing variabel. Untuk SPL , solusi untuk variabel , dapat dihitung dengan rumus:

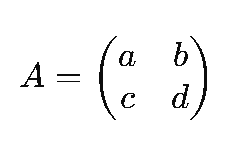
di mana adalah matriks yang telah digantikan kolom ke- dengan vektor , dan adalah determinan dari matriks . Metode ini hanya berlaku jika determinan matriks tidak sama dengan nol.

**2.1.2 Aplikasi SPL**Metode-metode dari Sistem Persamaan Linier (SPL) bisa diterapkan dalam masalah-masalah lain seperti interpolasi dan regresi. Dalam interpolasi, SPL digunakan untuk menemukan polinomial yang melalui sekumpulan titik data. Sedangkan dalam regresi, SPL digunakan untuk memodelkan hubungan antara variabel-variabel independen dan dependen.

## **2.2 Determinan**

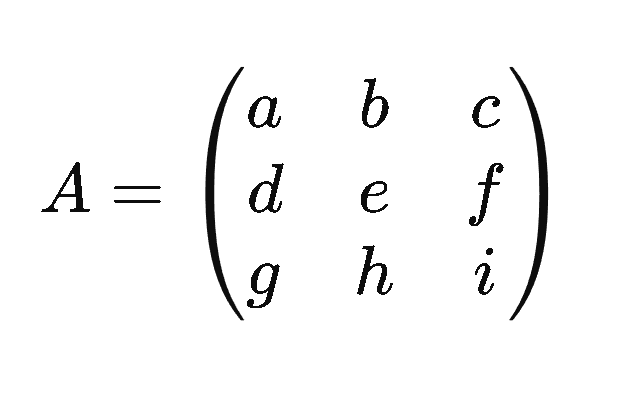
Determinan dari matriks adalah sebuah nilai skalar yang bisa dihitung dari matriks persegi. Determinan mempunyai banyak aplikasi dalam aljabar linear, contohnya seperti untuk mengetahui apakah matriks tersebut memiliki invers atau tidak. Jika determinan dari sebuah matriks adalah nol, maka matriks tersebut dianggap *singular*, yang artinya matriks tidak memiliki invers dan sistem persamaan yang terkait mungkin tidak memiliki solusi unik.. Cara menghitung determinan bergantung pada ukuran matriks, misalnya untuk matriks 2x2, kita dapat menghitungnya dengan rumus:

di mana adalah matriks 2x2 dengan elemen-elemen sebagai berikut:

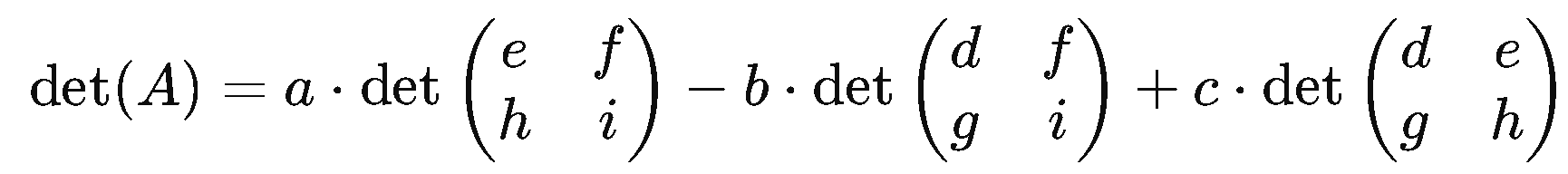


Namun, untuk matriks yang lebih besar seperti matriks 3x3 atau lebih, determinan dapat dicari dengan menggunakan ekspansi kofaktor yang memecah determinan menjadi sub-determinan dari matriks yang lebih kecil.

Misalkan pada matriks (3x3)



Determinannya adalah

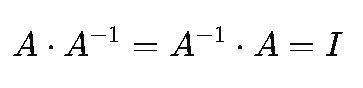


Setiap minor adalah matriks 2x2 yang dihasilkan dari menghapus baris dan kolom yang terkait dengan elemen tersebut. Jadi, ekspansi kofaktor memecah perhitungan determinan menjadi determinan dari matriks yang lebih kecil dan digunakan untuk ukuran matriks lebih besar dari 2x2.

## **2.3 Matriks *Invers***

Matriks *Invers* adalah kebalikan dari sebuah matriks yang apabila matriks tersebut dikalikan dengan inversnya, akan menjadi matriks identitas. Suatu matriks dikatakan memiliki invers jika determinan dari matriks tersebut tidak sama dengan nol. Matriks invers hanya ada untuk matriks persegi (jumlah baris sama dengan jumlah kolom) yang tidak singular (determinan tidak nol).

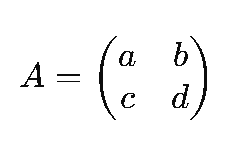
Misalkan adalah matriks, maka matriks inverse dari dilambangkan dengan memiliki sifat berikut:



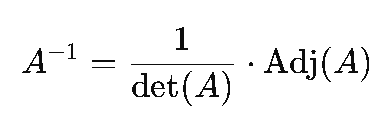
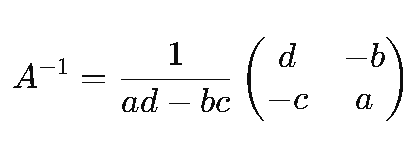
di mana adalah matriks identitas yang merupakan matriks persegi dengan elemen diagonal utamanya 1 dan elemen lainnya 0.

## 2.3.1 Cara Menghitung Invers Matriks

Untuk matriks berukuran kecil seperti matriks 2x2, inversnya dapat dihitung dengan menggunakan rumus sederhana. Misalkan kita memiliki matriks sebagai berikut:



Invers dari matriks atau dapat dihitung menggunakan rumus:

di mana adalah determinan dari matriks . Jika maka matriks tidak memiliki invers.

### 2.3.2 Aplikasi Matriks Invers

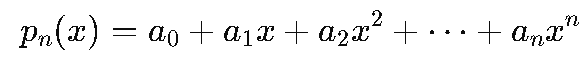
Matriks invers sering digunakan dalam aljabar linear untuk menyelesaikan sistem persamaan linear, di mana solusi dari SPL dapat ditemukan dengan:

Matriks invers membantu menyelesaikan SPL secara langsung, asalkan inverse dari matriks bukan nol atau dapat dihitung.

## **2.4 Interpolasi Polinom**

Interpolasi polinomial adalah metode untuk memperkirakan nilai suatu fungsi pada titik-titik yang tidak diketahui, berdasarkan data yang telah diketahui pada beberapa titik tertentu. Dalam konteks ini, kita diberikan sejumlah titik data ,,. Setelah itu, diminta untuk menentukan polinomial yang melewati segala titik tersebut sehingga untuk setiap

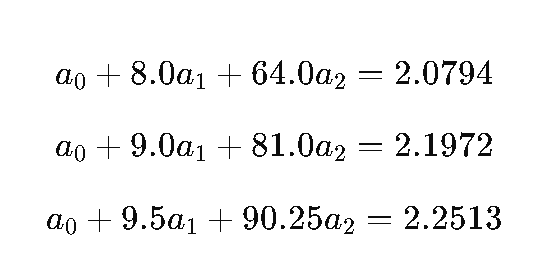
Setelah polinomial ditemukan, kita bisa menggunakannya untuk menghitung perkiraan nilai pada sembarang titik dalam selang . Polinomial interpolasi yang kita buat memiliki derajat , di mana derajat ini bergantung pada jumlah titik yang tersedia. Polinomial ini berbentuk:



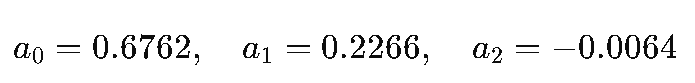
**Contoh Interpolasi Polinomial:**

Misalkan diberikan tiga masukan titik data: (8.0, 2,0794), (9.0, 2.1972), dan (9.5, 2.2513). Melalui tiga titik tersebut, diminta untuk membuat polinomial kuadratik yang menginterpolasi ketiga titik tersebut, kemudian mengestimasi nilai fungsi pada

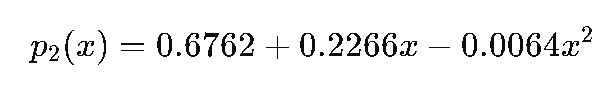
Untuk menemukan koefisien kita mensubtitusi titik-titik data yang diberikan ke deret polinomial:



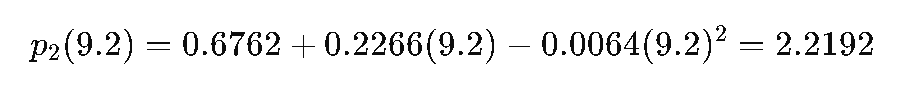
Dari sistem persamaan linear ini, dapat digunakan metode eliminasi Gauss untuk mencari solusi dari koefisien-koefisien . Hasilnya adalah:



Jadi, fungsi polinomial kuadratik yang menginterpolasi ketiga titik tersebut didapatkan berdasarkan koefisien-koefisien yang didapat yaitu:



Dari fungsi polinomial tersebut, dapat ditinjau atau diperkirakan nilai fungsi pada sebagai berikut:

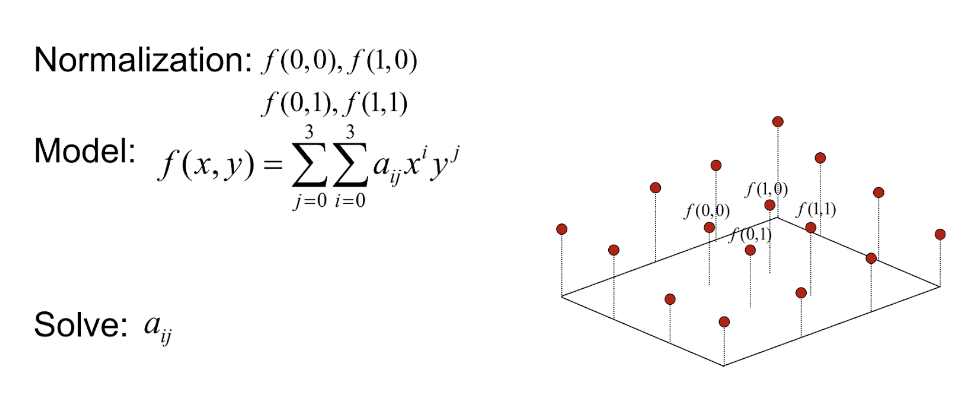


Jadi, nilai taksiran fungsi pada adalah .

