

TUGAS BESAR 1
IF2123 ALJABAR LINIER DAN GEOMETRI SISTEM PERSAMAAN LINIER
DETERMINAN, DAN APLIKASINYA
SEMESTER 1 TAHUN 2020/2021

Disusun oleh:

1. 13520110 Farrel Ahmad
2. 13520070 Raden Haryosatyo Wisjnunandono
3. 13520145 Steven Gianmarg H. Siahaan



BAB I

DESKRIPSI MASALAH

Tugas besar ini merupakan aplikasi dari konsep Sistem Persamaan Linier dan Determinan yang telah dipelajari di kuliah. Pada tugas besar 1 IF 2123 ini, mahasiswa diminta untuk 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, library tersebut digunakan dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

Berikut spesifikasi dari tugas besar 1 IF 2123 :

1. Buatlah pustaka dalam Bahasa Java untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.
2. Gunakan pustaka di atas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Pada tugas besar 1 IF 2123 ini juga diberikan beberapa spesifikasi untuk program,yaitu :

1. Program dapat menerima masukan (input) baik dari keyboard maupun membaca masukan dari file text. Untuk SPL, masukan dari keyboard adalah 3 m, n, koefisien a_{ij} , dan b_i . Masukan dari file berbentuk matriks augmented tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

$$\begin{matrix} 3 & 4.5 & 2.8 & 10 & 12 \\ -3 & 7 & 8.3 & 11 & -4 \\ 0.5 & -10 & -9 & 12 & 0 \end{matrix}$$

2. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari keyboard adalah n dan koefisien a_{ij} . Masukan dari file berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya

$$\begin{matrix} 3 & 4.5 & 2.8 & 10 \\ -3 & 7 & 8.3 & 11 \\ 0.5 & -10 & -9 & 12 \end{matrix}$$

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x_0 , y_0) , (x_1 , y_1) , ..., (x_n , y_n) , dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

8.0 2.0794

9.0 2.1972

9.5 2.2513

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), semua nilai-nilai x_{1i} , x_{2i} , ..., x_{ni} , nilai y_i , dan nilai-nilai x_k yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. Untuk persoalan SPL, luaran (output) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$.)
6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing
7. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing
8. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam file**.
9. Bahasa program yang digunakan adalah Java.
10. **Program tidak harus berbasis GUI**, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

MENU

1. Sistem Persamaan Linier
2. Determinan
3. Matriks balikan
4. Interpolasi Polinom
5. Regresi linier berganda
6. Keluar

Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:

1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Begitu juga untuk pilihan menu nomor 2 dan 3

BAB II

TEORI SINGKAT

2.1 Eliminasi Gauss

Eliminasi Gauss adalah suatu metode penyelesaian sistem persamaan dengan memanfaatkan matriks augmented yang ditemukan oleh Carl Friedrich. Eliminasi Gauss adalah suatu metode untuk mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana lagi dengan melakukan operasi baris elementer(OBE). Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Caranya dengan mengubah persamaan linear tersebut ke dalam matriks teraugmentasi dan mengoperasikannya. Dalam eliminasi Gauss kemudian dilakukan operasi baris elementer (OBE) terhadap matriks augmented tersebut sehingga diperoleh matriks eselon baris. Setelah menjadi matriks baris, lakukan substitusi balik untuk mendapatkan nilai dari variabel-variabel tersebut. Pada metode Gauss ini terdapat ciri-ciri yang harus selalu diingat diantaranya :

1. Jika suatu baris tidak semua nol,maka bilangan pertama yang tidak nol adalah 1(1 utama)
2. Baris nol terletak paling bawah
3. 1 utama baris berikutnya berada di kanan 1 utama baris diatasnya
4. Dibawah 1 utama harus nol

Pada setiap SPL terdapat 3 pilihan solusi yaitu :

1. Solusi Tunggal
2. Memiliki banyak solusi
3. Tidak Memiliki Solusi

Berikut contoh penerapannya(SPL dengan banyak solusi) :

$$2x + 10y + 4z = -2$$

$$x + 4y + 5z = -3$$

$$3x + 15y + 6z = -3$$

Representasikan dalam bentuk matriks :

$$\left[\begin{array}{ccc|c} 2 & 10 & 4 & -2 \\ 1 & 4 & 5 & -3 \\ 3 & 15 & 6 & -3 \end{array} \right]$$

Dalam eliminasi Gauss kemudian dilakukan operasi baris elementer (OBE) terhadap matriks augmented tersebut sehingga diperoleh matriks eselon baris. Lalu ubah kembali menjadi bentuk SPL

$$\left[\begin{array}{ccc|c} 1 & 5 & 2 & -1 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 0 & 0 \end{array} \right] \rightarrow \begin{aligned} x + 5y + 2z &= -1 \dots (1) \\ y - 3z &= 2 \dots (2) \end{aligned}$$

Perhatikan persamaan (2) :

$$y - 3z = 2 \Leftrightarrow y = 2 + 3z$$

Subtitusikan ke persamaan (1) dan diperoleh :

$$x + 5y + 2z = -1$$

$$\Leftrightarrow x = -1 - 5y - 2z$$

$$\Leftrightarrow x = -1 - 5(2 + 3z) - 2z$$

$$\Leftrightarrow x = -1 - 10 - 15z - 2z$$

$$\Leftrightarrow x = -11 - 17z$$

2.2 Eliminasi Gauss Jordan

Eliminasi Gauss-Jordan adalah lanjutan dari metode eliminasi Gauss yang dikembangkan oleh Wilhelm Jordan. Metode ini memanfaatkan bentuk matriks eselon baris dari eliminasi Gauss dan melakukan OBE lebih lanjut sehingga diperoleh bentuk matriks eselon baris tereduksi

Bentuk-bentuk tersebut adalah sebagai berikut.

Solusi unik

Solusi banyak

Tidak ada solusi

$$\left[\begin{array}{cccc|c} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{array} \right] \sim \left[\begin{array}{cccc|c} 1 & 3 & -2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1/3 \end{array} \right] \sim \left[\begin{array}{cccc|c} 1 & 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]$$

Contoh penerapannya(dengan melanjutkan contoh pada 2.1) :

$$\left[\begin{array}{ccc|c} 1 & 5 & 2 & -1 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

Dengan membuat bentuk matriks eselon baris tereduksi :

$$\left[\begin{array}{ccc|c} 1 & 0 & 17 & -11 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

Didapatkan :

$$x = -11 - 17z$$

$$y = 2 + z$$

$z = \text{semua anggota bilangan real}$ (banyak solusi)

2.3 Determinan

Determinan ialah sebuah nilai yang dapat dihitung dari unsur suatu matriks persegi.

Determinan matriks A ditulis dengan tanda $\det(A)$, $\det A$, atau $|A|$. Determinan dapat dianggap sebagai faktor penskalaan transformasi yang digambarkan oleh matriks.

Misalkan A adalah sebuah matriks berukuran $n \times n$,

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Maka, determinan dari matriks A dapat dilambangkan sebagai

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

Determinan dari sebuah matriks $n \times n$ dapat dicari dengan memanfaatkan matriks segitiga (upper triangular atau lower triangular).

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

Hal-hal yang menjadi teorema dari determinan matriks adalah sebagai berikut:

1. Jika A mengandung baris / kolom yang semua anggotanya nol, maka $\det(A) = 0$
2. Jika A^T adalah transpose matriks A, maka $\det(A) = \det(A^T)$
3. Jika $A = BC$, maka $\det(A) = \det(B)\det(C)$
4. **Matriks memiliki balikan jika $\det(A) \neq 0$**
5. $\det(A^{-1}) = 1/\det(A)$

2.4 Matriks Balikan

Matriks Balikan Matriks balikan atau invers adalah teorema matriks yang menyatakan bahwa jika suatu matriks A memiliki balikan berupa matriks B sedemikian sehingga $AB = BA = I$. Matriks invers dapat disimbolkan dengan tanda A^{-1} , sehingga sifat matriksnya menjadi $AA^{-1} = A^{-1}A = I$. Untuk suatu matriks 2×2 balikannya dapat diperoleh dengan cara berikut

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

dengan syarat $ad - bc \neq 0$

$ad - bc$ dalam perhitungan tersebut adalah determinan dari matriks A, sehingga apabila sebuah matriks memiliki determinan 0, maka matriks tersebut tidak memiliki balikan. Untuk matriks berukuran $n \times n$, balikannya dapat dihitung dengan menggunakan memanfaatkan metode GaussJordan.

$$\left(\begin{array}{ccc|cc} 1 & 2 & 3 & 1 & 0 \\ 2 & 5 & 3 & 0 & 1 \\ 1 & 0 & 8 & 0 & 1 \end{array} \right)$$

Dari bentuk tersebut, dilakukan OBE sehingga diperoleh :

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array} \right) = (I | A^{-1})$$

Maka, balikan dari matriks A adalah :

$$A^{-1} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

2.5 Matriks Kofaktor dan Adjoint

Matriks Misalkan A adalah matriks $n \times n$ dan C_{ij} adalah kofaktor entri a_{ij} , maka kofaktor matriks A adalah

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

Untuk mencari nilai kofaktor terlebih dahulu kita harus mencari nilai minor dari setiap elemen matrik. Untuk memudahkan, selanjutnya minor kita beri simbol dengan huruf M dan minor untuk setiap elemen matrik akan kita beri simbol dengan M_{ij} dimana i adalah letak baris dan j adalah letak kolom dari setiap elemen matrik.

contoh:

diketahui matrik A sebagai berikut:

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 4 & 1 & 6 \\ 1 & 4 & 0 \end{bmatrix}$$

maka minor elemen 2 yang terletak pada baris ke 1 kolom ke 1 diberi simbol dengan M₁₁. Untuk mencari harga minornya dapat kita lakukan dengan mencoret atau menghilangkan baris ke 1 dan kolom ke 1 sehingga didapatkan matrik baru seperti berikut:

$$\begin{vmatrix} 1 & 6 \\ 4 & 0 \end{vmatrix}$$

jadi minor elemen 2 (M₁₁) adalah :

$$\begin{vmatrix} 1 & 6 \\ 4 & 0 \end{vmatrix} = 0 - 24 = -24$$

Serupa dengan cara di atas , minor elemen 3 (M₁₂) adalah :

$$\begin{vmatrix} 4 & 6 \\ 1 & 0 \end{vmatrix} = 0 - 6 = -6$$

Untuk nilai M₁₃, M₂₁, M₂₂, M₂₃, M₃₁, M₃₂ dan M₃₃ didapatkan hasil sebagai berikut:

$$M_{13} = \begin{vmatrix} 4 & 1 \\ 1 & 4 \end{vmatrix} = (16 - 1) = 15$$

$$M_{21} = \begin{vmatrix} 3 & 5 \\ 4 & 0 \end{vmatrix} = (0 - 20) = -20$$

$$M_{22} = \begin{vmatrix} 2 & 5 \\ 1 & 0 \end{vmatrix} = (0 - 5) = -5$$

$$M_{23} = \begin{vmatrix} 2 & 3 \\ 1 & 4 \end{vmatrix} = (8 - 3) = 5$$

$$M_{31} = \begin{vmatrix} 3 & 5 \\ 1 & 6 \end{vmatrix} = (18 - 5) = 13$$

$$M_{32} = \begin{vmatrix} 2 & 5 \\ 4 & 6 \end{vmatrix} = (12 - 20) = -8$$

$$M_{33} = \begin{vmatrix} 2 & 3 \\ 4 & 1 \end{vmatrix} = (2 - 12) = -10$$

Kofaktor

Setelah mendapatkan harga minor dari masing-masing elemen matriks kita dapat menentukan nilai atau harga dari kofaktor. Cara mencarinya adalah dengan mengalikan masing-masing nilai minor di atas dengan tanda tempat masing-masing elemen. Adapun tanda tempatnya dapat dilihat pada gambar berikut:

$$\begin{array}{cccc|c} + & - & + & \dots \\ - & + & - & \dots \\ + & - & + & \dots \\ \dots & \dots & \dots & \dots \end{array}$$

Jadi berdasarkan tanda tempat di atas kita dapat mencari nilai kofaktor dari masing-masing elemen matriks. Untuk selanjutnya kita akan berikan simbol untuk nilai kofaktor masing-masing elemen dengan C_{ij} , dimana i menandakan baris dan j menandakan kolom. jadi untuk setiap elemen di atas kita dapatkan harga kofaktornya sebagai berikut:

$$C_{11} = + \begin{vmatrix} 1 & 6 \\ 4 & 0 \end{vmatrix} = + (0 - 24) = -24$$

$$C_{12} = + \begin{vmatrix} 4 & 6 \\ 1 & 0 \end{vmatrix} = - (0 - 6) = 6$$

$$C_{13} = + \begin{vmatrix} 4 & 1 \\ 1 & 4 \end{vmatrix} = + (16 - 1) = 15$$

$$C_{21} = - \begin{vmatrix} 3 & 5 \\ 4 & 0 \end{vmatrix} = - (0 - 20) = 20$$

$$C_{22} = + \begin{vmatrix} 2 & 5 \\ 1 & 0 \end{vmatrix} = + (0 - 5) = -5$$

$$C_{23} = - \begin{vmatrix} 2 & 3 \\ 1 & 4 \end{vmatrix} = - (8 - 3) = -5$$

$$C_{31} = + \begin{vmatrix} 3 & 5 \\ 1 & 6 \end{vmatrix} = + (18 - 5) = 13$$

$$C_{32} = - \begin{vmatrix} 2 & 5 \\ 4 & 6 \end{vmatrix} = - (12 - 20) = 8$$

$$C_{33} = + \begin{vmatrix} 2 & 3 \\ 4 & 1 \end{vmatrix} = + (2 - 12) = -10$$

Setelah kita mendapatkan harga atau nilai kofaktor dari masing-masing elemen matrik di atas, maka kita sekarang akan menyusun setiap nilai kofaktor tersebut sesuai dengan alamat tempatnya masing-masing. Susunan masing-masing elemen dari nilai kofaktor ini akan menghasilkan sebuah matrik baru yang kita namakan dengan matrik kofaktor. Untuk selanjutnya matrik kofaktor akan kita beri simbol dengan huruf C . Jadi matrik kofaktor (C) dari matrik di atas adalah:

$$C = \begin{bmatrix} -24 & 6 & 15 \\ 20 & -5 & -5 \\ 13 & 8 & -10 \end{bmatrix}$$

Adjoint dari matriks A adalah transpose dari matriks kofaktor A . Adjoin matriks dapat dimanfaatkan untuk mencari matriks balikan, menggunakan persamaan sebagai berikut

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

Adjoin Matrik Bujur Sangkar

Jika kita sudah mendapatkan matrik kofaktor (C) maka kita sudah bisa mendapatkan adjoin dari matrik tersebut. adjoin matrik bujur sangkar sama nilainya dengan transpose dari matrik kofaktor, jadi dengan mencari transpose dari matrik kofaktor kita sudah mendapatkan nilai adjoin matrik. Transpose dari matrik C adalah :

$$C^T = \begin{bmatrix} -24 & 20 & 13 \\ 6 & -5 & 8 \\ 15 & -5 & -10 \end{bmatrix}$$

Maka matrik transpose dari matrik kofaktor dinamakan dengan matrik adjoin dari matrik A. Jadi untuk memperoleh adjoin dari suatu matrik bujur sangkar A kita harus

- Membentuk matrik kofaktor C
- Menuliskan transpose dari matrik C yaitu CT

2.6 Kaidah Cramer

Aturan Cramer atau kaidah Cramer, ditemukan oleh matematikawan Swiss, Gabriel Cramer, adalah salah satu prosedur untuk menyelesaikan sistem persamaan linear. Dasar metode ini adalah determinan dan matriks.

Jika $Ax = b$ adalah SPL yang terdiri dari n persamaan linier dan n variabel / peubah, sedemikian sehingga $\det(A) \neq 0$, maka SPL tersebut memiliki solusi unik, yaitu

$$x_1 = \frac{\det(A_1)}{\det(A)}, \quad x_2 = \frac{\det(A_2)}{\det(A)}, \quad \dots, \quad x_n = \frac{\det(A_n)}{\det(A)}$$

Penerapan Metode Cramer pada SPL 2 variabel :

Diberikan sistem persamaan linear dua variabel:

$$\begin{aligned} a_1x + b_1y &= c_1 \\ a_2x + b_2y &= c_2 \end{aligned}$$

yang dapat ditulis dalam bentuk matriks:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Misalkan $a_1b_2 - b_1a_2$ tidak nol, maka nilai x dan y dapat dicari dengan menggunakan aturan Cramer, yaitu membagi determinan matriks yang telah dimodifikasi dengan determinan matriks awal:

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \frac{c_1b_2 - b_1c_2}{a_1b_2 - a_2b_1}, \quad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}$$

Penerapan Metode Cramer pada SPL 3 Variabel :

$$\begin{aligned} a_1x + b_1y + c_1z &= d_1 \\ a_2x + b_2y + c_2z &= d_2 \\ a_3x + b_3y + c_3z &= d_3 \end{aligned}$$

yang dapat ditulis dalam bentuk matriks:

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Misalkan determinan matriks ordo 3 tersebut tidak nol, maka x , y dan z dapat dicari dengan cara membagi determinan matriks yang telah dimodifikasi dengan determinan matriks awal:

$$x = \frac{\begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}, \quad y = \frac{\begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}, \quad z = \frac{\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}$$

2.7 Interpolasi Polinom

Interpolasi Polinomial adalah sebuah metode yang digunakan untuk menemukan nilai dari y sebagai fungsi dari x yang belum diketahui bentuk dan persamaan fungsinya. Penentuan nilai

dari y yang terkorenpodensi dari x hanya bisa dilakukan berdasarkan data-data yang didapat dari percobaan mulai $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Nilai x dan y ini bisa digambarkan dalam bidang kartesian x dan y, sehingga nilai $y(x)$ bisa ditentukan berdarkan model kurva yang terbentuk. Hal ini membuat metode interpolasi juga dikenal sebagai metode pencocokan Kurva. Pada interpolasi polinom, diberikan sebuah $n+1$ buah titik $(X_0, Y_0), \dots, (X_n, Y_n)$ sedemikian sehingga terdapat $P_n(X_i)$ yang menginterpolasikan himpunan titik-titik $\{(X_0, Y_0), \dots, (X_n, Y_n)\}$. $P_n(X_i)$ dapat digunakan untuk memprediksi nilai Y dari suatu nilai X pada selang $[X_0 \dots X_n]$. Dalam persamaan Lagrange, polynomial $p(x)$ dapat diperoleh sebagai berikut

$$p(x) = \frac{(x - x_1)(x - x_2) \cdots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)} y_0 + \frac{(x - x_0)(x - x_2) \cdots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \cdots (x_1 - x_n)} y_1 + \dots + \frac{(x - x_0)(x - x_1) \cdots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})} y_n$$

Dengan menggunakan matriks, interpolasi polinomial dapat dituliskan sebagai berikut

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & \dots & x_1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & x_n^{n-2} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

2.8 Regresi Linier Berganda

Regresi berganda adalah model regresi atau prediksi yang melibatkan lebih dari satu variabel bebas atau prediktor. Istilah regresi berganda dapat disebut juga dengan istilah multiple regression. Kata multiple berarti jamak atau lebih dari satu variabel. Banyak para mahasiswa yang salah kaprah dalam memahami istilah tersebut. Dimana tidak bisa membedakan antara multiple regression dengan multivariat regression. Perbedaannya adalah jika multiple regression atau regresi berganda adalah adanya lebih dari satu **variabel** prediktor (variabel bebas/variabel independen).

