LAPORAN TUGAS BESAR I

IF2123 ALJABAR LINIER DAN GEOMETRI

SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

Dosen : Ir. Rila Mandala, M.Eng., Ph.D.

Disusun Oleh :

Kelompok 16, HURUHARA Semaxximal Mungkin

Wardatul Khoiroh	13523001
Ranashahira Reztaputri	13523007
Heleni Gratia Meitrina	13523107

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

OKTOBER 2024

DAFTAR ISI

BAB I

DESKRIPSI MASALAH

1.1 Deskripsi Data

A. Sistem Persamaan Linier (SPL)

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan (x = A-1b), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

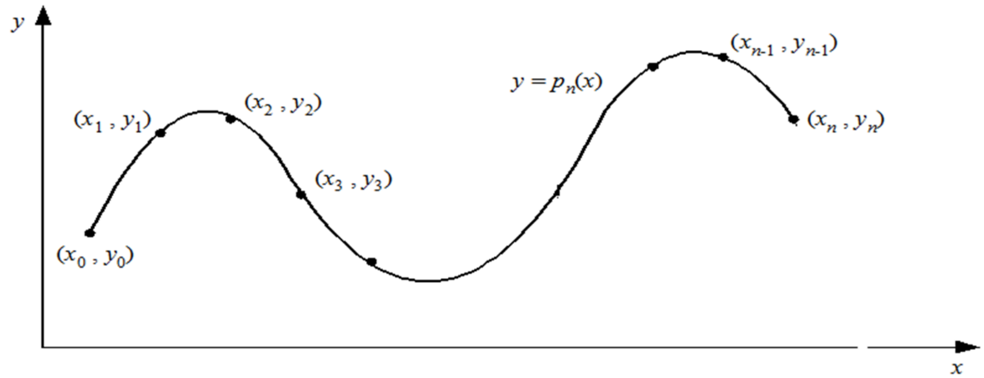
$$\begin{bmatrix} 0 & \mathbf{2} & 1 & -1 \\ 0 & 0 & \mathbf{3} & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad \begin{bmatrix} 0 & \mathbf{1} & 0 & -\frac{2}{3} \\ 0 & 0 & \mathbf{1} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss-Jordan dengan matriks eselon baris tereduksi.

Di dalam Tugas Besar 1 ini, Anda diminta membuat satu atau lebih library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, gunakan library tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

B. Interpolasi Polinomial

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, …, n.



Gambar 2. Ilustrasi beberapa titik yang diinterpolasi secara polinomial.

Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang $[x_0, x_n]$.

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ 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 $p_2(x) = a_0 + a_1x + a_2x^2$ 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 $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, 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 (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk $i = 0, 1, 2, \ldots, n$, akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \ldots, a_n$,

$$
\begin{aligned}
a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n &= y_0 \\
a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n &= y_1 \\
\ldots \qquad\qquad \ldots & \\
a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n &= y_n
\end{aligned}
$$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0, a_1, …, a_n, 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 $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

Penyelesaian sistem persamaan dengan 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 $p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$. Dengan menggunakan polinom ini, maka nilai fungsi pada $x = 9.2$ dapat ditaksir sebagai berikut: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

C. Regresi Berganda

Regresi (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Pada tugas besar ini, anda diminta untuk membuat 2 jenis regresi yaitu Regresi Linier Berganda dan Regresi Kuadratik Berganda.

1. Regresi Linier Berganda

Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

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

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

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

2. **Regresi Kuadratik Berganda**

Dalam kasus ini, proses mengubah data-data dalam regresi kuadratik berganda cukup berbeda dengan Regresi Linier Berganda. Bentuk persamaan dari regresi kuadratik ada 3, yaitu:

a. Variabel Linier: Variabel dengan derajat satu seperti X, Y, dan Z
b. Variabel Kuadrat: Variabel dengan derajat dua seperti X2
c. Variabel Interaksi: 2 Variabel dengan derajat satu yang dikalikan dengan satu sama lain seperti XY, YZ, dan XZ

Setiap n-peubah, jumlah variabel linier, kuadrat, dan interaksi akan berbeda-beda. Perhatikan contoh regresi kuadratik 2 variabel peubah sebagai berikut!

$$\begin{pmatrix} N & \sum u_i & \sum v_i & \sum u_i^2 & \sum u_i v_i & \sum v_i^2 \\ \sum u_i & \sum u_i^2 & \sum u_i v_i & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i v_i^2 \\ \sum v_i & \sum u_i v_i & \sum v_i^2 & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum v_i^3 \\ \sum u_i^2 & \sum u_i^3 & \sum u_i^2 v_i & \sum u_i^4 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 \\ \sum u_i v_i & \sum u_i^2 v_i & \sum u_i v_i^2 & \sum u_i^3 v_i & \sum u_i^2 v_i^2 & \sum u_i v_i^3 \\ \sum v_i^2 & \sum u_i v_i^2 & \sum v_i^3 & \sum u_i^2 v_i^2 & \sum u_i v_i^3 & \sum v_i^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i u_i \\ \sum y_i v_i \\ \sum y_i u_i^2 \\ \sum y_i u_i v_i \\ \sum y_i v_i^2 \end{pmatrix}$$

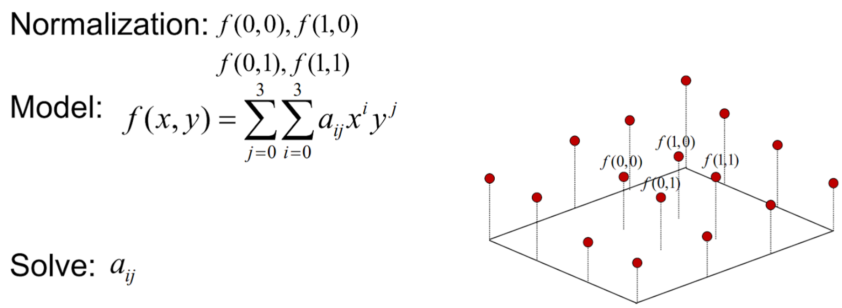
N menandakan jumlah peubah, terdapat 2 variabel linier yaitu ui dan vi, 2 variabel kuadrat yaitu ui2 dan vi2, dan 1 variabel interaksi yaitu uv. Untuk setiap n-peubah, akan terdapat 1 konstan N (Terlihat di bagian atas kiri gambar), n variabel linier, n variabel kuadrat, dan C2n variabel linier (dengan 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.

Kedua model regresi yang dijadikan sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

D. Bicubic Spline Interpolation

Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. *Bicubic spline interpolation* melibatkan konsep *spline* dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. Pendekatan ini menciptakan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat daripada metode interpolasi linear.

Dalam pemrosesan menggunakan interpolasi *bicubic spline* digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.



Gambar 3. Pemodelan interpolasi *bicubic spline*.

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

$$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j$$

$$f_x(x,y) = \sum_{j=0}^{3}\sum_{i=1}^{3} a_{ij} i x^{i-1} y^j$$

$$f_y(x,y) = \sum_{j=1}^{3}\sum_{i=0}^{3} a_{ij} j x^i y^{j-1}$$

$$f_{xy}(x,y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij} ij x^{i-1} y^{j-1}$$

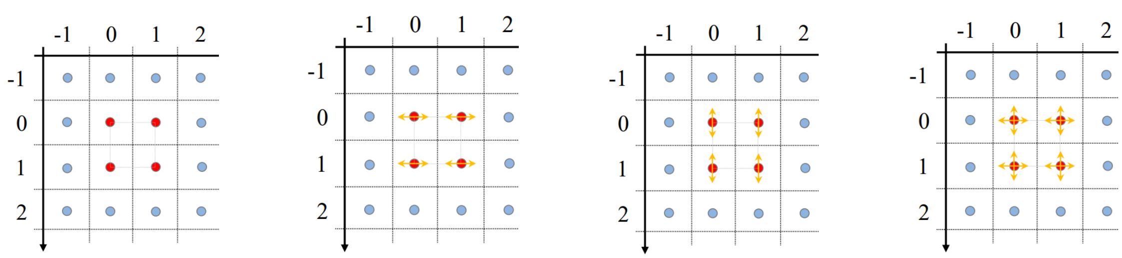
Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi X yang membentuk persamaan penyelesaian sebagai berikut.

$$y = Xa$$

$$\begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f_x(0,0) \\ f_x(1,0) \\ f_x(0,1) \\ f_x(1,1) \\ f_y(0,0) \\ f_y(1,0) \\ f_y(0,1) \\ f_y(1,1) \\ f_{xy}(0,0) \\ f_{xy}(1,0) \\ f_{xy}(0,1) \\ f_{xy}(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 4 & 6 & 0 & 3 & 6 & 9 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Perlu diketahui bahwa elemen pada matriks X adalah nilai dari setiap komponen koefisien a_{ij} yang diperoleh dari persamaan fungsi maupun persamaan turunan yang telah dijelaskan sebelumnya. Sebagai contoh, elemen matriks X pada baris 8 kolom ke 2 adalah koefisien dari a_{10} pada ekspansi sigma untuk $f_x(1, 1)$ sehingga diperoleh nilai konstanta 1 x 1^{1-1} x 1^0 = 1, sesuai dengan isi matriks X.