## **2.5 Interpolasi *Bicubic Spline***

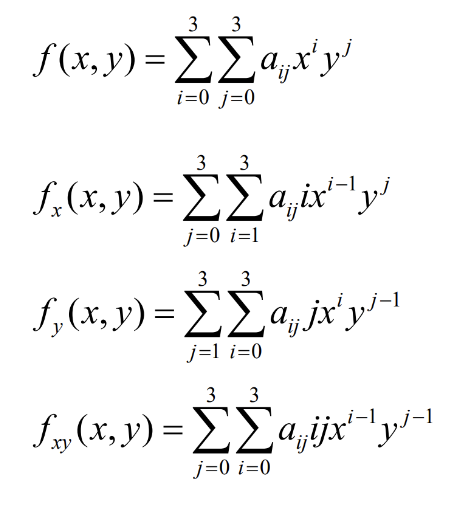
Interpolasi Bicubic Spline adalah metode untuk memperkirakan nilai fungsi di antara titik-titik data dua dimensi yang diketahui. Metode ini menggunakan polinomial kubik untuk membangun permukaan yang halus dan kontinu dalam dua arah (horizontal dan vertikal).

Metode ini melibatkan 16 titik data, yang terdiri dari 4 titik referensi utama di pusat dan 12 titik di sekitarnya. Dengan menggunakan nilai fungsi dan turunan berarah dari titik-titik ini, interpolasi bicubic menghasilkan permukaan yang lebih akurat dibandingkan dengan metode interpolasi lainnya, seperti interpolasi bilinear.



**Gambar xx** Pemodelan interpolasi *bicubic spline*

Selain melibatkan model dasar, juga digunakan model turunan berarah dari kedua sumbu, baik terhadap sumbu x, sumbu y, maupun keduanya. Persamaan polinomial yang digunakan adalah sebagai berikut.



Interpolasi Bicubic Spline banyak digunakan dalam pengolahan citra, grafis komputer, dan bidang lain yang membutuhkan representasi permukaan yang halus dan presisi tinggi. Metode ini memiliki banyak kelebihan seperti memberikan hasil interpolasi yang lebih halus dan akurat.

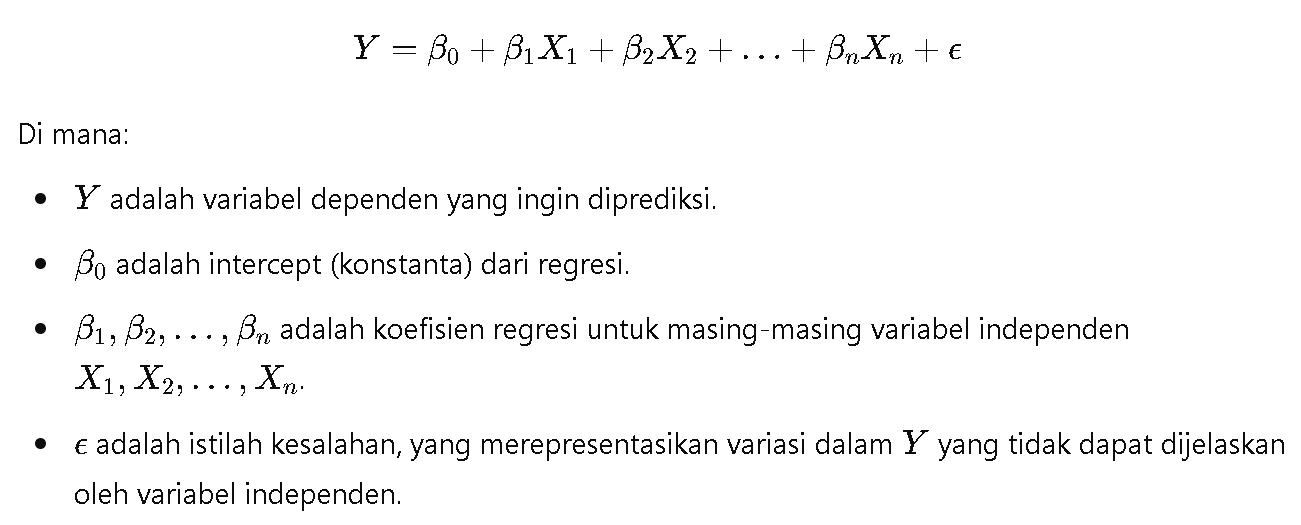
## **2.6 Regresi** Berganda

Regresi Linear Berganda adalah metode statistika yang digunakan untuk memprediksi nilai variabel dependen berdasarkan dua atau lebih variabel independen. Metode ini merupakan perluasan dari regresi linear sederhana, yang hanya melibatkan satu variabel independen. Regresi linear berganda digunakan untuk menggambarkan hubungan antara variabel dalam situasi di mana banyak faktor dapat mempengaruhi hasil.

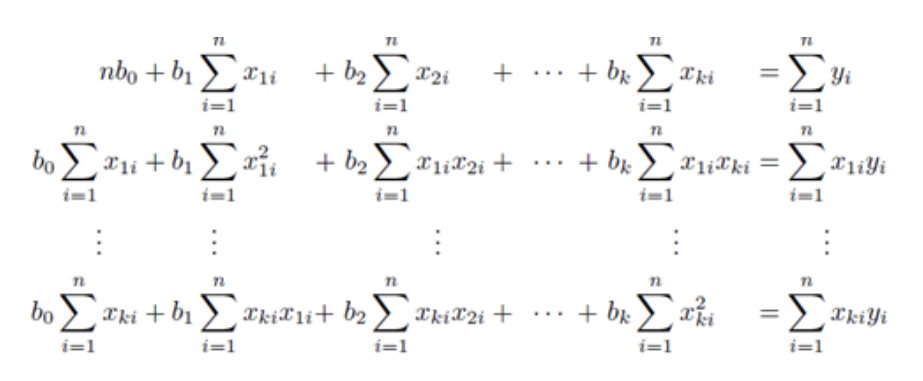
**2.6.1 Regresi Linear Berganda**

Regresi linier berganda merupakan alat yang sangat berguna dalam analisis data, memungkinkan peneliti untuk memahami dan memprediksi hubungan kompleks antara beberapa variabel. Dengan pendekatan yang tepat, regresi ini dapat digunakan dalam berbagai bidang, termasuk ekonomi, ilmu sosial, dan ilmu alam, untuk memberikan wawasan yang lebih dalam tentang fenomena yang diteliti.

Walaupun telah tersedia rumus untuk menghitung regresi linear sederhana, ada juga persamaan umum untuk regresi linear yang dapat diterapkan pada regresi linear berganda.



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

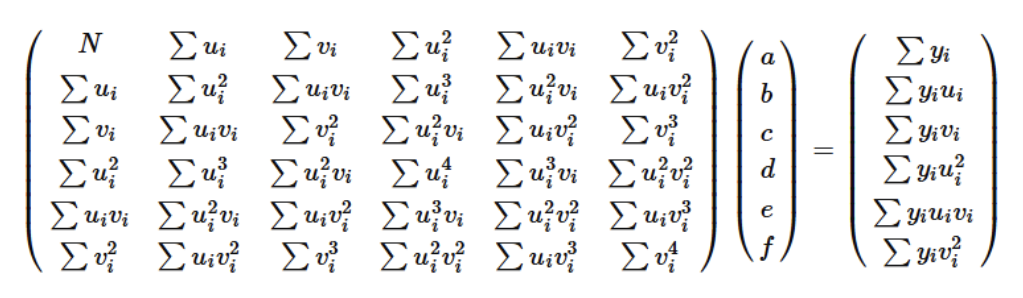


**2.6.2 Regresi Kuadratik Berganda**

Dalam regresi kuadratik berganda, terdapat 3 bentuk persamaan dari regresi kuadratik, yaitu:

1. Variabel Linear: Variabel dengan derajat satu seperti X, Y, Z.
2. Variabel Kuadrat: Variabel dengan derajat dua seperti
3. Variabel Interaksi 2 Variabel dengan derajat satu yang dikalikan dengan satu sama lain seperti XY, YZ, dan XZ.

Kedua model regresi yang dijadikan sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss. Setiap n-peubah, jumlah variabel linier, kuadrat, dan interaksi akan berbeda-beda. Berikut adalah contoh regresi kuadratik 2 variabel peubah.



N menandakan jumlah peubah, terdapat 2 variabel linear yaitu dan , 2 variabel kuadrat dan , dan 1 variabel interaksi yaitu . Untuk setiap n-peubah, akan terdapat 1 konstan N. n variabel linear, n variabel kuadrat, dan variabel linear (dengan syarat n > 1). Tentu dengan bertambahnya peubah n, ukuran matriks akan bertumbuh lebih besar dibandingkan regresi linier berganda tetapi solusi tetap bisa didapat dengan menggunakan SPL.

# BAB III

IMPLEMENTASI PROGRAM

## 3.1 Class Main

### 3.1.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| - | - |

### 3.1.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| displayMenu() | Menampilkan menu utama dari program |
| main(String[] args) | Metode utama yang menangani alur program berdasarkan input pengguna. |

## 3.2 Class OBE

### 3.2.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| row1List | Array tipe double, tempat penyimpanan row1 |
| row2List | Array tipe double, tempat penyimpanan row2 |
| mOut | Matriks tipe double, menyimpan parameter matriks dan mengembalikkan sebagai output |
| m | Matriks tipe double, matriks biasa |
| row1 | Array tipe double, tempat penyimpanan baris yang diambil dari matriks dan akan digunakan pada fungsi yang membutuhkan 2 baris. |
| row2 | Array tipe double, tempat penyimpanan baris yang diambil dari matriks dan akan digunakan pada fungsi yang membutuhkan 2 baris. |
| row | Array tipe double, baris biasa yang diambil dari matriks |
| c | Variabel tipe double, variabel pengali / pertambahan |
| n | Matriks tipe double, matriks kedua yang digunakan sebagai parameter yang membutuhkan 2 matriks |
| jumlah | Variabel tipe double, jumlah dari perhitungan |
| list | Array tipe double, array biasa |
| col | Array tipe double, kolom biasa yang diambil dari matriks |
| rowEff | Variabel tipe integer, berupa banyaknya baris yang efektif |
| colEff | Variabel tipe integer, berupa banyaknya kolom yang efektif |
| m2 | Matriks tipe double, matriks baru yang dibuat dalam fungsi |
| currentRow | Array tipe double, baris yang sedang direferensi |
| currentCol | Array tipe double, kolom yang sedang direferensi |
| i | Variabel tipe integer, untuk for loop |
| j | Variabel tipe integer, untuk for loop |
| k | Variabel tipe integer, untuk for loop |