Regresi linier berganda digunakan untuk menjelaskan hubungan suatu variabel tak bebas (Y) dengan dua atau lebih variabel bebas ($X_1, X_2, X_3, \dots, X_n$). Regresi linier berganda bertujuan untuk memprediksi nilai dari variabel tak bebas (Y), jika nilai-nilai variabel bebasnya (X) diketahui. Secara matematik, regresi linier berganda dapat dituliskan sebagai berikut

$$Y = a + b_1 X_1 + b_2 X_2 + \dots + b_n X_n$$

Regresi linier berganda juga dapat diselesaikan dengan menggunakan matriks. Misalkan suatu persamaan regresi sebagai berikut

$$a n + b_1 \sum X_1 + b_2 \sum X_2 = \sum Y$$

$$a \sum X_1 + b_1 \sum X_1^2 + b_2 \sum X_1 X_2 = \sum X_1 Y$$

$$a \sum X_2 + b_1 \sum X_2 X_1 + b_2 \sum X_2^2 = \sum X_2 Y$$

Sebagai contoh, persamaan tersebut dapat dituliskan seperti ini :

$$m_{11}a + m_{12}b_1 + m_{13}b_2 = h_1$$

$$m_{21}a + m_{22}b_1 + m_{23}b_2 = h_2$$

$$m_{31}a + m_{32}b_1 + m_{33}b_2 = h_3$$

Dalam tugas ini, bentuk persamaan tersebut dapat dibuat matriks yang digunakan untuk melakukan penyelesaian dengan metode eliminasi Gauss

BAB 3

IMPLEMENTASI

Untuk lebih lengkapnya dapat dicek pada :
<https://github.com/handono206/Algeo01-20070>

3.1 Class Matrix

```
public class Matrix {  
    double[][] Mat;  
    private int Row;  
    private int Col;  
    int tukar; //jumlah penukaran  
  
    //KONSTRUKTOR  
    Matrix (String path){      //konstruktor dengan file  
        readMatrixFile(path);  
        this.tukar = 0;  
    }  
    Matrix(int Row, int Col){ //konstruktor tanpa file  
        this.Row = Row;  
        this.Col = Col;  
        this.Mat = new double[Row][Col];  
        this.tukar = 0;  
    }  
    //getter dan setter  
    public int getRow(){  
        return this.Row;  
    }  
    public int getCol(){  
        return this.Col;  
    }  
    public void setRow(int Row){  
        this.Row = Row;  
    }  
  
    public void setCol(int Col) {  
        this.Col = Col;  
    }  
  
    public int getTukar()  
    {  
        ...  
    }  
}
```

```

//Prosedur isi matrix
public void isiMatrix(){
    int i, j;
    Scanner input = new Scanner(System.in);
    for (i=0; i<this.Row; i++){
        for (j=0; j<this.Col; j++){
            // System.out.print("Masukkan Baris ke-"+(i+1)+" Kolom ke-"+(j+1));
            this.Mat[i][j] = input.nextDouble();
            // System.out.println();
        }
    }
}

//Prosedur tulis isi matrix
public void tulisMatrix(){
    int i, j;
    for (i=0; i<this.Row; i++){
        for (j=0; j<this.Col; j++){
            String elem = String.format("%.3f", this.Mat[i][j]);
            System.out.print(elem + " ");
        }
        System.out.println();
    }
    System.out.println();
}

public String tulisMatrixString(){
    int i, j;
    String s = "";
    for (i=0; i<this.Row; i++){
        for (j=0; j<this.Col; j++){
            String elem = String.format("%.3f", this.Mat[i][j]);
            s += (elem + " ");
        }
    }
}

public void readMatrixFile(String path)
{
    this.Row = 0;
    this.Col = 0;
    //membaca size matrix dari file
    try {
        File file = new File(path);
        Scanner reader = new Scanner(file);
        int i = 0;
        while (reader.hasNextLine())
        {
            this.Row+=1;
            Scanner colReader = new Scanner(reader.nextLine());
            while(colReader.hasNextDouble())
            {
                if (i == 0){this.Col+=1;}
                colReader.nextDouble();
                i += 1;
            }
            reader.close();
        }
        catch (FileNotFoundException e)
        {
            System.out.println("File not found.");
            e.printStackTrace();
        }
    }

    //isi matrix
    this.Mat = new double[this.Row][this.Col];
    File file = new File("../\\test\\matrix.txt");
    try{
        Scanner rowReader = new Scanner(file);
        for (int i = 0 ; i<this.Row ; i++)
    
```

```

        this.Mat = new double[this.Row][this.Col];
        File file = new File("../test\\matrix.txt");
        try{
            Scanner rowReader = new Scanner(file);
            for (int i = 0 ; i<this.Row ; i++)
            {
                Scanner colReader = new Scanner(rowReader.nextLine());
                for (int j = 0 ; j<this.Col ; j++)
                {
                    double data = colReader.nextDouble();
                    this.Mat[i][j] = data;
                }
                colReader.close();
            }
            rowReader.close();
        }
        catch (FileNotFoundException e)
        {
            System.out.println("File not found.");
            e.printStackTrace();
        }
    }

    public Matrix transpose(){
        int i, j;
        Matrix TMat = new Matrix(this.getCol(), this.getRow());
        for (i=0; i<this.Row; i++){
            for (j=0; j<this.Col; j++){
                TMat.Mat[j][i] = this.Mat[i][j];
            }
        }
        return TMat;
    }
}

```

```

public void kaliKons(int row, double x){
    for (int j=0; j<this.Col; j++){
        this.Mat[row-1][j] = this.Mat[row-1][j]*x;
    }
}

public void tukarBaris(int row1, int row2){
    double[] temp = new double[this.Col];
    for (int i=0; i < this.Col; i++){
        temp[i] = this.Mat[row1][i];
        this.Mat[row1][i] = this.Mat[row2][i];
    }
    for (int i=0; i<this.Col; i++){
        this.Mat[row2][i] = temp[i];
    }
    this.tukar+=1;
}

public double getElmt(int row, int col){
    return this.Mat[row][col];
}

public int getIndexCol(int row, double x){
    int j = 0;
    boolean found = false;
    while (!found && j<this.getCol()){
        if (this.Mat[row][j]==x){
            found = true;
        }
    }
    return j;
}

public boolean isOnRow (int row, double x){
    boolean flag = false;
    int j = 0;
    while (!flag && j<this.getCol()){
        if (this.Mat[row][j]== x){

```

```

public void roundMatrix ( double constraint){
    for (int i=0; i<this.getRow(); i++){
        for (int j=0; j<this.getCol(); j++){
            if (Math.abs(this.getElmt(i, j)) < constraint){
                this.Mat[i][j] = 0;
            }
        }
    }
}
public Matrix copyMatrix(){
    Matrix copyMat = new Matrix(this.Row, this.Col);
    for (int i=0; i<this.Row; i++){
        for (int j = 0; j<this.Col; j++){
            copyMat.Mat[i][j] = this.Mat[i][j];
        }
    }
    return copyMat;
}
public boolean isRowAllZero(int row){
    int j = 0;
    boolean flag = true;
    if (this.Mat[row][j] != 0){
        flag = false;
    }
    while (j<this.getCol()&& flag){
        if (this.Mat[row][j] != 0){
            flag = false;
        }
        else{
            j++;
        }
    }
    return flag;
}
public Matrix matCofactor()
{
    Matrix mc = new Matrix(this.getRow(), this.getCol());
    int a = 0, b = 0; //index m1
    int multiply = 1;
    Determinant det = new Determinant();
    double deter;
    for (int i = 0;i<this.getRow();i++)
    {
        if (i%2==0){multiply = 1;}
        else{multiply = -1;}
        for (int j = 0;j<this.getCol();j++)
        {
            Matrix m1 = new Matrix(this.getRow()-1, this.getCol()-1);
            for (int c = 0 ; c < this.getRow() ; c++)
            {
                for (int d = 0 ; d < this.getCol() ; d++)
                {
                    if (c!=i && d!=j)
                    {
                        m1.Mat[a][b] = this.getElmt(c, d);
                        if (b+1<m1.getCol()){b++;}
                        else if (a+1<m1.getRow()){a++;b=0;}
                    }
                }
            }
            a = 0;b=0;
            deter = det.detCofactor(m1);
            mc.Mat[i][j] = multiply*deter;
            multiply *= -1;
        }
    }
    return mc;
}

```

```

public Matrix adj()
{
    if (this.getRow() == 1 && this.getCol() ==1){
        this.Mat[0][0]=1; return this;}
    else{
        Matrix madj;
        madj = this.matCofactor();
        return madj.transpose();}
}

public Matrix inverse()
{
    Determinant det = new Determinant();
    double deter = det.detCofactor(this);
    Matrix Minverse = this.adj();
    for (int i=0 ; i < Minverse.getRow() ; i++)
    {
        for (int j=0 ; j < Minverse.getCol() ; j++)
        {
            Minverse.Mat[i][j]=(1/deter);
        }
    }
    return Minverse;
}
public int FirstNonZero (int i){
    int j = 0;
    if (this.Mat[i][j]==0){
        while (this.Mat[i][j] == 0 && j<this.Col){
            j++;
        }
    }
    if (j==this.Col){[]
        return -1;
    }
    else {[]
        return j;
    }
}
public boolean isKolomAllZero (int col){
    int i = 0;
    while (i<this.Row){
        if (this.Mat[i][col] != 0){
            return false;
        }
        i++;
    }
    return true;
}
public boolean fRowIsKolomAllZero (int row, int col){
    boolean flag = true;
    int i = row;
    while (flag && i<this.Row){
        if (this.Mat[i][col]!=0){
            flag = false;
        }
        i++;
    }
    return flag;
}
public int NonZeroElmt(int n, int m){
    int count = 0;
    for(int i=0;i<=m;i++){
        if (this.Mat[n][i]!=0){
            count++;
        }
    }
    return count;
}

```

```

public int Kemungkinan (Matrix matrix){
    int i = matrix.getRow()-1;
    if (matrix.isRowAllZero(i)){
        while (matrix.isRowAllZero(i)&&i>=0){
            i--;
        }
    }
    //mencari index pertama yg bukan 0 pada baris
    int x = matrix.FirstNonZero(i);
    if (x == matrix.getCol()-1){
        return 3;
    }
    else {
        int row = 0;
        int flag = matrix.FirstNonZero(row);
        // System.out.println(flag);
        if (flag == -1){
            return 2;
        }
        if (matrix.Mat[row][flag] != 0){
            while (row< matrix.getRow()&&matrix.Mat[row][flag]!=0&&flag< matrix.getCol()){
                row++;
                flag++;
            }
            // System.out.println(row);
            // System.out.println(flag);
            if (row == matrix.getCol()-1){
                return 1;
            }
            else{
                return 2;
            }
        }
        else {
            return 2;
        }
    }
}

public Matrix ElimMaju(Matrix matrix){
    int col = matrix.getCol();
    int row = matrix.getRow();

    for (int k=0; k<row; k++){
        //mencari dan mengurutkan row yang memiliki nol dari kiri
        // int max = 0;
        int [] arrayZero = new int[matrix.getRow()];
        for (int i = k; i< matrix.getRow(); i++){
            int j = 0;
            if (matrix.Mat[i][j] == 0){
                while (matrix.Mat[i][j]==0 && j< matrix.getCol()){
                    j++;
                    if(j>= matrix.getCol()){
                        break;
                    }
                }
            }
            arrayZero[i] = j;
        }
        //mengurutkan matriks dari menjadi matriks eselon
        // System.out.println(Arrays.toString(arrayZero));
        // System.out.println();
        Matrix temp = matrix.copyMatrix();
        // temp.tulisMatrix();
        // System.out.println();
    }
}

```

```

Matrix temp = matrix.copyMatrix();
int[] sortedZero = Arrays.copyOf(arrayZero, arrayZero.length);
Arrays.sort(sortedZero);
if (!Arrays.equals(arrayZero, sortedZero)){
    for (int i = k; i< arrayZero.length; i++){
        int x = 0;

        while((arrayZero[i] != sortedZero[x])){
            x++;
        }

        sortedZero[x] = matrix.getCol();
        matrix.Mat[x] = temp.Mat[i];
        matrix.tukar += 1;
    }
}

if (col>row){
    for (int i = k+1; i<row; i++){
        double divider;
        int tempRow = k;
        //scan index row pertama yang bukan 0
        if (matrix.Mat[k][tempRow] == 0){
            while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getRow()){
                tempRow++;
                if (tempRow>= temp.getRow()){
                    break;
                }
            }
        }

        if (matrix.Mat[k][tempRow] != 0){
            divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
            for (int j = k+1; j<col; j++){
                matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                Rounder(matrix, 1e-9);
            }
            matrix.Mat[i][k] = 0;
        }
    }
}
else {
    for (int i = k+1; i<col; i++){
        double divider;
        int tempRow = k;
        //scan index row pertama yang bukan 0
        if (matrix.Mat[k][tempRow] == 0){
            while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getCol()){
                tempRow++;
                if (tempRow>= temp.getRow()){
                    break;
                }
            }
        }

        if (matrix.Mat[k][tempRow] != 0){
            divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
            for (int j = 0; j<col; j++){
                matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                Rounder(matrix, 1e-9);
            }
            matrix.Mat[i][k] = 0;
        }
    }
}
}

```

```

        for (int r = 0; r< matrix.getRow(); r++){
            int c = 0;
            if (matrix.Mat[r][c] == 0){
                while (matrix.Mat[r][c]==0 && c< matrix.getCol()){
                    c++;
                    if (c>= matrix.getCol()){
                        break;
                    }
                }
            }

            if (c< matrix.getCol()){
                double divider = matrix.Mat[r][c];
                for (int tcol = 0; tcol< matrix.getCol(); tcol++){
                    matrix.Mat[r][tcol]=matrix.Mat[r][tcol]/divider;
                }
            }
            //else berarti barisnya isinya 0 semua
        }
        Rounder(matrix, 1e-9);
        return matrix;
    }
    public void Rounder ( Matrix matrix, double constraint){
        for (int i=0; i<matrix.getRow(); i++){
            for (int j=0; j<matrix.getCol(); j++){
                if (Math.abs(matrix.Mat[i][j]) < constraint){
                    matrix.Mat[i][j] = 0;
                }
            }
        }
    }
}

```

3.2 Class Point.java

```

package com.matrixxx;

public class Point {
    double x;
    double y;

    //Konstruktor
    Point(){}
    Point(double X, double Y)
    {
        this.x = X;
        this.y = Y;
    }

    public double getX()
    {
        return this.x;
    }

    public double getY()
    {
        return this.y;
    }

    public void setX(double x_arg)
    {
        this.x = x_arg;
    }
}

```

```
    public void setY(double y_arg)
    {
        this.y = y_arg;
    }
}
```

3.3 Class Determinant.java

```
package com.matrixxx;

import java.util.Arrays;
import java.lang.Math;

public class Determinant {
    public void call(int choice, Matrix mat) //1 = kofaktor, 2 = reduksi baris
    {
        if (choice == 1)
        {
            double det = detCofactor(mat);
            String strdet = String.format("%.3f",det); //3 decimals floating point
            System.out.println("Determinan = " + strdet);
        }

        else if (choice == 2)
        {

            double det = detReduction(mat);
            String strdet = String.format("%.3f",det); //3 decimals floating point
            System.out.println("Determinan = " + strdet);
        }
    }

    public double detCofactor(Matrix m)
    {
        int row = m.getRow();
        int col = m.getCol();
        double det = 0.0;
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                double minor = getMinor(m, i, j);
                double cofactor = Math.pow(-1, i + j) * minor;
                det += cofactor;
            }
        }
        return det;
    }

    private double getMinor(Matrix m, int row, int col)
    {
        int newRow = 0;
        int newRowCount = 0;
        int newCol = 0;
        int newColCount = 0;
        for (int i = 0; i < m.getRow(); i++)
        {
            if (i == row)
            {
                continue;
            }
            for (int j = 0; j < m.getCol(); j++)
            {
                if (j == col)
                {
                    continue;
                }
                if (newRowCount < m.getRow() - 1)
                {
                    newRow += 1;
                    newCol += 1;
                }
                else
                {
                    newCol += 1;
                }
            }
            newRowCount += 1;
        }
        Matrix minor = new Matrix(newRow, newCol);
        for (int i = 0; i < minor.getRow(); i++)
        {
            for (int j = 0; j < minor.getCol(); j++)
            {
                minor.setElement(i, j, m.getElement(i, j));
            }
        }
        return minor.det();
    }
}
```

```

public double detCofactor(Matrix m)
{
    int row = m.getRow();
    int col = m.getCol();
    double det = 0, res;
    if (row==1 && col==1) //base 1
    {
        double a = m.getElmt( row: 0, col: 0);
        return a;
    }

    else if (row==2 && col==2) //base 2
    {
        double a;
        a = ((m.getElmt( row: 0, col: 0)*m.getElmt( row: 1, col: 1)) - (m.getElmt( row: 0, col: 1)*m.getElmt( row: 1, col: 0)));
        return a;
    }

    else //rekurens
    {

        int i;
        for (i=0;i<col;i++)
        {
            if (i%2==0)
            {
                Matrix m1 = new Matrix( Row: row-1, Col: col-1);
                int a,b,c,d;
                c = 0;d = 0;
                else //rekurens
                {

                    int i;
                    for (i=0;i<col;i++)
                    {
                        if (i%2==0)
                        {
                            Matrix m1 = new Matrix( Row: row-1, Col: col-1);
                            int a,b,c,d;
                            c = 0;d = 0;
                            for (a=0;a<row;a++)
                            {
                                for (b=0;b<col;b++)
                                {
                                    if (!(a==0 || b==i))
                                    {
                                        m1.Mat[c][d] = m.Mat[a][b];
                                        if (d<m1.getCol()-1){d+=1;}
                                        else{c+=1;d=0;}
                                    }
                                }
                            }
                            // res = ELM(m,0,i)*determinant(m1);
                            res = m.Mat[0][i]*detCofactor(m1);
                            det+= res;
                        }
                    }
                }
            }
        }
    }
}

```

```

        else
    {
        Matrix m1;
        m1 = new Matrix( Row: row-1, Col: col-1);
        int a,b,c,d;
        c = 0;d = 0;
        for (a=0;a<row;a++)
        {
            for (b=0;b<col;b++)
            {
                if (!(a==0 || b==i))
                {
                    m1.Mat[c][d] = m.Mat[a][b];
                    if (d<m1.getCol()-1){d+=1;}
                    else{c+=1;d=0;}
                }
            }
            res = -(m.getElmt( row: 0 ,i)*detCofactor(m1));
            det += res;
        }
    }
    return det;
}

public double detReduction (Matrix m)
{
    double deter = 1;
    m = ElimMajuDet(m);
    for (int i = 0; i<m.getRow() ; i++)
    {
        deter *= m.getElmt(i, i);
    }
    deter *= Math.pow(-1,m.getTukar());
    return deter;
}

public int Kemungkinan (Matrix matrix){
    int i = matrix.getRow()-1;
    if (matrix.isRowAllZero(i)){
        while (matrix.isRowAllZero(i)&&i>=0){
            i--;
        }
    }
    //mencari index pertama yg bukan 0 pada baris
    int x = matrix.FirstNonZero(i);
    if (x == matrix.getCol()-1){
        return 3;
    }
    else {
        int row = 0;
        int flag = matrix.FirstNonZero(row);
        // System.out.println(flag);
        if (flag == -1){

