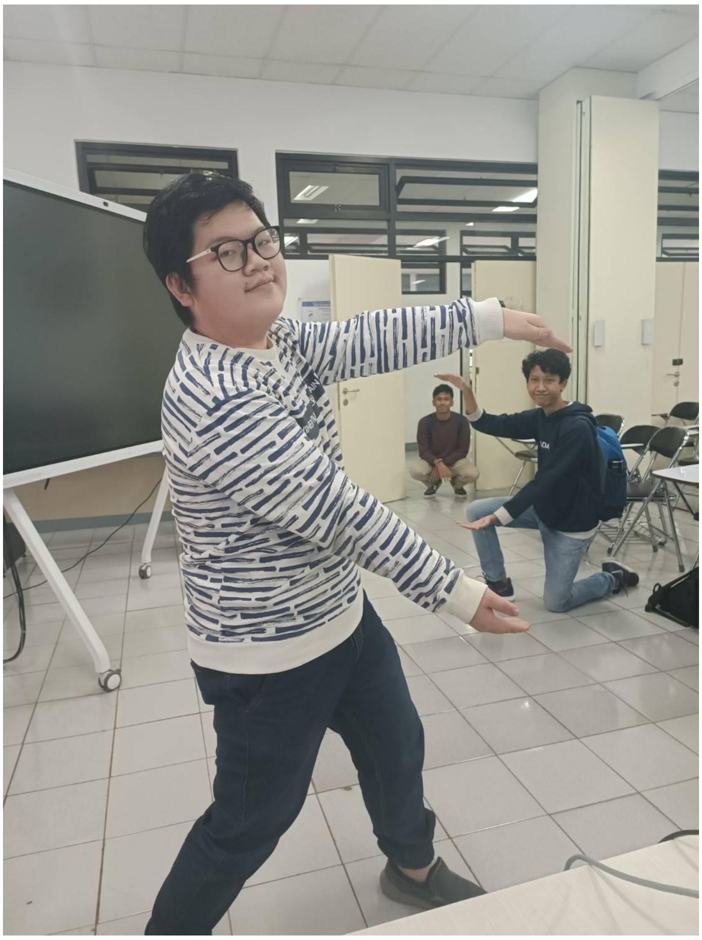
LAPORAN TUGAS BESAR

IF2123 ALJABAR LINIER DAN GEOMETRI



Go Dillon Audris	13521062
Salomo Reinhart Gregory Manalu	13521063
M. Dimas Sakti Widyaatmaja	13521160

Institut Teknologi Bandung

2022

DAFTAR ISI

BAB 1

Deskripsi Masalah

1. Interpolasi Polinom

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. 1 Setelah polinom interpolasi pn(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, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2 , …, a_n ,

$a_0 + a_1x_0 + a_2x_0^2 + ... + a_n.x_0^n = y_n.$

$a_0 + a_1x_1 + a_2x_1^2 + ... + a_n.x_1^n = y_1.$

...

...

$a_0 + a_1x_n + a_2x_n^2 + ... + a_n.x_n^n = y_n.$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0, a_1, a_2 , …, a_n, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah dipelajari. 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 2 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.

2. Bicubic Interpolation

Bicubic interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan dalam pembesaran citra yang merupakan pengembangan dari interpolasi linear dan cubic yang telah dipelajari pada kuliah metode numerik di aljabar geometri. Diberikan sebuah matriks awal, misal M, kita akan mencari persamaan interpolasi f(*x,y*) dengan pemodelan sebagai berikut: Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(*x,y*) akan menghasilkan sebuah matriks persamaan: Elemen pada matrix X adalah koefisien aij yang diperoleh dari persamaan f(*x,y*) di atas. Sebagai contoh, elemen pada baris 4 kolom ke 10 adalah koefisien dari a_{12} dan diperoleh dari 21*(-1)2 = 2, sesuai persamaan x_i * y_j . 3 Vektor *a* dapat dicari dari persamaan tersebut (menggunakan inverse), lalu vektor *a* digunakan sebagai nilai variabel dalam f(*x,y*). Sehingga terbentuk fungsi interpolasi bicubic sesuai model. Tugas Anda adalah menentukan persamaan f(*x,y*) lalu melakukan interpolasi berdasarkan f(*a*,*b*) dari masukan matriks 4 x 4. Nilai masukan a dan b dalam rentang [0,1]. Untuk studi kasus ini, buatlah matriks X menggunakan rumus yang ada (tidak hardcode) serta carilah invers matriks X dengan library kalian dalam penyelesaian masalah.

3. Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu. Untuk mendapatkan nilai dari setiap βi dapat digunakan Normal Estimation Equation for Multiple Linear Regression sebagai berikut: Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB 2

1. Metode Eliminasi Gauss

Metode ini dapat dimanfaatkan untuk memecahkan sistem persamaan linear dengan merepresentasikan (mengubah) menjadi bentuk matriks, matriks tersebut lalu diubah ke bentuk **Eselon Baris** melalui Operasi Baris Elementer. Kemudian sistem diselesaikan dengan substitusi balik.

2. Metode Eliminasi Gauss-Jordan

Metode ini merupakan pengembangan dari metode eliminasi Gauss. Operasi baris elementer (OBE) diterapkan pada matriks *augmented* sehingga menghasilkan matriks eselon baris tereduksi. Hasil OBE tidak perlu lagi disubstitusi secara mundur untuk memperoleh nilai-nilai variabel. Nilai variabel langsung diperoleh dari matriks augmented akhir.

3. Kaidah Cramer

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

$$x_i = \frac{\det(A_i)}{\det(A)} \qquad i = 1, \ldots, n$$

yang dalam hal ini, di mana A_i adalah matriks yang dibentuk dengan mengganti kolom ke-i dari A dengan vektor kolom b.

4. Matriks Balikan

Misalkan matriks A merupakan matriks persegi (ukuran n x n). Maka, invers matriks A adalah A^{-1}, sehingga $AA^{-1} = I$. Kita dapat menggunakan eliminasi Gauss-Jordan untuk mendapatkan matriks balikan dari A.

[A | I] —-(Gauss-Jordan)---> [I | A^{-1}]

Jika matriks A memiliki balikan, maka solusi pada Ax = b adalah unik. Sebaliknya, jika matriks A tidak memiliki balikan, solusi pada Ax = b tidak unik. Pada kasus lain, yaitu Ax = 0, jika matriks A memiliki balikan, maka SPL hanya memiliki solusi trivial. Jika tidak memiliki balikan, maka SPL memiliki solusi nontrivial.