### 3.2.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| switchRow(double[][] m, int row1, int row2) | Menukar dua baris pada sebuah matriks. Menerima input matriks m dan dua indeks baris (row 1 dan row2), lalu mengembalikan matriks hasilnya. |
| multiplyRow(double[][], int row, double c) | Mengalikan semua elemen pada baris row di matriks m dengan konstanta c, lalu mengembalikan matriks hasilnya. |
| multiplyBetweenMatrix(double[][] m, double[][] n) | Mengalikan dua matriks m dan n (perkalian matriks standar), dan mengembalikan matriks hasilnya. |
| multiplyMatrix(double[][] m, double c) | Mengalikan setiap elemen di seluruh matriks m dengan konstanta c, dan mengembalikan matriks hasilnya. |
| addRow(double[][] m, int row1, int row2, double c) | Menambahkan hasil perkalian baris row2 dengan konstanta c ke baris row1 dalam matriks m. Mengembalikan matriks hasilnya. |
| getRow(double[][] m, int row) | Mengambil baris ke-row dari matriks m dan mengembalikannya sebagai array. |
| getCol(double[][] m, int col) | Mengambil kolom ke-col dari matriks m dan mengembalikannya sebagai array. |
| getRowEff(double[][] m) | Mengembalikan jumlah baris efektif dalam matriks m.. |
| getColEff(getColEff(double[][] m) | Mengembalikan jumlah kolom efektif dalam matriks m. |
| deleteRow(double[][] m, int row) | Menghapus baris ke-row dari matriks m, dan mengembalikan matriks yang telah dihapus barisnya. |
| deleteCol(double[][] m, int col) | Menghapus kolom ke-col dari matriks m, dan mengembalikan matriks yang telah dihapus kolomnya. |
| insertCol(double[][] m, int idx, double[][] inputM) | Memasukkan kolom dari matriks inputM ke indeks idx pada matriks m, mengembalikan matriks hasilnya. |
| printMatrix(double[][] m) | Menampilkan matriks m ke terminal/console |
| printList(double[] list) | Mencetak array list ke terminal/console dengan format dua angka di belakang koma. |
| readMatrix |  |

## 3.3 Class Eselon

### 3.3.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| m | Matriks tipe double, matriks biasa |
| mOut | Matriks tipe double, menyimpan parameter matriks dan mengembalikkan sebagai output |
| colIdx | Variabel tipe integer, index acuan untuk mengurangi baris menjadi 0 pada OBE. |
| idx | Variabel tipe integer, index acuan untuk OBE |
| temp | Variabel tipe double, nilai yang digunakan dalam perkalian untuk pengurangan elemen menjadi 0 |
| c | Variabel tipe double, nilai yang digunakan untuk pengurangan pada colElim |
| x | Variabel tipe double, nilai acuan dalam membentuk satu utama |
| ctr | Variabel tipe integer, nilai untuk pengecekan perhitungan baris |
| swapped | Variabel tipe boolean, untuk memeriksa bila baris sudah ditukar |
| i | Variabel tipe integer, untuk for loop |
| j | Variabel tipe integer, untuk for loop |

### 3.3.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| colElim(double[][] m, int idx) | Mengeliminasi kolom pada indeks idx sehingga semua elemen di bawah elemen utama pada kolom tersebut menjadi nol, menggunakan OBE. |
| colElim2(double[][] m, int idx) | Digunakan dalam proses eselon tereduksi, dimana elemen di atas elemen utama pada kolom diubah menjadi nol dengan operasi baris elementer. |
| rowNorm(double[][] m, int idx) | Membuat elemen pertama yang bukan nol di setiap baris menjadi 1. |
| MatriksEselon(double[][] m) | Menghasilkan bentuk eselon dari matriks dengan menormalkan setiap baris matriks menggunakan metode rowNorm(). |
| MatriksEselonTereduksi(double[][] m) | Menghasilkan bentuk eselon tereduksi dari matriks dengan menggunakan colElim2() untuk mengeliminasi elemen di atas elemen utama. |
| ReduksiBaris(double[][] m) | Mengkombinasikan operasi pengurutan baris dan eliminasi kolom untuk mengubah matriks menjadi bentuk eselon. |
| CountSwapRed(double[][] m) | Menghitung jumlah berapa kali matriks mengalami operasi pertukaran baris selama proses reduksi baris. |
| countZero(double[][] m, int idx) | Menghitung jumlah angka 0 (*leading zero*) berturut-turut dari kiri pada baris ke-idx dalam matriks |
| countSwap(double[][] m) | Menghitung berapa kali pertukaran baris yang diperlukan untuk mengurutkan matriks berdasarkan jumlah *leading zero.* |
| SortMatriks(double[][] m) | Mengurutkan baris-baris matriks berdasarkan jumlah *leading zero.* |

## 3.4 Class SPL

### 3.4.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| m | Matriks tipe double, matriks biasa |
| mOut | Matriks tipe double, menyimpan parameter matriks dan mengembalikkan sebagai output |
| x | Matriks tipe double, matriks hasil perkalian matriks |
| inverseA | Matriks tipe double, matriks hasil inverse |
| A | Matriks tipe double, matriks yang akan di invers |
| b | Matriks tipe double, matriks yang akan dikalikan dengan matriks invers |
| rowCount | Variabel tipe integer, panjang baris |
| colCount | Variabel tipe integer, panjang kolom |
| allZero | Variabel tipe boolean, variabel yang memeriksa bila semua elemen 0 |
| stringTemp | Variabel tipe string, string berupa Xi untuk ditampilkan |
| satuUtama | Variabel tipe boolean, variabel untuk memeriksa bila ada satu utama pada kolom |
| result | Array tipe double, matriks yang menyimpan parameter matriks dan mengembalikkan sebagai output |
| row | Variabel tipe integer, panjang baris |
| col | Variabel tipe integer, panjang kolom |
| det | Variabel tipe double, determinan matriks utama |
| copy | Matriks tipe double, copy dari matriks utama |
| tempDet | Variabel tipe double, determinan matriks kramer |
| i | Variabel tipe integer, untuk for loop |
| j | Variabel tipe integer, untuk for loop |
| k | Variabel tipe integer, untuk for loop |

### 3.4.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| elimGaussJordan(double[][] m) | Menggunakan metode eliminasi Gauss Jordan pada matriks m, mengembalikan matriks dalam bentuk eselon baris tereduksi. |
| inverseSPL(double[][] A, double[][] b) | Menggunakan metode matriks balikan untuk menyelesaikan sistem persamaan linear menggunakan invers dari matriks. Matriks A adalah koefisien, sedangkan b adalah konstanta. |
| isSolution(double[][] m) | Mengecek apakah sistem persamaan linier memiliki solusi. |
| isInfiniteSolution | Mengecek apakah sistem persamaan linier memiliki solusi tak terhingga. |
| printParametric(double[][] m) | Menampilkan solusi parametrik dari matriks m |
| listSolution(double[][] m) | Mengambil solusi unik dari matriks m ke dalam array |
| copyMatrix(double[][] m) | Menerima matriks m dan mengembalikkan copy dari matrix m |
| Cramer(double[][] m, double[] n) | Menerima matriks m dan array n dan menggunakan kaidah cramer untuk mendapatkan solusi SPL yang dikembalikkan dalam bentuk array. |

## 3.5 Class Determinan

### 3.5.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| m | Matriks tipe double, Matriks biasa |
| swap | Variabel tipe integer, Banyaknya operasi pertukaran baris dalam OBE |
| result | Variabel tipe double, nilai determinan dengan matriks segitiga atas |
| ctr | Variabel tipe integer, nilai perhitungan dalam determinan kofaktor |
| i | Variabel tipe integer, untuk for loop |

### 3.5.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| minorMatriks(double[][] m) | Menerima matriks m dan mengembalikkan determinan dari m yang 2x2 |
| DeterminanReduksiBaris(double[][] m) | Menerima matriks m dan mengembalikkan determinannya dengan proses mereduksi baris menjadi matriks segitiga atas. |
| DeterminanKofaktor(double[][] m) | Menerima matriks m dan mengembalikkan determinannya dengan proses ekspansi kofaktor |

## 3.6 Class Kofaktor

### 3.6.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| m | Matriks tipe double, matriks biasa |
| i | Variabel tipe integer, untuk for loop dan sebagai index dalam kofaktor |
| j | Variabel tipe integer, untuk for loop dan sebagai index dalam kofaktor |
| m2 | Matriks tipe double, parameter matriks kedua |

### 3.6.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| isMatriks2x2(double[][] m) | Menerima matriks m dan mengembalikan true jika matriks berukuran 2x2 |
| kofaktor(double[][] m, int i, int j) | Menghitung nilai kofaktor dari suatu elemen di baris ke-i dan kolom ke-j dari matriks input m |
| Matrikskofaktor(double[][] m) | menghitung matriks kofaktor dari suatu matriks input m |

## 3.7 Class Inverse

### 3.7.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| m | Matriks tipe double, matriks biasa |
| n | Variabel tipe integer, baris dan kolom dari matriks nxn |
| mOut | Matriks tipe double, menyimpan parameter matriks dan mengembalikkan sebagai output |
| t | Matriks tipe double, matriks hasil transpos |
| adjoin | Matriks tipe double, matriks adjoin |
| newM | Matriks tipe double, matriks hasil inverse |
| det | Variabel tipe double, nilai determinan |
| row | Variabel tipe integer, panjang baris |
| col | Variabel tipe integer, panjang kolom |
| identity | Matriks tipe double, matriks identitas |
| process | Matriks tipe double, matriks gabungan matriks yang akan diinvers dan identitas |
| i | Variabel tipe integer, untuk for loop |
| j | Variabel tipe integer, untuk for loop |

### 3.7.2. Metode

| Transpose(double[][] m) | Menerima matriks m dan mengembalikkan m yang telah di transpos |
| --- | --- |
| InverseAdjoin(double[][] m) | Menerima matriks m dan mengembalikkan balikan matriks m dengan perkalian 1/det dan adjoin |
| identityMatrix(int n) | Menerima panjang baris dan kolom matriks nxn dan mengembalikkan matriks identitas sedemikian ukuran |
| InverseIdentity(double[][] m) | Menerima matriks m dan mengembalikkan balikan matriks m dengan melakukan OBE gabungan matriks m dan identitas |

## 3.8 Class InterpolasiPolinomial

### 3.8.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| points | Struktur data abstrak (ADT), merepresentasikan titik x dan titik y yang akan digunakan untuk melakukan interpolasi. |
| a | Matriks tipe double, berisi koefisien untuk polinomial interpolasi |
| matrix | Matriks tipe double yang menyimpan bentukan matriks augmented yang didapat dari input titik. |
| x | Nilai absis yang merupakan bagian dari ADT points. |
| y | Nilai ordinat yang merupakan bagian dari ADT points. |
| solvedMatrix | Matriks tipe double yang merupakan hasil eliminasi Gauss Jordan dari matrix. |
| solution | Array yang menyimpan hasil solusi dari solvedMatrix. |
| result | Variabel tipe double, jumlah dari metode interpolasi (mencari fungsi Pn(x) atau Y) |
| n | Variabel tipe integer, length atau panjang dari points. |
| N | Variabel tipe integer, masukan jumlah input titik. |
| interpolation | Variabel tipe InterpolasiPolinomial, menyimpan referensi ke objek yang baru dari kelas. |
| valX | Variabel tipe double, inputan titik X |
| valY | Variabel tipe double, inputan titik Y |
| i | Variabel tipe integer, untuk for loop |
| j | Variabel tipe integer, untuk for loop |