```

```

        if (_flag == -1){
            return 2;
        }
        if (matrix.Mat[_row][_flag] != 0){
            while (_row < matrix.getRow() && matrix.Mat[_row][_flag] != 0 && _flag < matrix.getCol()){
                _row++;
                _flag++;
            }

            if (_row == matrix.getCol()-1){
                return 1;
            }
            else{
                return 2;
            }
        }
        else {
            return 2;
        }
    }
}

public Matrix ElimMajuDet(Matrix matrix){
    int col = matrix.getCol();
    int row = matrix.getRow();

    for (int k=0; k<row; k++){
        for (int k=0; k<row; k++){
            //mencari dan mengurutkan row yang memiliki nol dari kiri
            // int max = 0;
            int [] arrayZero = new int[matrix.getRow()];
            for (int i = k; i< matrix.getRow(); i++){
                int j = 0;
                if (matrix.Mat[i][j] == 0){
                    while (matrix.Mat[i][j]==0 && j< matrix.getCol()){
                        j++;
                        if(j>= matrix.getCol()){
                            break;
                        }
                    }
                }
                arrayZero[i] = j;
            }
            //mengurutkan matriks dari menjadi matriks eselon

            Matrix temp = matrix.copyMatrix();

            int[] sortedZero = Arrays.copyOf(arrayZero, arrayZero.length);
            Arrays.sort(sortedZero);

            if (!Arrays.equals(arrayZero, sortedZero)){
                for (int i = k; i< arrayZero.length; i++){
                    int x = 0;

                    while((arrayZero[i] != sortedZero[x])){

```

```

        while((arrayZero[i] != sortedZero[x])){
            x++;
        }
        sortedZero[x] = matrix.getCol();
        matrix.Mat[x] = temp.Mat[i];
        matrix.tukar += 1;
    }
}

if (col>row){
    for (int i = k+1; i<row; i++){
        double divider;
        int tempRow = k;
        //scan index row pertama yang bukan 0
        if (matrix.Mat[k][tempRow] == 0){
            while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getRow()){
                tempRow++;
                if (tempRow>= temp.getRow()){
                    break;
                }
            }
        }
        if (matrix.Mat[k][tempRow] != 0){
            divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
            for (int j = k+1; j<col; j++){
                matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                Rounder(matrix, constraint: 1e-9);
            }
        }
        matrix.Mat[i][k] = 0;
    }
}
}

```

```

        for (int r = 0; r< matrix.getRow(); r++){
            int c = 0;
            if (matrix.Mat[r][c] == 0){
                while (matrix.Mat[r][c]==0 && c< matrix.getCol()){
                    c++;
                    if (c>= matrix.getCol()){
                        break;
                    }
                }
            }
            if (c< matrix.getCol()){
                // double divider = matrix.Mat[r][c];
                for (int tcol = 0; tcol< matrix.getCol(); tcol++){
                    // matrix.Mat[r][tcol]=matrix.Mat[r][tcol]/divider;
                }
            }
            //else berarti barisnya isinya 0 semua
        }
        Rounder(matrix, constraint: 1e-9);
        return matrix;
    }

    public void Rounder ( Matrix matrix, double constraint){
        for (int i=0; i<matrix.getRow(); i++){
            for (int j=0; j<matrix.getCol(); j++){
                if (Math.abs(matrix.Mat[i][j]) < constraint){
                    matrix.Mat[i][j] = 0;
                }
            }
        }
    }
}

```

3.4 Class Points.java

```

package com.matrixxx;

import java.util.Scanner;
import java.io.FileNotFoundException;
import java.io.File;

public class Points {
    Point[] points; //array of Point

    //Konstruktor
    Points(){
        readPointsFile(); //dengan file points.txt
    }
    Points(int n)
    {
        this.points = new Point[n]; //tanpa file points.txt
    }

    public void readPoints()
    {

        System.out.println("Masukkan titik: ");
        Scanner sc = new Scanner(System.in);
        for(int i = 0 ; i<this.points.length ; i++)
        {
            double x = sc.nextDouble();
            double y = sc.nextDouble();
            Point P = new Point(x,y);
            this.points[i] = P;
        }
    }
}

```

```

        }

        sc.close();
    }

    public Point getElmt(int i) { return this.points[i]; }

    public void writePoints()
    {
        for (int i = 0 ; i<this.points.length ; i++)
        {
            String x = String.format("%.3f", getElmt(i).getX()); //3 decimals floating point
            String y = String.format("%.3f", getElmt(i).getY());
            System.out.println("(" + x + " , " + y + ")");
        }
    }

    public void readPointsFile()
    {
        File file = new File( pathname: "txt\\points.txt");
        try{
            Scanner lineReader = new Scanner(file);
            int n = 0;
            while(lineReader.hasNextLine())
            {
                lineReader.nextLine();
                n+=1;
            }
            this.points = new Point[n];
            lineReader.close();

            Scanner reader = new Scanner(file);
            for (int i=0 ; i<n ; i++)
            {
                Scanner num = new Scanner(reader.nextLine());
                double x_in = num.nextDouble();
                double y_in = num.nextDouble();
                Point P = new Point(x_in, y_in);
                this.points[i] = P;
                num.close();
            }
            reader.close();
        }
        catch (FileNotFoundException e)
        {
            System.out.println("File not found.");
            e.printStackTrace();
        }
    }
}

```

3.5 Class Cramer.java

```

package com.matrixxx;

public class Cramer {
    public double[] call(Matrix mat)
    {
        double[] solusi = new double[mat.getCol()-1];
        Matrix mat2 = new Matrix(mat.getCol()-1,mat.getCol()-1);

        double[] y = new double[mat.getRow()];
        int i,j;

        for (i=0;i<mat.getRow();i++) //menyimpan semua nilai y
        {
            y[i] = mat.getElmt(i, mat.getCol()-1);
        }

        for (i=0 ; i<mat2.getRow();i++) //mat2 sebagai isi x nya saja dari mat
        {
            for (j=0 ; j<mat2.getCol();j++)
            {
                mat2.Mat[i][j] = mat.getElmt(i, j);
            }
        }

        Determinant determinant = new Determinant();
        double detCram = 0.0;
        double det = determinant.detCofactor(mat2);
        if (det==0.0){
            solusi = new double[0];
            return solusi;} //determinan nol tidak ada solusi

        Matrix mat3 = mat2.copyMatrix(); //copy mat2 ke mat3 untuk perubahan matriks di cramer
    }
}

```

```

for(i=0 ; i<mat2.getCol() ;i++) //perhitungan cramer
{
    for (j=0 ; j<mat2.getRow() ;j++)
    {
        mat3.Mat[j][i] = y[j];      //penggantian elemen dengan y
    }
    detCram = determinant.detCofactor(mat3);
    solusi[i] = detCram/det; //pengisian solusi x1, x2, ..., xn
    mat3 = mat2.copyMatrix(); //kembalikan seperti semula
}

return solusi;
}
}

```

3.6 Class Gauss.java

```
public class Gauss {

    public static int Kemungkinan (Matrix matrix){
        int i = matrix.getRow()-1;
        if (matrix.isRowAllZero(i)){
            while (matrix.isRowAllZero(i)&&i>=0){
                i--;
            }
        }
        //mencari index pertama yg bukan 0 pada baris
        int x = matrix.FirstNonZero(i);
        if (x == matrix.getCol()-1){
            return 3;
        }
        else {
            int row = 0;
            int flag = matrix.FirstNonZero(row);
            // System.out.println(flag);
            if (flag == -1){
                return 2;
            }
            if (matrix.Mat[row][flag] != 0){
                while (row< matrix.getRow()&&matrix.Mat[row][flag]!=0&&flag< matrix.getCol()){
                    row++;
                    flag++;
                }
                // System.out.println(row);
                // System.out.println(flag);
                if (row == matrix.getCol()-1){
                    return 1;
                }
                else{
                    return 2;
                }
            }
        }
    }
}
```

```

        }
    else {
        return 2;
    }
}

public static void GaussSolver(Matrix matrix, String write, String toFile, Save file)
{
    int x = Kemungkinan(matrix);
    if (x==3){
        if (write == "SPL"){
            System.out.println("SPL Tidak Memiliki Solusi!");
            toFile += "Tidak Memiliki Solusi!\n";
            file.write(toFile);
        }
        else {
            System.out.println("Polinom Interpolasi tidak dapat dicari untuk titik-titik ini!");
            toFile += "Polinom Interpolasi tidak dapat dicari untuk titik-titik ini!\n";
            file.write(toFile);
        }
    }
    else if (x==1){
        double[] solusi = new double[matrix.getCol()-1];
        for (int sol = 0; sol<matrix.getCol()-1; sol++){
            solusi[sol] = matrix.Mat[sol][matrix.getCol()-1];
        }
        for (int i = solusi.length-1; i >=0; i--){
            for (int j = matrix.getCol()-2; j > i; j--){
                solusi[i] -= solusi[j]*matrix.Mat[i][j];
            }
        }
        if (write == "SPL"){
            System.out.println("SOLUSI:");
        }
    }
}

```

```

        System.out.println("SOLUSI:");
        toFile += "\nSOLUSI:\n";
        for (int i=0; i< solusi.length; i++){
            System.out.println("x"+(i+1)+" = "+solusi[i]);
            toFile += "x"+(i+1)+" = "+solusi[i] +"\n";
        }
        file.write(toFile);
    }

    else if (write == "polinom"){
        System.out.println("Polinom interpolasi yang melewati titik-titik tersebut adalah:");
        toFile += "\nPolinom interpolasi yang melewati titik-titik tersebut adalah:\n";
        System.out.print("P(x) = ");
        toFile += "P(x) = ";
        String tanda;
        for (int i=0; i<solusi.length; i++){
            if (solusi[i]!=0){
                if (solusi[i] < 0 ){
                    tanda = " - ";
                }
                else {
                    if (i == 0){
                        tanda = " ";
                    }
                    else {
                        tanda = " + ";
                    }
                }
                if (i!=0){
                    if (Math.abs(solusi[i]) != 1){
                        System.out.print(tanda + Math.abs(solusi[i]) + "x^" + i);
                        toFile += tanda + Math.abs(solusi[i]) + "x^" + i;
                    }
                    else {
                        System.out.print(tanda + "x^" + i);
                        toFile += tanda + "x^" + i;
                    }
                }
                else {
                    System.out.print(tanda + Math.abs(solusi[i]));
                    toFile += tanda + Math.abs(solusi[i]);
                }
            }
            //System.out.print(solusi[i]);
        }
        tofile += "\n";
        System.out.println();
        boolean flag = true;
        double solve;
        Scanner sc = new Scanner(System.in);
        while (flag){
            double sum = 0;
            System.out.print("Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): ");
            toFile += "\nMasukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): ";
            solve = sc.nextDouble();
            toFile += solve + "\n";
            if (solve == 999.999){
                System.out.println("-----KEMBALI KE MENU UTAMA-----");
                toFile += "-----KEMBALI KE MENU UTAMA-----\n";
                flag = true;
                break;
            }
            else [
                System.out.print("P("+solve+") = ");
                toFile += "P("+solve+") = ";
                for (int i = 0; i<solusi.length; i++){
                    sum += solusi[i]*Math.pow(solve, i);
                }
            ]
        }
    }
}

```

```

        }
        System.out.println(sum);
        toFile += sum + "\n";
    }
    file.write(toFile);
}
else {
    //mencari row pertama dari bawah yang elemennya tidak 0 (solusi banyak)
    int idxNotZero = matrix.getRow()-1;
    while (matrix.isRowAllZero(idxNotZero)&&idxNotZero>=0){
        idxNotZero--;
    }
    int countNonZero = idxNotZero+1;
    System.out.println(idxNotZero);
    // Solusi disimpan dalam bentuk array 2d dengan jml baris = jml variabel yang dicari
    // dan kolom adalah jumlah baris yang elemennya tidak nol pada matriks eselon + 4.
    //kolom 1 = apakah variabel tersebut dijadikan solusi parametrik?
    //kolom 2 = apakah variabel tersebut memiliki solusi tunggal?
    //kolom 3 = urutan variabel parametrik
    //kolom 4 = nilai solusi tunggal
    // *benar == 1, salah == 0 (untk kolom 1 & 2)

    double[][] solusi = new double[matrix.getCol()-1][matrix.getCol()-countNonZero+3];
    boolean[] declared = new boolean[matrix.getCol()-1];
    System.out.println(Arrays.toString(declared));
    System.out.println(Arrays.deepToString(solusi));

    int xParam = 0;
    int cr2 = countNonZero-1;
    int cc2;

    //kolom yang isinya nol semua pasti variabel pada kolom tersebut parametrik
    for (int j = 0; j< matrix.getCol()-1;j++){
        //kolom yang isinya nol semua pasti variabel pada kolom tersebut parametrik
        for (int j = 0; j< matrix.getCol()-1;j++){
            if (matrix.isKolomAllZero(j)){
                declared[j] = true;
                solusi[j][matrix.getCol()-countNonZero-1] = 1;
                solusi[j][matrix.getCol()-countNonZero+1] = xParam;
                xParam++;
            }
        }

        //mencari variabel lain yang juga merupakan variabel parametrik
        while(xParam < matrix.getCol()-countNonZero-1) {
            cc2 = matrix.getCol() - 2;
            while (cc2 >= 0 && xParam < matrix.getCol() - countNonZero - 1) {
                if (!declared[cc2] && matrix.Mat[cr2][cc2] != 0) {
                    if (matrix.NonZeroElmt(cr2, matrix.getCol() - 2) == 1) {

                        declared[cc2] = true;
                        solusi[cc2][matrix.getCol() - countNonZero] = 1;
                        solusi[cc2][matrix.getCol() - countNonZero + 2] = matrix.Mat[cr2][matrix.getCol() - 1] / matrix.Mat[cr2][cc2];

                    } else if (cc2 != matrix.FirstNonZero(cr2) && matrix.fRowIsKolomAllZero(cc2 + 1, cc2)) {
                        declared[cc2] = true;
                        solusi[cc2][matrix.getCol() - countNonZero - 1] = 1;
                        solusi[cc2][matrix.getCol() - countNonZero + 1] = xParam;
                        xParam++;
                    }
                }
            cc2--;
        }
        cr2--;
    }
}

```

```

        for (int i=cr2;i>=0;i--) {
            cc2 = matrix.FirstNonzero(i);
            if (!declared[cc2]){
                declared[cc2] = true;
                solusi[cc2][matrix.getCol()-countNonZero+2] = matrix.Mat[i][matrix.getCol()-1];
                System.out.println(Arrays.deepToString(solusi));
                for (int k= matrix.getCol()-2; k>=cc2+1 ;k--){
                    if (solusi[k][matrix.getCol()-countNonZero-1]==0 && solusi[k][matrix.getCol()-countNonZero]==0){
                        for (int j=0; j<=xParam-1; j++){
                            solusi[cc2][j] += solusi[k][j] * -(matrix.Mat[i][k]);
                        }
                        solusi[cc2][matrix.getCol()-countNonZero+2] += solusi[k][matrix.getCol()-countNonZero+2]*-(matrix.Mat[i][k]);
                    }
                    else if (solusi[k][matrix.getCol()-countNonZero-1]==1 && solusi[k][matrix.getCol()-countNonZero]==0){
                        solusi[cc2][(int)solusi[k][matrix.getCol()-countNonZero+1]]+=matrix.Mat[i][k];
                    }
                    else{
                        solusi[cc2][matrix.getCol()-countNonZero+2] -= solusi[k][matrix.getCol()-countNonZero+2]*matrix.Mat[i][k];
                    }
                }
            }
        }
    }
    // tulis hasil solusi
    DecimalFormat df = new DecimalFormat("#.##");
    String[] kons = {"r","s","t","u","v","w","x","y","z"};
    System.out.println("SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi: ");
    for (int i = 0; i<= matrix.getCol()-2; i++){
        if(solusi[i][matrix.getCol()-countNonZero-1]==0 && solusi[i][matrix.getCol()-countNonZero]==0){
            // tulis hasil solusi
            DecimalFormat df = new DecimalFormat("#.##");
            String[] kons = {"r","s","t","u","v","w","x","y","z"};
            System.out.println("SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi: ");
            for (int i = 0; i<= matrix.getCol()-2; i++){
                if(solusi[i][matrix.getCol()-countNonZero-1]==0 && solusi[i][matrix.getCol()-countNonZero]==0){
                    System.out.print("x"+(i+1)+" = ");
                    toFile += "x"+(i+1)+" = ";
                    if (solusi[i][matrix.getCol()-countNonZero+2]!=0){
                        System.out.print(df.format(solusi[i][matrix.getCol()-countNonZero+2]));
                        toFile += df.format(solusi[i][matrix.getCol()-countNonZero+2]);
                    }
                    for (int j=0; j<=xParam-1;j++){
                        int output = 0;
                        if (solusi[i][j]!=0){
                            if (-solusi[i][j]>0 && (output!=0 || solusi[i][matrix.getCol()-countNonZero+2]!=0)){
                                System.out.print("+");
                                toFile += "+";
                            }
                            if (-solusi[i][j]==-1){
                                System.out.print("-");
                                toFile += "-";
                            }
                            if (Math.abs(solusi[i][j])!=1){
                                System.out.print(df.format(-(solusi[i][j])));
                                toFile += df.format(-(solusi[i][j]));
                            }
                            System.out.print(kons[j]);
                            toFile += kons[j];
                            output++;
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        System.out.println();
        toFile += "\n";
    }
    else if (solusi[i][matrix.getCol()-countNonZero-1]==1 && solusi[i][matrix.getCol()-countNonZero]==0){
        System.out.print("x"+(i+1)+" = "+kons[(int)solusi[i][matrix.getCol()-countNonZero+1]]);
        toFile += "x"+(i+1)+" = "+kons[(int)solusi[i][matrix.getCol()-countNonZero+1]];
        System.out.println();
        toFile += "\n";
    }
    else {
        System.out.print("x"+(i+1)+" = "+df.format(solusi[i][matrix.getCol()-countNonZero+2]));
        toFile += "x"+(i+1)+" = "+df.format(solusi[i][matrix.getCol()-countNonZero+2]);
        System.out.println();
        toFile += "\n";
    }
}

}

public static double[] GaussSolverFunction(Matrix matrix){
    int x = Kemungkinan(matrix);
    double[] solusi = {};
    if (x==3){
        System.out.println("SPL Tidak Memiliki Solusi!");
        return solusi;
    }
    else if (x==1){
        solusi = new double[matrix.getCol()-1];
        for (int sol = 0; sol<matrix.getCol()-1; sol++){
            solusi[sol] = matrix.Mat[sol][matrix.getCol()-1];
        }
    }
}

```

```

        }
        for (int i = solusi.length-1; i >=0; i--){
            for (int j = matrix.getCol()-2; j > i; j--){
                solusi[i] -= solusi[j]*matrix.Mat[i][j];
            }
        }
        return solusi;
    }
    else {
        return solusi;
    }
}

public static void ElimMaju(Matrix matrix){
    int col = matrix.getCol();
    int row = matrix.getRow();

    for (int k=0; k<row; k++){
        //mencari dan mengurutkan row yang memiliki nol dari kiri
        int max = 0;
        int [] arrayZero = new int[matrix.getRow()];
        for (int i = k; i< matrix.getRow(); i++){
            int j = 0;
            if (matrix.Mat[i][j] == 0){
                while (matrix.Mat[i][j]==0 && j< matrix.getCol()){
                    j++;
                    if(j>= matrix.getCol()){
                        break;
                    }
                }
                arrayZero[i] = j;
            }
        }
        for (int i = k+1; i< matrix.getRow(); i++){
            if (arrayZero[i] < max)
                continue;
            for (int j = 0; j< matrix.getCol(); j++){
                if (matrix.Mat[i][j] != 0)
                    matrix.Mat[i][j] /= matrix.Mat[i][arrayZero[i]];
                matrix.Mat[i][j] *= matrix.Mat[k][arrayZero[i]];
            }
        }
    }
}

