Laporan
Tugas Besar 1 IF2123 Aljabar Linier dan Geometri

Sistem Persamaan Linier dan Aplikasinya

oleh

Kinantan Arya Bagaspati (13519044)
Widya Anugrah Putra (13519105)
Kadek Surya Mahardika (13519165)

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2020

BAB I
Deskripsi Masalah

Pada tugas besar pertama mata kuliah Aljabar Linier dan Geometri ini, kami diminta untuk membuat sebuah program menggunakan bahasa Java yang bisa digunakan untuk:

1. Menghitung 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).
2. Menyelesaikan persoalan interpolasi dan regresi linier.
3. Menghitung matriks balikan
4. Menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Lalu spesifikasi programnya sebagai berikut:

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

```
3 4.5 2.8 10 12
-3 7 8.3 11 -4
0.5 -10 -9 12 0
```

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,

```
3 4.5 2.8 10
-3 7 8.3 11

0.5 -10 -9 12
```

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 polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.
8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.
9. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

```
MENU
1. Sistem Persamaaan 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

1. **Metode Eliminasi Gauss**

Metode eliminasi gauss adalah salah satu metode atau algoritma penyelesaian persamaan linier banyak variabel. Cara kerja metode ini adalah mengubah persamaan-persamaan linier menjadi sebuah matriks *augmented* yang merupakan matriks gabungan dari koefisien persamaan linier tersebut dan nilai setiap persamaan. Setelah menjadi sebuah matriks *augmented,* dilakukan operasi baris elementer (OBE) pada matriks tersebut sehingga berubah menjadi matriks eselon. Matriks *eselon* adalah matriks yang pada setiap barisnya, elemen terkiri yang tidak nol dari baris tersebut bernilai satu. Operasi baris elementer (OBE) adalah operasi yang bertujuan untuk membuat elemen-elemen matriks dibawah diagonal utama bernilai nol, terdapat tiga jenis OBE yakni mengalikan sebuah baris dengan sebuah konstanta, menukar dua baris, dan menambah sebuah baris dengan kelipatan dari baris lain sesuai elemen yang bersesuaian. Setelah terbentuk matriks *eselon*, ada tiga interpretasi yang dapat ditarik dari baris paling akhir, jika kolom terakhir dan dua kolom terakhir bernilai nol berarti persamaan linier yang kita definisikan memiliki tak hingga solusi, jika dua kolom terakhir bernilai nol dan kolom terakhir tidak nol, dapat disimpulkan persamaan linier tidak memiliki solusi, dan jika kedua kolom nilainya tidak nol, persamaan linier tersebut memiliki solusi unik. Jika persamaan linier tersebut memiliki solusi, akan dilakukan substitusi atau penyulihan mundur, sehingga didapatkan suatu solusi unik untuk setiap variabel atau solusi tak hingga yang ditulis dalam bentuk parametrik.

2. **Metode Eliminasi Gauss-Jordan**

Metode eliminasi Gauss-Jordan metode eliminasi Gauss dengan beberapa tambahan. Jika dalam metode eliminasi Gauss, kita berhenti melakukan operasi baris elementer (OBE) saat matriks sudah menjadi matriks *eselon*, dalam eliminasi Gauss-Jordan matriks *eselon* tersebut tetap dilakukan OBE sampai terbentuk matriks eselon tereduksi. Matriks *eselon* tereduksi adalah matriks *eselon* dengan satu syarat tambahan yaitu pada setiap barisnya elemen terkiri yang bernilai satu, kolom yang bersesuaian dengan elemen tersebut semua elemennya bernilai nol kecuali pada elemen terkiri yang bernilai satu tersebut. Proses penentuan solusi dalam metode ini sama dengan metode eliminasi gauss, perbedaanya adalah dengan metode ini kita tidak perlu melakukan substitusi/penyulihan mundur ketika persamaan memiliki solusi unik, solusi setiap variabel adalah elemen pada kolom terakhir.

3. **Determinan**

Determinan adalah suatu nilai skalar yang dapat di extract dari suatu matriks persegi, determinan suatu matriks dapat digunakan untuk menentukan balikan/invers dari suatu matriks, yang nantinya nilai tersebut berguna untuk pencarian solusi persamaan linier. Untuk menentukan determinan dari suatu matriks persegi, ada dua metode yang dapat digunakan yakni metode dengan OBE dan metode dengan ekspansi kofaktor. Metode dengan OBE sesuai namanya adalah metode dengan menerapkan operasi baris elementer sehingga terbentuk suatu matriks segitiga atas. Matriks segitiga atas adalah matriks dengan elemen-elemen dibawah diagonal utama bernilai nol. Setelah menjadi matriks segitiga atas, determinan dari matriks tersebut adalah perkalian dari diagonal utama. Untuk metode dengan ekspansi kofaktor, determinan ditentukan dengan penjumlahan dari perkalian elemen-elemen yang bersesuaian pada suatu baris/kolom dengan kofaktor yang bersesuaian dari elemen-elemen tersebut.

4. **Matriks Balikan**

Matriks balikan dari suatu matriks jika dikalikan dengan matriks tersebut akan menghasilkan matriks identitas. Karena karakteristik ini, kita dapat mencari solusi x pada Ax = B dengan mengalikan invers A dengan ruas kiri dan invers A dengan ruas kanan, sehingga didapatkan $Ix = A^{-1}B \Leftrightarrow x = A^{-1}B$. Metode yang digunakan untuk mencari matriks balikan adalah metode OBE, dengan cara meng-augmentasi-kan matriks A dan matriks identitasnya, lalu lakukan OBE sehingga bagian kiri matriks augment tersebut berbentuk matriks identitas. Selanjutnya ada metode kofaktor yang akan dijelaskan di bawah ini.

5. **Matriks Kofaktor**

Matriks kofaktor adalah matriks yang dibentuk oleh elemen-elemen minor dari suatu matriks. Minor elemen pada baris ke-i dan kolom ke-j merupakan hasil penghitungan determinan submatriks dengan mengabaikan baris ke-i dan kolom ke-j pada matriks awal. Minor yang sudah dihitung dapat menentukan kofaktor dari elemennya. Jika elemen berada baris ke-i dan kolom ke-j maka kofaktor dari elemen itu = $(-1)^{(i+j)}$*minor elemen. Dengan kata lain jika baris dan kolom sama-sama genap atau sama-sama ganjil maka kofaktornya sama dengan minornya. Sebaliknya, maka kofaktornya berlawanan tanda dengan minornya.

Matriks kofaktor bisa digunakan untuk mencari determinan juga. Caranya dengan menjumlahkan perkalian elemen matriks dengan elemen kofaktornya yang bersesuaian dalam suatu baris atau kolom tertentu. Oleh karena itu ketika menghitung minor untuk mencari kofaktor dari suatu matriks, maka penghitungan kofaktor ini bersifat rekursif jika menghitung determinan dari submatriks-nya juga menggunakan metode kofaktor. Sifat rekursif ini memiliki basis ketika matriks/submatriks-nya berisi satu elemen dan berukuran 1x1. Determinan dari matriks/submatriks berukuran 1x1 tersebut adalah elemen yang berada di dalamnya.

Setelah bisa menentukan determinan dari suatu matriks menggunakan matriks kofaktornya, matriks kofaktor juga bisa digunakan untuk mencari invers dari matriks. Selanjutnya akan dijelaskan pada bagian matriks adjoin.

6. **Matriks Adjoin**

Matriks adjoin merupakan matriks yang dihasilkan dari matriks kofaktor yang telah ditranspose. Matriks ini dapat digunakan untuk mencari matriks invers. Caranya adalah dengan mengalikan matriks adjoin dengan 1/determinan matriks asli. Determinannya bisa dicari dengan menggunakan metode OBE atau metode matriks kofaktor juga.

7. **Kaidah Cramer**

Kaidah cramer merupakan salah satu yang relatif mudah ketika ingin mencari solusi SPL. Namun, kaidah cramer hanya bisa digunakan jika SPL memiliki n buah persamaan untuk n buah variabel sedemikian sehingga determinan dari matriks tidak sama dengan nol. Misal SPL-nya berbentuk Ax = B, jika sudah bisa dipastikan A memiliki n buah persamaan dengan n buah variabel sedemikian sehingga det(A) tidak sama dengan 0, maka kita bisa mencari x1,x2,..,xn dengan

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

yang dalam hal ini, A_j adalah matriks yang diperoleh dengan mengganti entri pada kolom ke-j dari A dengan entri dari matriks

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

8. Interpolasi Polinom

Interpolasi polinom merupakan salah satu cara untuk memperkirakan suatu nilai. Jika kita mempunyai n+1 titik, yaitu (x_0,y_0), (x_1,y_1), (x_2,y_2), … , (x_n,y_n), kita dapat menentukan polinom $p_n(x)$ yang melewati semua titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk setiap i = 0, 1, 2, … , n. Polinom yang melewati titik-titik tersebut berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Misal jika hanya ada dua titik maka polinomnya adalah $p_1(x) = a_0 + a_1x$ yang berupa persamaan garis lurus. Lalu jika ada tiga titik maka polinomnya adalah $p_2(x) = a_0 + a_1x + a_2x^2$ yang berupa persamaan kuadrat dan kurvanya berbentuk parabola, demikian seterusnya