Untuk mendapatkan penyelesaian SPL (x) dengan matriks balikan, caranya adalah x = A^{-1} **b**

5. Matriks Kofaktor

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 C_{ij}.

$C_{ij} = (-1)^{i+j}.M_{ij}$

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 (letak) kofaktor-kofaktornya.

6. Matriks Adjoin

Adjoin matriks merupakan transpose dari matriks kofaktor. Adjoin sering disingkat dengan Adj. Misalkan matriks A, maka adjoin A ditulis Adj(A). Transpose sendiri maksudnya adalah pertukaran elemen pada baris menjadi kolom atau kolom menjadi baris. Adjoin matriks digunakan dalam menentukan invers matriks.

7. Determinan

Determinan pada sebuah matriks merupakan selisih antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder, dengan syarat matriksnya merupakan matriks persegi. Untuk mendapatkan nilai determinan dari suatu matriks, ada beberapa cara yang dapat dilakukan :

a. Untuk matriks 2 x 2

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

maka det(A) = $a_{11}a_{22} - a_{12}a_{21}$

b. Untuk matriks 3 x 3

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \left[\begin{array}{ccc|cc} a_{11} & a_{12} & a_{13} & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} & a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} & a_{31} & a_{32} \end{array}\right]$$

maka,

det(A) = $(a_{11}a_{22}a_{33} + a_{12}a_{21}a_{31} + a_{13}a_{21}a_{32}) - (a_{13}a_{22}a_{31} + a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33})$

c. Cara segitiga

Kita dapat mengubah suatu matriks dengan OBE sehingga setengah dari matriks tersebut (segitiga) adalah 0. Nilai determinan dari matriks tersebut adalah hasil perkalian semua elemen pada diagonal utama.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

det(A) = $a_{11}a_{22}\ a_{33}a_{44}$

d. Kofaktor

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.

Diberikan n+1 buah titik berbeda yaitu (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.

Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di x = a, yaitu y = $p_n(a)$. Jika $x_0 < a < x_n$ maka $y_k = p(x_k)$ disebut nilai interpolasi (*interpolated value*). Jika x_0 < xk atau $x_0 < x_n$ maka $y_k = p(x_k)$ disebut nilai ekstrapolasi (*extrapolated value*)

9. Interpolasi Bikubik

Interpolasi bicubic merupakan sebuah metode interpolasi yang menggunakan 16 pixel dalam pixel 4x4 tetangga terdekat pada citra aslinya. Dengan menggunakan metode interpolasi bicubic ini dapat membuat tepi-tepi citra hasil lebih halus. Sehingga metode interpolasi bicubic sering digunakan dalam pengeditan perangkat lunak dan banyak kamera digital lainnya.

10. Regresi Linier Berganda

Regresi linear berganda merupakan model regresi yang melibatkan lebih dari satu variabel independen. Analisis regresi linear berganda dilakukan untuk mengetahui arah dan seberapa besar pengaruh variabel independen terhadap variabel dependen (Ghozali, 2018).

Untuk mendapatkan nilai dari setiap βi dapat digunakan Normal Estimation Equation for Multiple Linear Regression. Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB 3

1. Class

a. Cramer

Class : Mencari solusi SPL sebuah matrix dengan kaidah Cramer

i. Methods :

1. int counter(double[][] matrix) :
   Memberi keluaran banyak elemen pada matrix.
2. double[][] changeColXWithHasil(double[][] matrix, double[][] hasil, int ColX) :
   Mengganti kolom ke x dari matrix A dengan matrix hasil (b).
3. Object[] cramerElimination(matrix[][]) :
   Menghasilkan solusi persamaan linier dengan kaidah Cramer.

b. DeterminanKofaktor

Class : Mencari determinan matrix dengan metode ekspansi kofaktor

i. Methods :

1. double determinanKofaktor (double[][] m) :
   Memberi keluaran nilai determinan matrix m dengan cara ekspansi kofaktor.

c. DeterminanReduksiBaris

Class : Mencari determinan matrix dengan metode gauss

i. Methods :

1. double[][] copyMatrix(double[][] m) :
   Memberikan keluaran sebuah matrix yang merupakan hasil copy dari matrix m.
2. double[][] swapRow(double[][] m, int r1, int r2) :
   Menukar baris r1 dan r2.
3. boolean isRowZero(double[][] m, int r) :

Mengecek apakah semua elemen dalam sebuah baris adalah 0.

4. double determinanReduksiBaris(double[][] m) :

   Mengembalikan nilai determinan sebuah matrix dengan gauss.

d. Gauss

Class : Mencari solusi SPL sebuah matrix dengan metode gauss

i. Methods :

1. double[][] swapRow(double[][] m, int baris1, int baris2) :

   Menukar baris baris1 dengan baris baris2.

2. void addRowWithMultipleOfRow(double[][] m, int baris1, int baris2, double constant) :

   Elemen m pada baris berindeks baris1 akan ditambah dengan constant*elemen pada baris berindeks baris2.

3. void divideRowWithConstant(double[][] m, int barisX, double constant) :

   Elemen m pada baris berindeks barisX akan dibagi dengan constant.

4. boolean isRowZero(double[][] m, int barisX) :

   Mengecek apakah semua elemen dalam barisX adalah 0.

5. boolean isRowAnException(double[][] m, int barisX) : Menghasilkan true jika barisX adalah suatu baris dengan semua elemen 0 kecuali pada.

6. int indexOfPivotRow(double[][] m, int barisX, int kolomX) : Menghasilkan indeks suatu pivot row, dimana pivot row ditentukan dengan membandingkan elemen pada kolomX untuk baris mulai dari barisX hingga baris terakhir m.

7. int arrangeMatrix(double[][] m) :

   Seluruh baris Zero dan Exception berada di bagian bawah matriks, juga mengembalikan banyak baris yang bukan Zero.

8. void gaussElimination(double[][] m, boolean interpol) :

   Mengubah matrix m menjadi matrix eselon baris.