```

```

}
//mengurutkan matriks dari menjadi matriks eselon
Matrix temp = matrix.copyMatrix();
int[] sortedZero = Arrays.copyOf(arrayZero, arrayZero.length);
Arrays.sort(sortedZero);
if (!Arrays.equals(arrayZero, sortedZero)){
    for (int i = k; i< arrayZero.length; i++){
        int x = 0;
        System.out.println(arrayZero[i]);
        while((arrayZero[i] != sortedZero[x])){
            x++;
        }
        sortedZero[x] = matrix.getCol();
        matrix.Mat[x] = temp.Mat[i];
    }
}

if (col>row){
    for (int i = k+1; i<row; i++){
        double divider;
        int tempRow = k;
        //scan index row pertama yang bukan 0
        if (matrix.Mat[k][tempRow] == 0){
            while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getRow()){
                tempRow++;
                if (tempRow>= temp.getRow()){
                    break;
                }
            }
        }
        if (matrix.Mat[k][tempRow] != 0){
            divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
        }
    }
}

```

```

        for (int j = k+1; j<col; j++){
            System.out.println(matrix.Mat[i][j]);
            System.out.print(matrix.Mat[k][j] + " ");
            System.out.println(divider);
            matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
            Rounder(matrix, 1e-9);
            System.out.println(matrix.Mat[i][j]);
            System.out.println();
        }
    }
    matrix.Mat[i][k] = 0;
}
else {
    for (int i = k+1; i<col; i++){
        double divider;
        int tempRow = k;
        //scan index row pertama yang bukan 0
        if (matrix.Mat[k][tempRow] == 0){
            while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getCol()){
                tempRow++;
                if (tempRow>= temp.getRow()){
                    break;
                }
            }
        }
        if (matrix.Mat[k][tempRow] != 0){
            divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
            for (int j = 0; j<col; j++){
                System.out.println(matrix.Mat[i][j]);
                System.out.print(matrix.Mat[k][j] + " ");
                System.out.println(divider);
                matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                Rounder(matrix, 1e-9);

                for (int j = k+1; j<col; j++){
                    System.out.println(matrix.Mat[i][j]);
                    System.out.print(matrix.Mat[k][j] + " ");
                    System.out.println(divider);
                    matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                    Rounder(matrix, 1e-9);
                }
            }
            matrix.Mat[i][k] = 0;
        }
    }
    else {
        for (int i = k+1; i<col; i++){
            double divider;
            int tempRow = k;
            //scan index row pertama yang bukan 0
            if (matrix.Mat[k][tempRow] == 0){
                while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getCol()){
                    tempRow++;
                    if (tempRow>= temp.getRow()){
                        break;
                    }
                }
            }
            if (matrix.Mat[k][tempRow] != 0){
                divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
                for (int j = 0; j<col; j++){
                    System.out.println(matrix.Mat[i][j]);
                    System.out.print(matrix.Mat[k][j] + " ");
                    System.out.println(divider);
                    matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                    Rounder(matrix, 1e-9);
                }
            }
        }
    }
}

```

```

        Rounder(matrix, 1e-9);
        System.out.println(matrix.Mat[i][j]);
        System.out.println();
    }
    matrix.Mat[i][k] = 0;
}
else {
    for (int i = k+1; i<col; i++){
        double divider;
        int tempRow = k;
        //scan index row pertama yang bukan 0
        if (matrix.Mat[k][tempRow] == 0){
            while (matrix.Mat[k][tempRow] == 0 && tempRow< matrix.getCol()){
                tempRow++;
                if (tempRow>= temp.getRow()){
                    break;
                }
            }
            if (matrix.Mat[k][tempRow] != 0){
                divider = matrix.Mat[i][tempRow]/matrix.Mat[k][tempRow];
                for (int j = 0; j<col; j++){
                    System.out.println(matrix.Mat[i][j]);
                    System.out.print(matrix.Mat[k][j] + " ");
                    System.out.println(divider);
                    matrix.Mat[i][j] -= matrix.Mat[k][j] * divider;
                    Rounder(matrix, 1e-9);
                    System.out.println(matrix.Mat[i][j]);
                    System.out.println();
                }
            }
            matrix.Mat[i][k] = 0;
        }
    }
}

```

```
        }
        matrix.Mat[i][k] = 0;
    }

}

for (int r = 0; r< matrix.getRow(); r++){
    int c = 0;
    if (matrix.Mat[r][c] == 0){
        while (matrix.Mat[r][c]==0 && c< matrix.getCol()){
            c++;
            if (c>= matrix.getCol()){
                break;
            }
        }
    }

    if (c< matrix.getCol()){
        double divider = matrix.Mat[r][c];
        for (int tcol = 0; tcol< matrix.getCol(); tcol++){
            matrix.Mat[r][tcol]=matrix.Mat[r][tcol]/divider;
        }
    }
    //else berarti barisnya isinya 0 semua
}
Rounder(matrix, 1e-9);
}
```

```

public static void Rounder ( Matrix matrix, double constraint){
    for (int i=0; i<matrix.getRow(); i++){
        for (int j=0; j<matrix.getCol(); j++){
            if (Math.abs(matrix.Mat[i][j]) < constraint){
                matrix.Mat[i][j] = 0;
            }
        }
    }
}

//membuat matriks eselon tereduksi (eliminasi gauss Jordan)
public static void reducedEF (Matrix mat){
    for (int i = mat.getRow()-1; i>0; i--){
        boolean flag = false;
        int j = 0;
        while (!flag && j< mat.getCol()){
            if (mat.Mat[i][j]==1){
                for (int n = i-1; n>=0; n--){
                    double multiplier = -(mat.Mat[n][j]/mat.Mat[i][j]);
                    for (int m = 0; m< mat.getCol(); m++){
                        mat.Mat[n][m] += mat.Mat[i][m] * multiplier;
                    }
                }
                flag = true;
            }
            j++;
        }
    }
}

```

3.7 Class InverseSPL.java

```

package com.matrixxx;

public class InverseSPL {
    public double[] inverseSPL(Matrix mat)
    {
        //Ax = B --> x = (A^-I) B
        double[] solusi = new double[mat.getCol()-1]; //solusi x1, x2, x3,...,xn
        Matrix mat2 = new Matrix(mat.getCol()-1,mat.getCol()-1); //Matriks A
        double[] B = new double[mat.getRow()];

        int i,j;
        double sum;

        for (i=0 ; i<mat.getRow() ; i++) //matriks B
        {
            B[i] = mat.getElmt(i, mat.getCol()-1);
        }

        for (i=0 ; i<mat2.getRow();i++) //Pengisian Matriks A dari mat
        {
            for (j=0 ; j<mat2.getCol();j++)
            {
                mat2.Mat[i][j] = mat.getElmt(i, j);
            }
        }

        mat2 = mat2.inverse();

        for (i=0 ; i<mat2.getRow() ; i++) //(A^-I) B = x
        {
            sum = 0;
            for (j=0 ; j<mat2.getCol() ; j++)
            {
                sum += (mat2.getElmt(i, j) * B[j]);
            }
            solusi[i] = sum;
        }

        return solusi;
    }
}

```

3.8 Class Main.java

The image shows a Windows desktop environment with two separate browser windows open, both displaying the same Java code from a GitHub repository. The top window is positioned above the taskbar, and the bottom window is below it. Both windows have the URL <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>.

```
1 package com.matrixxx;
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args)
7     {
8         int choice = 0;
9         Scanner sc = new Scanner(System.in);
10        do
11        {
12            showMainMenu();
13            showCommand();
14            choice = sc.nextInt();
15            MainMenuRoute(choice);
16        }
17        while(choice != 6);
18        sc.close();
19    }
20
21    public static void showMainMenu()
22    {
```

```
21    public static void showMainMenu()
22    {
23        System.out.println("MENU");
24        System.out.println("1. Sistem Persamaan Linear");
25        System.out.println("2. Determinan");
26        System.out.println("3. Matriks Balikan");
27        System.out.println("4. Interpolasi Polinom");
28        System.out.println("5. Regresi Linear Berganda");
29        System.out.println("6. Keluar");
30    }
31
32    public static void showCommand()
33    {
34        System.out.print(">>> ");
35    }
36
37    public static void showSubSPLMenu()
38    {
39        System.out.println("1. Metode Eliminasi Gauss");
40        System.out.println("2. Metode Eliminasi Gauss-Jordan");
41        System.out.println("3. Metode Matriks Balikan");
42        System.out.println("4. Kaidah Cramer");
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both tabs is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program with a menu system. The top window has line numbers 39 to 60 visible, while the bottom window has line numbers 45 to 66 visible. Both windows show the same code structure, including a main menu route method and various case statements for different menu options.

```
39     System.out.println("1. Metode Eliminasi Gauss");
40     System.out.println("2. Metode Eliminasi Gauss-Jordan");
41     System.out.println("3. Metode Matriks Balikan");
42     System.out.println("4. Kaidah Cramer");
43   }
44
45   public static void mainMenuRoute(int c)
46   {
47     switch (c)
48     {
49       case 1 :
50         SPL();
51         break;
52
53       case 2 :
54         determinan();
55         break;
56
57       case 3:
58         Inverse();
59         break;
60
61       case 4:
62         Interpolasi();
63         break;
64
65       case 5:
66         mlr();
```

```
65         case 5:
66             mlr();
67             break;
68
69         case 6: break;
70
71     default:
72         System.out.println("Invalid Command");
73         break;
74     }
75 }
76
77 public static void mlr()
78 {
79     int c=0;
80     showInputType();
81     showCommand();
82     Scanner sc = new Scanner(System.in);
83     c = sc.nextInt();
84     if (c==1) //keyboard input - mlr
85     {
86
87         Save file = new Save("regresi.txt");
88         String content = "";
89         MLR mlr = new MLR();
90         Matrix mat = mlr.callInputMatrix();
91         System.out.println("Matriks Regresi Linear Berganda: ");
92         content+="Matriks Regresi Linear Berganda: \n";
93         mat.tulisMatrix();
94         content+=mat.tulisMatrixString();content+="\n";
95         mat = mat.ElimMaju(mat);
96         System.out.println("Matriks Epsilon: ");
97         content+="\nMatriks Epsilon: \n";
98         mat.tulisMatrix();
99         System.out.println();
```

The image shows a Windows desktop with three browser windows open in a horizontal row. Each window displays a different line of Java code from a GitHub repository. The top window shows lines 100 to 112, the middle window shows lines 113 to 121, and the bottom window shows lines 113 to 134. The code is part of a Java program for solving matrix equations.

```

100     content+=mat.tulisMatrixString(); content+="\n";
101
102     double[] solusi = {};
103     Gauss gauss = new Gauss();
104     solusi = gauss.GaussSolverFunction(mat);
105
106     if (solusi.length==0)
107     {
108         System.out.println("Tidak memiliki solusi/Solusi Banyak, Y tidak bisa ditaksir");
109         content += "\nTidak memiliki solusi/Solusi Banyak, Y tidak bisa ditaksir";
110         file.write(content);
111     }
112
113     else
114     {
115         int i;
116         double[] x = new double[solusi.length];
117         x[0] = 1;
118         for(i = 1 ; i<x.length ; i++)
119         {
120             System.out.print("Masukkan X"+i+": ");
121             x[i] = sc.nextDouble();
122         }
123         System.out.print("\ny = ");
124         content += "\ny = ";
125         for (i=0 ; i<solusi.length;i++)
126         {
127             if (i==0)
128             {
129                 String strsol = String.format("%.3f", solusi[i]);
130                 System.out.print(strsol + " + ");
131                 content += (strsol + " + ");
132             }
133             else
134             {

```

```

133         else
134     {
135         String strsol = String.format("%.3f", solusi[i]);
136         System.out.print(strsol+" x"+(i));
137         content += strsol+" x"+(i);
138         if (i != solusi.length-1){System.out.print(" + ");
139             content+=" + "; }
140     }
141
142 }
143 System.out.println("\n");
144 content += "\n";
145
146 double y = 0;
147 for (i = 0; i<solusi.length;i++)
148 {
149     if (i==0) {y+=solusi[i];}
150     else{
151         y = y + x[i]*solusi[i];
152     }
153 }
154 for(i = 1 ; i<x.length ; i++)

```



```

154     for(i = 1 ; i<x.length ; i++)
155     {
156         System.out.println("X"+i+" = " + x[i]);
157         content += ("X"+i+" = " + x[i]);
158         content += "\n";
159     }
160     System.out.println("Y = " + y);
161     content += ("Y = " + y);
162     System.out.println();
163     content += "\n";
164     file.write(content);
165 }
166
167 }
168 else if (c==2) //file input - mlr
169 {
170     Save file = new Save("regresi.txt");
171     String content = "";
172     MLR mlr = new MLR();
173     Matrix mat = mlr.callFileMatrix("../test\\mlrpoints.txt");
174     double[] solusi;
175     solusi = mlr.callFile("../test\\mlrpoints.txt");

```

```

168     else if (c==2) //file input - mlr
169     {
170         Save file = new Save("regresi.txt");
171         String content = "";
172         MLR mlr = new MLR();
173         Matrix mat = mlr.callFileMatrix("../test\\mlrpoints.txt");
174         double[] solusi;
175         solusi = mlr.readFile("../test\\mlrpoints.txt");
176         content+="Matriks Regeresi Linear Berganda: \n";
177         content += mat.tulisMatrixString();
178         mat = mat.ElimMaju(mat);
179         content+="\nMatriks Eselon:\n";
180         content+=mat.tulisMatrixString();
181         if (solusi.length==0)
182         {
183             System.out.println("Tidak memiliki solusi/Solusi Banyak, Y tidak bisa ditaksir");
184             content += "\nTidak memiliki solusi/Solusi Banyak, Y tidak bisa ditaksir";
185             file.write(content);
186         }
187         else
188         {
189             int i;
190             else
191             {
192                 int i;
193                 double[] x = new double[solusi.length];
194                 x[0] = 1;
195                 for(i = 1 ; i<x.length ; i++)
196                 {
197                     System.out.print("Masukkan X"+i+": ");
198                     x[i] = sc.nextDouble();
199                 }
200                 System.out.print("\ny = ");
201                 content += "\ny = ";
202                 for (i=0 ; i<solusi.length;i++)
203                 {
204                     if (i==0)
205                     {
206                         String strsol = String.format("%.3f", solusi[i]);
207                         System.out.print(strsol + " + ");
208                         content += (strsol + " + ");
209                     }
210                 }
211             }
212         }

```

The screenshot shows a Windows desktop environment. In the center is a Microsoft Edge browser window displaying a Java source code file from GitHub. The URL in the address bar is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code itself is as follows:

```
207         else
208     {
209         String strsol = String.format("%.3f", solusi[i]);
210         System.out.print(strsol+" x"+(i));
211         content += strsol+" x"+(i);
212         if (i != solusi.length-1){System.out.print(" + ");
213             content+=" + ";}
214     }
215
216     }
217     System.out.println("\n");
218     content += "\n";
219
220     double y = 0;
221     for (i = 0; i<solusi.length;i++)
222     {
223         if (i==0) {y+=solusi[i];}
224         else{
225             y = y + x[i]*solusi[i];
226         }
227     }
228     for(i = 1 ; i<x.length ; i++)
```

Below the browser window, the Windows taskbar is visible, featuring the Start button, a search bar with the placeholder "Type here to search", and various pinned application icons. The system tray shows the date and time as "10/1/2021 3:57 PM".

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>.

The code in the browser windows is as follows:

```
228         for(i = 1 ; i<x.length ; i++)
229     {
230         System.out.println("X"+i+" = "+x[i]);
231         content += ("X"+i+" = "+x[i]);
232         content += "\n";
233     }
234     System.out.println("Y = " + y);
235     content += ("Y = " + y);
236     System.out.println();
237     content += "\n";
238     file.write(content);
239 }
240 }
241 else{System.out.println("Invalid Input");}
242 }
243
244 public static void cramer()
245 {
246     showInputType();
247     showCommand();
248     int c = 0;
249     Scanner sc = new Scanner(System.in);
244     public static void cramer()
245     {
246         showInputType();
247         showCommand();
248         int c = 0;
249         Scanner sc = new Scanner(System.in);
250         c = sc.nextInt();
251         if (c==1)
252         {
253             int row,col;
254             System.out.print("Masukkan jumlah baris: ");
255             row = sc.nextInt();
256             System.out.print("Masukkan jumlah kolom: ");
257             col = sc.nextInt();
258             if (row == col-1)
259             {
260                 System.out.println("Masukkan Matriks: ");
261                 Matrix mat = new Matrix(row, col);
262                 mat.isiMatrix();
263                 Save file = new Save("CramerResult.txt");
264                 String content = "";
265                 Cramer cramer = new Cramer();
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program that handles matrix operations and solves systems of linear equations using Cramer's rule. It includes imports for `java.util.Scanner`, `com.matrixxx.Matrix`, and `com.matrixxx.Cramer`. The program reads input from a file named `matrix.txt`, processes it, and writes the results to a file named `CramerResult.txt`.

```
260     System.out.println("Masukkan Matriks: ");
261     Matrix mat = new Matrix(row, col);
262     mat.isiMatrix();
263     Save file = new Save("CramerResult.txt");
264     String content = "";
265     Cramer cramer = new Cramer();
266     double[] solusi = cramer.call(mat);
267
268     System.out.println("Matriks: ");
269     content+="Matriks: \n";
270
271     mat.tulisMatrix();System.out.println();
272     content+=mat.tulisMatrixString(); content+="\n";
273     content+="\nSolusi:\n";
274     for (int i = 0 ; i<solusi.length ; i++)
275     {
276         System.out.println("X"+(i+1)+" = " +solusi[i]);
277         content+=( "X"+(i+1)+" = " +solusi[i]+"\n");
278     }
279     file.write(content);
280     System.out.println();
281 }
```

```
282     else
283     {
284         System.out.println("Tidak bisa menggunakan Kaidah Cramer");
285     }
286 }
287
288 else if (c==2)
289 {
290     Matrix mat = new Matrix("../test\\matrix.txt");
291     if (mat.getRow()==(mat.getCol()-1))
292     {
293         Save file = new Save("CramerResult.txt");
294         String content = "";
295         Cramer cramer = new Cramer();
296         double[] solusi = cramer.call(mat);
297
298         System.out.println("Matriks: ");
299         content+="Matriks: \n";
300
301         mat.tulisMatrix();System.out.println();
302         content+=mat.tulisMatrixString(); content+="\n";
303         content+="\nSolusi:\n";
```

The image shows a Windows desktop environment with three separate browser windows open, each displaying a different section of a Java program from a GitHub repository. The browser windows are arranged vertically, and the desktop taskbar is visible at the bottom.