### 3.8.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| InterpolasiPolinomial | Konstruktor yang menginisialisasi atribut points dengan nilai yang diberikan sebagai parameter, yaitu pasangan titik yang diinputkan oleh pengguna, serta menginisialisasi atribut a dengan memanggil fungsi calculateCoefficient(). |
| calculateCoefficient | Menghitung koefisien polinomial interpolasi dari titik-titik input yang disimpan dalam points. |
| interpolate | menghitung nilai interpolasi Y atau Pn(x) berdasarkan nilai x yang diberikan pengguna. |

## 3.9 Class RegresiBerganda

### 3.9.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| X | Matriks tipe double, matriks berisi elemen X1ᵢ, X2ᵢ, dll. |
| Y | Array tipe double, matriks berisi nilai Yᵢ. |
| m | Variabel tipe integer, panjang baris |
| n | Variabel tipe integer, panjang kolom |
| XT | Matriks tipe double, matriks X yang ditranspos |
| XTX | Matriks tipe double, matriks hasil dari perkalian matriks XT dengan X |
| XTY | Array tipe double, array hasil dari perkalian matriks XT dengan Y |
| XTXInv | Matriks tipe double, matriks XTX yang di invers |
| coefficients | Array tipe double, array berisi nilai-nilai koefisien (Bi) |
| X2 | Matriks tipe double, matriks yang berisi elemen matriks X yang dikuadratkan |
| newX | Matriks tipe double, matriks gabungan X dengan X2 menjadi X pada regresi kuadratik |
| choose | Variabel tipe integer, nilai penentu jenis regresi berganda |
| linear | Variabel tipe boolean, variabel untuk memeriksa bila pilihan berupa linear |
| xk | Variabel tipe double, nilai x yang mau diprediksi |
| yk | Variabel tipe double, nilai y yang mau diprediksi |
| i | Variabel tipe integer, untuk for loop |
| j | Variabel tipe integer, untuk for loop |

### 3.9.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| calculateLinearCoefficients(double[][] X, double[][]Y) | Menghitung koefisien dari regresi linear berganda dari matriks X yang berisi elemen Xij dan matriks Y yang berisi elemen Yi dan mengembalikan array berisi koefisien |
| calculateQuadraticCoefficients(double[][] X, double[][]Y) | Menghitung koefisien dari regresi kuadratik berganda dari matriks X yang berisi elemen Xij dan matriks Y yang berisi elemen Yi dan mengembalikan array berisi koefisien |
| MultipleRegression() | Prosedur Regresi Berganda dengan penginputan jumlah variabel, sampel, elemen Xij, elemen Yi, dan menampilkan fungsi setelah regresi dan nilai Yk (Y prediksi). |

## 3.10 Class Bicubic Spline

### 3.10.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| i | index, representasi baris |
| j | index, representasi kolom |
| x | Nilai variabel untuk koordinat x |
| y | Nilai variabel untuk koordinat y |
| m | Matriks yang digunakan sebagai input atau referensi data, bisa berupa matriks grid yang berisi nilai piksel gambar. |
| Y | Matriks kolom yang menyimpan hasil perhitungan, biasanya berhubungan dengan nilai fungsi interpolasi pada grid. |
| X | Matriks yang digunakan untuk menyimpan basis fungsi interpolasi bicubic |
| grid | Matriks yang menyimpan nilai atau data yang akan diinterpolasi menggunakan metode bicubic spline. |
| mOut | Matriks baru hasil output metode seperti interpolasi, derivatif, atau lainnya. |
| idx | Index |
| a | Array yang menyimpan koefisien hasil solusi persamaan linier setelah proses SPL Gauss-Jordan. |
| result | Variabel untuk menyimpan hasil sementara dari perhitungan, seperti hasil fungsi interpolasi atau operasi aritmatika lainnya. |
| indexA | Index, mengambil elemen dari array a yang menyimpan koefisien interpolasi berdasarkan posisi matriks. |

### 3.10.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| function(int i, int j, int x, int y) | Menghitung nilai basis fungsi berdasarkan pangkat i dan j, serta kordinat x dan y. |
| functionX(int i, int j, int x, int y) | Menghitung turunan fungsi basis terhadap variabel x. |
| functionY(int i, int j, int x, int y) | Menghitung turunan fungsi basis terhadap variabel y. |
| functionXY(int i, int j, int x, int y) | Menghitung turunan parsial kedua dari fungsi basis, terhadap variabel x dan y. |
| matriksY(double[][] m) | Membuat matriks kolom Y yang menyimpan hasil perhitungan pada 16 titik dari grid untuk interpolasi. Saat ini belum diimplementasikan penuh. |
| derivativeX(double[][] m, int i, int j) | Menghitung turunan numerik pada arah x di titik (i, j) dalam matriks m. |
| derivativeY(double[][] m, int i, int j) | Menghitung turunan numerik pada arah y di titik (i, j) dalam matriks m. |
| derivativeXY(double[][] m, int i, int j) | Menghitung turunan numerik pada arah x dan y sekaligus di titik (i, j) dalam matriks m, menggunakan metode perbedaan terbagi. |
| addMatriksX() | Membuat matriks X yang berukuran 16x16, yang berisi nilai fungsi basis dan turunannya (untuk interpolasi bicubic spline) berdasarkan nilai x dan y pada grid. |
| addMatriksY(double[][] m, double[][] grid): | Memodifikasi matriks m dengan menambahkan kolom ke-17 (di indeks 16) untuk menyimpan nilai turunan dan interpolasi dari grid. |
| getListOfA(double[][] m) | Mengambil array koefisien a dari hasil solusi sistem persamaan linier (SPL). |
| finalFunction(double x, double y, double[] a) | Menghitung nilai fungsi interpolasi bicubic pada koordinat x dan y berdasarkan koefisien a. |

## 3.11 InputFile

### 3.11.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| inputFile | Variabel tipe File. |

### 3.11.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| getFile(String filename) | Membuat objek tipe File berdasarkan nama file masukan |
| checkFile(String filename) | Memeriksa apakah file dengan nama yang diberikan ada di direktori. |

## 3.12 WriteFile

### 3.12.1. Atribut

| **Nama Atribut** | **Penjelasan Singkat** |
| --- | --- |
| option | Variabel tipe string, string “Y”/”N” untuk mengkonfirmasi |
| currentDirectory | Variabel tipe string, directory untuk melokasikan filepath file output |
| fileName | Variabel tipe string, nama file output |
| kalimat | Variabel tipe string, string di dalam file output |
| directory | Variabel File, directory untuk file output |
| file | Variabel File, file output |

### 3.12.2. Metode

| **Nama Metode** | **Penjelasan Singkat** |
| --- | --- |
| write(String kalimat) | Mengkonfirmasi untuk write atau tidak. Bila iya, fungsi membuat file dengan nama custom berisi kalimat yang berupa parameter dan menaruh file tersebut di directory\test\output. |

# 

# 

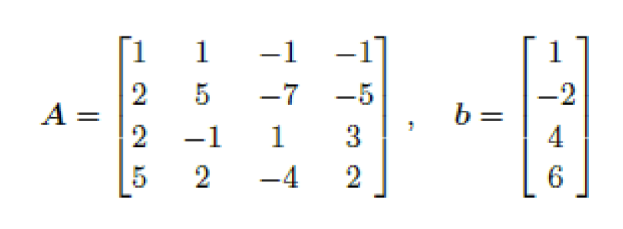
# 

# 

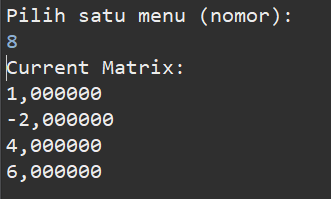
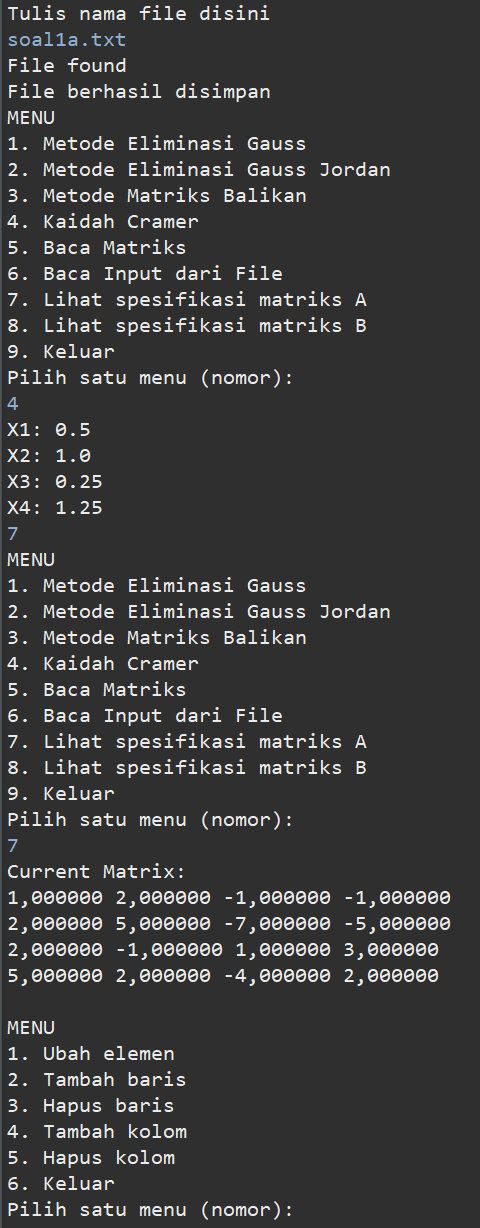
# BAB IV

EKSPERIMEN

Untuk menguji program yang telah dibuat, penulis melakukan *testing* beberapa tes kasus SPL, persoalan interpolasi polinom, dan matriks-matriks sebagai berikut.

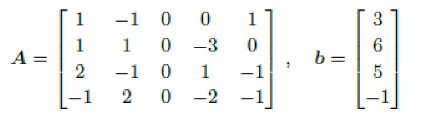
1. Mencari SPL Ax = b dari matriks berikut:

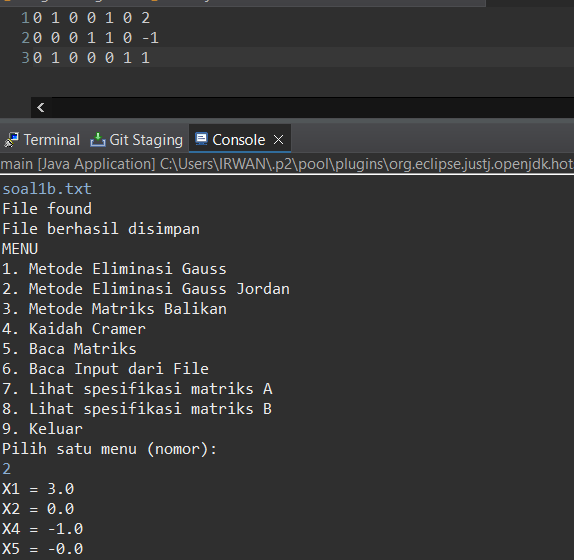
a.