9. Object[] gaussEliminationSolution(double[][] matrix) : Menghasilkan solusi SPL dari matrix eselon baris.

e. GaussJordan

Class : Mencari solusi SPL sebuah matrix dengan kaidah Gauss-Jordan

i. Methods :

1. int countRowToWorked(double[][] m) :

   Menghasilkan jumlah baris yang bukan baris zero dalam matrix m.
2. int indexOfLeadingOne(double[][] m, int barisX) :

   Mengembalikan nilai indeks dari leading one disuatu baris
3. boolean isThereLeadingOne(double[][] matrix, int kolomX) : Mengecek apakah pada kolomX suatu matrix ada leadingOne.
4. void gaussJordanElimination(double[][] m) :

   Mengubah matrix m menjadi matrix eselon baris tereduksi.
5. Object[] gaussEliminationSolution(double[][] matrix) : Menghasilkan solusi SPL dari matrix eselon baris tereduksi.

f. InterpolasiBikubik

Class : Mencari interpolasi bikubik

i. Methods :

1. double[][] makeMatrixCoefficient() :

   Menghasilkan suatu matriks koefisien (16 x 16) yang dibutuhkan untuk menyelesaikan interpolasi bikubik dengan input matriks 4 x 4.
2. double[][] changeMatrixDimension(double[][] m) :

   Menghasilkan matriks berukuran 16 x 1 dari matriks 4 x 4.
3. double[][] integrateMatrixAandB(double[][] mA, double mB) : Menghasilkan matriks augmented 16 x 17 hasil penggabungan matriks koefisien dengan matriks nilai
4. double[] interpolationByBicubic(double[][] m, double a, double b): Menghasilkan suatu list a, yang merupakan koefisien dari suku-suku persamaaan interpolasi bikubik.

g. InterpolasiPolinom

Class : Mencari interpolasi polinom dari beberapa titik.

i. Methods:

1. int inputJumlahTitik():
   Membaca input jumlah titik.
2. double inputNilaiTaksiranX():
   Membaca input nilai x yang ingin ditaksir.
3. double[][] makePersamaanMatrix(double[][] matrix):
   Membuat matriks yang setiap barisnya adalah polinom.
4. Object[] interpolasiPolinom (double[][] titik, int jumlahTitik):
   Menghitung hasil interpolasi polinom

h. InversAdjoint

Class : Menghasilkan invers dari matriks m dengan metode adjoint

i. Methods:

1. double[][] transposeMatrix (double[][] m):
   Menghasilkan matriks transpose dari m, dimana mTranspose[i][j] = m[j][i]
2. void multiplyMatrixByConst (double[][] m, double constant):
   Prosedur Kali Matriks dengan Konstanta
3. double[][] makeMatrixCofactor (double[][] m):
   Membuat suatu matriks kofaktor dari matriks m
4. double[][] inversByAdjoint (double[][] m):
   Menghasilkan invers dari matriks m dengan metode adjoint, Jika matriks tidak memiliki invers, mengembalikan suatu m yang elemennya 0 semua.

i. InversReduksiBaris

Class : Menghasilkan invers dari matriks m dengan cara reduksi baris

i. Methods:

1. double[][] makeIdentityMatrix (int n):
   Menghasilkan matriks identitas berukuran n x n.

2. double[][] copyMatrix(double[][] m):

   Mengeluarkan salinan m dengan dimensi dan elemen yang sama.

3. double[][] inversByReduksiBaris (double[][] m):

   Menghasilkan invers dari matriks m dengan cara reduksi baris. Jika matriks tidak memiliki invers, maka mengembalikan matriks yang elemennya 0 semua.

j. MainApp

Class : Algoritma Utama

i. Method:

1. void subMenuSPL() :

   Merupakan submenu untuk penyelesaian SPL dengan 4 metode

2. void solveSPLWithGauss() :

   Merupakan kode untuk menyelesaikan SPL dengan eliminasi Gauss

3. void solveSPLWithGaussJordan() :

   Merupakan kode untuk menyelesaikan SPL dengan eliminasi Gauss-Jordan

4. void solveSPLWithMetodeInveres() :

   Merupakan kode untuk menyelesaikan SPL dengan metode invers

5. void solveSPLWithCramer() :

   Merupakan kode untuk menyelesaikan SPL dengan kaidah cramer

6. void subMenuDeterminan() :

   Merupakan submenu untuk mendapatkan determinan suatu matriks dengan 2 metode

7. void solveDeterminanWithReduksiBaris() :

   Merupakan kode untuk menentukan determinan matriks dengan metode reduksi baris

8. void solveDeterminanWithEkspansiKofaktor() :

Merupakan kode untuk menentukan determinan matriks dengan metode ekspansi kofaktor

9. void subMenuInvers() :

   Merupakan submenu untuk mendapatkan invers suatu matriks dengan 2 metode

10. void solveInversWithReduksiBaris() :

    Merupakan kode untuk menyelesaikan invers dengan metode reduksi baris

11. void solveInversWithAdjoint() :

    Merupakan kode untuk menyelesaikan matriks dengan metode adjoint matriks

12. void subMenuInterpolasiPolinom() :

    Merupakan kode untuk menyelesaikan permasalahan terkait interpolasi polinom

13. void subMenuInterpolasiBikubik() :

    Merupakan kode untuk menyelesaikan permasalahan terkait interpolasi bikubik

14. void main(String[] args) :

    Merupakan program utama untuk mengatasi berbagai persoalan terkait matriks

k. MatriksBalikan

Class : Mencari solusi SPL sebuah matrix dengan metode matriks balikan

i. Methods:

1. double[][] splitHasil(double[][] matrix):

   Memisahkan matriks augmented dan menghasilkan matriks hasil

2. double[][] splitMainMatrix(double[][] matrix):

   Memisahkan matriks augmented dan menghasilkan matriks yang memiliki persamaan

3. double[][] multiplyMatrixbyMatrix(double[][] m1, double[][] m2):

   Memberi keluaran matriks hasil perkalian matriks m1 dan m2

4. Object[] inversElimination(double[][] matrix):
   Menghasilkan solusi SPL dengan metode matriks balikan

l. Matrix

Class : Berisi primitif-primitif untuk mengerjakan fungsi-fungsi lainnya

i. Attributes:
   1. class RowCol: Merepresentasikan pasangan jumlah baris dan kolom suatu matriks.

ii. Methods:
   1. int getnRows(double[][] m) :
      Memberi keluaran banyak baris pada matriks m.
   2. int getnCols(double[][] m) :
      Memberi keluaran banyak kolompada matriks m.
   3. int getLastIdxRows(double[][] m):
      Memberi keluaran indeks baris terakhir matriks m.
   4. int getLastIdxCols(double[][] m):
      Memberi keluaran indeks kolomterakhir matriks m.
   5. void printSolusi(Object[] hasilx):
      Menulis solusi SPL ke layar atau file.
   6. void readMatrixFromKeyboard(double[][] m):
      Membaca masukan elemen matriks dari keyboard.
   7. void printMatrixToScreen(double[][] m):
      Mencetak matriks m ke layar.
   8. double[] colRowNumbersFromFile(RowCol rc, String pathname):
      Mencari tahu jumlah row dan column matriks dalam file.
   9. void readMatrixFromFile(double[][] m, String pathname):
      Membaca input matriks dari file.
   10. void writeMatrixToFile(double[][] m, String pathname):
       Menlis matriks m ke file.
   11. boolean isMatrixEqual(double[][] m1, double[][] m2:

Menghasilkan true jika ukuran m1 sama dengan m2 dan seluruh elemen pada indeks yang sama bernilai sama.