Nilai dari vektor a dapat dicari dari persamaan $y = Xa$, lalu vektor a tersebut digunakan sebagai nilai variabel dalam $f(x, y)$, sehingga terbentuk fungsi interpolasi bicubic sesuai model. Tugas Anda pada studi kasus ini adalah membangun persamaan $f(x, y)$ yang akan digunakan untuk melakukan interpolasi berdasarkan nilai $f(a, b)$ dari masukan matriks 4 x 4. Nilai masukan a dan b berada dalam rentang [0, 1]. Nilai yang akan diinterpolasi dan turunan berarah disekitarnya dapat diilustrasikan pada titik berwarna merah pada gambar di bawah.



Gambar 4. Nilai fungsi yang akan di interpolasi pada titik merah, turunan berarah terhadap sumbu x, terhadap sumbu y, dan keduanya (kiri ke kanan).

Untuk studi kasus ini, buatlah matriks X menggunakan persamaan yang ada (tidak *hardcode*) serta carilah invers matriks X dengan *library* yang telah kalian buat dalam penyelesaian masalah.

BAB II

TEORI SINGKAT

2.1 Metode Eliminasi Gauss

Metode Eliminasi Gauss adalah metode eliminasi yang ditemukan oleh Gauss yang dilakukan dengan cara mengubah matriks augmented menjadi matriks eselon baris melalui rangkaian Operasi Baris Elementer (OBE), kemudian mendapatkan solusi dari persamaan linear dengan penyulihan mundur. Algoritma yang dapat digunakan untuk melakukan metode Eliminasi Gauss pada komputer adalah:

1) Cari baris dengan nilai absolut terbesar dimulai pada kolom pertama, lalu tukar baris tersebut dengan baris pertama.
2) Penukaran tersebut memberikan pivot yaitu pada baris pertama kolom pertama, bagi baris pertama dengan pivot sehingga elemen yang menjadi pivot berubah menjadi bernilai satu. Kita mendapatkan satu utama pada baris pertama.
3) Gunakan kelipatan dari baris pertama untuk menghilangkan semua nilai yang berada pada kolom yang sama (yang berada dibawah) supaya menjadi nol.
4) Ulangi langkah diatas untuk kolom yang tersisa.
5) Lakukan penyulihan mundur untuk mendapatkan hasil dari sistem persamaan linear.

2.2 Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan merupakan perluasan dari metode Eliminasi Gauss yang diperkenalkan oleh Jordan. Hal yang membedakan metode eliminasi ini dengan metode Eliminasi Gauss adalah matriks augmented diubah menjadi matriks eselon baris tereduksi sehingga akan lebih mudah karena nilai variabel langsung didapat tanpa adanya substitusi mundur. Algoritma yang dapat digunakan untuk melakukan metode Eliminasi Gauss-Jordan pada komputer adalah:

1) Cari pivot mulai dari kolom pertama dengan mencari nilai bukan nol pada tiap baris.
2) Tukarkan baris tersebut dengan baris pertama.
3) Penukaran tersebut memberikan pivot yaitu pada baris pertama kolom pertama, bagi baris pertama dengan pivot sehingga elemen yang menjadi pivot berubah menjadi bernilai satu. Kita mendapatkan satu utama pada baris pertama.
4) Gunakan kelipatan dari baris pertama untuk menghilangkan semua nilai yang berada pada kolom yang sama (kecuali pivot) supaya menjadi nol.
5) Ulangi langkah diatas untuk kolom yang tersisa.
6) Solusi sistem persamaan linear langsung didapat tanpa melakukan substitusi mundur.

2.3 Determinan

Determinan adalah nilai skalar yang dihasilkan fungsi dari entri-entri suatu matriks persegi. Determinan dari matriks A umumnya dinyatakan dengan notasi det(A), det A, atau |A|. Determinan dapat dianggap sebagai faktor penskalaan transformasi yang digambarkan oleh matriks. Perhitungan determinan berbeda-beda untuk matriks yang berbeda ukuran. Namun, terdapat dua yang bersifat umum sehingga dapat digunakan untuk matriks n x n, diantaranya:

1) Reduksi baris dengan OBE sehingga didapat matriks segitiga
2) Ekspansi kofaktor

2.4 Matriks Balikan

Jika A adalah matriks persegi non-singular, maka terdapat matriks A^{-1} berukuran n x n, yang disebut matriks invers dari A sehingga memenuhi sifat:

$$AA^{-1} = A^{-1}A = I, \text{ di mana I adalah matriks identitas}$$

Perlu diperhatikan bahwa untuk mencari matriks invers, matriks persegi harus non-singular yang nilai determinannya tidak sama dengan nol. Terdapat dua metode yang dapat digunakan untuk mencari matriks balikan, yaitu:

1) Menggunakan eliminasi Gauss-Jordan

$$[A \mid I] \sim [I \mid A^{-1}],$$

2) Menggunakan adjoin

$$A^{-1} = \text{adj}(A)/\det(A)$$

2.5 Matriks Kofaktor

Untuk memahami matriks kofaktor, perlu diketahui istilah minor dan kofaktor. Minor suatu matriks *A* dilambangkan dengan Mij adalah determinan matriks bagian dari matriks A yang diperoleh dengan cara menghilangkan elemen – elemennya pada baris ke-i dan elemen elemen pada kolom ke-j. Kofaktor adalah hasil perkalian minor dengan suatu angka yang besarnya menuruti suatu aturan yaitu $(-1)^{i+j}$ dimana i adalah baris dan j adalah kolom. Kofaktor suatu elemen baris ke-i dan kolom ke-j dari matriks A dilambangkan dengan Cij. Jadi, matriks kofaktor merupakan matriks yang terdiri dari kofaktor-kofaktor matriks itu sendiri. Jadi, misalkan terdapat suatu matriks katakanlah matriks A, maka matriks kofaktor A merupakan matriks yang terdiri dari kofaktor-kofaktor dari matriks A. Susunan elemen matriks kofaktor juga mengikuti susunan kofaktor-kofaktornya.

2.6 Matriks Adjoin

Matriks adjoin berkaitan erat dengan matriks kofaktor. Matriks adjoin merupakan hasil transpose dari matriks kofaktor. Transpose matriks merupakan matriks yang dioperasikan dengan melakukan pertukaran elemen baris menjadi kolom dan elemen kolom menjadi baris dari matriks awalnya.

2.7 Kaidah *Cramer*

Kaidah *Cramer* adalah salah satu metode yang digunakan untuk menyelesaikan sistem persamaan. Kaidah ini melibatkan determinan. yaitu, nilai-nilai variabel dalam sistem ditemukan dengan bantuan determinan. Kaidah Cramer dapat digunakan untuk mencari solusi dari sistem persamaan linear. Mari kita perhatikan sistem persamaan n variabel x_1, x_2, x_3, ..., x_n yang ditulis dalam bentuk matriks Ax = b (b adalah matriks kolom dengan konstanta), dengan langkah-langkah sebagai berikut:

1) Cari determinan A dan lambangkan dengan det(A)
2) Cari det(A_1), det(A_2), …, det(A_n), di mana det(A_i) adalah determinan dari matriks A yang kolom ke-i nya telah digantikan dengan matriks kolom b
3) Bagi setiap det(A_i) dengan det(A) untuk mendapatkan nilai x_i

2.8 Interpolasi Polinom

Interpolasi polinomial merupakan teknik interpolasi dengan mengasumsikan pola data yang kita miliki mengikuti pola polinomial baik berderajat satu (linier) maupun berderajat tinggi. Interpolasi dengan metode ini dilakukan dengan terlebih dahulu membentuk persamaan polinomial. Persamaan polinomial yang terbentuk selanjutnya digunakan untuk melakukan interpolasi dari nilai yang diketahui atau ekstrapolasi (prediksi) dari nilai diluar rentang data yang diketahui**.** 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) = $a_0 + a_1x + a_2x_2 + \ldots + a_nx_n$ untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2, …, a_n:

$$a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n = y_0$$
$$a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n = y_1$$
$$\ldots \qquad \ldots$$
$$a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n = y_n$$

Cari nilai koefisien a_0, a_1, a_2, …, a lalu kembalikan ke bentuk persamaan:

$$a_0 + a_1x + a_2x^2 + \ldots + a_n x^n = y$$

Lalu substitusikan nilai x yang ingin diinterpolasikan ke dalam persamaan polinom tersebut.