Gambar 1.1. Program uji kasus a dan solusi matriks menggunakan metode cramer

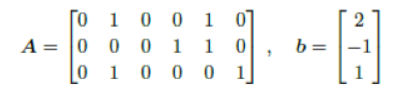
b.

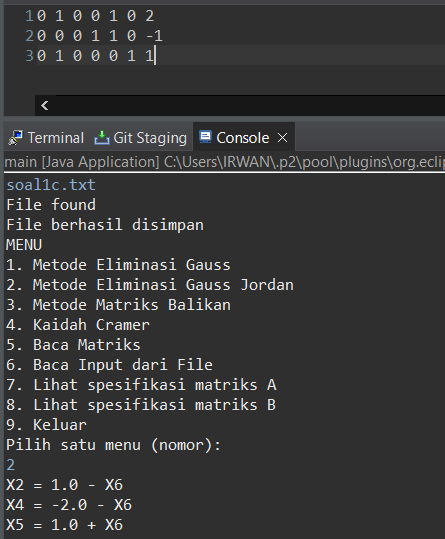




Gambar 1.2. Program uji kasus 1b dan solusi matriks menggunakan metode eliminasi Gauss Jordan.

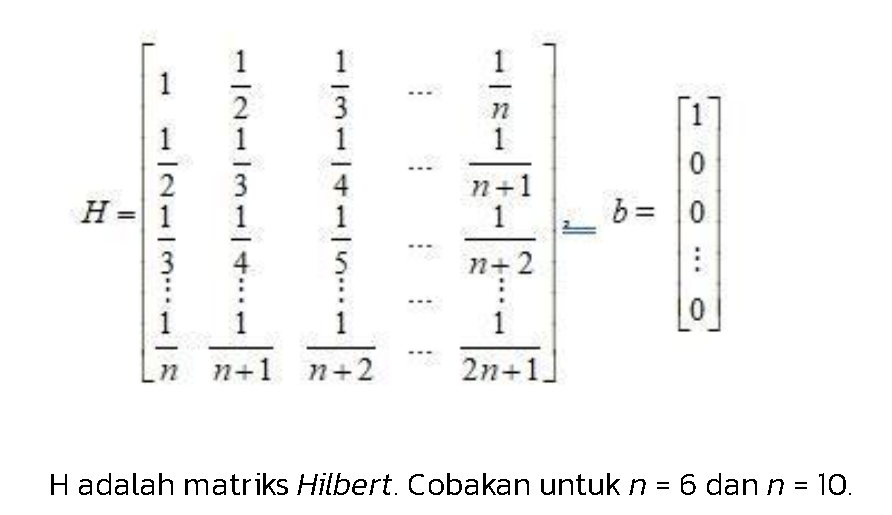
c.

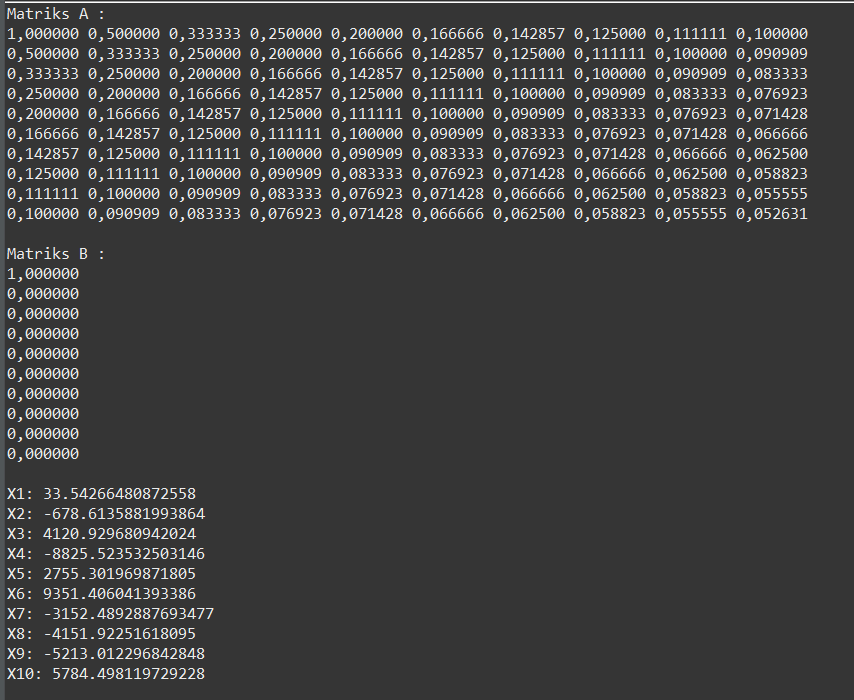




Gambar 1.3. Program uji kasus 1c dan solusi matriks menggunakan metode eliminasi Gauss Jordan.

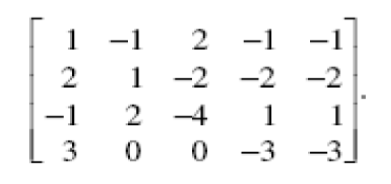
d.

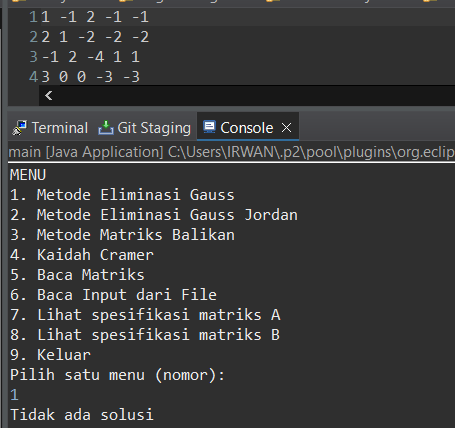




Gambar 1.4. Program uji kasus b dan solusi matriks menggunakan metode cramer balikan (n = 10)

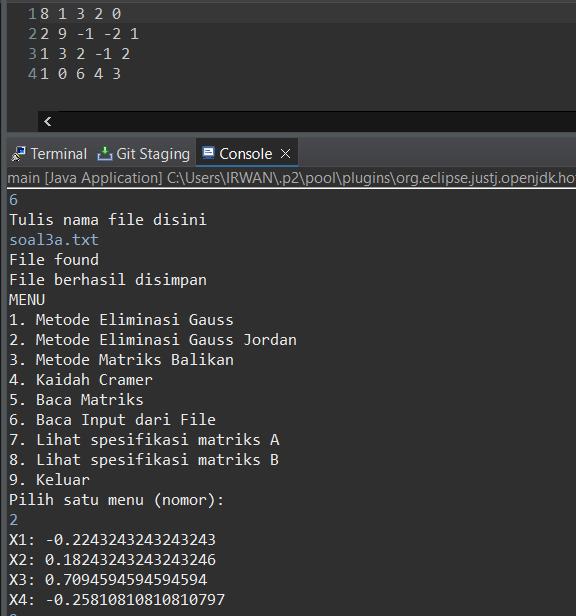
1. Mencari solusi SPL berbentuk matriks augmented





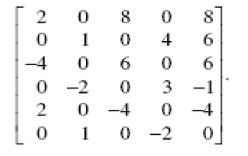
Gambar 2.1. Program uji kasus 2a dan solusi SPL berbentuk matriks augmented menggunakan eliminasi Gauss Jordan

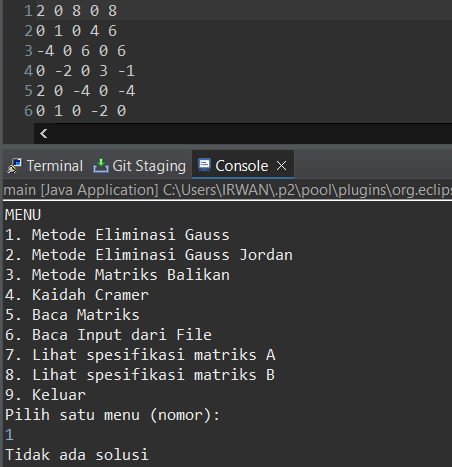
b.



Gambar 2.2. Program uji kasus 3a dan solusi SPL berbentuk matriks augmented menggunakan eliminasi Gauss Jordan

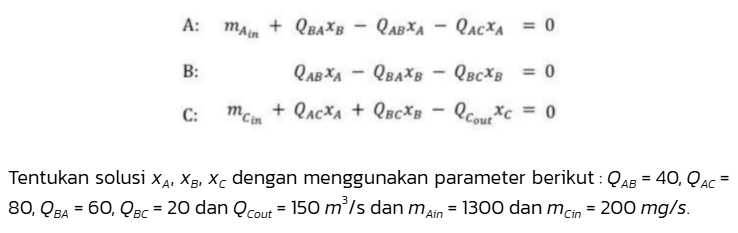
c.

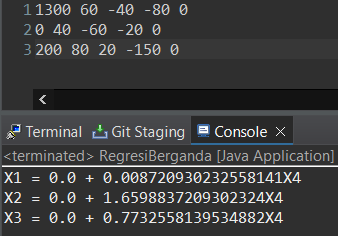




Gambar 2.3. Program uji kasus 2b dan solusi SPL berbentuk matriks augmented menggunakan eliminasi Gauss Jordan

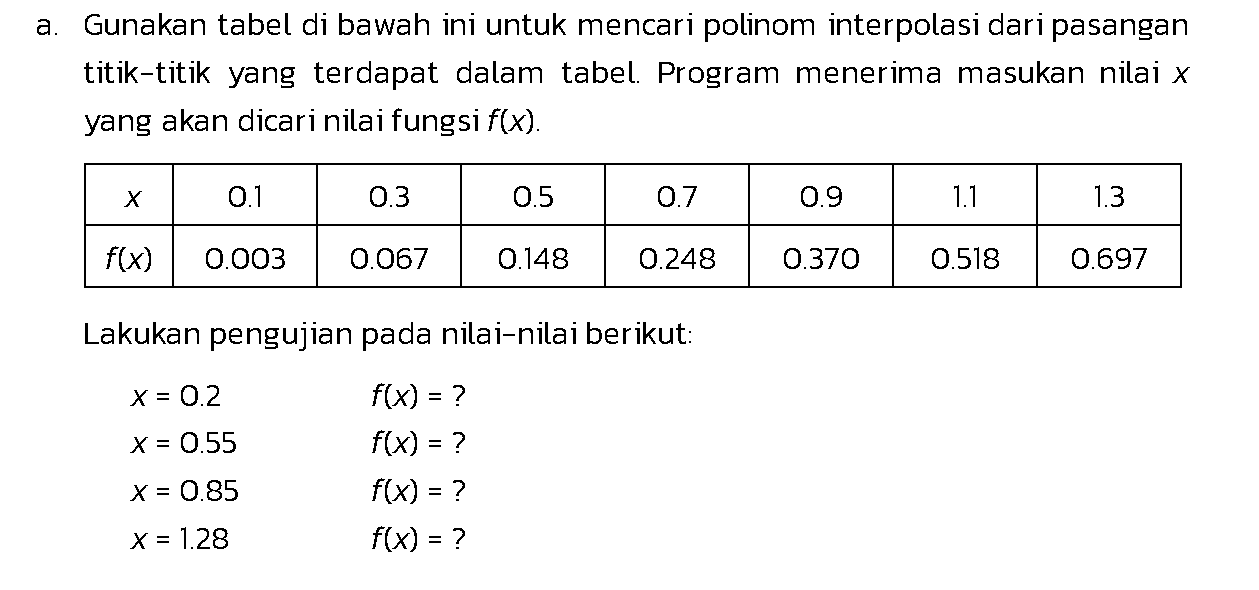
d.