12. double[] readFileForInterpolasiBikubik(double[][] m, String pathname):
    Membaca input dari file untuk permasalahan interpolasi bikubik.
13. void writeGeneralStringToFile(String out, String pathname):
    Menulis output ke file untuk suatu string.

m. RegresiLinierBerganda

Class: Berisi algoritma untuk melakukan regresi linear berganda

i. Attributes: Tidak ada attribute pada class ini.
ii. Methods: Terdapat satu method untuk menulis taksiran fungsi hasil regresi.
    1. void regresiLinierBerganda() :
       menulis taksiran fungsi hasil regresi ke layar atau ke file.

2. Garis Besar Program

Saat mulai menjalankan program *Main.java*, user akan diberi prompt untuk memilih menu aplikasi yang ingin dijalankan oleh pengguna. Aplikasi-aplikasi tersebut antara lain:

1. Penyelesaian Sistem Persamaan Linear
2. Determinan Matriks
3. Balikan Matriks
4. Interpolasi Polinom
5. Interpolasi Bikubik
6. Regresi Linear Berganda
0. Keluar

Pengguna harus memasukkan kode angka dari menu yang ingin digunakan. Jika pengguna memberi input yang salah, pengguna akan diminta lagi untuk menginput dengan kode angka yang benar.

Jika pengguna memilih 1, pengguna akan diminta untuk memilih metode yang akan digunakan untuk menyelesaikan sistem persamaan linear. Ada 4 pilihan yang diberikan : 1. Metode Eliminasi Gauss, 2. Metode Eliminasi Gauss-Jordan, 3. Metode Matriks Balikan, dan 4. Kaidah Cramer. Misalkan pengguna memilih nomor 1, selanjutnya pengguna akan diminta untuk memilih cara menginput matriks yang akan dicari solusinya. Pilihannya adalah 1. Input melalui keyboard dan 2. Input melalui file. Bila memilih nomor 1, maka pengguna akan menginput matriks dengan cara mengetikkan di keyboard. Bila memilih nomor 2, pengguna akan diminta untuk mengetikkan nama file yang berisi matriks yang akan dicari solusinya. Setelah itu, solusi akan diberikan, lalu pengguna akan ditanya apakah output ingin disimpan ke file. Jika iya, pengguna akan dimintai nama file tempat penyimpanan hasil output. Apabila inputan saat memilih pilihan-pilihan salah, maka pengguna akan diminta untuk memasukkan angka yang benar. Hal ini juga berlaku untuk 3 metode lainnya.

Jika pengguna memilih 2, pengguna akan diminta untuk memilih metode yang akan digunakan untuk mencari determinan dari suatu matriks. Ada 2 pilihan yang diberikan : 1. Metode Reduksi Baris dan 2. Metode Ekspansi Kofaktor. Setelah memilih salah satu pilihan, pengguna akan diminta untuk memilih cara menginput matriks yang akan dicari solusinya. Pilihannya adalah 1. Input melalui keyboard dan 2. Input melalui file. Bila memilih nomor 1, maka pengguna akan menginput matriks dengan cara mengetikkan di keyboard. Bila memilih nomor 2, pengguna akan diminta untuk mengetikkan nama file yang berisi matriks yang akan dicari determinannya. Setelah itu, determinan akan diberikan, lalu pengguna akan ditanya apakah output ingin disimpan ke file. Jika iya, pengguna akan dimintai nama file tempat penyimpanan hasil output. Apabila inputan saat memilih pilihan-pilihan salah, maka pengguna akan diminta untuk memasukkan angka yang benar.

Jika pengguna memilih 3, pengguna akan diminta untuk memilih metode yang akan digunakan untuk mencari invers dari suatu matriks. Ada 2 pilihan yang diberikan : 1. Metode Reduksi Baris dan 2. Metode Adjoint. Setelah memilih salah satu pilihan, pengguna akan diminta untuk memilih cara menginput matriks yang akan dicari solusinya. Pilihannya adalah 1. Input melalui keyboard dan 2. Input melalui file. Bila memilih nomor 1, maka pengguna akan menginput matriks dengan cara mengetikkan di keyboard. Bila memilih nomor 2, pengguna akan diminta untuk mengetikkan nama file yang berisi matriks yang akan dicari inversnya. Setelah itu, invers akan diberikan, lalu pengguna akan ditanya apakah output ingin disimpan ke file. Jika iya, pengguna akan dimintai nama file tempat penyimpanan hasil output. Apabila inputan saat memilih pilihan-pilihan salah, maka pengguna akan diminta untuk memasukkan angka yang benar.

Jika pengguna memilih 4, pengguna artinya ingin menyelesaikan permasalahan terkait interpolasi polinom. Pertama-tama user akan dimintai tipe input yang diinginkan, lewat keyboard dengan kode nomor 1 atau lewat file dengan kode nomor 2, dan 0 untuk kembali ke menu utama. Jika input salah pengguna akan diminta untuk menginput ulang kode angka. Jika pengguna memilih input dengan file, pengguna akan diminta untuk menginput nama file yang sudah ditaruh di folder test. Jika file tidak ada, pengguna akan diberi tahu bahwa terjadi error dan akan dikembalikan ke pilihan di submenu. Jika pengguna memilih input via keyboard, pengguna akan diminta untuk memasukkan elemen-elemen matriks. Setelah itu, program akan menampilkan output hasil interpolasi polinom ke layar, lalu pengguna akan ditanya apakah output ingin disimpan ke file. Jika iya, pengguna akan dimintai nama file tempat penyimpanan hasil output. Program lalu akan kembali ke bagian awal sub menu.

Jika pengguna memilih 5, pengguna artinya ingin menyelesaikan permasalahan terkait interpolasi bikubik. Pertama-tama user akan dimintai tipe input yang diinginkan, lewat keyboard dengan kode nomor 1 atau lewat file dengan kode nomor 2, dan 0 untuk kembali ke menu utama. Jika input salah pengguna akan diminta untuk menginput ulang kode angka. Jika pengguna memilih input dengan file, pengguna akan diminta untuk menginput nama file yang sudah ditaruh di folder test. Jika file tidak ada, pengguna akan diberi tahu bahwa terjadi error dan akan dikembalikan ke pilihan di submenu. Jika pengguna memilih input via keyboard, pengguna akan diminta untuk memasukkan nilai x dan y. Jika input tidak valid, pengguna akan diminta untuk memasukkan ulang input. Setelah itu, program akan menampilkan output hasil interpolasi bikubik ke layar, lalu pengguna akan ditanya apakah output ingin disimpan ke file. Jika iya, pengguna akan dimintai nama file tempat penyimpanan hasil output. Program lalu akan kembali ke bagian awal sub menu.