Dengan menyulihkan/mensubstitusi titik (x_i,y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ untuk i = 0,1,2…,n akan diperoleh n+1 buah SPL dalam a_0, a_1, a_2, …, a_n. Lalu dengan menggunakan metode eliminasi gauss, kita dapat menentukan polinom interpolasinya. Terakhir, dengan menggunakan polinom interpolasi tersebut, kita bisa memperkirakan nilai y di sembarang titik dengan x berada pada rentang $[x_0, x_n]$.

9. Regresi Linier Berganda

Regresi linier berganda merupakan cara lain selain interpolasi polinom yang juga berfungsi memperkirakan suatu nilai. Namun perbedaannya, interpolasi menerima masukan 1 variabel peubah x dengan pasangan nilai fungsinya yakni y dalam tiap sampel datanya. Sementara regresi linear menerima masukan k buah peubah x dengan hasil fungsinya yakni y dalam setiap sampel datanya. Kemudian perbedaan selanjutnya adalah interpolasi polinom menghasilkan fungsi polinom $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$, sedangkan regresi linear menghasilkan fungsi linear dengan k peubah $f(x) = B_0 + B_1x_1 + B_2x_2 + \dots + B_kx_k$.

Proses yang digunakan ialah mencari Sistem Persamaan Linear untuk menemukan rumus regresinya terlebih dahulu, dari n buah sample data yang ada. Sehingga apabila matriks data yang kita punyai ialah

$$\begin{bmatrix} x_{11} & \cdots & x_{k1} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{1n} & \cdots & x_{kn} & y_n \end{bmatrix}$$

Maka Sistem Persamaan Linear yang kita dapat ialah menggunakan rumus regresi:

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

Sebagai contoh apabila kita mempunyai data matrix 3x3:

$$
\begin{bmatrix} x_{11} & x_{21} & y_1 \\ x_{12} & x_{22} & y_2 \\ x_{13} & x_{23} & y_3 \end{bmatrix}
$$

Maka kita akan mempunyai matrix SPL

$$
\begin{bmatrix}
3 & x_{11}+x_{12}+x_{13} & x_{21}+x_{22}+x_{23} & y_1+y_2+y_3 \\
x_{11}+x_{12}+x_{13} & x_{11}^2+x_{12}^2+x_{13}^2 & x_{11}x_{21}+x_{12}x_{22}+x_{13}x_{23} & x_{11}y_1+x_{12}y_2+x_{13}y_3 \\
x_{21}+x_{22}+x_{23} & x_{21}x_{11}+x_{22}x_{12}+x_{23}x_{13} & x_{21}^2+x_{22}^2+x_{23}^2 & x_{21}y_1+x_{22}y_2+x_{23}y_3
\end{bmatrix}
$$

Dengan mencoba kasus kecil, serta mencocokkan dengan rumus yang diberikan, kami mendapatkan bahwa menghitung matriks SPL dari data yang diberikan tidaklah sesusah yang dibayangkan. Terdapat metode mendapatkan matriks SPL dari matriks sampel data dengan 3 operasi. Misalkan matriks data dinamakan A, tambahkan 1 kolom berisi 1 di sebelah kiri A menjadi matrix B

$$
B = \begin{bmatrix} 1 & x_{11} & \cdots & x_{k1} & y_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{1n} & \cdots & x_{kn} & y_n \end{bmatrix}
$$

Kemudian hapus kolom terkanan B dan transpose menjadi matrix C

$$
C = \begin{bmatrix} 1 & \cdots & 1 \\ x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{kn} \end{bmatrix}
$$

Maka matriks SPL dapat diperoleh dari mengalikan matriks B dengan matriks C, menghasilkan matriks (k+1) x (k+2). Kemudian diselesaikan dengan metode invers atau crammer akan membuahkan k+1 bilangan $B_0, ..., B_k$ yang digunakan dalam fungsi regresi.

BAB III
Implementasi Program

1. Struktur Class

Pada proyek ini kami menggunakan dua kelas, dua kelas tersebut kami pisahkan ke dalam dua file berbeda yaitu Main.java dan Matrix.java. Setiap kelas memiliki struktur yang berbeda:

a. Kelas Main

Pada kelas main terdapat 13 public static atribut, 3 diantaranya berupa scanner dengan scanner pertama (sc) digunakan untuk input pilihan menu, scanner kedua (file) digunakan untuk input fileName (nama dari file), dan scanner ketiga (fileInput) digunakan untuk input fileWriter, yaitu input dengan untuk membaca file masukan. Mengapa scannernya dibedakan menjadi 3 dan tidak disatukan saja? Karena jika disatukan maka scanner bisa bermasalah/terjadi konflik. Lalu ada 3 atribut dengan class Matrix, satu atribut (input) berguna untuk menerima semua input matriks yang akan dioperasikan SPL-nya, sedangkan atribut yang kedua (sample) hanya berguna untuk menerima matriks sample ketika menu regresi linier dipilih, dan terakhir atribut interpolasiMatrix yang akan menerima matriks dari menu interpolasi . Selanjutnya ada dua atribut double, yang pertama bernama hasilDeterminan yang akan digunakan untuk menyimpan hasil determinan dari matriks dan yang kedua bernama yTaksiran yang digunakan untuk menyimpan hasil taksiran dari interpolasi. Selanjutnya ada 3 atribut bertipe integer, yaitu metodeDeterminan, metodeSPL, dan metodeInverse. Ketiga-tiganya menerima input berupa pilihan metode yang akan dilakukan. Misal jika metodeDeterminan bernilai 1 maka akan diberikan determinan dari matriks dengan metode reduksi baris, jika bernilai 2 akan diberikan determinan dengan metode kofaktor. Terakhir ada 1 atribut FileWriter dan 1 atribut PrintWriter.

Pada kelas ini, terdapat 7 public static method, 6 diantaranya berupa method menu/submenu, sedangkan method yang berbeda merupakan method power yang mempunyai 2 parameter, misal a yang bertipe double dan b yang bertipe integer, lalu mengembalikan a^b, method ini digunakan untuk melakukan interpolasi . Method yang berupa menu/submenu di antara lain, mainMenu(), splMenu(), determinanMenu(), inverseOfMatrixMenu(), inputMenu(String operasi) dan outputMenu(String operasi). Method mainMenu() digunakanketika pengguna ingin melihat menu-menu utama, lalu method splMenu() digunakan ketika pengguna memilih submenu “Sistem Persamaan Linier”, selanjutnya method determinanMenu() digunakan ketika pengguna memilih submenu “Determinan”, lalu method inverseOfMatrixMenu() digunakan ketika pengguna memilih submenu “Matriks balikan”. Terakhir ada method inputMenu dan outputMenu yang digunakan ketika menerima input dan mengeluarkan output. Input/output diimplementasikan berbeda untuk tiap operasi/menu, oleh sebab itu kedua method ini memerlukan 1 parameter string yang akan menunjukkan operasi/menu apa yang akan diinput/outputkan. Selanjutnya akan dibahas pada bagian Garis Besar Program.

b. Kelas Matrix

Pada kelas matrix terdapat empat atribut yang bersifat private yaitu variabel nRow dan nCol bertipe integer yang berfungsi sebagai penampung banyaknya baris dan kolom dari matriks, variable matriks bertipe array dua dimensi, variabel epsilon bertipe double yang berfungsi sebagai threshold nilai yang mendekati nol, yang berarti ketika suatu nilai berada diantara threshold epsilon, nilai tersebut dianggap bernilai nol. Kemudian, ada satu variabel bersifat public yaitu isFileExist yang berfungsi sebagai penanda apakah file masukan user terdapat di folder test. Terakhir ada variabel input yakni variabel static bertipe scanner yang berfungsi sebagai scanner/alat untuk input beberapa nilai.

Setelah atribut, di dalam kelas matriks ada tiga konstruktor matriks yakni konstruktor yang mengkonstruksi matriks kosong, lalu matriks berukuran nRow, nCol (dari input user) dan matriks identitas berukuran nSize (dari input user). Ada pula selektor untuk atribut-atribut yang bersifat private agar bisa diakses di luar kelas seperti getNRow, getNCol, dan getMatrix. Selanjutnya ada 2 method readMatrixFromFile dan readMatrixFromConsole, berguna untuk membaca input dari user melalui file atau melalui konsol. Ada pula satu method yang mengeluarkan matriks ke dalam file, yaitu outputMatrixFromFile. Lalu, ada method copyMatrix yang menyalin matriks dari matriks lain. Selanjutnya ada method printMatrix yang mengeluarkan matriks di layar.

Kemudian, ada beberapa method yang digunakan untuk transformasi matriks sederhana (seperti hilangkan kolom/baris, tukar kolom/baris,mengganti suatu kolom, menambah baris satu dengan baris yang lain setelah dikali multiplier dan transpose) yaitu cutOneCol, cutOneRow, swapRow, swapCol, replaceOneCol, addRowToRow, dan transpose.

Selanjutnya ada kelompok method yang digunakan untuk mentransformasi matriks secara kompleks, yang pertama ada method toTopTriangular yang mengubah matriks menjadi matriks eselon yang bilangan tak nol terkiri dari tiap barisnya belum dibuat menjadi 1, serta mengembalikan banyaknya pertukaran baris yang terjadi. Method ini dipanggil pada method eliminasiGauss dan fungsi determinantByReduction, karena program dapat menghitung determinantByReduction dari topTriangular dengan mudah serta bentuk eselon juga sangat mudah dicapai dari TopTriangular, sehingga method ini pantas untuk disertakan. Selanjutnya terdapat method eliminasiGauss yang mengubah matriks menjadi matriks eselon dan method eliminasiGaussJordan yang mengubah matriks menjadi matriks eselon tereduksi. Kelompok method ini juga menjadi kumpulan method-method dasar yang digunakan pada method-method lain.

Kemudian ada kelompok method yang digunakan untuk melakukan operasi matriks yang menggunakan lebih dari 2 matriks. Di antaranya ada method augmentRight yang akan mengembalikan matriks augmented hasil augmentasi 2 matriks dan ada pula method dotProduct yang menghasilkan matriks hasil perkalian 2 matriks.