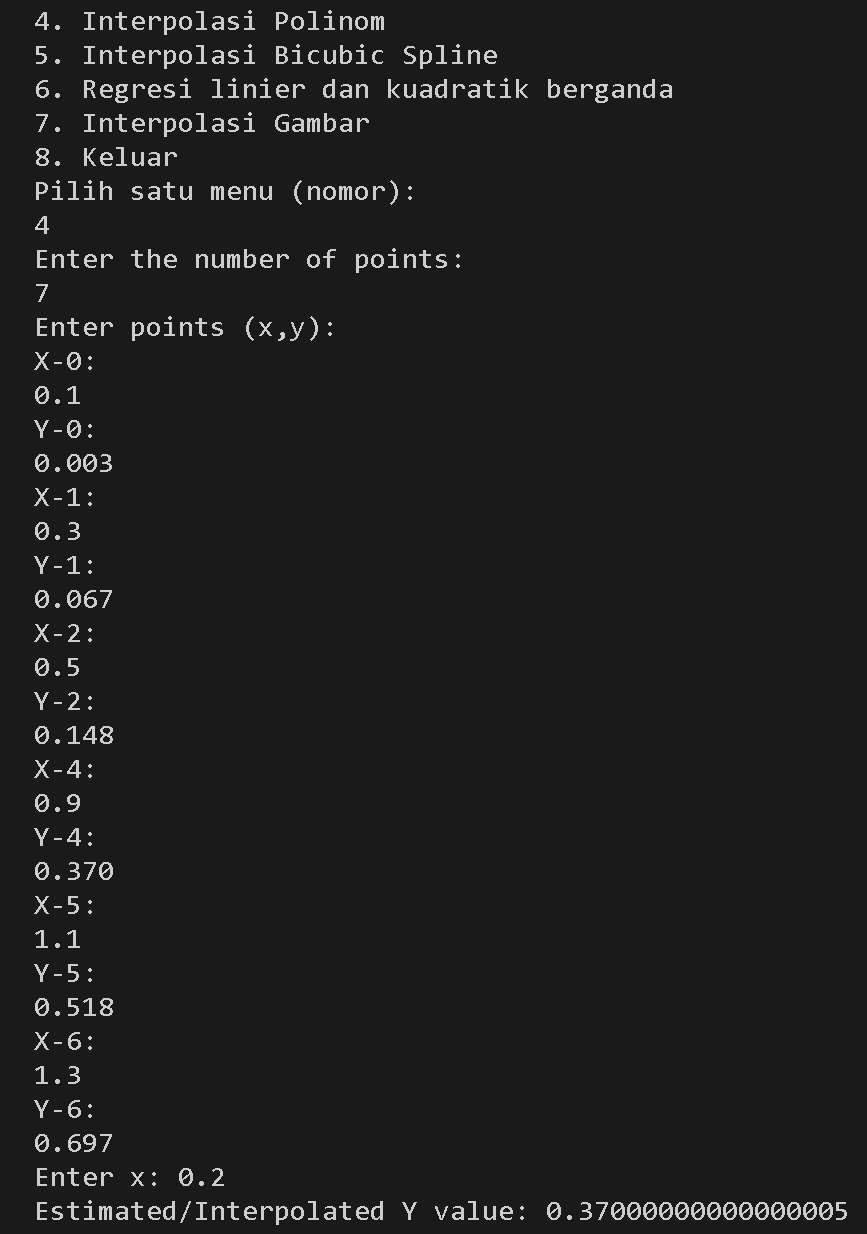




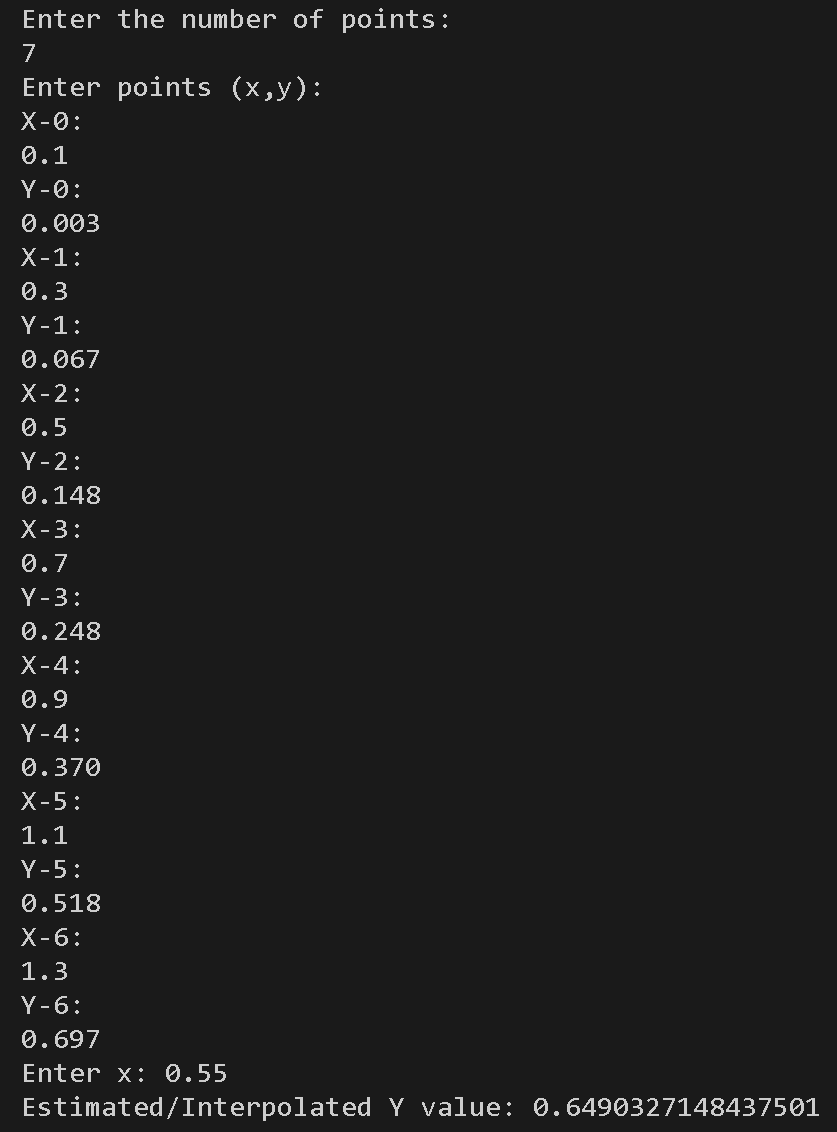
Gambar 2.4. Program uji kasus 4 dan solusi SPL berbentuk matriks augmented menggunakan eliminasi Gauss Jordan