Jika pengguna memilih 6, pengguna artinya ingin menyelesaikan permasalahan terkait regresi linier berganda. Pertama-tama user akan dimintai tipe input yang diinginkan, lewat keyboard dengan kode nomor 1 atau lewat file dengan kode nomor 2, dan 0 untuk kembali ke menu utama. Jika input salah pengguna akan diminta untuk menginput ulang kode angka. Jika pengguna memilih input dengan file, pengguna akan diminta untuk menginput nama file yang sudah ditaruh di folder test. Jika file tidak ada, pengguna akan diberi tahu bahwa terjadi error dan akan dikembalikan ke pilihan di submenu. Jika pengguna memilih input via keyboard, pengguna akan diminta untuk elemen-elemen matriks XY terlebih dahulu, setelah itu, pengguna akan diminta untuk menginput nilai x_k yang ingin ditaksir. Jika input tidak valid, pengguna akan diminta untuk memasukkan ulang input. Setelah itu, program akan menampilkan output hasil regresi linier berganda ke layar, lalu pengguna akan ditanya apakah output ingin disimpan ke file. Jika iya, pengguna akan dimintai nama file tempat penyimpanan hasil output. Program lalu akan kembali ke bagian awal sub menu.

Yang terakhir, jika pengguna memilih 0, program akan berakhir.

BAB 4

1. Solusi SPL Ax = b

   a. File Studi Kasus:

```
test > testCase1a.txt
1   1 1 -1 -1 1
2   2 5 -7 -5 -2
3   2 -1 1 3 4
4   5 2 -4 2 6
```

   Hasil:

```
Masukkan nama dari file : testCase1a.txt

Matriks tidak memiliki penyelesaian (Jika menggunakan eliminasi Gauss atau Gauss-Jordan)
Atau tidak dapat diselesaikan dengan metode yang digunakan (Jika menggunakan metode Invers atau Kaidah Cramer)

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase1a.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

   Analisis:

   Jadi setelah dicari solusi dari matriks Ax = B menggunakan eliminasi Gauss atau Gauss-Jordan, matriks tersebut tidak memiliki penyelesaian. Hal ini dikarenakan adanya baris exception atau baris yang berbentuk

   0 0 0 .. 0 0 0 x (Dengan x tidak sama dengan 0)

   b. File studi Kasus :

```
test > testCase1b.txt
1   1 -1 0 0 1 3
2   1 1 0 -3 0 6
3   2 -1 0 1 -1 5
4   -1 2 0 -2 -1 -1
```

   Hasil :

```
Masukkan nama dari file : testCase1b.txt

Solusi SPL adalah :
Nilai x1 adalah 3.0 + (a)
Nilai x2 adalah 0.0 + 2.0*(a)
Nilai x3 adalah b
Nilai x4 adalah -1.0 + (a)
Nilai x5 adalah a

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase1b.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Analisis:

Jadi setelah dicari solusi dari matriks Ax = B menggunakan eliminasi Gauss atau Gauss-Jordan. Masing-masing nilai variabel yang didapat antara lain:

$x_1 = 3 + a$,

$x_2 = 2a$,

$x_3 = b$,

$x_4 = -1 + a$,

$x_5 = a$

c. File studi Kasus :

```
test > testCase1c.txt
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1
```

Hasil :

```
Masukkan nama dari file : testCase1c.txt

Solusi SPL adalah :
Nilai x1 adalah a
Nilai x2 adalah 1.0 - (c)
Nilai x3 adalah b
Nilai x4 adalah -2.0 - (c)
Nilai x5 adalah 1.0 + (c)
Nilai x6 adalah c

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase1c.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Jadi setelah dicari solusi dari matriks Ax = B menggunakan eliminasi Gauss atau Gauss-Jordan. Masing-masing nilai variabel yang didapat antara lain:

$x_1 = a$

$x_2 = 1 - c$

$x_3 = b$

$x_4 = -2 - c$

$x_5 = 1 + c$

$x_6 = c$

d. File studi Kasus :

```
test > testCase1d6.txt
1 0.5 0.333 0.25 0.2 0.167 1
0.5 0.333 0.25 0.2 0.167 0.143 0
0.333 0.25 0.2 0.167 0.143 0.125 0
0.25 0.2 0.167 0.143 0.125 0.111 0
0.2 0.167 0.143 0.125 0.111 0.1 0
0.167 0.143 0.125 0.111 0.1 0.091 0
```

Hasil :

```
Masukkan nama dari file : testCase1d6.txt

Solusi SPL adalah :
Nilai x1 adalah 8.108597803067992
Nilai x2 adalah -23.651100350012037
Nilai x3 adalah -31.672058910096013
Nilai x4 adalah 108.002005556
Nilai x5 adalah -43.693012
Nilai x6 adalah -17.953

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase1d6.txt
File berhasil dibuat: testCase1d6.txt
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Jadi setelah dicari solusi dari matriks Ax = B menggunakan eliminasi Gauss atau Gauss-Jordan. Masing-masing nilai variabel yang didapat antara lain:

$x_1 = 8.11$

$x_2 = -23.65$

$x_3 = -31.67$

$x_4 = 108.00$

$x_5 = -43.69$

$x_6 = -17.95$

e. File studi Kasus :

```
test > testCase1d10.txt
1 0.5 0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 1
0.5 0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0
0.333 0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0
0.25 0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0
0.2 0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0
0.167 0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0
0.143 0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.0625 0
0.125 0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.0625 0.059 0
0.111 0.1 0.091 0.083 0.077 0.071 0.067 0.0625 0.059 0.056 0
0.1 0.091 0.083 0.077 0.071 0.067 0.0625 0.059 0.056 0.053 0
```