Top Window:

```
288     else if (c==2)
289     {
290         Matrix mat = new Matrix("../test/matrix.txt");
291         if (mat.getRow()==(mat.getCol()-1))
292         {
293             Save file = new Save("CramerResult.txt");
294             String content = "";
295             Cramer cramer = new Cramer();
296             double[] solusi = cramer.call(mat);
297
298             System.out.println("Matriks: ");
299             content+="Matriks: \n";
300
301             mat.tulisMatrix();System.out.println();
302             content+=mat.tulisMatrixString(); content+="\n";
303             content+="\nSolusi:\n";
304             for (int i = 0 ; i<solusi.length ; i++)
305             {
306                 System.out.println("X"+(i+1)+" = " +solusi[i]);
307                 content+=( "X"+(i+1)+" = " +solusi[i]+"\n");
308             }
309             file.write(content);
```

Middle Window:

```
308         }
309         file.write(content);
310         System.out.println();
311     }
312     else
313     {
314         System.out.println("Tidak bisa menggunakan Kaidah Cramer");
315     }
316 }
```

Bottom Window:

```
317
318     else{
319         System.out.println("Invalid Input");
320     }
321 }
322 public static void ElimGauss(){
323     int c;
324     showInputType();
325     showCommand();
326     Scanner sc = new Scanner(System.in);
327     c = sc.nextInt();
328     if (c==1){ //keyboard input
329         Save file = new Save("ElimGauss.txt");
```

The image shows a Windows desktop environment with three overlapping browser windows. All three windows have the same URL: <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The browser is Microsoft Edge. The taskbar at the bottom of the screen also displays the same URL and shows various pinned icons and open applications.

```
322     public static void ElimGauss(){
323         int c;
324         showInputType();
325         showCommand();
326         Scanner sc = new Scanner(System.in);
327         c = sc.nextInt();
328         if (c==1){ //keyboard input
329             Save file = new Save("ElimGauss.txt");
330             String content = "";
331
332             System.out.print("Masukkan jumlah baris: ");
333             Scanner scanner = new Scanner(System.in);
334             int baris = scanner.nextInt();
335             System.out.print("Masukkan jumlah kolom: ");
336             int kolom = scanner.nextInt();
337             System.out.println("Masukkan elemen matriks: ");
338
339             Matrix mat = new Matrix(baris, kolom);
340             mat.isiMatrix();
341             content += "Matriks: \n";
342             content += mat.tulisMatrixString(); content+="\n\n";
343             System.out.println("Matriks:");
344             mat.tulisMatrix();
345             System.out.println();
346             content += "Akan dicari penyelesaian SPL (Gauss) untuk matriks berikut ini \n";
347             Gauss.ElimMaju(mat);
348             System.out.println("Matriks Eselon: ");
349             content+="\nMatriks Eselon: \n";
350             mat.tulisMatrix();
351             System.out.println();
352             content+=mat.tulisMatrixString(); content+="\n";
353             Gauss.GaussSolver(mat, "SPL", content, file);
354
355     }
356
357     else if (c==2) //file input gauss
358     {
359         Save file = new Save("ElimGauss.txt");
360         String content = "";
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

Top Browser Window:

```

357     else if (c==2) //file input gauss
358     {
359         Save file = new Save("ElimGauss.txt");
360         String content = "";
361         Matrix mat = new Matrix("../test\\matrix.txt");
362         content += "Matriks: \n";
363         content += mat.tulisMatrixString(); content+="\n\n";
364         System.out.println("Matriks:");
365         mat.tulisMatrix();
366         System.out.println();
367         content += "Akan dicari penyelesaian SPL (Gauss) untuk matriks berikut ini \n";
368         Gauss.ElimMaju(mat);
369         System.out.println("Matriks Eselon: ");
370         content+="\nMatriks Eselon: \n";
371         mat.tulisMatrix();
372         System.out.println();
373         content+=mat.tulisMatrixString(); content+="\n";
374         Gauss.GaussSolver(mat, "SPL", content, file);
375
376     }
377
378     else

```

Bottom Browser Window:

```

378     else
379     {
380         System.out.println("Invalid input");
381     }
382 }
383
384 public static void GaussJordan(){
385     int c;
386     showInputType();
387     showCommand();
388     Scanner sc = new Scanner(System.in);
389     c = sc.nextInt();
390     if (c==1){ //keyboard input
391         Save file = new Save("ElimGaussJordan.txt");
392         String content = "";
393         Gauss gauss = new Gauss();
394
395         System.out.print("Masukkan jumlah baris: ");
396         Scanner scanner = new Scanner(System.in);
397         int baris = scanner.nextInt();
398         System.out.print("Masukkan jumlah kolom: ");
399         int kolom = scanner.nextInt();

```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program for solving systems of linear equations using Gaussian elimination. It includes imports for Scanner, Matrix, Gauss, and Save classes, and handles user input for matrix dimensions and elements. The code is annotated with line numbers (e.g., 384, 402, 410, 420) and includes comments explaining the logic.

```
384     public static void GaussJordan(){
385         int c;
386         showInputType();
387         showCommand();
388         Scanner sc = new Scanner(System.in);
389         c = sc.nextInt();
390         if (c==1){ //keyboard input
391             Save file = new Save("ElimGaussJordan.txt");
392             String content = "";
393             Gauss gauss = new Gauss();
394
395             System.out.print("Masukkan jumlah baris: ");
396             Scanner scanner = new Scanner(System.in);
397             int baris = scanner.nextInt();
398             System.out.print("Masukkan jumlah kolom: ");
399             int kolom = scanner.nextInt();
400             System.out.println("Masukkan elemen matriks:");
401
402             Matrix mat = new Matrix(baris, kolom);
403             mat.isiMatrix();
404             content += "Matriks: \n";
405             content += mat.tulisMatrixString(); content += "\n";
406
407             Gauss.ElimMaju(mat);
408             Gauss.reducedEF(mat);
409             System.out.println("Matriks Eselon Tereduksi: ");
410             mat.tulisMatrix();
411             content += "Matriks Eselon Tereduksi:\n";
412             content += mat.tulisMatrixString(); content += "\n";
413             Gauss.GaussSolver(mat, "SPL", content, file);
414
415         }
416
417         else if (c==2) //file input gaussjordan
418     {
419         Save file = new Save("ElimGaussJordan.txt");
420         String content = "";
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both tabs is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program for solving systems of linear equations using Gaussian elimination. It includes methods for reading matrices from files, performing row operations, and printing results. The code spans from line 420 to line 462.

```
420     else if (c==2) //file input gaussjordan
421     {
422         Save file = new Save("ElimGaussJordan.txt");
423         String content = "";
424         Gauss gauss = new Gauss();
425
426         Matrix mat = new Matrix("../test\\matrix.txt");
427
428         content += "Matriks: \n";
429         content += mat.tulisMatrixString(); content += "\n";
430         System.out.println("Matriks:");
431         mat.tulisMatrix();
432         System.out.println();
433
434         Gauss.ElimMaju(mat);
435         Gauss.reducedEF(mat);
436         System.out.println("Matriks Eselon Tereduksi: ");
437         mat.tulisMatrix();
438         content += "Matriks Eselon Tereduksi:\n";
439         content += mat.tulisMatrixString(); content += "\n";
440         Gauss.GaussSolver(mat, "SPL", content, file);
441     }
442
443     else
444     {
445         System.out.println("Invalid input");
446     }
447 }
448
449 public static void SPL()
450 {
451     //SPL ISI DI SINI
452     int choice = 0;
453     Scanner sc = new Scanner(System.in);
454     showSubSPLMenu();
455     showCommand();
456     choice = sc.nextInt();
457
458     switch (choice)
459     {
460         case 1:
461             ElimGauss();
462             break;
```

The image shows three separate windows on a Windows desktop, each displaying a portion of a Java program. The windows are arranged vertically.

Top Window:

```
449 public static void SPL()
450 {
451     //SPL ISI DI SINI
452     int choice = 0;
453     Scanner sc = new Scanner(System.in);
454     showSubSPLMenu();
455     showCommand();
456     choice = sc.nextInt();
457
458     switch (choice)
459     {
460         case 1:
461             ElimGauss();
462             break;
463
464         case 2:
465             GaussJordan();
466             break;
467
468         case 3:
469             //SPL MATRIKS BALIKAN DI SINI
470             invspl();
```

Middle Window:

```
468         case 3:
469             //SPL MATRIKS BALIKAN DI SINI
470             invspl();
471             break;
472
473         case 4:
474             //CRAMER DI SINI
475             cramer();
476             break;
477
478         default:
479             System.out.println("Invalid Input");
480             break;
481
482     }
483
484 }
```

Bottom Window:

```
486     public static void invspl()
487     {
488         int c = 0;
489         showInputType();
```

```

486     public static void invspl()
487     {
488         int c = 0;
489         showInputType();
490         showCommand();
491         Scanner sc = new Scanner(System.in);
492         c = sc.nextInt();
493         if (c==1)      //keyboard input - inverse spl
494         {
495             System.out.print("Masukkan jumlah baris: ");
496             int row = sc.nextInt();
497             System.out.print("Masukkan jumlah kolom: ");
498             int col = sc.nextInt();
499
500             if (row == (col-1))
501             {
502                 Matrix mat = new Matrix(row,col);
503                 System.out.println("Masukkan elemen matriks: ");
504                 mat.isiMatrix();
505                 int i,j;
506                 Matrix mat2 = new Matrix(mat.getCol()-1,mat.getCol()-1); //Matriks A
507                 for (i=0 ; i<mat2.getRow();i++) //Pengisian Matriks A dari mat
508                     for (j=0 ; j<mat2.getCol();j++)
509                     {
510                         mat2.Mat[i][j] = mat.getElmt(i, j);
511                     }
512             }
513             Determinant determinant = new Determinant();
514             double det = determinant.detCofactor(mat2);
515             if (det!=0)
516             {
517
518                 Save file = new Save("InverseSPL.txt");
519                 String content = "";
520
521                 InverseSPL splinv = new InverseSPL();
522                 double[] solusi = splinv.inverseSPL(mat);
523
524
525                 System.out.println("Matriks: ");
526                 mat.tulisMatrix();
527                 content+="Matriks:\n";
528                 content+=mat.tulisMatrixString();content+="\n\n";

```

The image shows three separate browser windows side-by-side, all displaying the same Java code from a GitHub repository. The URL in the address bar of each window is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program that performs matrix operations. The top window shows lines 525 to 541, the middle window shows lines 545 to 561, and the bottom window shows lines 563 to 566. The code includes printing matrices A and A^-1, calculating matrix B, and printing the solution vector solusi.

```
525     System.out.println("Matriks: ");
526     mat.tulisMatrix();
527     content+="Matriks:\n";
528     content+=mat.tulisMatrixString();content+="\n\n";
529
530     System.out.println("Matriks A: ");
531     mat2.tulisMatrix();
532     content+="Matriks A:\n";
533     content+=mat2.tulisMatrixString();content+="\n\n";
534
535     double[] B = new double[mat.getRow()];
536     for (i=0 ; i<mat.getRow() ; i++) //matriks B
537     {
538         B[i] = mat.getElmt(i, mat.getCol()-1);
539     }
540
541
542     System.out.println("Matriks A^-I: ");
543     mat2 = mat2.inverse();
544     mat2.tulisMatrix();
545     content+="Matriks A^-I:\n";
546     content+=mat2.tulisMatrixString();content+="\n\n";
547
548     System.out.println("Matriks B: ");
549     content+="Matriks B: \n";
550     for (i = 0;i<B.length;i++)
551     {
552         System.out.print(B[i]);
553         content+=B[i];
554         if (i!=B.length-1)
555         {
556             System.out.print(" ");
557             content+=" ";
558         }
559     }
560     System.out.println("\n");
561     content+="\n\n";
562
563     System.out.println("Solusi: ");
564     content += "Solusi: \n";
565     for (i=0 ; i<solusi.length ; i++)
566     {
```

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Algeo01-20070/Main.java at main · Algeo01-20070 · GitHub
- Address Bar:** https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java
- Content Area:** Displays Java code for matrix inversion. The code handles various cases based on matrix properties (size, determinant) and writes solutions or error messages to a file named 'solusi'. It uses System.out.println for output and file.write for writing to a file.
- Taskbar:** Shows the Windows Start button, a search bar with 'Type here to search', and pinned application icons for File Explorer, Edge, File History, OneDrive, Mail, Spotify, Control Panel, File Explorer, Task View, Settings, and File Explorer.
- System Tray:** Shows battery level (100%), signal strength, volume, and a power icon.
- Bottom Right:** Displays the date and time as 10/1/2021 4:02 PM.

```
564     content += "Solusi: \n";
565     for (i=0 ; i<solusi.length ; i++)
566     {
567         content += ("X"+(i+1)+" = " + solusi[i]+"\n");
568         System.out.println("X"+(i+1)+" = " + solusi[i]);
569     }
570     file.write(content);
571 }
572
573 else
574 {
575     System.out.println("Tidak bisa menggunakan metode matriks balikan");
576     System.out.println("Determinan = 0");
577 }
578
579 }
580 else
581 {
582     System.out.println("Tidak bisa menggunakan metode matriks balikan");
583     System.out.println("Matriks haruslah N x M dengan N = M-1");
584 }
585 }
```

```

587     else if (c==2)      //file input - inverse SPL
588     {
589         Matrix mat = new Matrix("../test\\matrix.txt");
590         if (mat.getRow() == mat.getCol()-1)
591         {
592             int i,j;
593             Matrix mat2 = new Matrix(mat.getCol()-1,mat.getCol()-1); //Matriks A
594             for (i=0 ; i<mat2.getRow();i++) //Pengisian Matriks A dari mat
595             {
596                 for (j=0 ; j<mat2.getCol();j++)
597                 {
598                     mat2.Mat[i][j] = mat.getElmt(i, j);
599                 }
600             }
601             Determinant determinant = new Determinant();
602             double det = determinant.detCofactor(mat2);
603             if (det!=0)
604             {
605
606                 Save file = new Save("InverseSPL.txt");
607                 String content = "";
608
609                 InverseSPL splinv = new InverseSPL();
610                 double[] solusi = splinv.inverseSPL(mat);
611
612                 System.out.println("Matriks: ");
613                 mat.tulisMatrix();
614                 content+="Matriks:\n";
615                 content+=mat.tulisMatrixString();content+="\n\n";
616
617                 System.out.println("Matriks A: ");
618                 mat2.tulisMatrix();
619                 content+="Matriks A:\n";
620                 content+=mat2.tulisMatrixString();content+="\n\n";
621
622                 double[] B = new double[mat.getRow()];
623                 for (i=0 ; i<mat.getRow() ; i++) //matriks B
624                 {
625                     B[i] = mat.getElmt(i, mat.getCol()-1);
626                 }
627
628                 System.out.println("Matriks A^-I: ");
629                 mat2 = mat2.inverse();

```

The screenshot shows a Windows desktop environment. In the center is a Microsoft Edge browser window displaying a Java code snippet from a GitHub repository. The code is part of a file named Main.java, specifically lines 629 to 650. The code prints matrices A^-I and B to the console, and then prints the solution matrix. The browser's address bar shows the URL: https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java. Below the browser is the Windows taskbar, which includes the Start button, a search bar, pinned icons for various apps like File Explorer, Edge, and Mail, and system status icons for battery, signal, and volume. The system tray shows the date and time as 10/1/2021 4:02 PM.

```
629     System.out.println("Matriks A^-I: ");
630     mat2 = mat2.inverse();
631     mat2.tulisMatrix();
632     content+="Matriks A-I:\n";
633     content+=mat2.tulisMatrixString();content+="\n\n";
634
635     System.out.println("Matriks B: ");
636     content+="Matriks B: \n";
637     for (i = 0;i<B.length;i++)
638     {
639         System.out.print(B[i]);
640         content+=B[i];
641         if (i!=B.length-1)
642         {
643             System.out.print(" ");
644             content+=" ";
645         }
646     }
647     System.out.println("\n");
648     content+="\n\n";
649
650     System.out.println("Solusi: ");
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program with several sections of logic, primarily using System.out.println() statements to output messages. The code spans lines 649 to 688.

```
650         System.out.println("Solusi: ");
651         content += "Solusi: \n";
652         for (i=0 ; i<solusi.length ; i++)
653         {
654             content += ("X"+(i+1)+" = " + solusi[i]+"\n");
655             System.out.println("X"+(i+1)+" = " + solusi[i]);
656         }
657         file.write(content);
658     }
659
660     else
661     {
662         System.out.println("Tidak bisa menggunakan metode matriks balikan");
663         System.out.println("Determinan = 0");
664     }
665 }
666
667 else
668 {
669     System.out.println("Tidak bisa menggunakan metode matriks balikan");
670     System.out.println("Matriks haruslah N x M dengan N = M-1");
671
672
673
674 else
675 {
676     System.out.println("Invalid input");
677 }
678 }
679
680 public static void Interpolasi(){
681     int c;
682     showInputType();
683     showCommand();
684     Scanner sc = new Scanner(System.in);
685     c = sc.nextInt();
686     if (c==1){ //keyboard input - interpolasi
687         Save file = new Save("interpolasi.txt");
688         String content = "";
```

The image shows a Windows desktop environment with three separate browser windows open, all displaying the same Java code from a GitHub repository. The code is a main method for a Java application named 'Main.java'.

```
680 public static void Interpolasi(){
681     int c;
682     showInputType();
683     showCommand();
684     Scanner sc = new Scanner(System.in);
685     c = sc.nextInt();
686     if (c==1){ //keyboard input - interpolasi
687         Save file = new Save("interpolasi.txt");
688         String content = "";
689
690         System.out.print("Masukkan jumlah titik: ");
691         int n = sc.nextInt();
692
693         Points points = new Points(n);
694         points.readPoints();
695         content = points.writePointstoFile(content);
696         file.write(content);
697
698         Matrix interMatrix;
699         interMatrix = points.toMatrix();
700
701
702         Gauss.ElimMaju(interMatrix);
703         Gauss.GaussSolver(interMatrix, "polinom", content, file);
704
705     }
706
707     else if (c==2) //file input - interpolasi
708     {
709         Points points = new Points("../test\\points.txt");
710         Save file = new Save("interpolasi.txt");
711         String content = "";
712         content = points.writePointstoFile(content);
713         Matrix interMatrix;
714         interMatrix = points.toMatrix();
715
716         Gauss.ElimMaju(interMatrix);
717         Gauss.GaussSolver(interMatrix, "polinom", content, file);
718     }
719
720     else
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program that prompts the user for the number of rows and columns of a matrix, creates a matrix object, and then calls its `isiMatrix()` method. The code spans lines 720 to 758.

```
720     else
721     {
722         System.out.println("Invalid input");
723     }
724 }
725
726 public static void Inverse()
727 {
728     Save file = new Save("inverse.txt");
729     String content = "";
730     int c;
731     showInputType();
732     showCommand();
733     Scanner sc = new Scanner(System.in);
734     c = sc.nextInt();
735     if (c==1) //keyboard input - inverse
736     {
737         int row,col;
738         do{
739             System.out.print("Masukkan Jumlah Baris: ");
740             do
741             {
742                 row = sc.nextInt();
743             }
744             while(row<0);
745
746             System.out.print("Masukkan Jumlah Kolom: ");
747             do
748             {
749                 col = sc.nextInt();
750             }
751             while(col<0);
752
753             if (row != col){System.out.println("Matrix harus N x N !");}
754         }
755         while(row != col);
756         System.out.println("Silakan isi matrix: ");
757         Matrix mat = new Matrix(row,col);
758         mat.isiMatrix();
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program for matrix operations, specifically calculating the inverse of a matrix. The code includes imports for `java.util.Scanner`, `java.util.ArrayList`, `java.util.List`, `java.util.Enumeration`, `java.util.Vector`, `java.util.Enumeration`, `java.util.List`, `java.util.ArrayList`, and `java.util.Scanner`. It defines a class `Main` with a main method that handles user input for matrix size, reads matrix elements from standard input, calculates the determinant and inverse, and writes the results to standard output. The code uses `Matrix` and `Determinant` classes from the `com.matrixxx` package.