2.9 Interpolasi *Bicubic Spline*

Interpolasi *Bicubic Spline* merupakan perpanjangan dari interpolasi kubik untuk menginterpolasi titik data pada grid reguler dua dimensi. Implementasinya sangat penting untuk melakukan interpolasi sesuai arah x, kemudian melakukan interpolasi pada arah y. Maka hasil yang diinginkan dapat diperoleh. Dalam perhitungan interpolasi *Bicubic Spline*, kita memperkirakan nilai suatu titik pada grid dua dimensi menggunakan 16 titik di sekitarnya. Berikut adalah langkah utama dalam proses perhitungannya:

1) Siapkan grid 4x4 yang berisi nilai-nilai fungsi f(x,y) di 16 titik yang diketahui. Selain nilai fungsi, nilai turunan berarah di sekitar titik-titik juga diperlukan
2) Buat sistem persamaan berdasarkan nilai fungsi f, serta turunannya fx, fy, dan fxy di 16 titik di sekitar lokasi yang ingin diinterpolasi. Simpan ke dalam matriks X
3) Selesaikan sistem persamaan linear y = Xa, carilah nilai koefisien a
4) Setelah mendapat nilai koefisien a, buat fungsi interpolasi bicubic

   $$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}.x^{i}.y^{j}$$

5) Substitusikan nilai yang ingin di interpolasikan

2.10 Regresi Linier

Regresi dapat digunakan untuk meneliti hubungan antar dua atau lebih variabel, dengan paling tidak satu variabel sebagai variabel dependen (respon) y dan variabel lainnya sebagai variabel independen (variabel prediktor) x. Regresi linear sederhana adalah analisis regresi linear yang hanya melibatkan dua variabel, yaitu satu variabel prediktor dan satu variabel respon. Pemodelan dari regresi linear sederhana adalah:

$$y_i = \beta_0 + \beta_1 x_1 + \epsilon_i$$

Sedangkan, regresi linear berganda adalah model regresi linear dengan melibatkan lebih dari satu variabel bebas atau prediktor. Regresi linear berganda memiliki model sebagai berikut:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_{2i} + \ldots + \beta_k x_{ki} + \epsilon_i$$

2.11 Kuadratik Berganda

Regresi kuadratik berganda adalah bentuk regresi yang menggabungkan variabel linier, kuadrat, dan interaksi untuk memodelkan hubungan antara variabel dependen dan beberapa variabel independen. Ini adalah perluasan dari regresi linier berganda, tetapi mencakup variabel kuadrat dan interaksi untuk menangkap hubungan yang lebih kompleks. Model regresi kuadratik berganda dapat dinyatakan sebagai:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \epsilon$$

Langkah-langkah penyelesaian dari regresi kuadratik berganda adalah:

1) Siapkan data yang berisi variabel independen dan dependen. Dalam kasus regresi kuadratik berganda, variabel independen tidak hanya mencakup variabel asli, tetapi juga variabel kuadrat (misalnya x_1^2) dan variabel interaksi (misalnya $x_1 x_2$).
2) Bangun matriks variabel. Variabel independen diperluas untuk memasukkan variabel linier, variabel kuadrat, dan interaksi. Sebagai contoh, untuk dua variabel independen x_1 dan x_2, matriks desain untuk regresi kuadratik akan berisi kolom
   - x_1, x_2 (variabel linier),
   - x_1^2, x_2^2 (variabel kuadrat),
   - $x_1 x_2$ (variabel interaksi).
3) Bentuk model regresi kuadratik lengkap dengan memasukkan semua variabel linier, kuadrat, dan interaksi ke dalam persamaan regresi, didapat model y = $x\beta + \epsilon$
4) Cari nilai koefisien β untuk mendapatkan sistem persamaan linear yang utuh

BAB III

IMPLEMENTASI PUSTAKA DAN PROGRAM

3.1 Implementasi Pustaka

Pustaka yang dibuat, yakni Abstract Data Type (ADT). ADT ini berisi fungsi-fungsi primitif untuk mengelola matriks yang digunakan untuk menyelesaikan berbagai persoalan lainnya (digunakan dalam fungsi lain).

1. **matriks.java**

   Berisi ADT yang digunakan dalam keberjalanan program.

   1) **matriks ()**

      Parameter: int Baris, int Kolom

      Membuat matriks kosong dengan panjang baris dan kolom sesuai input.

   2) **matriks ()**

      Parameter: double[][] mat

      Menyalin isi dari matriks yang bertipe double[][] ke tipe matriks.

   3) **matriks ()**

      Parameter: String file_name

      Membaca file yang berisi matriks dan menyimpannya nilainya dalam tipe matriks.

   4) **toDoubleArray()**

      Menyalin nilai dari matriks ke array.

   5) **matriksToString()**

      Parameter: double[] m

      Mengembalikan string hasil konversi dari tipe double [].

   6) **matriksToString()**

Parameter : double[][] m

Mengembalikan string hasil konversi dari tipe double [][].

7) getX ()

Mengembalikan nilai x yang disimpan dari pembacaan file.

8) getY()

Mengembalikan nilai y yang disimpan dari pembacaan file.

9) simpan()

Parameter: String content

Menyimpan string yang diinput pada suatu file.

10) GetFirstIdxBar

Parameter: matriks M

Mengembalikan indeks pertama dari baris matriks M.

11) GetFirstIdxCol

Parameter: matriks M

Mengembalikan indeks pertama dari kolom matriks M.

12) GetLastIdxBar

Parameter: matriks M

Mengembalikan indeks terakhir dari baris matriks M.

13) GetLastIdxCol

Parameter: matriks M

Mengembalikan indeks terakhir dari kolom matriks M.

14) GetElement

Parameter: int m, int n;

Mengembalikan nilai matriks di baris m dan kolom n;

15) SetElement()

Parameter: int m, int n, double value

Mengisi nilai matriks di baris ke-m dan kolom ke-n dengan value.

16) ReadMat()

Menginput nilai matriks dari keyboard (manual).

17) PrintMat()

Mengeluarkan (output) matriks ke layar.

18) Multiply ()

Parameter: matriks M, double k

Mengembalikan matriks hasil perkalian matriks M dan double K.

19) Multiply ()

Parameter: matriks M, matriks N

Mengembalikan matriks hasil perkalian matriks M dan matriks N.

20) Swap()

Parameter: int bar1, int bar2

Menukar letak 2 baris pada indeks bar1 dan bar2.

21) AddXBaris()

Parameter: int bar1, int bar2, double k

Melakukan penambahan pada baris indeks bar1 dengan hasil kali k dengan baris indeks bar2.

22) Identity()

Parameter: int N

Mengembalikan matriks identitas dengan dimensi N.

23) **ChangeVal()**

Parameter: int m, int n, double k

Mengisi nilai matriks dari baris m dan kolom n dengan k.

3.2 Program

Berisi pemrograman utama dan fungsi-fungsi untuk menyelesaikan persoalan di berbagai aplikasi terkait.

1. **menu.java**

   Menu adalah program utama yang menampilkan menu pilihan-pilihan permasalahan yang ingin diselesaikan. Menu yang tersedia berupa:

   1) Sistem Persamaan Linear
   2) Determinan
   3) Matriks Matriks Balikan
   4) Interpolasi Polinom
   5) Interpolasi Bicubic Spline
   6) Regresi Linier dan Kuadratik
   7) Keluar

   Penguna (*User*) akan diminta untuk memasukkan pilihan (opsi) permasalahan yang ingin diselesaikan.

2. **Bicubic.java**

   Bicubic.java memuat aplikasi dari pustaka untuk membuat interpolasi *Bicubic Spline.*

   a. **fungsi()**

      Parameter: matriks M, double x, double y, int baris, int kolom

      Mengisi nilai matriks M dengan perhitungan *double sigma,* yaitu persamaan polinomial dari 0 sampai 3.

   b. **turunanX()**

      Parameter: matriks M, double x, double y, int baris, int kolom

Mengisi nilai matriks dengan perhitungan polinomial turunan pertama x.

c. **turunanY()**

Parameter: matriks M, double x, double y, int baris, int kolom

Mengisi nilai matriks dengan perhitungan polinomial turunan pertama y.

d. **turunanXY()**

Parameter: matriks M, double x, double y, int baris, int kolom

Mengisi nilai matriks dengan perhitungan polinomial turunan pertama x dan y.

e. **solusiMatrix()**

Parameter: matriks M

Mengisi nilai dari fungsi(), turunanX(), turunanY(), dan turunanXY() ke susunan matriks 16x16.

f. **interpolate()**

Parameter: matriks M, double x, double y

Mengembalikan nilai hasil interpolasi setelah diketahui nilai matriks balikan dari matriks di solusiMatrix().

g. **hasil_Bicubic_Akhir()**

Parameter: matriks M_Input, double X, double Y

Mengembalikan nilai hasil dari interpolasi *bicubic spline* dengan melakukan perangkaian pada fungsi-fungsi sebelumnya.