Selanjutnya ada kelompok method yang digunakan untuk menghitung determinan sekaligus invers matriks dengan 2 cara yaitu melakukan OBE dan menggunakan ekspansi kofaktor. Untuk mencari determinan menggunakan OBE, method yang digunakan adalah determinantByReduction, sedangkan untuk mencari determinan menggunakan ekspansi kofaktor, method-method yang dipakai adalah determinantByCofactor yang memanggil cofactorMatrix dengan cofactorMatrix juga memanggil minorMatrix dan minorMatrix yang memanggil determinantByReduction. Hal ini disebabkan apabila kita memanggil determinantByCofactor, maka fungsi akan bersifat rekursif dan kompleksitasnya akan menjadi O(n!). Untuk mencari invers menggunakan OBE, method yang digunakan adalah inverseByAugment, dengan memanfaatkan method augmentRight dan method eliminasiGaussJordan. Terakhir, untuk menentukan invers menggunakan ekspansi kofaktor, method yang digunakan adalah method inverseByCofactor yang memanggil adjoinMatrix dan determinantByCofactor serta conMultiplyMatrix, untuk method adjointMatrix sendiri memanggil cofactorMatrix dan transpose. Kelompok method ini tidak lengkap apabila tidak ada method isSquareMatrix untuk mengecek apakah matriks yang akan dioperasikan berbentuk persegi atau tidak.

Kemudian, ada method interpolasi yang menghitung y-taksiran menggunakan metode interpolasi polinom. Lalu ada method regressionSPL yang menghitung regresi linier dengan melakukan transformasi-transformasi matriks, seperti augmentRight, cutOneCol, transpose dan dotProduct.

Selanjutnya, ada kelompok method solusiCrammer yang menghitung solusi SPL dari suatu persamaan Ax = B dengan menggunakan kaidah crammer dengan melakukan penghitungan determinan matriks A_i (matriks A dengan mengganti kolom ke i dengan matriks B) menggunakan method methodCrammer lalu membaginya dengan determinan matriks A. Sehingga didapati $x_i = \det(A_i) / \det(A)$.

Terakhir ada kelompok method yang mengeluarkan solusi SPL ke layar dan ke file. Kelompok method yang mengeluaran solusi SPL ke layar terdiri dari solutionFromGaussJordan (prekondisi : matriks yang akan dikeluarkan sudah dilakukan method eliminasiGaussJordan), solutionSPLInvers (Misal Ax = B, maka dengan mengalikan kedua ruas dengan A^{-1} didapat $x = A^{-1}B$) yang menggunakan method inverseByAugment pada matriks serta dotProduct antara matriks konstanta dengan matriks yang sudah dipotong kolom terakhirnya (asumsi matriksnya augmented, jadi harus diambil kolom terakhir sebagai konstanta dan matriks yang terpotong merupakan matriks A), lalu ada solutionFromRegression yang memanggil method regressionSPL di dalamnya, untuk solutionFromCrammer tidak direalisasikan di kelas Matrix.java namun pada method outputMenu pada kelas Main.java. Selanjutnya kelompok method yang mengeluarkan solusi SPL ke dalam file terdiri dari solutionFromGaussJordanFIle, solutionSPLInversFile, solutionFromRegressionFIle dan terakhir solutionCrammerFIle yang menggunakan method solusiCrammer.

Terakhir ada method main yang kami gunakan untuk melakukan debugging pada method-method yang ada pada kelas Matrix.java .

2. Garis Besar Program

Secara garis besar, kelas Main adalah kelas utama yang bergantung pada kelas Matrix. Kelas ini kami gunakan sebagai navigasi menu-menu dan input/output pada program utama. Kita menggunakan try-catch statement pada tiap menu/submenu yang ada, untuk menghindari adanya salah input yang diberikan pengguna. Jika terjadi salah input maka program akan meminta input ulang dari pengguna. Salah input bisa berarti input tidak berada pada range pilihan menu, atau tipe inputan tidak sesuai dengan yang diminta program.

Pada menu mainMenu, kita memberikan beberapa pilihan operasi, yaitu:

1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linier Berganda
6. Keluar

Jika pengguna menginput angka 1, maka akan masuk ke submenu splMenu, sedangkan jika menginput angka 2, maka akan masuk ke submenu determinanMenu, lalu jika menginput angka 3, maka akan masuk ke submenu inverseOfMatrixMenu, selanjutnya jika menginput angka 4 atau 5 maka akan masuk berturutan ke submenu inputMenu bagian/berparameter "Interpolasi Polinom" atau "Regresi Linier Berganda". Terakhir, jika inputnya angka 6, maka program akan behenti.

Pada submenu splMenu, kita memberikan beberapa pilihan metode yang akan dipakai, yaitu:

1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
5. Balik ke Main Menu

Jika pengguna menginput angka di rentang 1 -4, maka atribut metodeSPL akan berubah sesuai dengan masukan. Setelah itu program akan masuk ke submenu inputMenu bagian/berparameter "SPL". Sedangkan jika pengguna menginput angka 5, program akan kembali ke mainMenu.

Pada submenu determinanMenu, kita memberikan beberapa pilihan metode yang akan dipakai, yaitu:

1. Determinan Metode Reduksi
2. Determinan Metode Kofaktor
3. Balik ke Main Menu

Jika pengguna menginput angka 1 atau 2, maka atribut metodeDeterminan akan berubah sesuai dengan masukan. Setelah itu program akan masuk ke submenu inputMenu bagian/berparameter "Determinan". Sedangkan jika pengguna menginput angka 3, program akan kembali ke mainMenu.

Pada submenu inverseOfMatrixMenu, kita memberikan beberapa pilihan metode yang akan dipakai, yaitu:

1. Matriks Inverse dengan metode OBE
2. Matriks Inverse dengan metode Kofaktor
3. Balik ke Main Menu

Jika pengguna menginput angka 1 atau 2, maka atribut metodeInverse akan berubah sesuai dengan masukan. Setelah itu program akan masuk ke submenu inputMenu bagian/berparameter "Matriks Balikan". Sedangkan jika penguna menginput angka 3, program akan kembali ke mainMenu.

Pada submenu inputMenu, kita memberikan beberapa pilihan metode input yang akan dipakai, yaitu:

1. Baca dari File
2. Baca dari Konsol
3. Balik ke Main Menu

Jika pengguna menginput angka 1, maka pengguna harus menginput nama file yang akan menjadi masukan. Jika inputMenu berparameterkan "SPL", "Determinan", dan "Matriks Balikan", maka atribut input akan diisi matriks dari file tersebut, sedangkan jika parameternya "Interpolasi Polinom" maka atribut interpolasiMatrix akan diisi matriks dari file tersebut. Khusus parameter “Regresi Linier Berganda”, tak hanya atribut input akan diisi dengan matriks yang berada di file tersebut tetapi juga program akan meminta pengguna memasukan file matriks sample lalu mengisi atribut sample dengan matriks yang berada pada file tersebut. Jika pengguna menginput angka 2, maka pengguna akan menginput matriks-matriks yang dibutuhkan melalui konsol. Perlakuannya sama untuk tiap-tiap parameter ketika inputnya angka 1, tetapi yang menjadi pembeda adalah metode inputnya melalui konsol bukan dari file. Terakhir, jika pengguna menginput angka 3 maka program akan kembali ke mainMenu. Perlu diperhatikan bahwa hanya inputMenu berparameterkan “SPL” saja yang hanya menerima input tetapi tidak memprosesnya, pemrosesan matriks pada kasus ini dilakukan ketika pemanggilan outputMenu terjadi setelah input matriks dilakukan. Pada kasus inputMenu parameter yang lain, pemrosesan matriks dilakukan di bagian inputMenu, sedangkan pemanggilan outputMenu hanya akan menentukan bagaimana hasil matriks yang telah diolah ingin dioutputkan. Pemrosesan matriks tentu saja menggunakan method-method yang telah dibuat pada kelas Matrix,java dan disesuaikan dengan cara pilihan pengguna sebelumnya.

Terakhir, pada submenu outputMenu, kita memberikan pilihan metode output yang akan dipakai, yaitu:

1. Tulis hasil ke dalam file
2. Tulis hasil ke konsol
3. Balik ke menu input sebelumnya
4. Balik ke main menu

Jika pengguna menginput angka 1 maka pengguna harus menuliskan nama file output yang dikehendaki. Setelah itu program akan mengeluarkan output operasi ke dalam file tersebut. Jika pengguna menginput angka 2 maka program akan mengeluarkan output operasi ke dalam konsol/layar. Lalu jika pengguna menginput angka 3 maka program akan kembali ke inputMenu sebelumnya. Terakhir jika pengguna menginput angka 4 maka program akan kembali ke mainMenu. Output yang dihasilkan submenu ini berbeda-beda tergantung pada operasi/parameter apa yang dilakukan dan tidak bergantung pada tempat output itu disimpan. Jika parameter submenu ini adalah “SPL”, maka program akan mengoperasikan matriks input sesuai dengan metodeSPL yang telah ditentukan lalu mengeluarkan hasilnya. Sedangkan untuk kasus parameter yang lain, karena matriks telah dioperasikan pada submenu inputMenu, maka submenu outputMenu hanya akan mengeluarkan hasil operasi pada matriksnya. Keluaran pada tiap operasi disesuaikan dengan spesifikasi dari tugas.

BAB IV
Eksperimen

1. **Kasus 1 (Mencari solusi SPL dalam bentuk Ax = B)**
   a. Masukan:

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$

Keluaran:
(Metode Gauss)