```
755     while(row != col);
756     System.out.println("Silakan isi matrix: ");
757     Matrix mat = new Matrix(row,col);
758     mat.isiMatrix();
759     System.out.println("\nIsi Matriks:");
760     content += "Isi Matriks:\n";
761     mat.tulisMatrix();
762     content += mat.tulisMatrixString();
763     content+="\n";
764     Determinant determinant = new Determinant();
765     double det = determinant.detCofactor(mat);
766     if (closeZero(det))
767     {
768         System.out.println("Matrix tidak punya inverse/balikan");
769         content+="Matriks tidak punya inverse/balikan";
770     }
771     else
772     {
773         System.out.println("\nMatrix Inverse/Balikan: ");
774         content+="Matriks Inverse/Balikan:\n";
775         mat = mat.inverse();
776         mat.tulisMatrix();
777         content+=mat.tulisMatrixString();
778         content+="\n";
779     }
780     file.write(content);
781 }
782
783 else if (c==2) //file input - inverse
784 {
785     Matrix mat = new Matrix("../test\\matrix.txt");
786     System.out.println("\nIsi Matriks:");
787     content+="Isi Matriks:\n";
788     mat.tulisMatrix();
789     content+=mat.tulisMatrixString();content+="\n";
790     Determinant determinant = new Determinant();
791     double det = determinant.detCofactor(mat);
792     if (closeZero(det))
```

The screenshot shows a Windows desktop environment. In the center is a Microsoft Edge browser window displaying a Java file from GitHub. The URL in the address bar is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code in the browser is as follows:

```
792         if (closeZero(det))
793     {
794         System.out.println("Determinan = 0\nMatrix tidak punya inverse/balikan");
795         content+="Determinan = 0\nMatrix tidak punya inverse/balikan\n";
796     }
797     else
798     {
799         System.out.println("\nMatrix Inverse/Balikan: ");
800         content += "Matriks Inverse/Balikan:\n";
801         mat = mat.inverse();
802         mat.tulisMatrix();
803         content+=mat.tulisMatrixString();
804         content+='\n';
805     }
806     file.write(content);
807 }
808 else{System.out.println("Invalid input");}
809
810 }
811
812 public static boolean closeZero(double x)
813 {
```

Below the browser window, the Windows taskbar is visible, showing various pinned icons and the system tray. The system tray displays the date and time as 10/1/2021 4:03 PM.

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

The top window shows lines 812 through 333 of the Main.java file. The bottom window shows lines 832 through 853 of the same file. Both windows have the URL <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java> in the address bar. The taskbar at the bottom of the screen shows various pinned icons and the system tray indicating the date and time as 10/1/2021 at 4:03 PM.

```
812     public static boolean closeZero(double x)
813     {
814         return (-0.0000000000001 <= x && x <= 0.0000000000001);
815     }
816
817     public static void determinan()
818     {
819         String content="";
820         Save file = new Save("determinan.txt");
821         int c;
822         detSubMenu();
823         Scanner sc = new Scanner(System.in);
824         c = sc.nextInt();
825         if (c==1) //Metode Reduksi Baris
826         {
827             showInputType();
828             showCommand();
829             c = sc.nextInt();
830             if (c==1) //Keyboard Input - reduction
831             {
832                 int row,col;
833                 do{
834                     System.out.print("Masukkan Jumlah Baris: ");
835                     do
836                     {
837                         row = sc.nextInt();
838                     }
839                     while(row<0);
840
841                     System.out.print("Masukkan Jumlah Kolom: ");
842                     do
843                     {
844                         col = sc.nextInt();
845                     }
846                     while(col<0);
847
848                     if (row != col){System.out.println("Matrix harus N x N !");}
849                 }
850                 while(row != col);
851                 System.out.println("Silakan isi matrix: ");
852                 Matrix mat = new Matrix(row,col);
853                 mat.isiMatrix();
```

```

850         while(row != col);
851             System.out.println("Silakan isi matrix: ");
852             Matrix mat = new Matrix(row,col);
853             mat.isiMatrix();
854             System.out.println("\nIsi Matriks:");
855             content += "Isi Matriks:\n";
856             mat.tulisMatrix();
857             content += mat.tulisMatrixString();
858             Determinant determinant = new Determinant();
859
860             double det = determinant.detReduction(mat);
861             mat = determinant.ElimMajuDet(mat);
862
863             System.out.println("Matriks Segitiga Atas:");
864             content += "Matriks Segitiga Atas:\n";
865             mat.tulisMatrix();
866             content += mat.tulisMatrixString();
867             content += "\n";
868             String strdet = String.format("%.3f",det);
869             System.out.println("Determinan = " + strdet +"\n");
870             content += "Determinan = " + strdet +"\n";
871
872             file.write(content);
873
874     }
875     else if (c==2) //txt file input (matrix.txt) - reduction
876     {
877         Matrix mat = new Matrix("../\\test\\matrix.txt");
878         if (mat.getCol() != mat.getRow())
879         {
880             System.out.println("Matriks harus N x N !");
881         }
882         else{
883             content += "Isi Matriks:\n";
884             System.out.println("\nIsi Matriks:");
885             content += mat.tulisMatrixString();
886             content += "\n";
887             mat.tulisMatrix();
888             Determinant determinant = new Determinant();
889
890             content += "Matriks Segitiga Atas:\n";
891             double det = determinant.detReduction(mat);
892             mat = determinant.ElimMajuDet(mat);
893

```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program for matrix operations, specifically handling triangular matrices and cofactor methods. The code spans lines 890 to 928.

```
content += "Matriks Segitiga Atas:\n";
double det = determinant.detReduction(mat);
mat = determinant.ElimMajuDet(mat);

System.out.println("Matriks Segitiga Atas:");
mat.tulisMatrix();
content+=mat.tulisMatrixString();
content+="\n";
String strdet = String.format("%.3f",det);
System.out.println("Determinan = " + strdet+"\n");
content+="Determinan = " + strdet+"\n";
file.write(content);
}

else{System.out.println("Invalid input");}
}

else if (c==2) //Metode Kofaktor
{
showInputType();
showCommand();
c = sc.nextInt();
}

else if (c==2) //Metode Kofaktor
{
showInputType();
showCommand();
c = sc.nextInt();
if (c==1) //keyboard input - cofactor
{
int row,col;
do{
System.out.print("Masukkan Jumlah Baris: ");
do
{
row = sc.nextInt();
}
while(row<0);

System.out.print("Masukkan Jumlah Kolom: ");
do
{
col = sc.nextInt();
}
while(col<0);
}
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The top window shows lines 921 through 421 of the Main.java file, while the bottom window shows lines 938 through 959. The code is related to matrix operations and determinant calculations.

```
921     while(row<0);
922
923         System.out.print("Masukkan Jumlah Kolom: ");
924         do
925             {
926                 col = sc.nextInt();
927             }
928         while(col<0);
929
930         if (row != col){System.out.println("Matrix harus N x N !");}
931     }
932     while(row != col);
933     while(col<0);
934     System.out.println("Silakan isi matriks: ");
935     Matrix mat = new Matrix(row,col);
936     mat.isiMatrix();
937     content += "Isi Matriks:\n";
938     System.out.println("\nIsi Matriks:");
939     mat.tulisMatrix();
940     content += mat.tulisMatrixString();
941     content += "\n";
942     Determinant determinant = new Determinant();
943
944     System.out.println("\nIsi Matriks:");
945     mat.tulisMatrix();
946     content += mat.tulisMatrixString();
947     content += "\n";
948     Determinant determinant = new Determinant();
949     double det = determinant.detCofactor(mat);
950     String strdet = String.format("%.3f",det);
951     System.out.println("Determinan = " + strdet +"\n");
952     content += "Determinan = " + strdet +"\n";
953     file.write(content);
954
955     else if (c==2) //file input - cofactor
956     {
957         Matrix mat = new Matrix("../\\test\\matrix.txt");
958         if (mat.getRow() != mat.getCol())
959         {
960             System.out.println("Matriks harus N x N !");
961         }
962         else{
963             content += "Isi Matriks:\n";
964             System.out.println("\nIsi Matriks:");
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Main.java>. The code is a Java program with several sections of logic, including matrix operations and determinant calculations. The desktop taskbar at the bottom shows various pinned icons and the system tray.

```
957             else{
958                 content += "Isi Matriks:\n";
959                 System.out.println("\nIsi Matriks:");
960                 mat.tulisMatrix();
961                 content += mat.tulisMatrixString();
962                 Determinant determinant = new Determinant();
963                 double det = determinant.detCofactor(mat);
964                 String strdet = String.format("%.3f",det);
965                 System.out.println("Determinan = " + strdet+"\n");
966                 content += "Determinan = " + strdet+"\n";
967                 file.write(content);
968             }
969         }
970     }
971     else{System.out.println("Invalid Input");}
972 }
973 else{System.out.println("Invalid Input");}
974
975 }
976
977 public static void detSubMenu()
978 {
979     public static void detSubMenu()
980     {
981         System.out.println("1. Metode Reduksi Baris");
982         System.out.println("2. Metode Ekspansi Kofaktor");
983         showCommand();
984     }
985     public static void showInputType()
986     {
987         System.out.println("Input Types:");
988         System.out.println("1. Keyboard Input");
989         System.out.println("2. File Input");
990     }
991 }
```

© 2021 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#)
[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

3.9 Class MLR.java

A screenshot of a Windows desktop environment. At the top, there is a browser window titled "Algeo01-20070/MLR.java at main" with the URL "https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java". The page content shows Java code for a Multiple Linear Regression class. Below the browser is a taskbar with various pinned icons, including Microsoft Edge, File Explorer, Task View, and several other application icons. On the far right of the taskbar, there are system status icons like battery level, signal strength, and volume. The system tray shows the date and time as "10/1/2021 3:52 PM".

```
1 package com.matrixxx;
2
3 import java.io.FileNotFoundException;
4 import java.util.Scanner;
5 import java.io.File;
6
7 //Multiple Linear Regression
8 public class MLR {
9     public double[] callInput()
10    {
11        System.out.print("Masukkan Jumlah Data: ");
12        Scanner sc = new Scanner(System.in);
13        int data = sc.nextInt();
14        System.out.print("Masukkan Jumlah Peubah: ");
15        int ds = sc.nextInt(); //ds = datasets
16        Points[] arrpoints = new Points[ds];
17        double[] y = new double[data];
18
19        for(int i = 0;i<data;i++)
20        {
21            System.out.print("y" + i+1 +": ");
22            y[i] = sc.nextDouble();
23        }
24    }
25}
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

Top Browser Window:

```

21         System.out.print("Y" + i+1 +": ");
22         y[i] = sc.nextDouble();
23     }
24     for (int i = 0 ; i<ds ; i++)
25     {
26         System.out.println("Dataset ke-"+(i+1));
27
28         for (int j = 0 ; j<data ; j++)
29         {
30             System.out.println("X"+(i+1)+(j+1)+": ");
31             double x_in = sc.nextDouble();
32             double y_in = y[j];
33             arrpoints[i].points[j] = new Point(x_in,y_in);
34         }
35     }
36     Matrix mat = new Matrix(ds+1, ds+2);
37     int n = data;
38     int i,j,k;
39     for (i = 0 ; i<mat.getRow();i++)
40     {
41         for (j=0 ; j<mat.getCol();j++)
42         {

```

Bottom Browser Window:

```

36     MATRIX mat = new MATRIX(ds+1, ds+2);
37     int n = data;
38     int i,j,k;
39     for (i = 0 ; i<mat.getRow();i++)
40     {
41         for (j=0 ; j<mat.getCol();j++)
42         {
43             if (i==0 && j==0)
44             {mat.Mat[i][j] = n;}
45
46             else if (i==0 && j!=0 && j != ds+1)
47             {
48                 mat.Mat[i][j] = 0;
49                 // System.out.println(mat.getCol());
50                 for (k=0 ; k<n ; k++)
51                 {
52                     mat.Mat[i][j] += arrpoints[j-1].points[k].x;
53                 }
54             }
55         }
56     else if (i==0 && j == ds+1)

```

```

57         else if (i==0 && j == ds+1)
58     {
59         mat.Mat[i][j] = 0;
60         for (k=0;k<n;k++)
61     {
62         mat.Mat[i][j] += arrpoints[0].points[k].y;
63     }
64 }
65
66 else if (i!=0 && j==0)
67 {
68     mat.Mat[i][j] = 0;
69     for (k=0 ; k<n ; k++)
70     {
71         mat.Mat[i][j] += arrpoints[i-1].points[k].x;
72     }
73 }
74
75 else if (i!=0 && j!=0 && j!=ds+1)
76 {
77     mat.Mat[i][j] = 0;
78     for (k=0 ; k<n ; k++)
79     {
80         mat.Mat[i][j] += arrpoints[i-1].points[k].x * arrpoints[j-1].points[k].x;
81     }
82 }
83
84 else if (i!=0 && j==ds+1)
85 {
86     // System.out.println(i +" "+ j);
87     mat.Mat[i][j] = 0;
88     for (k=0 ; k<n ; k++)
89     {
90         mat.Mat[i][j] += arrpoints[0].points[k].y * arrpoints[i-1].points[k].x;
91     }
92 }
93
94 }
95 }
96 System.out.println("Matriks Regresi Linear Berganda: ");
97 mat.tulisMatrix(); //MATRIX MLR (Multiple Linear Regression)
98 mat = mat.ElimMaju(mat);

```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

The GitHub URL for both windows is: <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java>

The Java code is as follows:

```
96     System.out.println("Matriks Regresi Linear Berganda: ");
97     mat.tulisMatrix(); //MATRIX MLR (Multiple Linear Regression)
98     mat = mat.ElimMaju(mat);
99     System.out.println("Matriks Eselon: ");
100    mat.tulisMatrix(); //Matrix reduksi
101    Gauss gauss = new Gauss();
102    double[] solusi = gauss.GaussSolverFunction(mat);
103    return solusi;
104 }
105
106 public Matrix callInputMatrix()
107 {
108     System.out.print("Masukkan Jumlah Data: ");
109     Scanner sc = new Scanner(System.in);
110     int data = sc.nextInt();
111     System.out.print("Masukkan Jumlah Peubah: ");
112     int ds = sc.nextInt(); //ds = datasets
113     Points[] arrpoints = new Points[ds];
114     double[] y = new double[data];
115
116     for(int i = 0;i<data;i++)
117     {
118         System.out.print("Y" + (i+1) + ": ");
119         y[i] = sc.nextDouble();
120     }
121     for (int i = 0; i<ds ; i++)
122     {
123         arrpoints[i] = new Points(data);
124         System.out.println("Dataset ke-"+(i+1));
125         for (int j = 0 ; j<data ; j++)
126         {
127             System.out.print("X"+(i+1)+(j+1)+": ");
```

```

127         System.out.print("X"+(i+1)+(j+1)+" : ");
128         double x_in = sc.nextDouble();
129         double y_in = y[j];
130         Point a = new Point(x_in,y_in);
131         arrpoints[i].points[j] = a;
132     }
133 }
134 Matrix mat = new Matrix(ds+1, ds+2);
135 int n = data;
136 int i,j,k;
137 for (i = 0 ; i<mat.getRow();i++)
138 {
139     for (j=0 ; j<mat.getCol();j++)
140     {
141         if (i==0 && j==0)
142             {mat.Mat[i][j] = n;}
143
144         else if (i==0 && j!=0 && j != ds+1)
145         {
146             mat.Mat[i][j] = 0;
147             // System.out.println(mat.getCol());
148             for (k=0 ; k<n ; k++)
149             {
150                 mat.Mat[i][j] += arrpoints[i-1].points[k].x;
151             }
152
153         }
154
155         else if (i==0 && j == ds+1)
156         {
157             mat.Mat[i][j] = 0;
158             for (k=0;k<n;k++)
159             {
160                 mat.Mat[i][j] += arrpoints[0].points[k].y;
161             }
162
163         else if (i!=0 && j==0)
164         {
165             mat.Mat[i][j] = 0;
166             for (k=0 ; k<n ; k++)
167             {

```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java>.

The code is a Java class named MLR. It contains several methods and logic for calculating matrix operations. The visible portion of the code includes:

```
167         for (k=0 ; k<n ; k++)
168     {
169         mat.Mat[i][j] += arrpoints[i-1].points[k].x;
170     }
171 }
172
173 else if (i!=0 && j!=0 && j!=ds+1)
174 {
175     mat.Mat[i][j] = 0;
176     for (k=0 ; k<n ; k++)
177     {
178         mat.Mat[i][j] += (arrpoints[i-1].points[k].x * arrpoints[j-1].points[k].x);
179     }
180 }
181
182 else if (i!=0 && j==ds+1)
183 {
184     // System.out.println(i + " " + j);
185     mat.Mat[i][j] = 0;
186     for (k=0 ; k<n ; k++)
187     {
188         mat.Mat[i][j] += (arrpoints[0].points[k].y * arrpoints[i-1].points[k].x);
189     }
190 }
191
192 }
193 }
194 return mat;
195 }
196
197 public double[] callFile(String path)
198 {
199     //dataset counter
200     //counter : dataset counter, n : how many (x,y)
201     int counter = 0; boolean flag = false; int n=0;
202     File file = new File(path);
203     try{
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java>. The desktop taskbar at the bottom shows various pinned icons and the date/time as 3:53 PM 10/1/2021.