3. **Cramer.java**

Cramer.java memuat aplikasi dari pustaka untuk mencari solusi dari sistem persamaan linier dengan menggunakan metode Cramer.

a. **OperasiCramer**

Parameter: matriks M, int n, double[] x, int flag

Mengembalikan nilai flag yang merupakan penanda apakah sistem persamaan linier memiliki solusi unik, tak hingga, atau tak ada solusi.

4. **Determinan.java**

Determinan.java memuat aplikasi dari pustaka untuk membuat hasil determinan dari suatu matriks dengan pilihan 2 cara, yaitu menggunakan OBE dan ekspansi kofaktor.

a. **determinanOBE**

Parameter: matriks M, int n

Mengembalikan nilai determinan dengan melakukan Operasi Baris Elementer.

b. **Kofaktor**

Parameter: matriks M, int n, int p, int q

Mengembalikan nilai kofaktor dari suatu matriks yang disimpan dalam variabel bertipe matriks.

c. **determinanKofaktor**

Parameter: matriks M, int n

Mengembalikan nilai determinan dengan menggunakan fungsi kofaktor.

5. **Gauss.java**

Gauss.java memuat aplikasi dari pustaka untuk mencari solusi dari sistem persamaan linier dengan menggunakan metode Gauss.

a. **OperasiGauss**

Parameter: double[][] mat, int m, int n, int flag

Mengembalikan nilai flag yang merupakan penanda apakah sistem persamaan linier memiliki solusi unik atau tidak.

b. **Cek**

Parameter: double[][] mat, int m, int n, int flag

Mengembalikan nilai flag yang merupakan penanda apakah sistem persamaan linier memiliki solusi tak hingga atau tidak ada solusi.

c. **SubstitusiParametrik**

Parameter: double[][] mat, int m, int n

Menghasilkan luaran untuk kasus solusi tak hingga sehingga menampilkan solusi dalam bentuk parametrik.

6. **GaussJordan.java**

GaussJordan.java memuat aplikasi dari pustaka untuk mencari solusi dari sistem persamaan linier dengan menggunakan metode Gauss-Jordan.

a. **OperasiGaussJordan**

Parameter: double[][] mat, int m, int n, int flag

Mengembalikan nilai flag yang merupakan penanda apakah sistem persamaan linier memiliki solusi unik atau tidak.

b. **Cek**

Parameter: double[][] mat, int m, int n, int flag

Mengembalikan nilai flag yang merupakan penanda apakah sistem persamaan linier memiliki solusi tak hingga atau tidak ada solusi.

c. **SubstitusiParametrik**

Parameter: double[][] mat, int m, int n

Menghasilkan luaran untuk kasus solusi tak hingga sehingga menampilkan solusi dalam bentuk parametrik.

7. **InterPolinom.java**

InterPolinom memuat aplikasi dari pustaka untuk mencari solusi dari sistem persamaan lalu menyulihkan hasilnya sehingga terbentuk sistem persamaan yang utuh yang digunakan untuk menaksir nilai fungsi pada x tertentu yang menjadi masukan.

a. **OperasiPolinom**

Parameter: int n, double[][] mat

Menghasilkan nilai koefiesien dari persamaan.

b. Substitusi

Parameter: double[][] mat, int m, int n, double[] x

Menyimpan solusi dari eliminasi Gauss ke dalam x.

c. UbahKeAug

Parameter: int n, double[][] titik

Mengubah input yang berbentuk titik ke dalam matriks *augmented*.

d. Hitung

Parameter: int n, double[] a, double x

Menghitung nilai y dari dari suatu persamaan dengan nilai x yang sudah dimiliki.

8. inverseADJ.java

inverseADJ.java memuat aplikasi dari pustaka untuk membuat hasil matriks balikan dari suatu matriks dengan metode matriks adjoin.

a. Transpose

Parameter: matriks M

Mengembalikan matriks hasil transpose dari matriks M.

b. matriksKofaktor

Parameter: matriks M

Mengembalikan matriks hasil kofaktor dari matriks M.

c. inverseAdjoin

Parameter: matriks M

Mengembalikan matriks hasil invers/balikan dengan metode matriks adjoin.

9. **InverseOBE.java**

InverseOBE.java memuat aplikasi dari pustaka untuk membuat hasil matriks balikan dari suatu matriks dengan metode Operasi Baris Elementer/OBE.

a. **EliminasiGaussJordan**

Parameter: double[][] mat, int m, int n, int flag

Mengembalikan nilai flag yang menandakan ada tidaknya pivot.

b. **inverseGaussJordan**

Parameter: double[][] mat, int m, int n

Mengembalikan matriks hasil invers/balikan dengan metode OBE/Operasi Baris Elementer atau Gauss Jordan.

10. **InverseSPL.java**

InverseSPL.java memuat aplikasi dari pustaka untuk mencari solusi dari sistem persamaan linier dengan menggunakan matriks balikan.

a. **determinant**

Parameter: double[][] matrix

Mengembalikan determinan dari suatu matriks.

b. **OperasiInverse**

Parameter: double[][] mat, int m, int n

Mengembalikan solusi dari sistem persamaan linier.

c. **isConsistent**

Parameter: double[][] A, double[] b, int m

Mengecek apakah sistem persamaan linier tersebut memiliki konsisten (memiliki solusi yang unik) atau tidak.

11. **RegresiLinierBerganda.java**

RegresiKuadratikBerganda.java memuat aplikasi dari pustaka untuk membuat hasil matriks balikan dari suatu matriks dengan metode Operasi Baris Elementer/OBE.

a. **EliminasiGauss**

   Parameter: double[][] mat, int m, int n, double[] x

   Mengubah matriks awal menjadi matriks eselon dengan Eliminasi Gauss.

b. **multipleLinierRegression**

   Parameter: double[][] data_X, double[] data_Y

   Mengembalikan koefisien regresi linier berganda dengan penyelesaian SPL.

12. RegresiKuadratikBerganda.java

RegresiLinierBerganda.java memuat aplikasi dari pustaka untuk membuat hasil matriks balikan dari suatu matriks dengan metode Operasi Baris Elementer/OBE.

a. **EliminasiGauss**

   Parameter: double[][] mat, int m, int n, double[] x

   Mengubah matriks awal menjadi matriks eselon dengan Eliminasi Gauss.

b. **buildSampelMatriks**

   Parameter: double[][] variabel

   Menghasilkan satu matriks sampel dari data_X.

c. **tambahMatriks**

   Parameter: double[][] x, double[][] y

   Mengembalikan hasil penjumlahan dari dua matriks.

d. **calculateHasilX**

   Parameter: double[][] data_X

   Mengembalikan matriks hasil_X sebagai penjumlahan semua matriks sampel.

e. **calculateHasilY**

Parameter: double[][] data_X, double[] data_Y

Mengembalikan matriks hasil_Y sesuai perhitungan.

f. multipleQuadraticRegression

Parameter: double[][] data_X, double[] data_Y

Mengembalikan koefisien regresi kuadratik berganda dengan penyelesaian SPL.

BAB IV

EKSPERIMEN

4.1 Tampilan Program

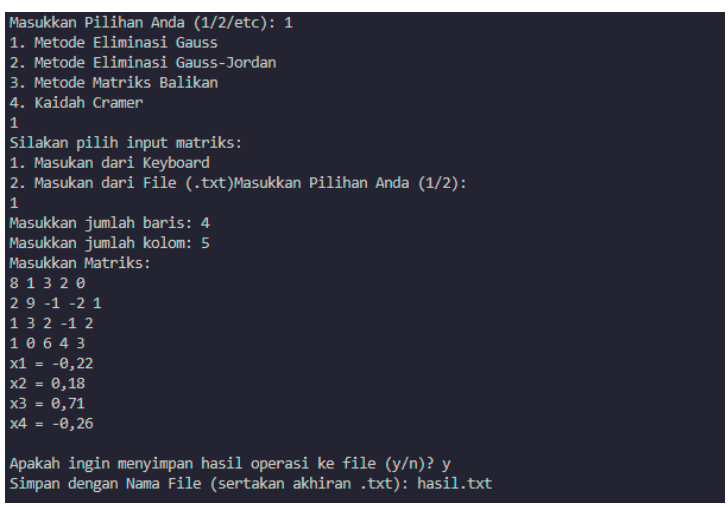
1. Tampilan Menu

Gambar 1. Tampilan Awal dan Menu

2. Solusi Sistem Persamaan Linier

a) Metode Eliminasi Gauss

1) Input Keyboard (Manual) dan Simpan dalam File



Gambar 2. Tampilan Pemilihan Menu SPL Gauss dan Input Manual.