3. Studi Kasus Interpolasi

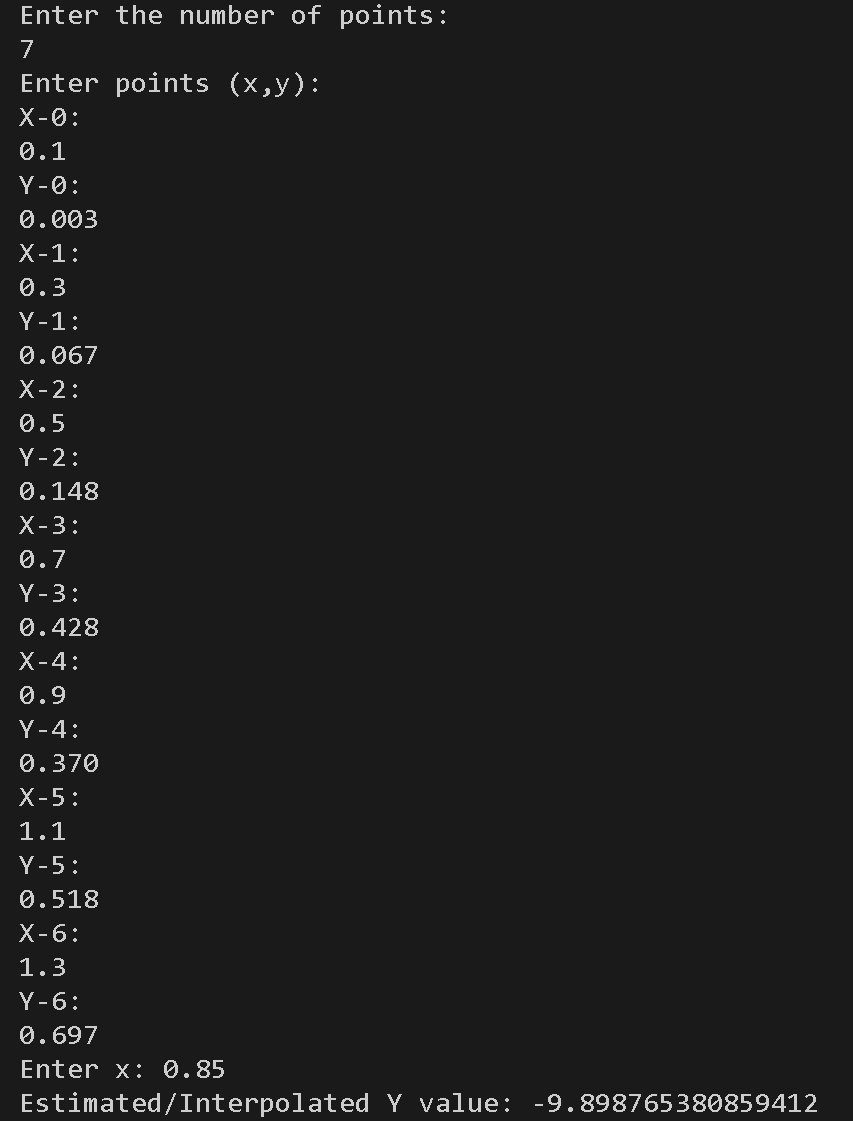




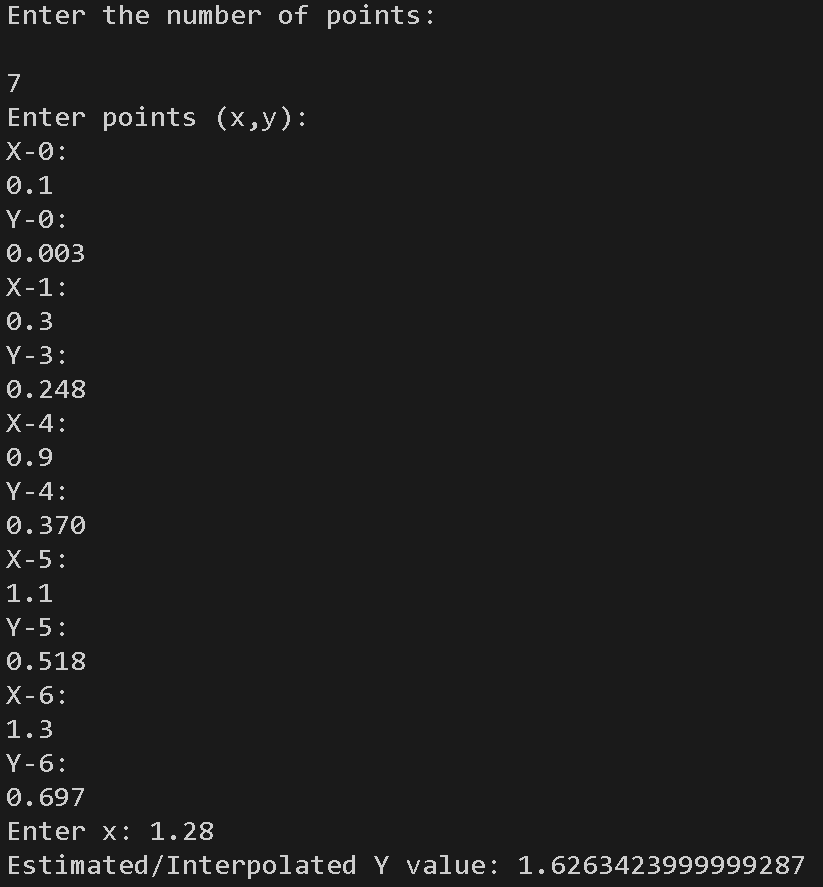
Gambar 3.1. Program uji kasus interpolasi polinomial a, x=0.2, y=0.370



Gambar 3.2. Program uji kasus interpolasi polinomial a, x=0.55, y=0.649



Gambar 3.3. Program uji kasus interpolasi polinomial a, x=0.85, y=-9.899



Gambar 3.4. Program uji kasus interpolasi polinomial a, x=1.28, y=1.626

Berdasarkan uji kasus di atas, didapatkan estimasi interpolasi Y atau P₆(x) untuk setiap x yaitu:

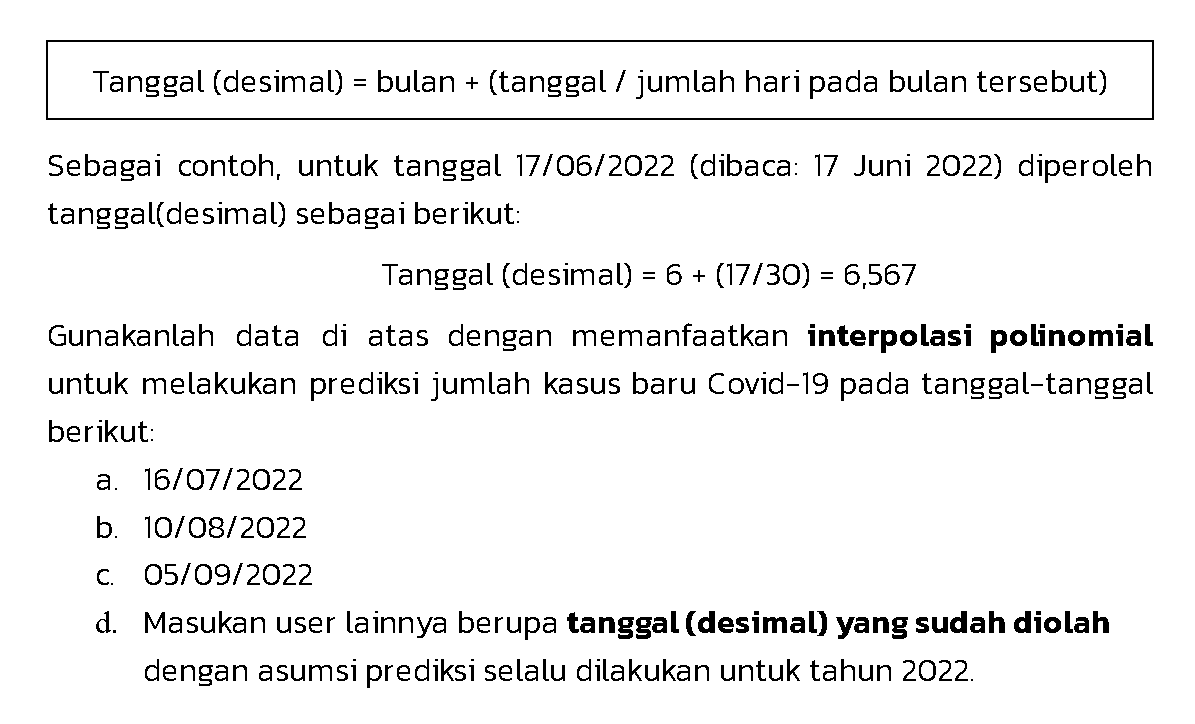
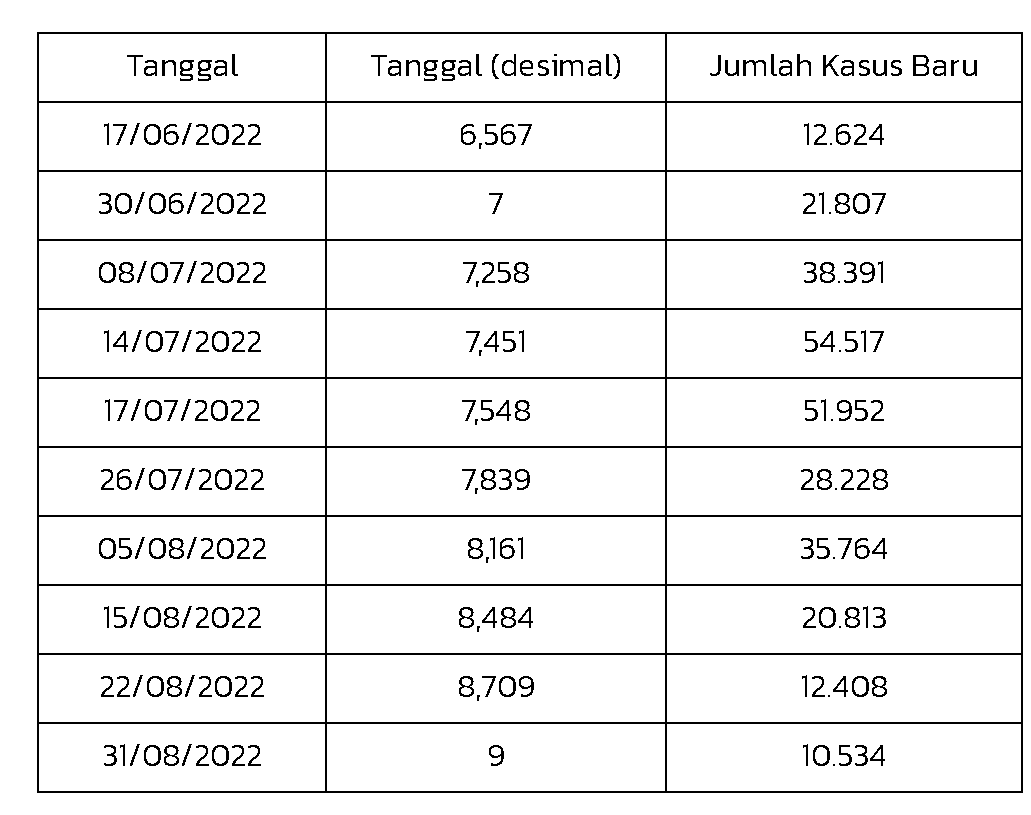
P₆(0.2) = 0.370

P₆(0.55) = 0.649

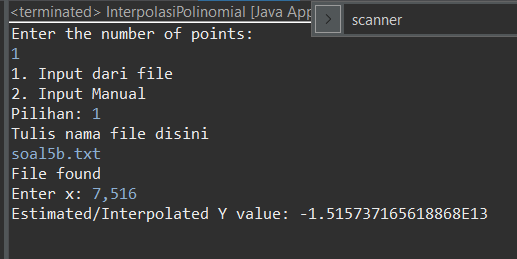
P₆(0.85) = -9.899

P₆(1.28) = 1.626

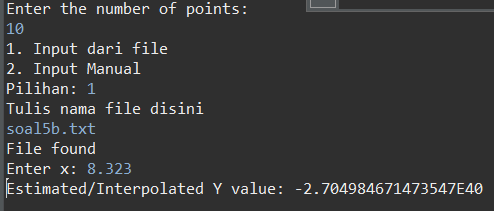
b.