Hasil :

```
Masukkan nama dari file : testCase1d10.txt

Solusi SPL adalah :
Nilai x1 adalah 2.0187463570458597
Nilai x2 adalah 35.180707645231394
Nilai x3 adalah -133.7793788443629
Nilai x4 adalah 96.098657208149
Nilai x5 adalah 6.3424057911380665
Nilai x6 adalah -69.5750038069106
Nilai x7 adalah 111.462427531472
Nilai x8 adalah 105.59523134400001
Nilai x9 adalah -91.10286400000001
Nilai x10 adalah -67.608

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase1d10.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

x_1 =2.02,

x_2 = 35.18,

x_3 = -133.78,

x_4 = 96.10,

x_5 = 6.34,

x_6 = -69.58,

x_7 = 111.46,

x_8 = 105.60,

x_9 = -91.10,

x_{10} = -67.61,

2. SPL Berbentuk matriks *Augmented*

   a. File studi Kasus :

```
test > testCase2a.txt
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3
```

Hasil :

```
Masukkan nama dari file : testCase2a.txt

Solusi SPL adalah :
Nilai x1 adalah -1.0 + (a)
Nilai x2 adalah 0.0 + 2.0*(b)
Nilai x3 adalah b
Nilai x4 adalah a

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase2a.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Jadi setelah dicari solusi dari matriks augmented menggunakan eliminasi Gauss atau Gauss-Jordan. Masing-masing nilai variabel yang didapat antara lain:

$x_1 = -1 + a$,

$x_2 = 2b$,

$x_3 = b$,

$x_4 = a$.

b. File studi Kasus :

```
test > testCase2b.txt
1   2 0 8 0 8
2   0 1 0 4 6
3   -4 0 6 0 6
4   0 -2 0 3 -1
5   2 0 -4 0 -4
6   0 1 0 -2 0
```

Hasil :

```
Masukkan nama dari file : testCase2b.txt

Solusi SPL adalah :
Nilai x1 adalah 0.0
Nilai x2 adalah 2.0
Nilai x3 adalah 1.0
Nilai x4 adalah 1.0
Nilai x5 adalah b
Nilai x6 adalah a

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase2b.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Jadi setelah dicari solusi dari matriks augmented menggunakan eliminasi Gauss atau Gauss-Jordan. Masing-masing nilai variabel yang didapat antara lain:

$x_1 = 0$,

$x_2 = 2$,

$x_3 = 1$,

$x_4 = 1$,

$x_5 = b$,

$x_6 = a$.

3. SPL Berbentuk
   a. File studi Kasus :

```
test > testCase3a.txt
1   8 1 3 2 0
2   2 9 -1 -2 1
3   1 3 2 -1 2
4   1 0 6 4 3
```

Hasil :

```
Masukkan nama dari file : testCase3a.txt

Solusi SPL adalah :
Nilai x1 adalah -0.2257654499999998
Nilai x2 adalah 0.1827951499999999
Nilai x3 adalah 0.70969
Nilai x4 adalah -0.258

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase3a.txt
File berhasil dibuat: testCase3a.txt
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Analisis :

Dengan menggunakan metode eliminasi Gauss-Jordan, solusi SPL-nya adalah x_1 = -0.2257654499999998, x_2 = 0.1827951499999999, x_3 = 0.70969, dan x_4 = -0.258

b. File studi Kasus :

```
test > testCase3b.txt
0 0 0 0 0 0 1 1 1 13
0 0 0 1 1 1 0 0 0 15
1 1 1 0 0 0 0 0 0 8
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.04289 0.75 0.04289 0 0.04289 0 0 3.81
0 0 1 0 0 1 0 0 1 18
0 1 0 0 1 0 0 1 0 12
1 0 0 1 0 0 1 0 0 6
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91241 0.25 0 0.25 0.91241 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04
```

Hasil :

```
Masukkan nama dari file : testCase3b.txt

Matriks tidak memiliki penyelesaian (Jika menggunakan eliminasi Gauss atau Gauss-Jordan)
Atau tidak dapat diselesaikan dengan metode yang digunakan (Jika menggunakan metode Invers atau Kaidah Cramer)

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke submenu Penyelesaian SPL : y

Masukkan nama file output : testCase3b.txt
File berhasil dibuat: testCase3b.txt
Berhasil menulis ke file.

Kembali ke submenu Penyelesaian SPL
```

Analisis :

Dengan menggunakan metode eliminasi Gauss-Jordan, matriks tersebut tidak memiliki penyelesaian. Dikarenakan adanya baris setelah eliminasi yang merupakan baris Exception. Sementara itu, kaidah cramer dan metode invers juga tidak dapat memberikan penyelesaian dari SPL ini, karena determinan matriks koefisien adalah 0.

4. Studi Kasus Interpolasi

a. File studi Kasus :

```
test > testCase4a1.txt
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
```

Hasil :

```
Masukkan nama dari file : testCase4a1.txt
Nilai x yang ingin ditaksir : 0.2

Persamaan Interpolasi Polinom berupa :
p(x) = -0.185 + 10.276x^1 + -163.916x^2 + 1220.855x^3 + -4346.314x^4 + 7102.399x^5 + -4212.435x^6

Nilai interpolasi pada x = 0.2 adalah 0.13

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4a1.txt
File berhasil dibuat: testCase4a1.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik x = 0.2 adalah 0.13. Hal ini sesuai dengan input file yang menunjukkan bahwa f(0.2) = 0.13

b. File studi Kasus :

```
test > testCase4a2.txt
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
```

Hasil :

```
Masukkan nama dari file : testCase4a2.txt
Nilai x yang ingin ditaksir : 0.55

Persamaan Interpolasi Polinom berupa :
p(x) = -0.185 + 10.276x^1 + -163.916x^2 + 1220.855x^3 + -4346.314x^4 + 7102.399x^5 + -4212.435x^6

Nilai interpolasi pada x = 0.55 adalah 2.138

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4a2.txt
File berhasil dibuat: testCase4a2.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik x = 0.55 adalah 2.138.

c. File studi Kasus :

```
test > testCase4a3.txt
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
```

Hasil :

```
Masukkan nama dari file : testCase4a3.txt
Nilai x yang ingin ditaksir : 0.85

Persamaan Interpolasi Polinom berupa :
p(x) = -0.185 + 10.276x^1 + -163.916x^2 + 1220.855x^3 + -4346.314x^4 + 7102.399x^5 + -4212.435x^6

Nilai interpolasi pada x = 0.85 adalah -66.27

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4a3.txt
File berhasil dibuat: testCase4a3.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik x = 0.85 adalah -66.27

d. File studi Kasus :

```
test > testCase4a4.txt
1  0.4 0.043
2  0.7 0.005
3  0.11 0.058
4  0.14 0.072
5  0.17 0.1
6  0.2 0.13
7  0.23 0.147
```

Hasil :