2) Input dari File (.txt) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
1
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): test11.txt
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
x1 = -0,22
x2 = 0,18
x3 = 0,71
x4 = -0,26

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 3. Tampilan Pemilihan Menu SPL Gauss dan Input dari File.

b) Metode Eliminasi Gauss-Jordan

1) Input Keyboard (Manual) dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Masukkan Matriks:
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
x1 = -1,00 + 1,00s
x2 = 0,00 + 2,00t
x3 = t
x4 = s

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): hasil.txt
```

Gambar 4. Tampilan Pemilihan Menu SPL Gauss-Jordan dan Input Manual.

2) Input dari File (.txt) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): test12.txt
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
x1 = -1,00 + 1,00s
x2 = 0,00 + 2,00t
x3 = t
x4 = s

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 5. Tampilan Pemilihan Menu SPL Gauss-Jordan dan Input dari File.

c) Metode Matriks Balikan

1) Input Keyboard (Manual) dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
3
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
1
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Masukkan Matriks:
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
Solusi dari SPL adalah:
x1 = -0,22
x2 = 0,18
x3 = 0,71
x4 = -0,26

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): hasil.txt
```

Gambar 6. Tampilan Pemilihan Menu SPL Inverse dan Input Manual.

2) Input dari File (.txt) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
3
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): test13.txt
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Solusi dari SPL adalah:
x1 = -0,22
x2 = 0,18
x3 = 0,71
x4 = -0,26

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 7. Tampilan Pemilihan Menu SPL Inverse dan Input dari File.

d) Kaidah Cramer

1) Input Keyboard (Manual) dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
4
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
1
Masukkan jumlah baris/kolom: 4
Masukkan Matriks:
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
x1 = -0,22
x2 = 0,18
x3 = 0,71
x4 = -0,26

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): hasil.txt
```

Gambar 8. Tampilan Pemilihan Menu SPL Cramer dan Input Manual.

2) Input dari File (.txt) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 1
1. Metode Eliminasi Gauss
2. Metode Eliminasi Gauss-Jordan
3. Metode Matriks Balikan
4. Kaidah Cramer
4
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): test13.txt
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
x1 = -0,22
x2 = 0,18
x3 = 0,71
x4 = -0,26

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 9. Tampilan Pemilihan Menu SPL Inverse dan Input dari File.

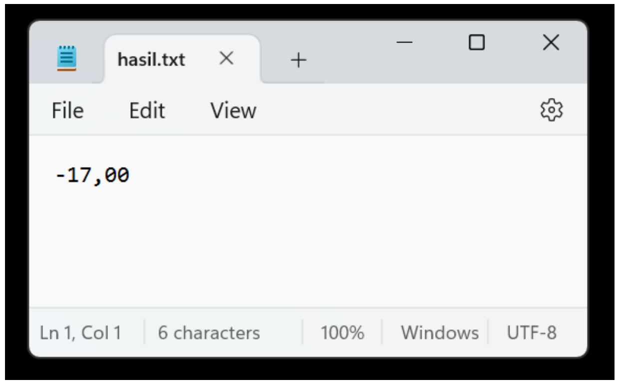
3. Solusi Determinan

a) Determinan Menggunakan OBE

1) Input Keyboard (Manual) dan Simpan dalam File

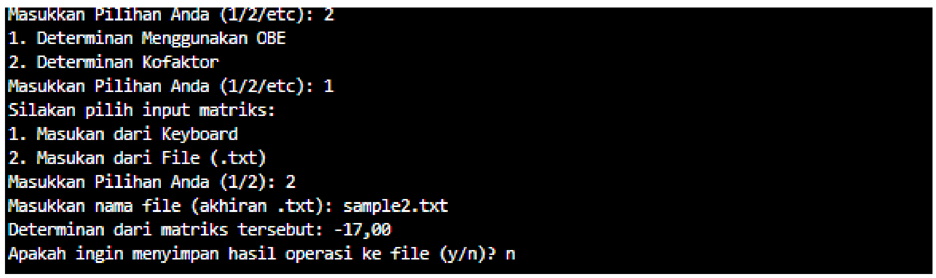
```
Masukkan Pilihan Anda (1/2/etc): 2
1. Determinan Menggunakan OBE
2. Determinan Kofaktor
Masukkan Pilihan Anda (1/2/etc): 1
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 1
Masukkan jumlah baris/kolom: 3
Masukkan Matriks:
2 1 3
1 4 2
3 1 2
Determinan dari matriks tersebut: -17,00
Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): hasil.txt
```

Gambar 10. Tampilan Pemilihan Menu Determinan Cara OBE dan Input Manual.

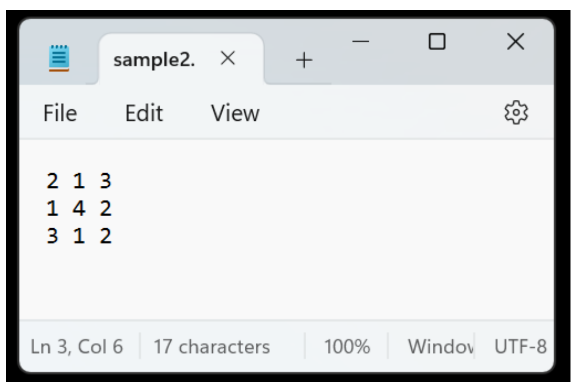


Gambar 11. Isi File Hasil Perhitungan yang Disimpan

2) Input dari File (.txt) dan Tidak Simpan dalam File



Gambar 12. Tampilan Pemilihan Menu Determinan Cara OBE dan Input dari File



Gambar 13. Isi File yang Dibaca

b) Determinan Kofaktor

1) Input Keyboard (Manual) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 2
1. Determinan Menggunakan OBE
2. Determinan Kofaktor
Masukkan Pilihan Anda (1/2/etc): 2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 1
Masukkan jumlah baris/kolom: 5
Masukkan Matriks:
3 4 6 8 9
10 11 2 4 3
3 6 7 8 1
1 1 1 1 4
9 8 7 3 1
Determinan dari matriks tersebut: 2556,00
Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 14. Tampilan Pemilihan Menu Determinan: Kofaktor dan Input Manual

2) Input File dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 2
1. Determinan Menggunakan OBE
2. Determinan Kofaktor
Masukkan Pilihan Anda (1/2/etc): 2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Masukkan nama file (akhiran .txt): test2.2.txt
Determinan dari matriks tersebut: 2556,00
Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): Hasil2.txt
```

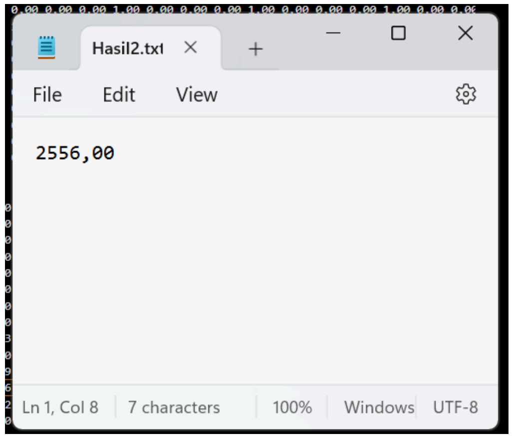
Gambar 15. Tampilan Pemilihan Menu Determinan: Kofaktor dan Input dari File

```
test2.2.txt
File   Edit   View

3 4 6 8 9
10 11 2 4 3
3 6 7 8 1
1 1 1 1 4
9 8 7 3 1

Ln 1, Col 1 | 51 characters | 100% | Windows (( | UTF-8
```

Gambar 16. Isi File yang Dibaca



Gambar 17. Isi File Hasil Perhitungan yang Disimpan

4. Solusi Matriks Balikan

a) Matriks Balikan Metode OBE

1) Input Keyboard (Manual) dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 3
1. Metode OBE
2. Metode Matriks Adjoin
1
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 1
Masukkan jumlah baris/kolom: 3
Masukkan Matriks:
3 5 7
3 3 3
2 3 1
-0.33 0.89 -0.33
0.17 -0.61 0.67
0.17 0.06 -0.33

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): output.txt
```

Gambar 18. Tampilan Pilih Menu Matriks Balikan Metode OBE dan Input Manual.