```
C:\WINDOWS\system32\cmd.exe
============================
Masukkan pilihan anda: 2

Menyelesaikan SPL dalam bentuk Ax = B
Keterangan:
A = matriks koefisien
B = matriks nilai setiap variabel x

Masukkan jumlah baris matriks koefisien: 4
Masukkan jumlah kolom matriks koefisien: 4

Masukkan matriks A (ukuran : baris x kolom)
Matrix masukan:
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2

Masukkan matriks B! (ukuran baris x 1)
Matrix masukan:
1 -2 4 6

============================
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
Berikut matrix setelah dilakukan eliminasi Gauss
1.0 1.0 -1.0 -1.0 1.0
0.0 1.0 -1.6666666666666667 -1.0 -1.3333333333333333
0.0 0.0 1.0 -1.0 1.0
0.0 0.0 0.0 0.0 1.0
Sistem ini tidak mempunyai solusi
Tekan enter untuk melanjutkan...
```

Analisis: Setelah dilakukan metode gauss didapatkan bahwa dua elemen pada baris terakhir tidak konsisten, sehingga benar bahwa sistem tersebut tidak mempunyai solusi.

   b. Masukan:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

Keluaran:
(Metode Gauss-Jordan)

```
C:\WINDOWS\system32\cmd.exe
Menyelesaikan SPL dalam bentuk Ax = B
Keterangan:
A = matriks koefisien
B = matriks nilai setiap variabel x

Masukkan jumlah baris matriks koefisien: 4
Masukkan jumlah kolom matriks koefisien: 5

Masukkan matriks A (ukuran : baris x kolom)
Matrix masukan:
1 -1 0 0 1
1 1 0 -3 0
2 -1 0 1 -1
-1 2 0 -2 -1

Masukkan matriks B! (ukuran baris x 1)
Matrix masukan:
3 6 5 -1

============================
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
Berikut matrix setelah dilakukan eliminasi Gauss Jordan
1.0 0.0 0.0 0.0 -1.0 3.0
0.0 1.0 0.0 0.0 -2.0 0.0
0.0 0.0 0.0 1.0 -1.0 -1.0
0.0 0.0 0.0 0.0 0.0 0.0
Sistem ini memiliki banyak solusi. Berikut solusi parametrik:
x1 = 3.0 + (1.0)a2
x2 = 0.0 + (2.0)a2
x3 = a1
x4 = -1.0 + (1.0)a2
x5 = a2
Tekan enter untuk melanjutkan...
```

Analisis: Untuk menyelesaikan SPL tersebut hanya bisa dilakukan dengan metode Gauss/Gauss-Jordan karena dimensi matriksnya yang tidak berbentuk (mx(m+1)), dimana m adalah jumlah baris. Kemudian, terlihat bahwa dua elemen pada baris terakhir bernilai nol yang berarti SPL tersebut tidak memiliki solusi tunggal, di dapatkanlah solusi seperti diatas. Catatan : a1, a2 merupakan variabel bebas yang nilainya adalah semua bilangan real.

c. Masukan:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

Keluaran:

(Metode Gauss)