```
Masukkan nama dari file : testCase4a4.txt
Nilai x yang ingin ditaksir : 1.28

Persamaan Interpolasi Polinom berupa :
p(x) = -0.185 + 10.276x^1 + -163.916x^2 + 1220.855x^3 + -4346.314x^4 + 7102.399x^5 + -4212.435x^6

Nilai interpolasi pada x = 1.28 adalah -3485.145

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4a4.txt
File berhasil dibuat: testCase4a4.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik x = 1.28 adalah -3485.145

e. File studi Kasus :

```
test > testCase4b1.txt
1    6.567 12264
2    7 21807
3    7.258 38391
4    7.451 54517
5    7.548 51952
6    7.839 28228
7    8.161 35764
8    8.484 20813
9    8.709 12408
10   9 10534
```

Hasil :

```
Masukkan nama dari file : testCase4b1.txt
Nilai x yang ingin ditaksir : 7.516

Persamaan Interpolasi Polinom berupa :
p(x) = 7181336719232.008 + -9340507355054.906x^1 + 5330942693282.656x^2 + -1755854913346.422x^3 + 368371026953.543x^4 + -51109339348.433x^5 + 4693924336.755x^6
+ -275373595.423x^7 + 9369693.474x^8 + -140949.901x^9

Nilai interpolasi pada x = 7.516 adalah 53537.752

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4b1.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Interpolasi Polinom
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik x = 7.516 adalah 53537.52

f. File Studi Kasus:

```
test > testCase4b2.txt
1    6.567 12264
2    7 21807
3    7.258 38391
4    7.451 54517
5    7.548 51952
6    7.839 28228
7    8.161 35764
8    8.484 20813
9    8.709 12408
10   9 10534
```

Hasil :

```
Masukkan nama dari file : testCase4b2.txt
Nilai x yang ingin ditaksir : 8.323

Persamaan Interpolasi Polinom berupa :
p(x) = 7181336719232.008 + -9340507355054.906x^1 + 5330942693282.656x^2 + -1755854913346.422x^3 + 368371026953.543x^4 + -51109339348.433x^5 + 4693924336.755x^6
+ -275373595.423x^7 + 9369693.474x^8 + -140949.901x^9

Nilai interpolasi pada x = 8.323 adalah 36294.531

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4b2.txt
File berhasil dibuat: testCase4b2.txt
Berhasil menulis ke file.

Kembali ke submenu Interpolasi Polinom
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik x = 8.323 adalah 36294.531

g. File Studi Kasus :

```
test > testCase4b3.txt
1   6.567 12264
2   7 21807
3   7.258 38391
4   7.451 54517
5   7.548 51952
6   7.839 28228
7   8.161 35764
8   8.484 20813
9   8.709 12408
10  9 10534
```

Hasil :

```
Masukkan nama dari file : testCase4b3.txt
Nilai x yang ingin ditaksir : 9.167

Persamaan Interpolasi Polinom berupa :
p(x) = 7181336719232.008 + -9340507355054.906x^1 + 5330942693282.656x^2 + -1755854913346.422x^3 + 368371026953.543x^4 + -51109339348.433x^5 + 4693924336.755x^6
+ -275373595.423x^7 + 9369693.474x^8 + -140949.901x^9

Nilai interpolasi pada x = 9.167 adalah -667672.078

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4b3.txt
File berhasil dibuat: testCase4b3.txt
Berhasil menulis ke file.

Kembali ke submenu Interpolasi Polinom
```

Analisis :

Dengan interpolasi polinom, kita mendapatkan jumlah kasus covid pada tanggal 5 September 2022 berjumlah negatif. Karena kasus covid tidak mungkin mengalami jumlah di bawah 0, dapat disimpulkan jumlah kasus covid adalah 0 atau sedang tidak ada kasus covid di hari itu.

h. File Studi Kasus :

```
test > testCase4c.txt
1    0 0
2    0.4 0.419
3    0.8 0.507
4    1.2 0.561
5    1.6 0.584
6    2 0.577
```

Hasil :

```
Masukkan nama dari file : testCase4c.txt
Nilai x yang ingin ditaksir : 0

Persamaan Interpolasi Polinom berupa :
p(x) = 2.039x^1 + -3.565x^2 + 3.253x^3 + -1.429x^4 + 0.238x^5

Nilai interpolasi pada x = 0.0 adalah 0

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase4c.txt
File berhasil dibuat: testCase4c.txt
Berhasil menulis ke file.

Kembali ke submenu Interpolasi Polinom
```

Analisis :

Nilai interpolasi polinom dari 7 titik di atas di titik 0.0 adalah 0

5. Studi Kasus Interpolasi Bikubik
   a. File studi Kasus :

```
test > testCase5a1.txt
1    153 59 210 96
2    125 161 72 81
3    98 101 42 12
4    21 51 0 16
5    0 0
```

Hasil :

```
Masukkan nama dari file : testCase5a1.txt

Persamaan Interpolasi Bikubik berupa :
f(x,y) = 161x^(0)y^(0) + -63.667x^(1)y^(0) + -62.5x^(2)y^(0) + 37.167x^(3)y^(0) + -7.667x^(0)y^(1) + -43.306x^(1)y^(1) + -33.833x^(2)y^(1) + 20.806x^(3)y^(1)
 + -81x^(0)y^(2) + 98.833x^(1)y^(2) + 108.25x^(2)y^(2) + -72.083x^(3)y^(2) + 28.667x^(0)y^(3) + -35.028x^(1)y^(3) + -42.917x^(2)y^(3) + 29.278x^(3)y^(3)

Nilai interpolasi pada x = 0.0, y = 0.0 adalah 161

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase5a1.txt
File berhasil dibuat: testCase5a1.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Dari keluaran program, diketahui nilai f(x,y) pada (0,0) adalah 161. Hal ini sesuai dengan matriks input, dimana pada koordinat (0,0) terdapat elemen bernilai 161.

b. File studi Kasus :

```
test > testCase5a2.txt
1   153 59 210 96
2   125 161 72 81
3   98 101 42 12
4   21 51 0 16
5   0.5 0.5
```

Hasil :

```
Masukkan nama dari file : testCase5a2.txt

Persamaan Interpolasi Bikubik berupa :
f(x,y) = 161x^(0)y^(0) + -63.667x^(1)y^(0) + -62.5x^(2)y^(0) + 37.167x^(3)y^(0) + -7.667x^(0)y^(1) + -43.306x^(1)y^(1) + -33.833x^(2)y^(1) + 20.806x^(3)y^(1)
 + -81x^(0)y^(2) + 98.833x^(1)y^(2) + 108.25x^(2)y^(2) + -72.083x^(3)y^(2) + 28.667x^(0)y^(3) + -35.028x^(1)y^(3) + -42.917x^(2)y^(3) + 29.278x^(3)y^(3)