```
output.txt U ×
output.txt
1  -0.33 0.89 -0.33
2  0.17 -0.61 0.67
3  0.17 0.06 -0.33
4
```

Gambar 19. Isi File Hasil Perhitungan yang Disimpan

2) **Input dari File (.txt) dan Tidak Simpan dalam File**

```
Masukkan Pilihan Anda (1/2/etc): 3
1. Metode OBE
2. Metode Matriks Adjoin
1
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Masukkan nama file (akhiran .txt): masukan.txt
-0.33 0.89 -0.33
0.17 -0.61 0.67
0.17 0.06 -0.33

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 20. Tampilan Pemilihan Menu Matriks Balikan Metode OBE dan Input dari File.

```
masukan.txt U ×
masukan.txt
1  3 5 7
2  3 3 3
3  2 3 1
```

Gambar 21. Isi File yang Dibaca

b) Matriks Balikan Metode Matriks Adjoin

1) **Input Keyboard (Manual) dan Tidak Simpan dalam File**

```
Masukkan Pilihan Anda (1/2/etc): 3
1. Metode OBE
2. Metode Matriks Adjoin
2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2):
1
Masukkan jumlah baris/kolom: 3
Masukkan Matriks:
3 5 7
3 3 3
2 3 1
-0.33 0.89 -0.33
0.17 -0.61 0.67
0.17 0.06 -0.33

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 22. Tampilan Pemilihan Menu Matriks Balikan Metode Matriks Adjoin dan Input Manual.

2) Input dari File (.txt) dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 3
1. Metode OBE
2. Metode Matriks Adjoin
2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): masukan.txt
-0.33 0.89 -0.33
0.17 -0.61 0.67
0.17 0.06 -0.33

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): output.txt
```

Gambar 23. Tampilan Pemilihan Menu Matriks Balikan Metode Matriks Adjoin dan Input dari File.

```
masukan.txt U ×
masukan.txt
1   3 5 7
2   3 3 3
3   2 3 1
```

Gambar 24. Isi File yang Dibaca

```
output.txt U ×
output.txt
1    -0.33 0.89 -0.33
2    0.17 -0.61 0.67
3    0.17 0.06 -0.33
4
```

Gambar 25. Isi File Hasil Perhitungan yang Disimpan

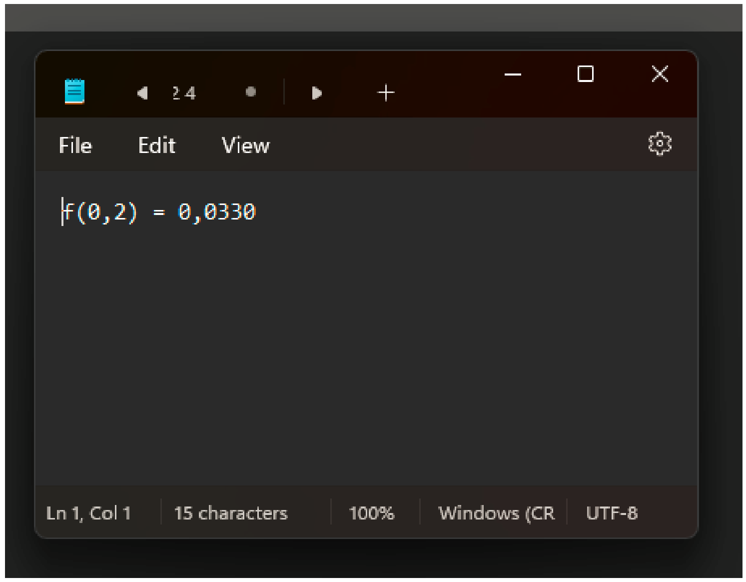
5) Solusi Interpolasi Polinom

a) Masukan dari Keyboard dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 4
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 1
Masukkan jumlah titik: 7
Masukkan Matriks:
0,1 0,003
0,3 0,067
0,5 0,148
0,7 0,248
0,9 0,370
1,1 0,518
1,3 0,697
Masukkan nilai x: 0,2
f(x) = -0,0000x^6 + 0,0000x^5 + 0,0260x^4 + 0,0000x^3 + 0,1974x^2 + 0,2400x + -0,0230
f(0,2) = 0,0330

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): hasil.txt
```

Gambar 26. Tampilan Pemilihan Menu Interpolasi Polinom dan Input Manual.



Gambar 27. Tampilan Pemilihan Menu Interpolasi Polinom dan Input dari File.

b) **Masukan dari File dan Tidak Simpan dalam File**

```
Masukkan Pilihan Anda (1/2/etc): 4
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Masukkan nama file (akhiran .txt): test4.txt
Masukkan jumlah titik: 7
f(x) = -0,0000x^6 + 0,0000x^5 + 0,0260x^4 + 0,0000x^3 + 0,1974x^2 + 0,2400x + -0,0230
f(0,2) = 0,0330

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

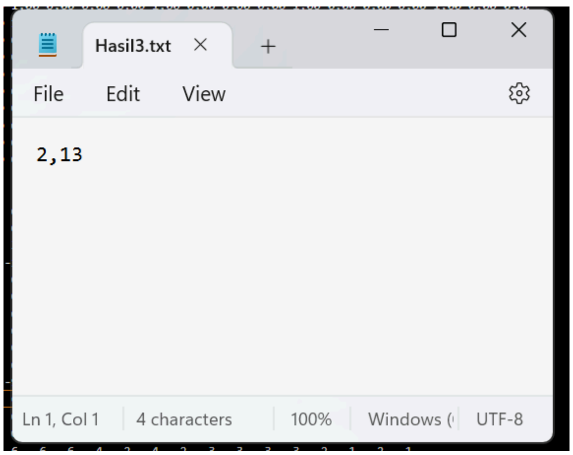
6) Solusi Interpolasi Bicubic Spline

a) Masukan dari Keyboard dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 5
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 1
Masukkan Matriks:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Masukkan x: 0,5
Masukkan y: 0,5
Hasil Binterpolasi Bicubic: 2,13

Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): Hasil3.txt
```

Gambar 28. Tampilan Pemilihan Menu Interpolasi Bicubic Spline dan Input Manual.



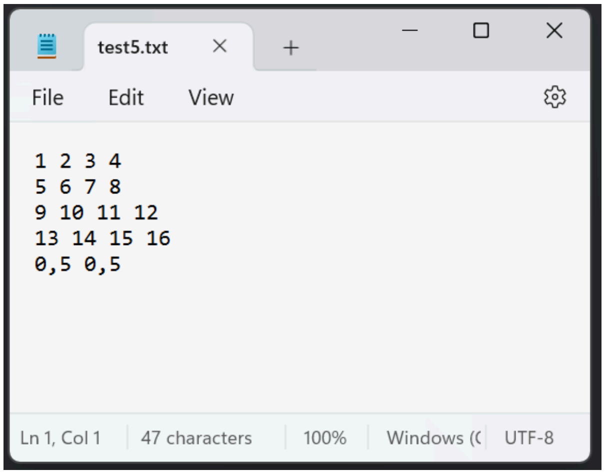
Gambar 29. Isi File Hasil Perhitungan yang Disimpan

b) Masukan dari File dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 5
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Silakan masukan file text (.txt): test5.txt
Hasil Binterpolasi Bicubic: 2,13

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 30. Tampilan Pemilihan Menu Interpolasi Bicubic Spline dan Input dari File



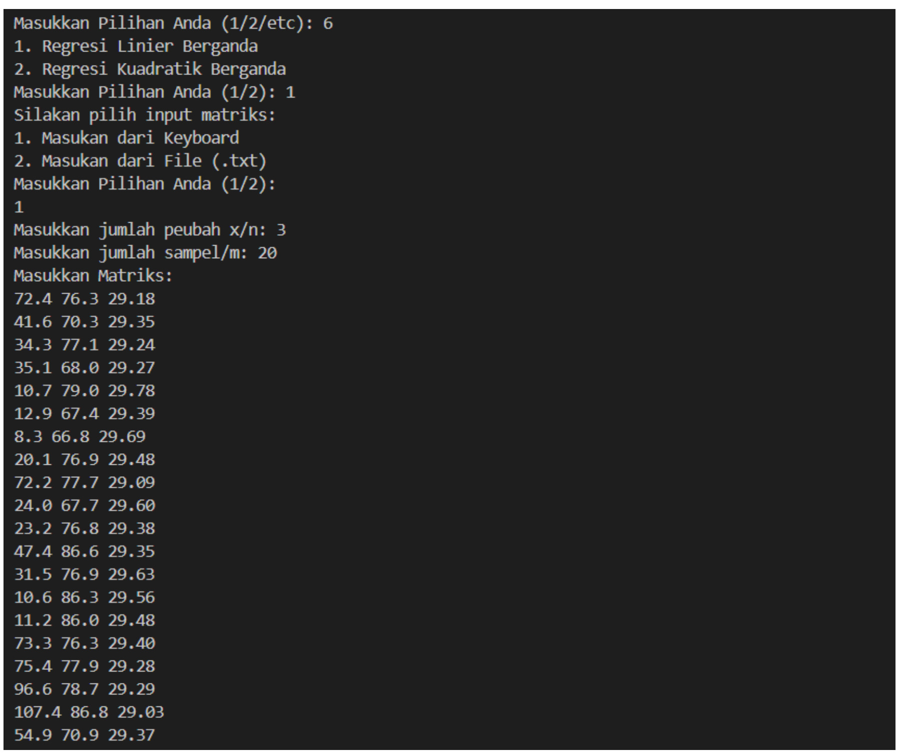
```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
0,5 0,5
```

Gambar 31. Isi File yang Dibaca

7) Solusi Regresi Linier dan Kuadratik Berganda

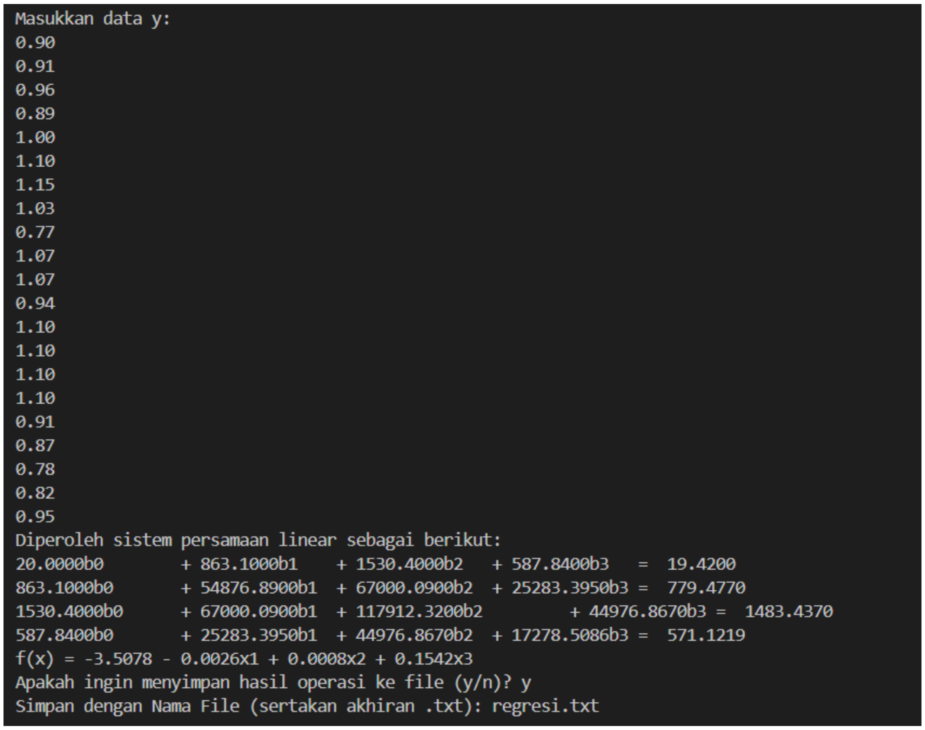
a) Regresi Linier Berganda

1) Input Keyboard (Manual) dan Simpan dalam File



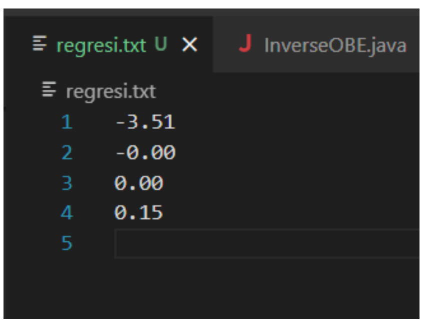
```
Masukkan Pilihan Anda (1/2/etc): 6
1. Regresi Linier Berganda
2. Regresi Kuadratik Berganda
Masukkan Pilihan Anda (1/2): 1
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2):
1
Masukkan jumlah peubah x/n: 3
Masukkan jumlah sampel/m: 20
Masukkan Matriks:
72.4 76.3 29.18
41.6 70.3 29.35
34.3 77.1 29.24
35.1 68.0 29.27
10.7 79.0 29.78
12.9 67.4 29.39
8.3 66.8 29.69
20.1 76.9 29.48
72.2 77.7 29.09
24.0 67.7 29.60
23.2 76.8 29.38
47.4 86.6 29.35
31.5 76.9 29.63
10.6 86.3 29.56
11.2 86.0 29.48
73.3 76.3 29.40
75.4 77.9 29.28
96.6 78.7 29.29
107.4 86.8 29.03
54.9 70.9 29.37
```

Gambar 32. Tampilan Pemilihan Menu Regresi Linier Berganda dan Input Manual.



```
Masukkan data y:
0.90
0.91
0.96
0.89
1.00
1.10
1.15
1.03
0.77
1.07
1.07
0.94
1.10
1.10
1.10
1.10
0.91
0.87
0.78
0.82
0.95
Diperoleh sistem persamaan linear sebagai berikut:
20.0000b0       + 863.1000b1    + 1530.4000b2    + 587.8400b3   =  19.4200
863.1000b0      + 54876.8900b1  + 67000.0900b2  + 25283.3950b3 =  779.4770
1530.4000b0     + 67000.0900b1  + 117912.3200b2         + 44976.8670b3 =  1483.4370
587.8400b0      + 25283.3950b1  + 44976.8670b2  + 17278.5086b3 =  571.1219
f(x) = -3.5078 - 0.0026x1 + 0.0008x2 + 0.1542x3
Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): regresi.txt
```

Gambar 33. Tampilan Pemilihan Menu Regresi Linier Berganda dan Input Manual.



```
regresi.txt U ×   InverseOBE.java
regresi.txt
1   -3.51
2   -0.00
3   0.00
4   0.15
5
```

Gambar 34. Isi File Hasil Perhitungan yang Disimpan

2) Input dari File (.txt) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 6
1. Regresi Linier Berganda
2. Regresi Kuadratik Berganda
Masukkan Pilihan Anda (1/2): 1
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): input.txt
Diperoleh sistem persamaan linear sebagai berikut:
20.0000b0       + 863.1000b1    + 1530.4000b2    + 587.8400b3    =  19.4200
863.1000b0      + 54876.8900b1  + 67000.0900b2   + 25283.3950b3 =  779.4770
1530.4000b0     + 67000.0900b1  + 117912.3200b2         + 44976.8670b3 =  1483.4370
587.8400b0      + 25283.3950b1  + 44976.8670b2   + 17278.5086b3 =  571.1219
f(x) = -3.5078 - 0.0026x1 + 0.0008x2 + 0.1542x3
Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 35. Tampilan Pemilihan Menu Regresi Linier Berganda dan Input dari File.

```
input.txt
You, 7 hours ago | 1 author (You)
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.8 29.03 0.82
54.9 70.9 29.37 0.95
```

Gambar 36. Isi File yang Dibaca

b) Regresi Kuadratik Berganda

1) Input Keyboard (Manual) dan Tidak Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 6
1. Regresi Linier Berganda
2. Regresi Kuadratik Berganda
Masukkan Pilihan Anda (1/2): 2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2):
1
Masukkan jumlah peubah x/n: 3
Masukkan jumlah sampel/m: 20
Masukkan Matriks:
72.4 76.3 29.18
41.6 70.3 29.35
34.3 77.1 29.24
35.1 68.0 29.27
10.7 79.0 29.78
12.9 67.4 29.39
8.3 66.8 29.69
20.1 76.9 29.48
72.2 77.7 29.09
24.0 67.7 29.60
23.2 76.8 29.38
47.4 86.6 29.35
31.5 76.9 29.63
10.6 86.3 29.56
11.2 86.0 29.48
73.3 76.3 29.40
75.4 77.9 29.28
96.6 78.7 29.29
107.4 86.8 29.03
54.9 70.9 29.37
```