```
C:\WINDOWS\system32\cmd.exe
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
===========================
Masukkan pilihan anda: 2
Berikut matrix setelah dilakukan eliminasi Gauss
0.0 1.0 0.0 0.0 1.0 0.0 2.0
0.0 0.0 0.0 1.0 1.0 0.0 -1.0
0.0 0.0 0.0 0.0 1.0 -1.0 1.0
Sistem ini memiliki banyak solusi. Berikut solusi parametrik:
x1 = a1
x2 = 1.0 + (-1.0)a3
x3 = a2
x4 = -2.0 + (-1.0)a3
x5 = 1.0 + (1.0)a3
x6 = a3
Tekan enter untuk melanjutkan...
```

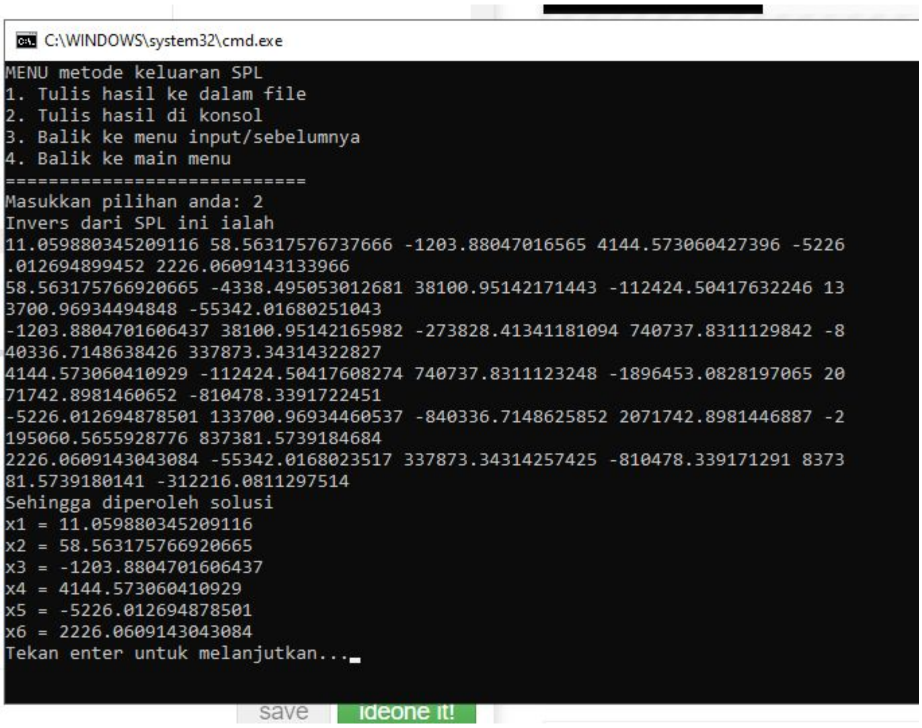
Analisis: Dalam kasus ini, karena ukuran matriks tidak berbentuk m * (m+1), dengan m adalah banyaknya baris, maka kami tidak bisa menggunakan cara determinan maupun cramer, sehingga kami menggunakan metode gauss, sehingga didapatkan solusi seperti diatas. Catatan: a1, a2, dan a3 adalah variabel bebas yang nilainya adalah semua bilangan real.

d. Masukan:

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

H adalah matriks Hilbert. Cobakan untuk n = 6 dan n = 10.

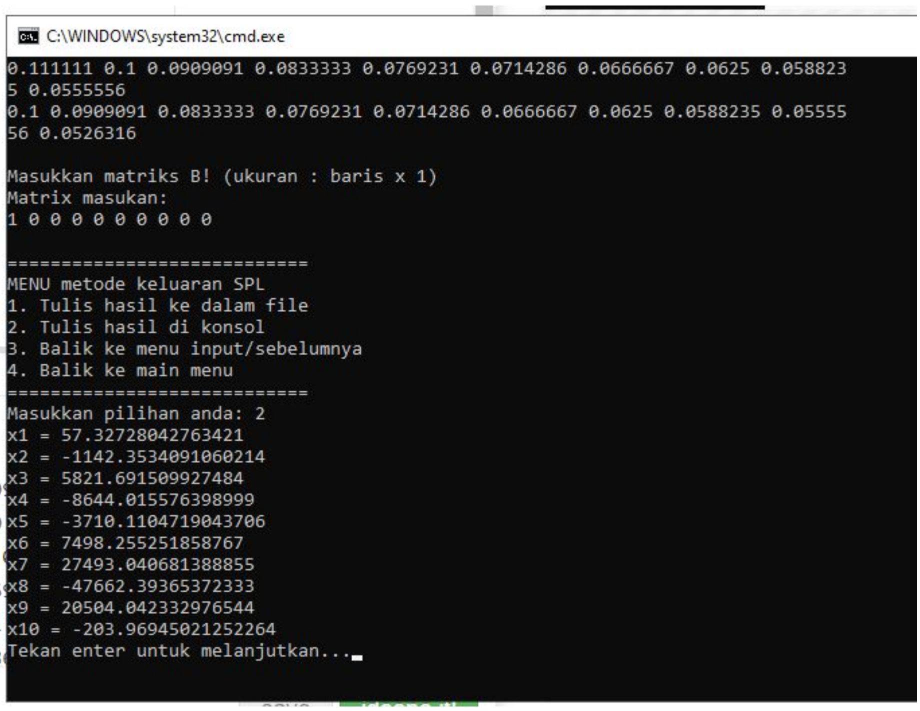
Keluaran (n=6) (Metode matriks balikan):



```
C:\WINDOWS\system32\cmd.exe
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
Invers dari SPL ini ialah
11.059880345209116 58.56317576737666 -1203.88047016565 4144.573060427396 -5226
.012694899452 2226.0609143133966
58.563175766920665 -4338.495053012681 38100.95142171443 -112424.50417632246 13
3700.96934494848 -55342.01680251043
-1203.8804701606437 38100.95142165982 -273828.41341181094 740737.8311129842 -8
40336.7148638426 337873.34314322827
4144.573060410929 -112424.50417608274 740737.8311123248 -1896453.0828197065 20
71742.8981460652 -810478.3391722451
-5226.012694878501 133700.96934460537 -840336.7148625852 2071742.8981446887 -2
195060.5655928776 837381.5739184684
2226.0609143043084 -55342.0168023517 337873.34314257425 -810478.339171291 8373
81.5739180141 -312216.0811297514
Sehingga diperoleh solusi
x1 = 11.059880345209116
x2 = 58.563175766920665
x3 = -1203.8804701606437
x4 = 4144.573060410929
x5 = -5226.012694878501
x6 = 2226.0609143043084
Tekan enter untuk melanjutkan...
```

Analisis: Untuk matriks hilbert karena dimensinya selalu berukuran (baris*(baris+1)) sehingga kita bisa menggunakan semua metode, disini kami menggunakan metode matriks balikan. Terlihat pada gambar diatas, bahwa matriks SPL diawal memiliki balikan, yang berarti determinannya tidak sama dengan nol sehingga SPL tersebut memiliki solusi.

Keluaran (n=10) (Metode Crammer):



```
C:\WINDOWS\system32\cmd.exe
0.111111 0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0.0625 0.058823
5 0.0555556
0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0.0625 0.0588235 0.05555
56 0.0526316

Masukkan matriks B! (ukuran : baris x 1)
Matrix masukan:
1 0 0 0 0 0 0 0 0 0

============================
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
x1 = 57.32728042763421
x2 = -1142.3534091060214
x3 = 5821.691509927484
x4 = -8644.015576398999
x5 = -3710.1104719043706
x6 = 7498.255251858767
x7 = 27493.040681388855
x8 = -47662.39365372333
x9 = 20504.042332976544
x10 = -203.96945021252264
Tekan enter untuk melanjutkan...
```

Analisis: Sama seperti kasus matriks hilbert n = 6, metode cramer dapat dipakai disini karena augmented matriksnya berukuran (baris*(baris+1)). Metode Crammer mirip dengan metode matriks balikan yakni jika solusi SPL ditemukan berarti matriks SPL awal determinannya tidak nol, jika bernilai nol, kita tidak bisa menentukan apakah SPL tersebut memiliki banyak solusi atau tidak memiliki solusi.

2. **Kasus 2 (Mencari solusi SPL dalam bentuk matriks augmented)**

   a. Masukan:

   $$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$$

   Keluaran:
   (Metode Gauss)

```
2aout - Notepad
File  Edit  Format  View  Help
Berikut matrix setelah dilakukan eliminasi Gauss
1.0 -1.0 2.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
Sistem ini memiliki banyak solusi. Berikut solusi parametrik:
x1 = -1.0 + (1.0)a1
x2 = 0.0
x3 = 0.0
x4 = a1
```

   Analisis: Sama seperti studi kasus 1, perbedaannya adalah output sekarang disimpan ke dalam file (.txt).

   b. Masukan:

   $$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$$

   Keluaran:
   (Metode Gauss-Jordan)

2bout - Notepad

File Edit Format View Help

```
Berikut matrix setelah dilakukan eliminasi Gauss Jordan
1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 2.0
0.0 0.0 1.0 0.0 1.0
0.0 0.0 0.0 1.0 1.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
Sistem ini memiliki tepat 1 solusi
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

Analisis: Dari keluaran diatas dapat disimpulkan bahwa dua baris terakhir adalah kelipatan dari baris-baris di atasnya, dan karena semua elemen diagonal ada nilainya, sehingga SPL diatas memiliki satu solusi yang terlihat seperti diatas.