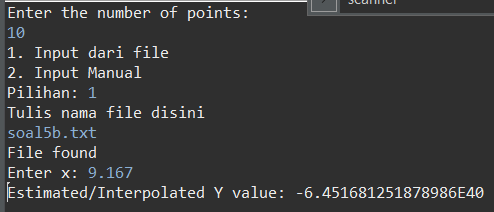
1. Tanggal(desimal) = 7 + (16/31) = 7.516



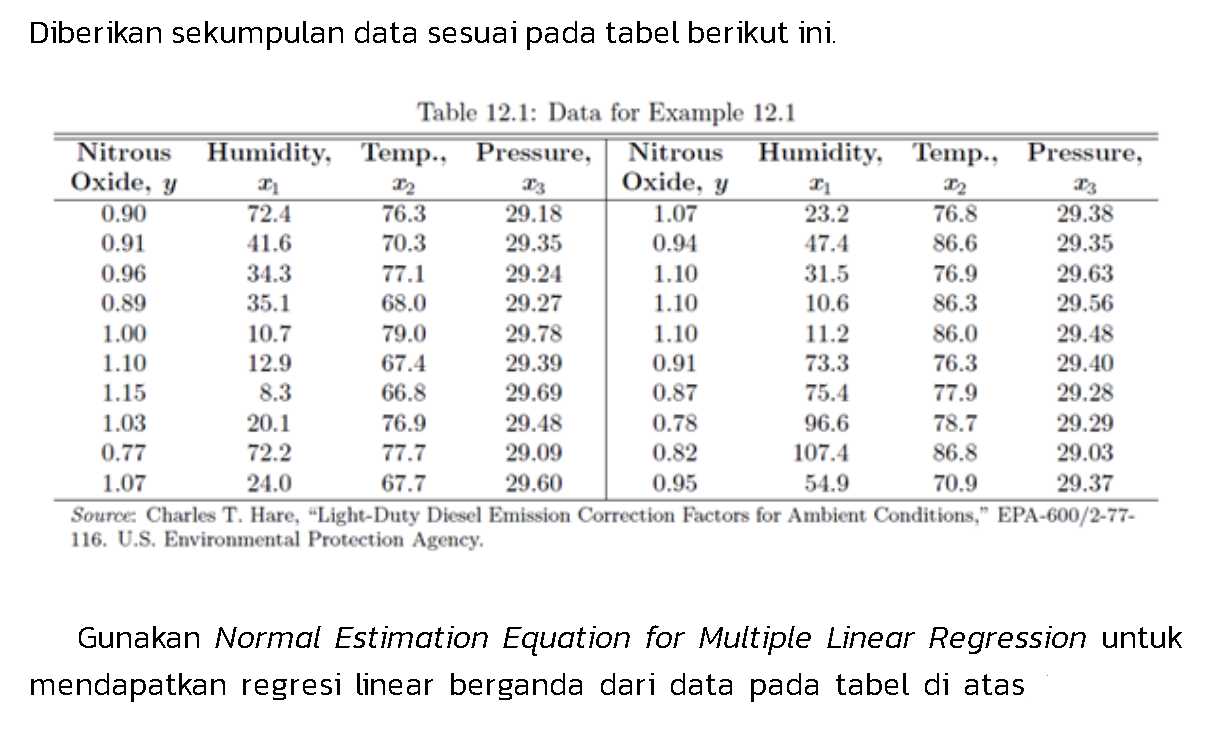
1. Tanggal(desimal) = 8 + (10/31) = 8.323

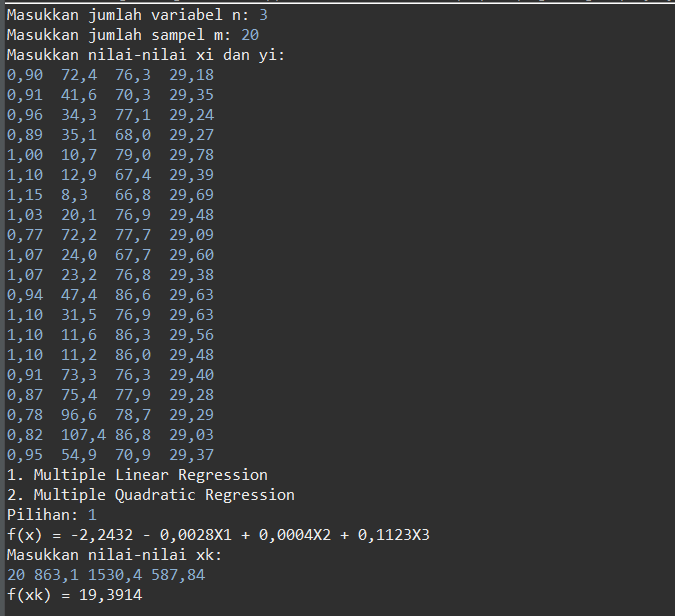


1. Tanggal(desimal) = 9 + (5/30) = 9.167

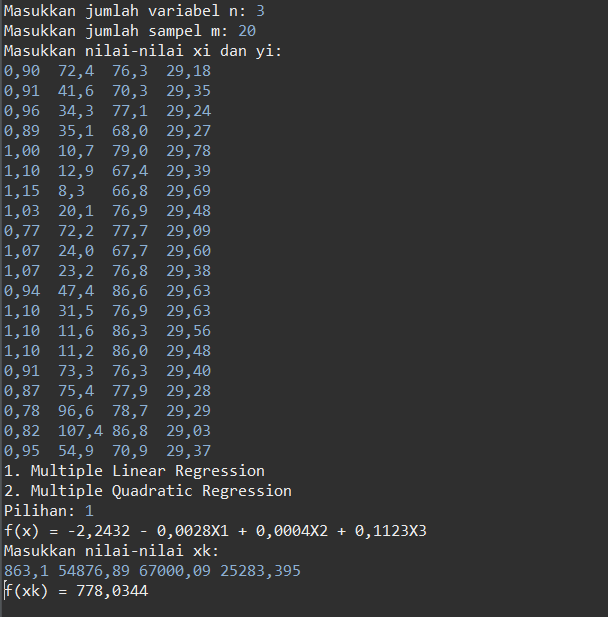


4. Studi Kasus Regresi Linear dan Kuadratik Berganda

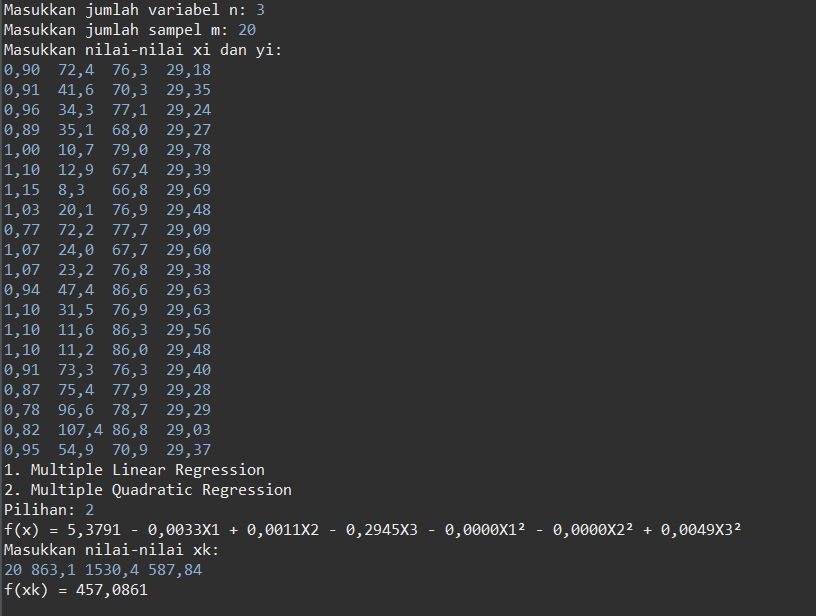




Gambar 4.1 Program uji kasus regresi linear berganda testcase 1



Gambar 4.2 Program uji kasus regresi linear berganda testcase 2



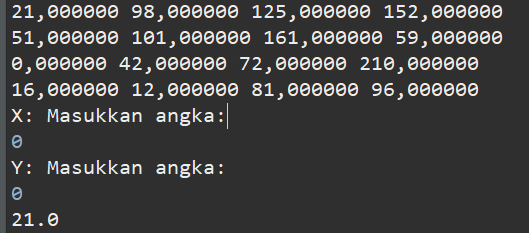
Gambar 4.3 Program uji kasus regresi kuadratik berganda test case 1

Dalam kasus ini, tersedia 20 sampel data nilai Nitrous Oxide dengan faktor yang memengaruhi adalah Humidity, Temperature, dan Pressure. Dengan menggunakan regresi linear berganda, dapat diprediksikan nilai Nitrous Oxide bila diberi data dari faktor-faktor yang memengaruhinya.

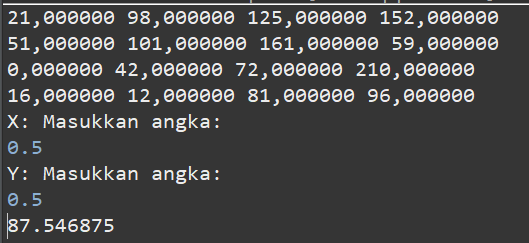
5. Studi Kasus Interpolasi *Bicubic Spline*



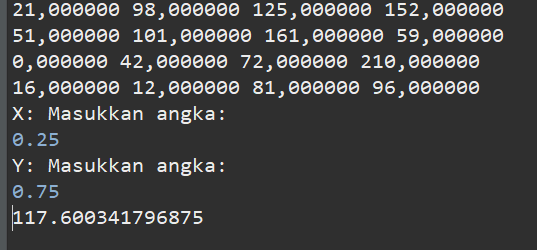
Input f(0,0) didapatkan hasil interpolasi yaitu 21.0



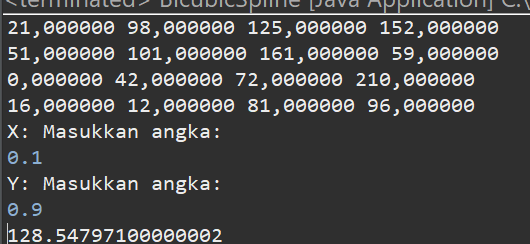
Input f(0.5,0.5) didapatkan hasil interpolasi yaitu 87.55



Input f(0.25,0.75) didapatkan hasil interpolasi yaitu 117.60



Input f(0.1,0.9) didapatkan hasil interpolasi yaitu 128.55



# BAB V

KESIMPULAN

Dalam pengerjaan tugas besar ini, penulis berhasil mengembangkan sebuah program aljabar linear berbasis terminal menggunakan bahasa pemrograman Java. Program ini mampu menyelesaikan berbagai permasalahan seperti sistem persamaan linear, perhitungan determinan dan matriks invers, interpolasi polinomial, interpolasi bicubic spline, serta regresi linear dan kuadratik berganda. Dengan mengimplementasikan metode-metode yang telah dipelajari selama perkuliahan, program ini diharapkan dapat membantu proses belajar, penelitian, maupun mempermudah serta memberikan solusi numerik yang tepat dalam waktu yang singkat.

Meskipun program yang dikembangkan sudah berjalan dengan baik, masih terdapat beberapa hal yang bisa ditingkatkan di masa depan. Salah satunya adalah penambahan antarmuka grafis (GUI) agar pengguna lebih mudah dalam mengoperasikan program. Melalui tugas besar ini, dapat disadari pentingnya pemrograman dalam menyelesaikan masalah matematika dan berharap program ini dapat menjadi alat yang berguna bagi pengguna lain.

# LAMPIRAN

1. <https://github.com/jehna/readme-best-practices/blob/master/README-default.md>
2. <https://www.youtube.com/watch?v=poY_nGzEEWM&ab_channel=Computerphile>
3. [(752) The Example of Gauss-Jordan Elimination for Finding Inverse of Matrix - YouTube](https://www.youtube.com/watch?v=UxImAfs_RBs)