```
197     public double[] callFile(String path)
198     {
199         //dataset counter
200         //counter : dataset counter, n : how many (x,y)
201         int counter = 0; boolean flag = false; int n=0;
202         File file = new File(path);
203         try{
204             Scanner lineReader = new Scanner(file);
205             if (lineReader.hasNextLine()){counter=1;}
206             else{counter=0;}
207             while(lineReader.hasNextLine())
208             {
209                 Scanner row = new Scanner(lineReader.nextLine());
210                 if (!row.hasNextDouble()){counter+=1;flag = true;}
211                 else if (!flag){n+=1;}
212             }
213         }
214         lineReader.close();
215     }
216
217     catch (FileNotFoundException e)
218     {
219         System.out.println("File not found.");
220         e.printStackTrace();
221     }
222
223     Points[] arrpoints = new Points[counter];
224
225     try{
226         int j = 0;
227         Scanner lineReader = new Scanner(file);
228         for (int i = 0 ; i<counter ; i++)
229         {
230             arrpoints[i] = new Points(n);
231
232             while(lineReader.hasNextLine())
233             {
234                 Scanner row = new Scanner(lineReader.nextLine());
235
236                 if (row.hasNextDouble())
237                 {
238                     double x = row.nextDouble();
239                     arrpoints[i].addPoint(x);
240                 }
241             }
242         }
243     }
244 }
```

```

236     if(!row.hasNextDouble())
237     {
238         double x = row.nextDouble();
239         double y = row.nextDouble();
240         arrpoints[i].points[j] = new Point(x,y);
241         j++;
242     }
243     else{j=0; break;}
244 }
245
246 }
247 lineReader.close();
248 }
249
250 catch (FileNotFoundException e)
251 {
252     System.out.println("File not found.");
253     e.printStackTrace();
254 }
255 int i; int j; int k;
256 Matrix mat = new Matrix(counter+1, counter+2);
257 for (i = 0 ; i<mat.getRow();i++)

```



```

255     int i; int j; int k;
256     Matrix mat = new Matrix(counter+1, counter+2);
257     for (i = 0 ; i<mat.getRow();i++)
258     {
259         for (j=0 ; j<mat.getCol();j++)
260         {
261             if (i==0 && j==0)
262             {mat.Mat[i][j] = n;}
263
264             else if (i==0 && j!=0 && j != counter+1)
265             {
266                 mat.Mat[i][j] = 0;
267                 // System.out.println(mat.getCol());
268                 for (k=0 ; k<n ; k++)
269                 {
270                     mat.Mat[i][j] += arrpoints[j-1].points[k].x;
271                 }
272             }
273
274             else if (i==0 && j == counter+1)
275             {

```

```

275         else if (i==0 && j == counter+1)
276     {
277         mat.Mat[i][j] = 0;
278         for (k=0;k<n;k++)
279     {
280         mat.Mat[i][j] += arrpoints[0].points[k].y;
281     }
282 }
283
284 else if (i!=0 && j==0)
285 {
286     mat.Mat[i][j] = 0;
287     for (k=0 ; k<n ; k++)
288     {
289         mat.Mat[i][j] += arrpoints[i-1].points[k].x;
290     }
291 }
292
293 else if (i!=0 && j!=0 && j!=counter+1)
294 {
295     mat.Mat[i][j] = 0;
296     for (k=0 ; k<n ; k++)
297     {
298         mat.Mat[i][j] += (arrpoints[i-1].points[k].x * arrpoints[j-1].points[k].x);
299     }
300 }
301
302 else if (i!=0 && j==counter+1)
303 {
304     // System.out.println(i +" "+ j);
305     mat.Mat[i][j] = 0;
306     for (k=0 ; k<n ; k++)
307     {
308         mat.Mat[i][j] += (arrpoints[0].points[k].y * arrpoints[i-1].points[k].x);
309     }
310 }
311
312 }
313 }
314 System.out.println("Matriks Regresi Linear Berganda: ");

```

A screenshot of a Windows desktop environment. In the center is a Microsoft Edge browser window displaying a GitHub page for a Java file named MLR.java. The code in the file is for a Multiple Linear Regression (MLR) algorithm using Gauss elimination. The browser's address bar shows the URL: https://github.com/handono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java. The browser interface includes standard controls like back, forward, and search, along with a tab bar and a status bar at the bottom indicating the date and time (10/1/2021, 3:54 PM). Below the browser is the Windows taskbar, which features the Start button, a search bar, and icons for various running applications.

```
314     System.out.println("Matriks Regresi Linear Berganda: ");
315     mat.tulisMatrix(); //MATRIX MLR (Multiple Linear Regression)
316     mat = mat.ElimMaju(mat);
317     System.out.println("Matriks Eselon: ");
318     mat.tulisMatrix(); //Matrix reduksi
319     Gauss gauss = new Gauss();
320     double[] solusi = gauss.GaussSolverFunction(mat);
321     return solusi;
322 }
323
324 public Matrix callFileMatrix(String path)
325 {
326     //dataset counter
327     //counter : dataset counter, n : how many (x,y)
328     int counter = 0; boolean flag = false; int n=0;
329     File file = new File(path);
330     try{
331         Scanner lineReader = new Scanner(file);
332         if (lineReader.hasNextLine()){counter=1;}
333         else{counter=0;}
334         while(lineReader.hasNextLine())
335     {
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository.

The top window displays lines 324 through 345 of the `MLR.java` file. The code defines a `callFileMatrix` method that reads a dataset from a file. It initializes a dataset counter, iterates through the file lines, and handles exceptions for file not found.

```
324     public Matrix callFileMatrix(String path)
325     {
326         //dataset counter
327         //counter : dataset counter, n : how many (x,y)
328         int counter = 0; boolean flag = false; int n=0;
329         File file = new File(path);
330         try{
331             Scanner lineReader = new Scanner(file);
332             if (lineReader.hasNextLine()){counter=1;}
333             else{counter=0;}
334             while(lineReader.hasNextLine())
335             {
336                 Scanner row = new Scanner(lineReader.nextLine());
337                 if (!row.hasNextDouble()){counter+=1;flag = true;}
338                 else if (!flag){n+=1;}
339             }
340         }
341         lineReader.close();
342     }
343
344     catch (FileNotFoundException e)
345     {
```

The bottom window displays lines 344 through 365 of the same file. This part of the code handles the `FileNotFoundException` by printing an error message and the stack trace to the console.

```
344     catch (FileNotFoundException e)
345     {
346         System.out.println("File not found.");
347         e.printStackTrace();
348     }
349
350     Points[] arrpoints = new Points[counter];
351
352     try{
353         int j = 0;
354         Scanner lineReader = new Scanner(file);
355         for (int i = 0 ; i<counter ; i++)
356         {
357             arrpoints[i] = new Points(n);
358
359             while(lineReader.hasNextLine())
360             {
361                 Scanner row = new Scanner(lineReader.nextLine());
362
363                 if (row.hasNextDouble())
364                 {
365                     double x = row.nextDouble();
```

```

363         if (row.hasNextDouble())
364     {
365         double x = row.nextDouble();
366         double y = row.nextDouble();
367         arrpoints[i].points[j] = new Point(x,y);
368         j++;
369     }
370     else{j=0; break;}
371 }
372 }
373 }
374 lineReader.close();
375 }
376
377 catch (FileNotFoundException e)
378 {
379     System.out.println("File not found.");
380     e.printStackTrace();
381 }
382 int i; int j; int k;
383 Matrix mat = new Matrix(counter+1, counter+2);
384 for (i = 0 ; i<mat.getRow();i++)

```



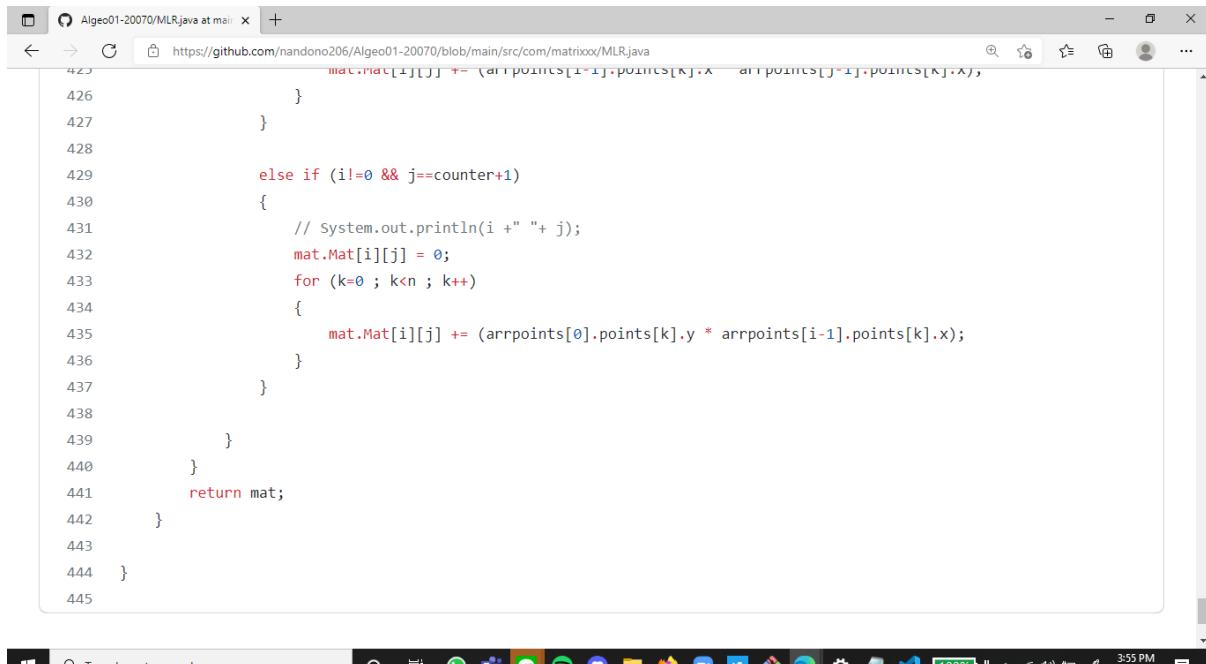
```

377     catch (FileNotFoundException e)
378     {
379         System.out.println("File not found.");
380         e.printStackTrace();
381     }
382     int i; int j; int k;
383     Matrix mat = new Matrix(counter+1, counter+2);
384     for (i = 0 ; i<mat.getRow();i++)
385     {
386         for (j=0 ; j<mat.getCol();j++)
387         {
388             if (i==0 && j==0)
389             {mat.Mat[i][j] = n;}
390
391             else if (i==0 && j!=0 && j != counter+1)
392             {
393                 mat.Mat[i][j] = 0;
394                 // System.out.println(mat.getCol());
395                 for (k=0 ; k<n ; k++)
396                 {
397                     mat.Mat[i][j] += arrpoints[j-1].points[k].x;
398                 }

```

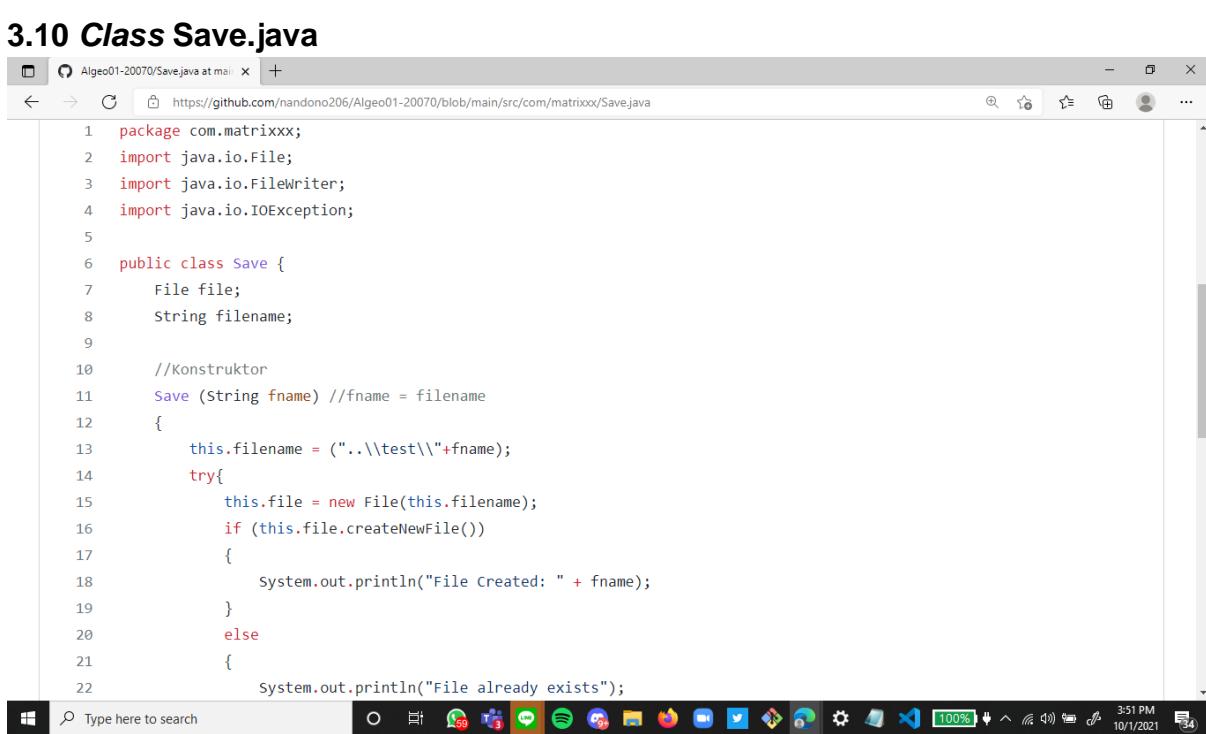
The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java>. The code is a method named `MLR` that takes a `Mat` object and an array of `arrpoints` as parameters. It iterates through the matrix elements, applying different formulas based on their indices (`i` and `j`) relative to the origin (0,0). The code uses nested loops and conditional statements to calculate values for non-zero elements. The Java syntax includes `if`, `else if`, `for`, and `return` statements. The code spans from line 400 to line 441.

```
400 }  
401  
402     else if (i==0 && j == counter+1)  
403     {  
404         mat.Mat[i][j] = 0;  
405         for (k=0;k<n;k++)  
406         {  
407             mat.Mat[i][j] += arrpoints[0].points[k].y;  
408         }  
409     }  
410  
411     else if (i!=0 && j==0)  
412     {  
413         mat.Mat[i][j] = 0;  
414         for (k=0 ; k<n ; k++)  
415         {  
416             mat.Mat[i][j] += arrpoints[i-1].points[k].x;  
417         }  
418     }  
419  
420     else if (i!=0 && j!=0 && j!=counter+1)  
421     {  
422         mat.Mat[i][j] = 0;  
423         for (k=0 ; k<n ; k++)  
424         {  
425             mat.Mat[i][j] += (arrpoints[i-1].points[k].x * arrpoints[j-1].points[k].x);  
426         }  
427     }  
428  
429     else if (i!=0 && j==counter+1)  
430     {  
431         // System.out.println(i +" "+ j);  
432         mat.Mat[i][j] = 0;  
433         for (k=0 ; k<n ; k++)  
434         {  
435             mat.Mat[i][j] += (arrpoints[0].points[k].y * arrpoints[i-1].points[k].x);  
436         }  
437     }  
438  
439 }  
440 }  
441 return mat;
```



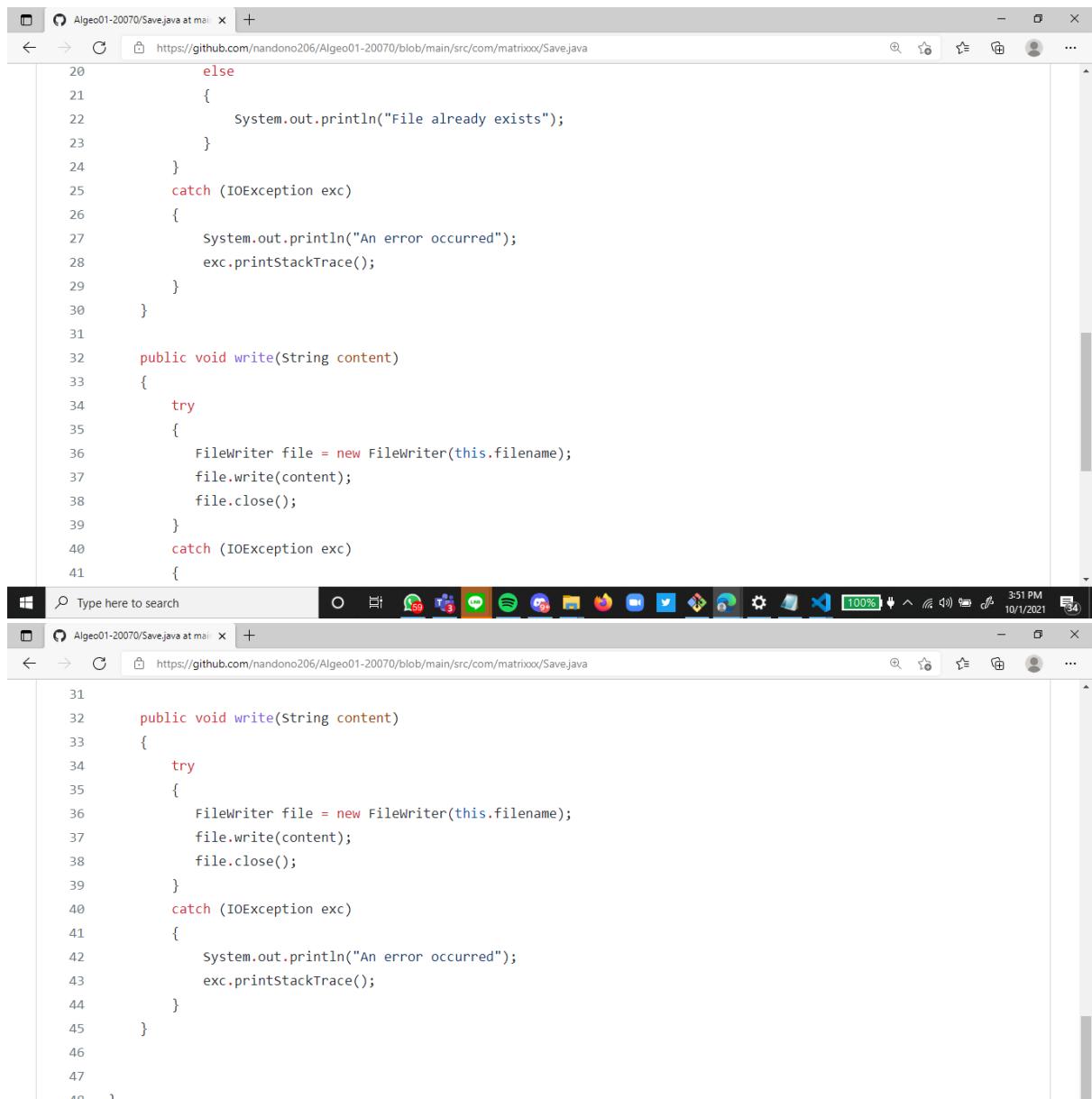
A screenshot of a Windows desktop environment. At the top is a taskbar with various icons. Below it is a browser window titled "Algeo01-20070/MLR.java at main". The URL in the address bar is "https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/MLR.java". The main content of the browser shows lines of Java code, starting from line 425 and ending at line 445. The code appears to be part of a class definition, involving loops and conditionals.

```
425         }
426     }
427 }
428
429 else if (i!=0 && j==counter+1)
430 {
431     // System.out.println(i +" "+ j);
432     mat.Mat[i][j] = 0;
433     for (k=0 ; k<n ; k++)
434     {
435         mat.Mat[i][j] += arrpoints[0].points[k].y * arrpoints[i-1].points[k].x;
436     }
437 }
438
439 }
440 }
441 return mat;
442 }
443
444 }
445
```



A screenshot of a Windows desktop environment. At the top is a taskbar with various icons. Below it is a browser window titled "Algeo01-20070/Save.java at main". The URL in the address bar is "https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Save.java". The main content of the browser shows lines of Java code, starting from line 1 and ending at line 22. The code defines a class named Save, which interacts with the Java file system to create or check for the existence of a file.

```
1 package com.matrixxx;
2 import java.io.File;
3 import java.io.FileWriter;
4 import java.io.IOException;
5
6 public class Save {
7     File file;
8     String filename;
9
10    //Konstruktor
11    Save (String fname) //fname = filename
12    {
13        this.filename = ("..\\\\test\\\\"+fname);
14        try{
15            this.file = new File(this.filename);
16            if (this.file.createNewFile())
17            {
18                System.out.println("File Created: " + fname);
19            }
20            else
21            {
22                System.out.println("File already exists");
```



The image shows a Windows desktop environment with two separate browser windows side-by-side. Both windows are displaying the same Java code from a GitHub repository. The URL in both browser bars is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Save.java>. The desktop taskbar at the bottom shows various pinned icons and the system tray indicating the date and time as 10/1/2021 at 3:51 PM.

```
20         else
21             {
22                 System.out.println("File already exists");
23             }
24         }
25     catch (IOException exc)
26     {
27         System.out.println("An error occurred");
28         exc.printStackTrace();
29     }
30 }
31
32 public void write(String content)
33 {
34     try
35     {
36         FileWriter file = new FileWriter(this.filename);
37         file.write(content);
38         file.close();
39     }
40     catch (IOException exc)
41     {
42
43
44
45
46
47
48 }
```



A screenshot of a Windows desktop showing the title "3.11 Class Test.java" in a large, bold, black font. The desktop taskbar and system tray are visible at the bottom, showing the date and time as 10/1/2021 at 3:51 PM.