Gambar 37. Tampilan Pemilihan Menu Regresi Kuadratik Berganda dan Input Manual.

```
Masukkan data y:
0.90
0.91
0.96
0.89
1.00
1.10
1.15
1.03
0.77
1.07
1.07
0.94
1.10
1.10
1.10
0.91
0.87
0.78
0.82
0.95
x1 = p
x2 = q
x3 = r
x4 = p^2
x5 = pq
x6 = pr
x7 = q^2
x8 = qr
x9 = r^2
f(x) = -859.9916 + 0.3062x1 - 0.2207x2 + 58.6211x3 - 0.0000x4 + 0.0001x5 - 0.0105x6 - 0.0002x7 + 0.0084x8 + 0.0000x9
Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

Gambar 38. Tampilan Pemilihan Menu Regresi Kuadratik Berganda dan Input Manual.

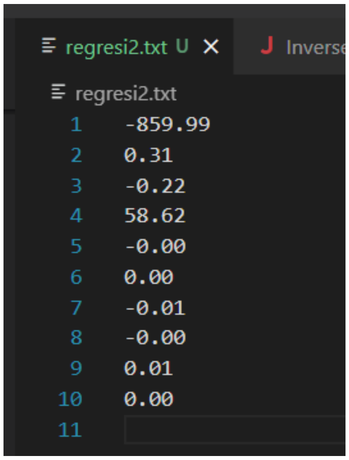
2) Input dari File (.txt) dan Simpan dalam File

```
Masukkan Pilihan Anda (1/2/etc): 6
1. Regresi Linier Berganda
2. Regresi Kuadratik Berganda
Masukkan Pilihan Anda (1/2): 2
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2):
2
Masukkan nama file (akhiran .txt): input.txt
x1 = p
x2 = q
x3 = r
x4 = p^2
x5 = pq
x6 = pr
x7 = q^2
x8 = qr
x9 = r^2
f(x) = -859.9916 + 0.3062x1 - 0.2207x2 + 58.6211x3 - 0.0000x4 + 0.0001x5 - 0.0105x6 - 0.0002x7 + 0.0084x8 + 0.0000x9
Apakah ingin menyimpan hasil operasi ke file (y/n)? y
Simpan dengan Nama File (sertakan akhiran .txt): regresi2.txt
```

Gambar 39. Tampilan Pemilihan Menu Regresi Kuadratik Berganda dan Input dari File.

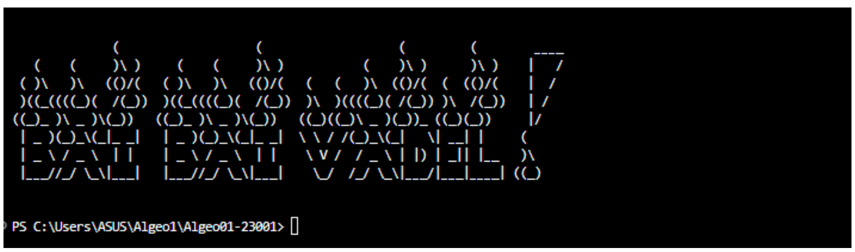
```
input.txt
You, 7 hours ago | 1 author (You)
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.8 29.03 0.82
54.9 70.9 29.37 0.95
```

Gambar 40. Isi File yang Dibaca



Gambar 41. Isi File Hasil Perhitungan yang Disimpan

8) Tampilan Akhir



Gambar 42. Tampilan Akhir

4. 2 Test Case

Beberapa test case diberikan pada file spesifikasi Tugas Besar 1.

1. **Interpolasi *Bicubic Spline***

$$\begin{pmatrix} 21 & 98 & 125 & 153 \\ 51 & 101 & 161 & 59 \\ 0 & 42 & 72 & 210 \\ 16 & 12 & 81 & 96 \end{pmatrix}$$

Tentukan nilai:

$f(0, 0)$ = ?

$f(0.5, 0.5)$ = ?

$f(0.25, 0.75)$ = ?

$f(0.1, 0.9)$ = ?

Maka, hasil dari program:

- f(0,0) = 21,00

```
Masukkan Pilihan Anda (1/2/etc): 5
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 1
Masukkan Matriks:
21 98 125 153
51 101 161 59
0 42 72 210
16 12 81 96
Masukkan x: 0
Masukkan y: 0
Hasil Binterpolasi Bicubic: 21,00

Apakah ingin menyimpan hasil operasi ke file (y/n)? n
```

- f(0,5 , 0,5) = 87,80

```
Masukkan Pilihan Anda (1/2/etc): 5
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Silakan masukan file text (.txt): test5.1.txt
Hasil Binterpolasi Bicubic: 87,80
```

- f(0,25 , 0,75) = 117,73

```
Masukkan Pilihan Anda (1/2/etc): 5
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Silakan masukan file text (.txt): test5.1.txt
Hasil Binterpolasi Bicubic: 117,73
```

- f(0,1 , 0,9) = 128,58

```
Masukkan Pilihan Anda (1/2/etc): 5
Silakan pilih input matriks:
1. Masukan dari Keyboard
2. Masukan dari File (.txt)
Masukkan Pilihan Anda (1/2): 2
Silakan masukan file text (.txt): test5.1.txt
Hasil Binterpolasi Bicubic: 128,58
```

BAB V

PENUTUP

5.1 Kesimpulan

Matriks adalah susunan bilangan-bilangan yang diatur dalam bentuk persegi panjang berdasarkan baris dan kolom. Matriks sering sekali dipakai untuk berbagai perhitungan dalam matematik (sistem persamaan linear) dan diterapkan di kehidupan sehari-hari. Pada matriks dapat dilakukan berbagai operasi, seperti perhitungan solusi sistem persamaan linear (SPL), determinan, invers matriks, interpolasi polinomial, interpolasi *bicubic spline*, dan regresi linear berganda.

Java merupakan salah satu bahasa pemrograman yang populer dan banyak digunakan. Dengan paradigma berbasis objek, program dapat dirancang dengan modularitas yang tinggi dan memungkinkan pengguna untuk mengelompokkan kode ke dalam kelas dan objek. Hal ini akan memudahkan untuk melakukan pengorganisasian fungsi-fungsi yang berbeda.

Persoalan perhitungan matriks dapat diselesaikan dengan penyusunan program java. Tidak hanya itu, fungsi yang diterapkan untuk menyelesaikan tiap persoalannya juga beragam. Seperti dalam penyelesaian sistem persamaan linear, dapat dipilih langkah penyelesaian dengan metode Gauss-Jordan atau kaidah Cramer.

Dari konsep perhitungan tersebut, maka dapat dibuat suatu program untuk menaksir suatu nilai dari beberapa data yang diberikan yang disebut interpolasi polinomial. Interpolasi *bicubic spline* digunakan untuk menghasilkan fungsi interpolasi yang halus dan kontinu melalui titik yang diberikan. Terakhir, regresi linear berganda dapat digunakan untuk menganalisis hubungan antar variabel.

5.2 Saran

Berdasarkan program yang telah disusun, maka disampaikan saran agar diberikan jawaban dari tes studi kasus pada dokumen spesifikasi agar tim dapat mengetahui kesesuaian jawaban (*output*) yang dimiliki dari program yang telah dirancang.

5.3 Komentar dan Refleksi

Dalam pengerjaan Tugas Besar 1 mengenai matriks dengan penerapan metode dan kaidah yang telah dijelaskan selama kelas, hal ini membantu tim penulis untuk belajar kembali dan memanfaatkan secara nyata pengetahuan yang telah diterima dengan menyusunnya menjadi suatu algoritma pemrograman. Tim penulis berhasil untuk menyelesaikan spesifikasi wajib yang diberikan dan memiliki kekurangan waktu untuk mengerjakan bonus yang ada. Hal ini tidak lepas dari padatnya tugas dan ujian yang dilakukan selama periode pengerjaan Tugas Besar. Tim penulis menyadari akan kekurangan dalam manajemen waktu yang baik dan seharusnya bisa lebih maksimal bila dikerjakan lebih awal.

LAMPIRAN

GeeksforGeeks.2023."Determinant of a Matrix". .https://www.geeksforgeeks.org/determinant-of-a-matrix/. Diakses pada 12 Oktober 2024

University, The Pennsylvania State. 2024. *A Matrix Formulation of The Multiple Regression Model*. https://online.stat.psu.edu/stat501/lesson/5/5.4. Diakses pada 18 Oktober 2024

Tutors, Varsity. 2024. *Quadratic Regression*. https://www.varsitytutors.com/hotmath/hotmath_help/topics/quadratic-regression. Diakses pada 19 Oktober 2024

Tautan Repository : https://github.com/mineraleee/Algeo01-23001.git
Tautan Video : https://youtu.be/VZfobE-lOo4?si=PDFCgIIGTVY15t08