3. **Kasus 3 (Mencari solusi SPL dalam bentuk persamaan-persamaan)**
   a. Masukan:

$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$

   Keluaran: (Metode Gauss)

```
C:\WINDOWS\system32\cmd.exe
Masukkan matriks A (ukuran : baris x kolom)
Matrix masukan:
8 1 3 2
2 9 -1 -2
1 3 2 -1
1 0 6 4

Masukkan matriks B! (ukuran : baris x 1)
Matrix masukan:
0 1 2 3

============================
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
Berikut matrix setelah dilakukan eliminasi Gauss
1.0 0.125 0.375 0.25 0.0
0.0 1.0 -0.2 -0.2857142857142857 0.11428571428571428
0.0 0.0 1.0 -0.19480519480519481 0.7597402597402597
0.0 0.0 0.0 1.0 -0.25810810810810797
Sistem ini memiliki tepat 1 solusi
x1 = -0.22432432432432436
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
Tekan enter untuk melanjutkan...
```

Analisis: Sama seperti studi kasus 1 dan 2, tinggal kita terjemahkan saja SPL diatas secara manual lalu gunakan salah satu dari 4 metode penyelesaian SPL, didapat hasil seperti diatas.

b. Masukan:

$$
\begin{aligned}
x_7 + x_8 + x_9 &= 13.00 \\
x_4 + x_5 + x_6 &= 15.00 \\
x_1 + x_2 + x_3 &= 8.00 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\
0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\
x_3 + x_6 + x_9 &= 18.00 \\
x_2 + x_5 + x_8 &= 12.00 \\
x_1 + x_4 + x_7 &= 6.00 \\
0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\
0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\
0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04
\end{aligned}
$$

Keluaran: (Metode Gauss Jordan)

3bout - Notepad

File Edit Format View Help

```
Berikut matrix setelah dilakukan eliminasi Gauss Jordan
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Sistem ini tidak mempunyai solusi
```

Analisis: Karena persamaan diatas cukup panjang, lebih baik disimpan ke sebuah file yang nantinya akan dipanggil,sehingga didapatkan keluaran seperti diatas, dan karena baris ketiga dari terakhir tidak konsisten, sedangkan 2 baris terakhir adalah kelipatan dari baris-baris di atasnya, karena satu saja tidak konsisten maka SPL tersebut tidak memiliki solusi.

4. **Kasus 4 (Mencari arus pada persamaan-persamaan dalam hukum ohm)**

   Dengan menyimpan sistem persamaan linear dalam file 4in.txt

test > 4in.txt

```
1 1 1 0 0 0 0 0 0 0 0
0 -1 0 1 -1 0 0 0 0 0 0
0 0 -1 0 0 1 0 0 0 0 0
0 0 0 0 1 -1 0 0 0 0 0
0 0 10 0 0 0 1 -1 0 0 0
0 0 0 0 0 10 0 1 -1 0 0
0 0 0 20 0 0 0 0 1 0 0
5 0 0 0 0 0 1 0 0 0 200
0 0 0 0 15 0 0 0 1 -1 0
0 10 0 0 0 0 1 0 0 -1 0
```

Dapat diperoleh solusi dengan metode apapun yang menyelesaikan SPL, misalkan Gauss Jordan:

```
C:\Windows\System32\cmd.exe - run.bat
Masukkan nama File (tanpa .txt di belakang): 4in

============================
MENU metode keluaran SPL
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
Berikut matrix setelah dilakukan eliminasi Gauss Jordan
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 6.7924528301886795
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -5.283018867924531
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -1.509433962264151
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 -6.7924528301886795
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 -1.509433962264151
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 -1.509433962264151
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 166.03773584905662
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 150.9433962264151
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 135.8490566037736
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 113.20754716981132
Sistem ini memiliki tepat 1 solusi
x1 = 6.7924528301886795
x2 = -5.283018867924531
x3 = -1.509433962264151
x4 = -6.7924528301886795
x5 = -1.509433962264151
x6 = -1.509433962264151
x7 = 166.03773584905662
x8 = 150.9433962264151
x9 = 135.8490566037736
x10 = 113.20754716981132
Tekan enter untuk melanjutkan...
```

Dengan x1 hingga x10 berturut-turut menyatakan i_{12}, i_{52}, i_{32}, i_{65}, i_{54}, i_{13}, V_2, V_3, V_4, V_5

5. **Kasus 5 (Mencari polinom interpolasi)**

Nilai taksiran saat x = 0.2

```
5out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: (0.24000000000001212)x + (0.197395833333264)x^2 +
(1.7853625840461486E-13)x^3 + (0.026041666666437653)x^4 + (1.4261776077240953E-13)x^5 + (-
3.4333068795385814E-14)x^6
Nilai taksiran di titik tersebut adalah 0.03296093750000009
```

Nilai taksiran saat x = 0.55

```
5out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: (0.24000000000001212)x + (0.197395833333264)x^2 + (1.7853625840461486E-13)x^3 + (0.026041666666437653)x^4 + (1.4261776077240953E-13)x^5 + (-3.4333068795385814E-14)x^6
Nilai taksiran di titik tersebut adalah 0.17111865234375
```

Nilai taksiran saat x = 0.85

```
5out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: (0.24000000000001212)x + (0.197395833333264)x^2 +
(1.7853625840461486E-13)x^3 + (0.026041666666437653)x^4 + (1.4261776077240953E-
13)x^5 + (-3.4333068795385814E-14)x^6
Nilai taksiran di titik tersebut adalah 0.3372358398437499
```

Nilai taksiran saat x = 1.28

```
5out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: (0.24000000000001212)x + (0.197395833333264)x^2 +
(1.7853625840461486E-13)x^3 + (0.026041666666437653)x^4 + (1.4261776077240953E-
13)x^5 + (-3.4333068795385814E-14)x^6
Nilai taksiran di titik tersebut adalah 0.6775418375
```

Analisis : pencarian taksiran untuk keempat nilai x di atas masih berada dalam rentang 0.1 sampai 1.3, sehingga rentang f(x) yang dihasilkan juga harus dalam rentang 0.003 sampai 0.697. Setelah program dijalankan didapatkan keluaran seperti di atas yang sudah sesuai dengan perkiraan kita.

6. **Kasus 6 (Mencari polinom interpolasi dari data yang diberikan)**

jumlah kasus saat 25/05/2020:

```
6out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: 2.2709632127282667E8 + (-4.158425643268657E8)x +
(3.181500180327195E8)x^2 + (-1.3600326765926722E8)x^3 + (3.61760352897562E7)x^4 + (-
6249553.921160528)x^5 + (704212.2235462547)x^6 + (-50061.95111238677)x^7 + (2041.9195997716974)x^8
+ (-36.47095471862471)x^9
Nilai taksiran di titik tersebut adalah 22804.245000958443
```

jumlah kasus saat 30/08/2020:

```
6out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: 2.2709632127282667E8 + (-4.158425643268657E8)x +
(3.181500180327195E8)x^2 + (-1.3600326765926722E8)x^3 + (3.61760352897562E7)x^4 + (-
6249553.921160528)x^5 + (704212.2235462547)x^6 + (-50061.95111238677)x^7 + (2041.9195997716974)x^8
+ (-36.47095471862471)x^9
Nilai taksiran di titik tersebut adalah 175759.35060882568
```

jumlah kasus saat 15/09/2020:

```
6out - Notepad
File Edit Format View Help
Polinomial yang terbentuk: 2.2709632127282667E8 + (-4.158425643268657E8)x +
(3.181500180327195E8)x^2 + (-1.3600326765926722E8)x^3 + (3.61760352897562E7)x^4 + (-
6249553.921160528)x^5 + (704212.2235462547)x^6 + (-50061.95111238677)x^7 + (2041.9195997716974)x^8
+ (-36.47095471862471)x^9
Nilai taksiran di titik tersebut adalah 68216.42822265625
```

jumlah kasus saat 02/10/2020:

6out - Notepad

```
Polinomial yang terbentuk: 2.2709632127282667E8 + (-4.158425643268657E8)x +
(3.181500180327195E8)x^2 + (-1.3600326765926722E8)x^3 + (3.61760352897562E7)x^4 +
(-6249553.921160528)x^5 + (704212.2235462547)x^6 + (-50061.95111238677)x^7 +
(2041.9195997716974)x^8 + (-36.47095471862471)x^9
Nilai taksiran di titik tersebut adalah -1200266.5605163574
```

Analisis : Saat tanggal yang dicari berada pada rentang data yang kita berikan, nilai taksirannya tentu saja juga mengikuti rentang range data. Namun ketika kita mencoba memprediksi nilai diluar rentang tersebut, kita bisa mendapatkan hasil yang acak, mengikuti polinomial yang terbentuk.

7. **Kasus 7 (Menyederhanakan fungsi dengan polinom interpolasi)**
   Misalkan kita sederhanakan fungsi tersebut dengan polinomial berderajat 3,5, dan 7:
   Untuk mencari tiap nilai f(x) yang akan diinputkan dalam interpolasi dengan masing-masing derajat (3, 5, dan 7), dapat dibuat program terpisah yang mengeluarkan x dan f(x) terurut mulai dari n=3, n=5, dan n=7 seperti berikut:

main.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

double f(double x){
    return (x*x + pow(x, 0.5))/(pow(2.71828, x)+x);
}

int main(){
    ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
    double n=3;
    while(n<8){
        cout << n << endl;
        double h=2/n;
        double cur=0;
        for(int i=0; i<=n; i++){
            cout <<cur << " " << f(cur) << endl;
            cur = cur+h;
        }
        cout << endl;
        n = n+2;
    }
}
```

C:\Users\Kinantan\Desktop\C++\CodeForces\...

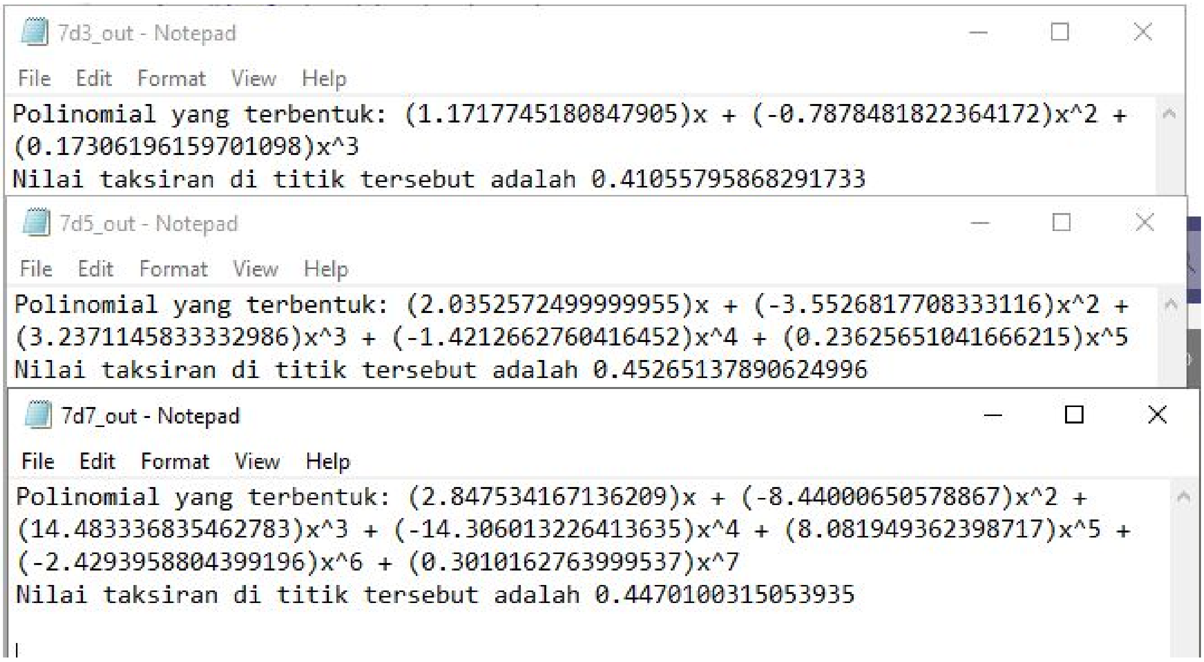
```
3
0 0
0.666667 0.482306
1.33333 0.571968
2 0.576652

5
0 0
0.4 0.418884
0.8 0.507158
1.2 0.560925
1.6 0.583686
2 0.576652

7
0 0
0.285714 0.381184
0.571429 0.462151
0.857143 0.516721
1.14286 0.555131
1.42857 0.577731
1.71429 0.584573
2 0.576652


Process returned 0 (0x0)   execution time : 0.065 s
Press any key to continue.
```

Katakanlah kita ingin mencari taksiran saat x = 0.5, didapatkan keluaran:



7d3_out - Notepad

File Edit Format View Help

```
Polinomial yang terbentuk: (1.1717745180847905)x + (-0.7878481822364172)x^2 +
(0.17306196159701098)x^3
Nilai taksiran di titik tersebut adalah 0.41055795868291733
```

7d5_out - Notepad

File Edit Format View Help

```
Polinomial yang terbentuk: (2.0352572499999955)x + (-3.5526817708333116)x^2 +
(3.2371145833332986)x^3 + (-1.4212662760416452)x^4 + (0.23625651041666215)x^5
Nilai taksiran di titik tersebut adalah 0.45265137890624996
```

7d7_out - Notepad

File Edit Format View Help