A screenshot of a Windows desktop environment. At the top, there is a browser window titled "Algeo01-20070/Test.java at main" showing the URL <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Test.java>. The code in the browser is:

```
1 package com.matrixxx;
2
3 import java.util.Scanner;
4
5 public class Test {
6
7     public static void main(String[] args) {
8
9         //determinant reduction tester
10        // Matrix mat = new Matrix("../test\\matrix.txt");
11        // Determinant det = new Determinant();
12        // det.call(1, mat);
13
14        // MLR mlr = new MLR();
15        // mlr.callFile("../test\\mlrpoints.txt");
16
17        //Cramer Test User Input
18        // Matrix mat = new Matrix(3, 4);
19        // mat.isiMatrix();
20        // Cramer cramer = new Cramer();
21        // double[] solusi = cramer.call(mat);
22        // for (int i = 0 ; i<solusi.length ; i++)
```

The taskbar at the bottom shows several pinned icons and the system tray with the date and time (3:50 PM, 10/1/2021).

The image shows three separate Windows desktop sessions, each with a taskbar at the bottom. Each session has a browser window open to the same GitHub URL: <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Test.java>. The code in the browser is identical across all three sessions.

```
22     // for (int i = 0 ; i<solusi.length ; i++)  
23     // {  
24     //     System.out.println("X"+(i+1)+" = " +solusi[i]);  
25     // }  
26  
27     //Cramer Test File  
28     Matrix mat = new Matrix("../test\\matrix.txt");  
29     mat.tulisMatrix();  
30     Cramer cramer = new Cramer();  
31     double[] solusi = cramer.call(mat);  
32     for (int i = 0 ; i<solusi.length ; i++)  
33     {  
34         System.out.println("X"+(i+1)+" = " +solusi[i]);  
35     }  
36  
37  
38     // System.out.println("Masukkan jumlah baris:");  
39     // Scanner scanner = new Scanner(System.in);  
40     // int baris = scanner.nextInt();  
41     // System.out.println("Masukkan jumlah kolom:");  
42     // int kolom = scanner.nextInt();  
43     // System.out.println("Masukkan elemen matriks:");  
44     /*  
        //file reader testing  
        //Matrix mat = new Matrix(); //tidak butuh argumen untuk konstruktor dengan file  
        mat.readMatrixFile();  
        Matrix mat = new Matrix(); //tidak butuh argumen untuk konstruktor dengan file  
        mat.tulisMatrix();  
        System.out.println(mat.getRow());  
        System.out.println(mat.getCol());  
51  
52        //Determinant tester  
53        Determinant det = new Determinant();  
54        det.call(1, mat);  
55  
56        //adjoint tester  
57        Matrix mat2;  
58        mat2 = mat.adj();  
59        mat2.tulisMatrix();
```

The image shows a Windows desktop environment with two separate browser windows side-by-side, both displaying the same Java code from a GitHub repository. The URL in both tabs is <https://github.com/nandono206/Algeo01-20070/blob/main/src/com/matrixxx/Test.java>. The code consists of several sections of Java code, numbered 56 through 96, which include matrix operations like adjoint, cofactor, and inverse, as well as point input and output. The desktop background is visible, along with the taskbar at the bottom containing various icons and system status indicators.

```
56     //adjoint tester
57     Matrix mat2;
58     mat2 = mat.adj();
59     mat2.tulisMatrix();
60
61     System.out.println();
62
63     //Mat Cofactor tester
64     Matrix mat3;
65     mat3 = mat.matCofactor();
66     mat3.tulisMatrix();
67
68     System.out.println();
69
70     //Mat Inverse tester
71     Matrix mat4;
72     mat4 = mat.inverse();
73     mat4.tulisMatrix();
74
75     //Points tester user input
76     Points P = new Points(3);
77     P.readPoints();
78
79
80     System.out.println();
81
82     //Points tester file input
83     Points P2 = new Points();
84     P2.writePoints();
85
86
87     // System.out.println("Masukkan jumlah baris:");
88     // Scanner scanner = new Scanner(System.in);
89     // int baris = scanner.nextInt();
90     // System.out.println("Masukkan jumlah kolom:");
91     // int kolom = scanner.nextInt();
92     // System.out.println("Masukkan elemen matriks:");
93     System.out.println(mat.getCol());/*
94     // Matrix mat = new Matrix(baris, kolom);
95     // mat.isiOtomatis2();
96     // mat.tulisMatrix();
```

```
95     // mat.isiUTOMATIS2();
96     // mat.tulisMatrix();
97     // System.out.println();
98     // Gauss.ElimMaju(mat);
99
100    // Matrix mat = new Matrix(baris, kolom);
101   // mat.isiMatrix();
102   // mat.tulisMatrix();
103   // System.out.println();
104   // Matrix newMat = new Matrix();
105   // newMat = mat.transpose();
106   // newMat.tulisMatrix();
107   // System.out.println();
108   // mat.bagiKons(2,0.10);
109   // mat.tulisMatrix();
110   // System.out.println();
111   // mat.tukarBaris(1, 2);
112   // mat.tulisMatrix();
113   // scanner.close();
114
115
116
```



```
97     // System.out.println();
98     // Gauss.ElimMaju(mat);
99
100    // Matrix mat = new Matrix(baris, kolom);
101   // mat.isiMatrix();
102   // mat.tulisMatrix();
103   // System.out.println();
104   // Matrix newMat = new Matrix();
105   // newMat = mat.transpose();
106   // newMat.tulisMatrix();
107   // System.out.println();
108   // mat.bagiKons(2,0.10);
109   // mat.tulisMatrix();
110   // System.out.println();
111   // mat.tukarBaris(1, 2);
112   // mat.tulisMatrix();
113   // scanner.close();
114
115
116
117 }
118 }
```

BAB IV

EKSPERIMENT

4.1 SPL

4.1.1 Test Case 1

Input	Output
<p>a.</p> $A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$ <p>Matriks Augmented</p> <pre>1.0 1.0 -1.0 -1.0 1.0 2.0 5.0 -7.0 -5.0 -2.0 2.0 -1.0 1.0 3.0 4.0 5.0 2.0 -4.0 2.0 6.0</pre>	Tidak Memiliki Solusi!
Catatan:	

4.1.2 Test Case 2

Input	Output
<p>b.</p> $A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$ <p>Matriks Augmented</p> <pre>1.0 -1.0 0.0 0.0 1.0 3.0 1.0 1.0 0.0 -3.0 0.0 6.0 2.0 -1.0 0.0 1.0 -1.0 5.0 -1.0 2.0 0.0 -2.0 -1.0 -1.0</pre>	SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi: x1 = 3+s x2 = 2s x3 = r x4 = -1+s x5 = s
Catatan:	

4.1.3 Test Case 3

Input	Output
<p>c.</p> $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$ <p>Matriks Augmented</p> <pre>0.0 1.0 0.0 0.0 1.0 0.0 3.0 0.0 0.0 0.0 1.0 1.0 0.0 -1.0 0.0 1.0 0.0 0.0 0.0 1.0 1.0</pre>	<p>SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi:</p> <p>x1 = r x2 = 1-t x3 = s x4 = -3-t x5 = 2+t x6 = t</p>
Catatan:	

4.1.4 Test Case 4

Input	Output
$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$	<p>SOLUSI:</p> <p>x1 = 0.0 x2 = 2.0 x3 = 1.0 x4 = 1.0</p>
Catatan:	

4.1.5 Test Case 5

Input	Output
<p>a.</p> $8x_1 + x_2 + 3x_3 + 2x_4 = 0$ $2x_1 + 9x_2 - x_3 - 2x_4 = 1$ $x_1 + 3x_2 + 2x_3 - x_4 = 2$ $x_1 + 6x_3 + 4x_4 = 3$ <p>Matriks Augmented:</p> <pre> 8.0 1.0 3.0 2.0 0.0 2.0 9.0 -1.0 -2.0 1.0 1.0 3.0 2.0 -1.0 2.0 1.0 0.0 6.0 4.0 3.0 </pre>	<p>SOLUSI:</p> <p>x1 = -0.2243243243243243 x2 = 0.18243243243243246 x3 = 0.7094594594594594 x4 = -0.25810810810810797</p>
Catatan:	

4.1.6 Test Case 6

Input	Output
<p>a.</p> $\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$ <p>Matriks Augmented:</p> <pre> 1.000 -1.000 2.000 -1.000 -1.000 2.000 1.000 -2.000 -2.000 -2.000 -1.000 2.000 -4.000 1.000 1.000 3.000 0.000 0.000 -3.000 -3.000 </pre>	<p>SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi:</p> <p>x1 = -1+s x2 = 2r x3 = r x4 = s</p>
Catatan:	

4.1.7 Test Case 7

Input	Output
<p>a.</p> $\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$ <p>Matriks Augmented:</p> <pre>1.000 -1.000 2.000 -1.000 -1.000 2.000 1.000 -2.000 -2.000 -2.000 -1.000 2.000 -4.000 1.000 1.000 3.000 0.000 0.000 -3.000 -3.000</pre>	<p>SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi:</p> <p>$x_1 = -1+s$</p> <p>$x_2 = 2r$</p> <p>$x_3 = r$</p> <p>$x_4 = s$</p>

4.1.7 Test Case 8

Input	Output
<p>a.</p> $\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$ <p>Matriks Augmented:</p> <pre>1.000 -1.000 2.000 -1.000 -1.000 2.000 1.000 -2.000 -2.000 -2.000 -1.000 2.000 -4.000 1.000 1.000 3.000 0.000 0.000 -3.000 -3.000</pre>	<p>SPL Memiliki Solusi Banyak (Parametrik) yang memenuhi:</p> <p>$x_1 = -1+s$</p> <p>$x_2 = 2r$</p> <p>$x_3 = r$</p> <p>$x_4 = s$</p>

4.1.8 Test Case 8

Input	Output
<p>Matriks Augmented: Masukkan jumlah baris: 6 Masukkan jumlah kolom: 7 Masukkan Matriks: 1 0.5 0.333 0.25 0.20 0.167 1 0.5 0.333 0.25 0.20 0.167 0.143 0 0.333 0.25 0.20 0.167 0.143 0.125 0 0.25 0.20 0.167 0.143 0.125 0.111 0 0.20 0.167 0.143 0.125 0.111 0.1 0 0.167 0.143 0.125 0.111 0.1 0.0909 0</p>	<p>Matriks: 1.000 0.500 0.333 0.250 0.200 0.167 1.000 0.500 0.333 0.250 0.200 0.167 0.143 0.000 0.333 0.250 0.200 0.167 0.143 0.125 0.000 0.250 0.200 0.167 0.143 0.125 0.111 0.000 0.200 0.167 0.143 0.125 0.111 0.100 0.000 0.167 0.143 0.125 0.111 0.100 0.091 0.000</p> <p>Solusi: X1 = 8.102596578792173 X2 = -24.43241666195989 X3 = -29.244862371310074 X4 = 110.00495737467713 X5 = -53.124304739108354 X6 = -12.121122598883732</p>
<p>Catatan: Menggunakan cramer</p>	

4.2 Determinan

4.2.1 Test Case 1

Input	Output
2.000 3.000 4.000 5.000 4.000 3.000 7.000 0.000 1.000	Determinan = -56.000

4.2.2 Test Case 2

Input	Output
Isi Matriks: 3.000 5.000 -2.000 6.000 1.000 2.000 -1.000 1.000 2.000 4.000 1.000 3.000 3.000 7.000 5.000 3.000	Matriks Segitiga Atas: 3.000 5.000 -2.000 6.000 0.000 0.333 -0.333 -1.000 0.000 0.000 3.000 1.000 0.000 0.000 0.000 0.000 Determinan = 0.000

4.3 Interpolasi Polinom

Input	Output
(8.000 , 2.079) (9.000 , 2.197) (9.500 , 2.251)	<p>Polinom interpolasi yang melewati titik-titik tersebut adalah:</p> $P(x) = 0.6762000000000088 + 0.22659999999998x^1 - 0.00639999999998875x^2$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 0</p> $P(0.0) = 0.6762000000000088$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 9.2</p> $P(9.2) = 2.2192239999999996$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 1</p> $P(1.0) = 0.896400000000007$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 2.2192</p> $P(2.2192) = 1.147551688704005$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 999.999</p> <p>-----KEMBALI KE MENU UTAMA-----</p>

Input	Output
Masukkan titik: 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	<p>0.100 0.003 0.300 0.067 0.500 0.148 0.700 0.248 0.900 0.370 1.100 0.518 1.300 0.697</p> <p>Polinom interpolasi yang melewati titik-titik tersebut adalah:</p> $P(x) = -0.022976562500000138 + 0.24000000000000216x^1 + 0.1973958333332392x^2 + 1.5064338665382593E-14x^3 + 0.0260416666665873x^4$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 0.2</p> $P(0.2) = 0.0329609375000002$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 0.55</p> $P(0.55) = 0.17111865234375$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 0.85</p> $P(0.85) = 0.33723583984374994$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 1.28</p> $P(1.28) = 0.6775418374999975$ <p>Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 999.999</p> <p>-----KEMBALI KE MENU UTAMA-----</p> <p>Output di interpolasi.txt</p>

Input

```
(6.567 , 12624.000)
(7.000 , 21807.000)
(7.258 , 38391.000)
(7.451 , 54517.000)
(7.548 , 51952.000)
(7.839 , 28228.000)
(8.161 , 35764.000)
(8.484 , 20813.000)
(8.709 , 12408.000)
(9.000 , 10534.000)
```

Output

Polinom interpolasi yang melewati titik-titik tersebut adalah:

$$P(x) = 7.189658216471242E12 - 9.350004058993094E12x^1 + 5.335755560850102E12x^2 - 1.7572765632607031E12x^3 + 3.686407609710408E11x^4 \\ - 5.114342929271088E10x^5 + 4.696794214320051E9x^6 - 2.7552878113789976E8x^7 + 9374584.419448344x^8 - 141018.35272207734x^9$$

```
Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 7.51629
P(7.51629) = 53526.27734375
Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 7
P(7.0) = 21807.0205078125
Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 8.32258
P(8.32258) = 36315.69921875
Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 9.167
P(9.167) = -667726.140625
Masukkan nilai X yang ingin ditaksir (masukkan 999.999 jika ingin keluar): 999.999
-----KEMBALI KE MENU UTAMA-----
```

4.4 Regresi Linear Berganda

Input	Output
<pre> 72.4 0.90 41.6 0.91 34.3 0.96 35.1 0.89 10.7 1.00 12.9 1.10 8.3 1.15 20.1 1.03 72.2 0.77 24.0 1.07 23.2 1.07 47.4 0.94 31.5 1.10 10.6 1.10 11.2 1.10 73.3 0.91 75.4 0.87 96.6 0.78 107.4 0.82 54.9 0.95 76.3 0.90 70.3 0.91 77.1 0.96 68.0 0.89 79.0 1.00 67.4 1.10 </pre> <p>input yi x1i, x2i, dan x3i ada di file mlrpoints.txt</p>	<pre> Matriks Regresi Linear Berganda: 20.000 863.100 1530.400 587.840 19.420 863.100 54876.890 67000.090 25283.395 779.477 1530.400 67000.090 117912.320 44976.867 1483.437 587.840 25283.395 44976.867 17278.509 571.122 Matriks Epsilon: 1.000 43.155 76.520 29.392 0.971 0.000 1.000 0.054 -0.005 -0.003 0.000 0.000 1.000 -0.000 0.001 0.000 0.000 0.000 1.000 0.154 y = -3.508 + -0.003 x1 + 0.001 x2 + 0.154 x3 X1 = 50.0 X2 = 76.0 X3 = 29.3 Y = 0.9384342262216672 </pre> <p>Output taksiran Y pada regresi.txt</p> <p>Hasil akan ditampilkan di terminal dan akan otomatis save file di directory /test pada file regresi.txt</p> <p> <i>Humidity (X1) = 50%</i> <i>Temperatur (X2) = 76 F</i> <i>Pressure (X3) = 29.30</i> <i>Nitrous Oxide (Y) = 0.9384</i> </p>

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

5.1. Kesimpulan

Setelah melalui proses penggerjaan dan pengujian, program untuk menghitung sistem persamaan linier, determinan, dan aplikasinya berhasil dibuat menggunakan bahasa pemrograman Java. Program ini dibuat dengan dapat menerima input dari pengguna ataupun melalui file. Seluruh fitur berhasil diimplementasikan pada program yang terdiri atas:

- 1) Memecahkan sistem persamaan linier menggunakan metode eliminasi Gauss, eliminasi Gauss-Jordan, matriks balikan, dan kaidah Cramer.
- 2) Melakukan invers matriks (membuat matriks balikan) dan menghitung determinan menggunakan metode reduksi baris dan ekspansi kofaktor.
- 3) Menghitung hasil fungsi menggunakan interpolasi polinom.
- 4) Melakukan regresi linear berganda dengan metode normal equation dan menghitung persamaan yang didapat menggunakan metode Gauss.

5.2 Saran

Tim penulis memiliki beberapa saran mengenai pengembangan program ini, yaitu

1. Dalam pengembangan program masih bisa dilakukan efisiensi kinerja karena algoritma yang dijelaskan dalam laporan tugas akhir ini masih memiliki banyak kemungkinan untuk dilakukan efisiensi.
2. Program yang telah dibuat ini masih dapat dikembangkan lebih lanjut dengan menambahkan antarmuka pengguna grafis untuk memudahkan pengguna berinteraksi dengan program ini, jadi pengguna tidak perlu mengetik teks perintah yang ingin dijalankan dan lebih user friendly.
3. Program ini sebaiknya di publikasikan ke khalayak ITB agar program ini memiliki nilai kebermanfaatan yang lebih besar.

5.3 Refleksi

Pembelajaran daring saat semester 3 ini membuat kami harus bekerja secara jarak jauh,namun kondisi ini telah berlangsung sejak kami semester 1 di ITB, sehingga kami sudah cukup terbiasa untuk berkoordinasi jarak jauh. Untuk berkoordinasi dan berkolaborasi kami memanfaatkan GitHub. Hanya saja,yang menyulitkan adalah bahasa baru yang harus dipelajari dalam waktu yang relatif singkat. Terdapat beberapa orang yang mungkin saja bisa langsung dengan mudah mengerti,namun ada beberapa orang yang mengalami sedikit kesulitan,apalagi jika ditambah dengan kesibukan diluar akademik yang cukup menyita waktu,karena dimasa ini kaderisasi lanjut di beberapa organisasi(unit dan semacamnya) sedang berjalan, dan mau tidak mau roda organisasi harus tetap berjalan.

Sebagai penutup refleksi, kami tim penyusun berharap kondisi pandemik ini cepat berlalu yang akan mempermudah kami baik itu dalam proses pembelajaran maupun dalam proses koordinasi dalam mengerjakan tugas besar seperti ini atau semacamnya.

DAFTAR PUSTAKA

- Anton, Howard, and Chris Rorres. Elementary linear algebra: applications version. John Wiley & Sons, 2013.
- Munir, Rinaldi. 2020. IF2123 Aljabar Geometri <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/algeo20-21.htm>. Diakses 25 September 2021
- <https://stackoverflow.com/>, diakses 24 September 2021 hingga 30 Oktober 2021
- <https://www.geeksforgeeks.org/>, diakses 24 September 2021 hingga 30 September 2021
- <https://www.w3schools.com/java/>, diakses 20 September 2021 hingga 30 Oktober 2021