Nilai interpolasi pada x = 0.5, y = 0.5 adalah 97.727

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase5a2.txt
File berhasil dibuat: testCase5a2.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Dari keluaran program, diketahui nilai f(x,y) pada (0.5,0.5) adalah 97.727. Hal ini dapat kita perkirakan melalui matriks input, terlihat bahwa koordinat (0.5, 0.5) berada di tengah matriks. Dan nilai 97.727 berada di antara 161 dan 42 dan di antara 101 dan 72.

c. File studi Kasus :

```
test > testCase5a3.txt
153 59 210 96
125 161 72 81
98 101 42 12
21 51 0 16
0.25 0.75
```

Hasil :

```
Persamaan Interpolasi Bikubik berupa :
f(x,y) = 161x^(0)y^(0) + -63.667x^(1)y^(0) + -62.5x^(2)y^(0) + 37.167x^(3)y^(0) + -7.667x^(0)y^(1) + -43.306x^(1)y^(1) + -33.833x^(2)y^(1) + 20.806x^(3)y^(1)
 + -81x^(0)y^(2) + 98.833x^(1)y^(2) + 108.25x^(2)y^(2) + -72.083x^(3)y^(2) + 28.667x^(0)y^(3) + -35.028x^(1)y^(3) + -42.917x^(2)y^(3) + 29.278x^(3)y^(3)

Nilai interpolasi pada x = 0.25, y = 0.75 adalah 105.515

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase5a3.txt
File berhasil dibuat: testCase5a3.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Dari keluaran program, diketahui nilai f(x,y) pada (0.25,0.75) adalah 105.515.

d. File studi Kasus :

```
test > testCase5a4.txt
153 59 210 96
125 161 72 81
98 101 42 12
21 51 0 16
0.1 0.9
```

Hasil :

```
Masukkan nama dari file : testCase5a4.txt

Persamaan Interpolasi Bikubik berupa :
f(x,y) = 161x^(0)y^(0) + -63.667x^(1)y^(0) + -62.5x^(2)y^(0) + 37.167x^(3)y^(0) + -7.667x^(0)y^(1) + -43.306x^(1)y^(1) + -33.833x^(2)y^(1) + 20.806x^(3)y^(1)
 + -81x^(0)y^(2) + 98.833x^(1)y^(2) + 108.25x^(2)y^(2) + -72.083x^(3)y^(2) + 28.667x^(0)y^(3) + -35.028x^(1)y^(3) + -42.917x^(2)y^(3) + 29.278x^(3)y^(3)

Nilai interpolasi pada x = 0.1, y = 0.9 adalah 104.229

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase5a4.txt
File berhasil dibuat: testCase5a4.txt
Berhasil menulis ke file.

Kembali ke menu utama
```

Analisis :

Dari keluaran program, diketahui nilai f(x,y) pada (0.1,0.9) adalah 104.229.

6. Studi Kasus Regresi Linier Berganda

   a. File studi Kasus :

```
test > testCase6XY.txt
72.4 76.3 29.18 0.9
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
```

```
test > testCase6Xk.txt
50 76 29.3
```

Hasil :

```
Masukkan nama file input XY: testCase6XY.txt
Masukkan nama file input Xk yang ingin ditaksir: testCase6Xk.txt

Persamaan Regresi berupa :
f(x) = -3.4535618210499885 + (-0.0027810899999999917)x1 + 2.3799999999984944E-4x2 + 0.154x3

f(Xk) = 0.9376716789500006

Apakah output ingin disimpan ke dalam file?
Input 'y' jika ya, atau input sembarang untuk kembali ke menu utama : y

Masukkan nama file output : testCase6.txt
File sudah ada, akan menulis ke file yang sudah ada.
Berhasil menulis ke file.

Kembali ke submenu Regresi Linier Berganda
```

Analisis:

Berdasarkan hasil regresi linier berganda, dapat dilihat bahwa tekanan udara merupakan variabel paling berpengaruh dalam nilai Nitrous Oxide, yaitu dengan koefisien sebesar 0.154. Sedangkan variabel yang paling tidak signifikan terhadap nilai Nitrous Oxide yaitu temperatur dengan koefisien sebesar 2.38E-4.

Lalu dengan data humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30, dengan regresi linier berganda dapat ditaksir bahwa nilai Nitrous Oxide-nya adalah 0.93767.

BAB 5

1. Kesimpulan

Dalam tugas besar ini, kami berhasil membuat kalkulator matriks dengan mengimplementasi berbagai algoritma untuk menyelesaikan SPL ke dalam bahasa Java. Kami juga berhasil menjawab beberapa studi kasus mengenai interpolasi, interpolasi polinomial, serta regresi linier berganda. Namun, kami belum bisa mengimplementasi soal bonus mengenai *image scaling* karena menurut kami soal ini belum bisa kami selesaikan dengan waktu dan kemampuan yang kami miliki saat ini.

2. Saran

Saran untuk kelompok kami diantaranya:

- Pengerjaan tugas besar bisa dimulai lebih awal.
- Lebih banyak mengeksplorasi tentang sintaks-sintaks Java.
- Bisa lebih memahami cara compile dan run di Java.

3. Refleksi

Dari tugas besar ini, kami dapat memperbaiki kinerja kelompok kami dalam berbagai hal seperti pengalokasian waktu, perencanaan pengerjaan tugas, serta penargetan progress dalam tugas dalam kurun waktu tertentu. Hal lain yang kami refleksikan dari tugas ini adalah pengalaman dan pengetahuan mengenai pemrograman dalam bahasa Java.

4. Tautan Repository

https://github.com/GoDillonAudris512/Algeo01-21062.git

DAFTAR REFERENSI

- Ghozali, Imam. 2018. Aplikasi Analisis Multivariate dengan Program IBM SPSS 25. Badan Penerbit Universitas Diponegoro: Semarang.
- https://en.wikipedia.org/wiki/Cramer%27s_rule
- https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf
- https://informatika.stei.itb.ac.id/~rinaldi.munir/MetNum/2010-2011/Interpolasi%20Polinom.pdf
- https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf
- https://bookdown.org/moh_rosidi2610/Metode_Numerik/interpolation.html#poliinterpolation
- https://www.madematika.id/2017/08/pengertian-minor-kofaktor-matriks.html
- https://www.gramedia.com/literasi/determinan/
- http://e-journal.uajy.ac.id/19481/4/TF079013.pdf