```
Polinomial yang terbentuk: (2.847534167136209)x + (-8.44000650578867)x^2 +
(14.483336835462783)x^3 + (-14.306013226413635)x^4 + (8.081949362398717)x^5 +
(-2.4293958804399196)x^6 + (0.3010162763999537)x^7
Nilai taksiran di titik tersebut adalah 0.4470100315053935
```

Analisis : dalam kasus ini rentang x yang mungkin adalah 0 sampai 2 (inklusif), dari masukan file, kita mengetahui bahwa rentang f(x) yang mungkin adalah 0 sampai 0.5, dari keluaran diatas terlihat nilai taksiran di 0.5 masih memenuhi rentang nilai tersebut.

8. Kasus 8 (Mencari regresi linear berganda dari data yang ada)

Berdasarkan data yang ada di studi kasus

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3	Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	1.10	11.2	86.0	29.48
1.10	12.9	67.4	29.39	0.91	73.3	76.3	29.40
1.15	8.3	66.8	29.69	0.87	75.4	77.9	29.28
1.03	20.1	76.9	29.48	0.78	96.6	78.7	29.29
0.77	72.2	77.7	29.09	0.82	107.4	86.8	29.03
1.07	24.0	67.7	29.60	0.95	54.9	70.9	29.37

Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$

$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$

$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$

$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$

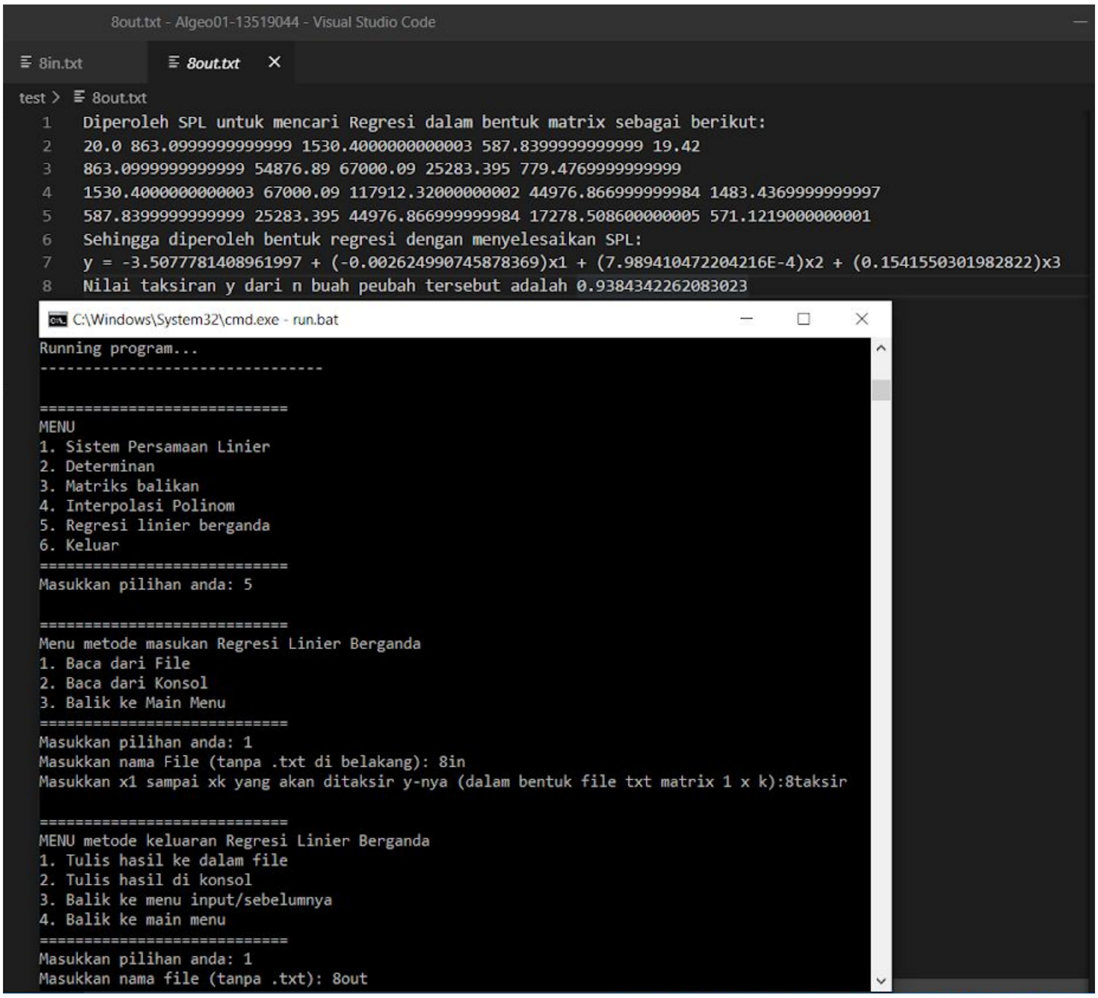
Input dan output dari konsol

C:\Windows\System32\cmd.exe - run.bat

```
Masukkan pilihan anda: 2
Masukkan k (banyak peubah x per sampel): 3
Masukkan n (banyak sampel): 20
Matrix masukan:
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.8 29.03 0.82
54.9 70.9 29.37 0.95
Masukkan x1 sampai xk yang akan ditaksir y-nya (dalam bentuk matrix 1xk):
Matrix masukan:
50 76 29.30

============================
MENU metode keluaran Regresi Linier Berganda
1. Tulis hasil ke dalam file
2. Tulis hasil di konsol
3. Balik ke menu input/sebelumnya
4. Balik ke main menu
============================
Masukkan pilihan anda: 2
Diperoleh SPL untuk mencari Regresi dalam bentuk matrix sebagai berikut:
20.0 863.0999999999999 1530.4000000000003 587.8399999999999 19.42
863.0999999999999 54876.89 67000.09 25283.395 779.4769999999999
1530.4000000000003 67000.09 117912.32000000002 44976.866999999984 1483.4369999999997
587.8399999999999 25283.395 44976.866999999984 17278.508600000005 571.1219000000001
Sehingga diperoleh bentuk regresi dengan menyelesaikan SPL:
y = -3.5077781408961997 + (-0.002624990745878369)x1 + (7.989410472204216E-4)x2 + (0.1541550301982822)x3
Nilai taksiran y dari n buah peubah tersebut adalah 0.9384342262083023
Tekan enter untuk melanjutkan...
```

Input dan output dari file



BAB V
Kesimpulan

1. Kesimpulan

Telah berhasil diimplementasikan sebuah objek berupa Matriks sesuai dengan definisi matriks pada umumnya yang dipelajari dalam mata kuliah Aljabar Linear dan Geometri. Hal mengenai matriks yang berhasil diimplementasi dalam program ini meliputi:

a. Sifat matriks: matriks persegi, matriks identitas, matriks eselon
b. Transformasi elementer: pertukaran kolom, perkalian dengan konstanta, penambahan dan pemotongan baris/kolom
c. Transformasi kompleks: reduksi eselon (gauss dan gauss jordan), inversi, transposisi
d. Properti: baris, kolom, determinan, minor/kofaktor

Semua implementasi ini kemudian berhasil digunakan untuk menyelesaikan metode yang ada dalam spesifikasi, diantaranya penyelesaian Sistem Persamaan Linear dengan metode eliminasi Gauss, Gauss Jordan, matriks inversi, dan cramer. Selanjutnya metode Sistem Persamaan Linear ini digunakan sebagai dasar dalam mencari koefisien polinom dan fungsi linear yang tepat berturut-turut dalam metode Interpolasi dan Regresi Linear Berganda.

2. Saran

Tiap struktur data memiliki keunggulan dan kerugiannya masing - masing, sehingga membuat setiap struktur data unggul dalam situasi yang sesuai dengannya. Matriks memiliki banyak properti, sifat, dan operasi dengan kegunaannya masing - masing yang menjadi keunggulannya dibandingkan struktur data yang lain. Properti, sifat, dan operasi inilah yang membuat matriks paling cocok diimplementasikan dalam bentuk Objek Oriented Programming.

Sangat direkomendasikan mengimplementasi matriks dalam struktur berorientasi objek. Berbagai sifat yang dimilikinya , transformasi elementer, transformasi kompleks, berbagai properti, dan lain sebagainya dapat direalisasikan dengan rapi dalam struktur objek ini. Pemakaian matriks dalam program tentunya dilakukan berulang kali dan tidak sedikit, sehingga bertambah satu alasan untuk menggunakan matriks sebagai objek dalam sebuah program.

3. Refleksi

Matriks merupakan sebuah struktur data yang termasuk salah satu struktur aljabar dan memiliki bermacam-macam kegunaan berkaitan dengan properti, sifat, dan operasi yang dibawa oleh matriks ini. Dalam mengerjakan tugas ini, banyak hal dan sudut pandang yang diperoleh dan dipelajari mengenai berbagai hal tentang matriks. Timbul apresiasi yang besar dalam diri kepada semua orang yang terlibat dalam bidang studi aljabar linear khususnya dalam hal matriks, yang dapat dibilang sebuah

Kegunaan matriks sebagian besar ada pada bidang keilmuan aljabar linear, namun perlu diketahui bahwa bidang keilmuan ini beririsan dengan berbagai bidang keilmuan yang ada. Bidang keilmuan yang tidak terlepas dari matriks misalnya statistika, elektronika, informatika, mesin, kimia, dan lain sebagainya. Ini dikarenakan bidang-bidang ini berurusan dengan teknik pengolahan data, melibatkan kalkulasi bilangan besar, sistem persamaan yang banyak, memperkirakan sebuah nilai atau sebuah fungsi dari informasi data yang ada, dan lainnya. Sifat matriks yang sarat akan kegunaan tentunya membuat salah satu ilmu dari bidang aljabar linear dan geometri ini patut dipelajari dan dibiasakan agar dapat diterapkan di kemudian hari.

DAFTAR REFERENSI

https://www.youtube.com/playlist?list=PLZS-MHyEIRo6V4_vk1s1NcM2HoW5KFG7i, diakses pada tanggal 19 sampai 26 September 2020.
Slide kuliah IF2123 Aljabar Linear dan Geometri tahun ajaran 2020/2021.