LAPORAN TUGAS BESAR
ALJABAR LINIER DAN GEOMETRI

Disusun oleh kelompok **JOIN SYNBIO GAIS!!:**

1. Steve Bezalel Iman Gustaman (13518018)
2. Yonatan Viody (13518120)
3. Muhammad Fauzan Rafi Sidiq Widjonarto (13518147)

Makalah yang dibuat untuk memenuhi tugas mata kuliah Aljabar Linier dan Geometri (IF2123)

TAHUN PELAJARAN 2019/2020

DAFTAR ISI

DAFTAR ISI ..... **ii**

BAB I DESKRIPSI MASALAH **1**

1.1 Masalah pertama 1
1.2 Masalah kedua 1
1.3 Masalah ketiga 2
1.4 Masalah keempat 2

BAB II LANDASAN TEORI **4**

2.1 Metode Eliminasi Gauss 4
2.2 Metode Eliminasi Gauss-Jordan 5
2.3 Determinan Matriks 5
2.4 Matriks Balikan 5
2.5 Matriks Kofaktor 6
2.6 Matriks Adjoin 7
2.7 Kaidah Kramer 7
2.8 Interpolasi Polinom 8

BAB III IMPLEMENTASI PROGRAM **9**

3.1 Deskripsi Singkat Program 9
3.2 Metode Matriks 9
3.3 Metode DriverMatriks 12
3.4 Garis Besar Program 13

BAB IV EKSPERIMEN **15**

4.1 Studi Kasus 1 15
4.2 Studi Kasus 2 19
4.3 Studi Kasus 3 22
4.4 Studi Kasus 4 24
4.5 Studi Kasus 5 26
4.6 Studi Kasus 6 28

BAB V KESIMPULAN **31**

5.1 Kesimpulan 31
5.2 Refleksi 31
5.3 Saran 31

DAFTAR REFERENSI **32**

BAB I
DESKRIPSI MASALAH

1.1 Masalah pertama

Diberikan sistem persamaan linier (SPL) Ax = b dengan n peubah (variabel) dan m persamaan adalah berbentuk:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

yang dalam hal ini x_i adalah peubah, a_{ij} dan b_i adalah koefisien R. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}b$), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

1.2 Masalah kedua

Menentukan determinan dari matriks yang berukuran $n \times n$. Misalkan A adalah matriks tersebut:

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix}$$

maka determinannya adalah:

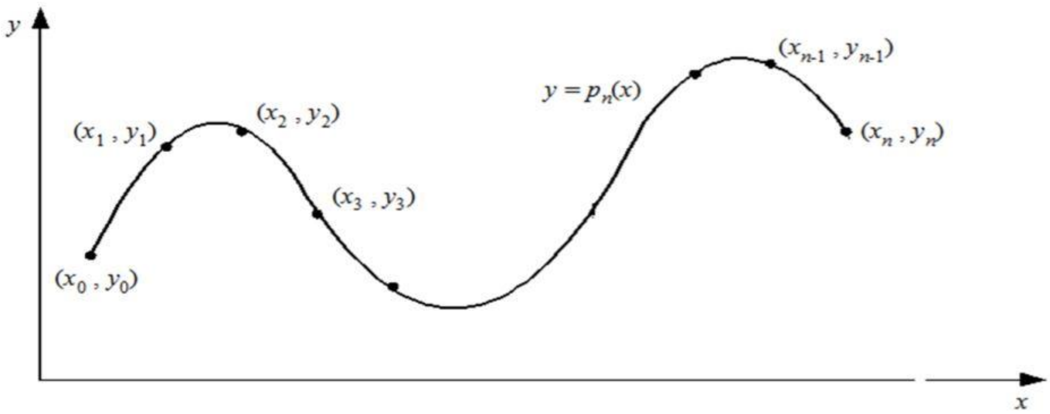
$$\det(M) = \begin{vmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{vmatrix}$$

1.3 Masalah ketiga

Balikan (*inverse*) matriks M berukuran $n \times n$ dapat dihitung dengan banyak cara, salah satunya menggunakan metode eliminasi Gauss-Jordan dan menggunakan matriks adjoin.

1.4 Masalah keempat

Masalah ini memerlukan solusi gabungan dari ketiga masalah sebelumnya, yaitu menginterpolasi sebuah fungsi dari titik-titik yang diberikan. SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, salah satunya adalah mengestimasi nilai fungsi dengan interpolasi polinom. Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n buah titik berbeda, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). Tentukan polinom p_n(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga y_i = $p_n(x_i)$ untuk i = 0, 1, 2, …, n.



Setelah polinom interpolasi p_n(x) ditemukan, p_n(x) dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x_0, x_n]. Polinom interpolasi derajat n yang menginterpolasi titik-titik (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). adalah berbentuk p_n(x) = a_0 + a_1x + a_2x^2 + … + a_nx^n.

Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah p1(x) = a0 + a1x yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0, y_0), (x_1, y_1), dan (x_2, y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah p_2(x) = a_0 + a_1x + a_2x^2 atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0, y_0), (x_1, y_1), (x_2, y_2), dan (x_3, y_3), polinom yang menginterpolasi keempat titik tersebut adalah p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (x_i, y_i) ke dalam persamaan polinomp_n(x) = a_0 + a_1x + a_2x^2 + … + a_nx^n untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a_0, a_1, a_2, …, a_n,

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

Solusi sistem persamaan lanjar ini, yaitu nilai a_0, a_1, …, a_n, diperoleh dengan menggunakan metode eliminasi Gauss yang telah dipelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk p_2(x) = a_0 + a_1x + a_2x^2. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

$$
\begin{aligned}
a_0 + 8.0a_1 + 64.00a_2 &= 2.0794 \\
a_0 + 9.0a_1 + 81.00a_2 &= 2.1972 \\
a_0 + 9.5a_1 + 90.25a_2 &= 2.2513
\end{aligned}
$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan a_0 = 0.6762, a_1 = 0.2266, dan a_2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah p_2(x) = 0.6762 + 0.2266x - 0.0064x^2 . Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064$(9.2)^2$ = 2.2192.

BAB II
LANDASAN TEORI

2.1 Metode Eliminasi Gauss

Metode Eliminasi Gauss, juga dikenal sebagai reduksi baris, adalah algoritma dalam aljabar linier untuk memecahkan sistem persamaan linear. Biasanya dipahami sebagai urutan operasi yang dilakukan pada matriks koefisien yang sesuai. Metode ini juga dapat digunakan untuk menemukan peringkat matriks, menghitung determinan matriks, dan menghitung kebalikan (*inverse*) dari matriks kuadrat terbalik. Metode ini dinamai Carl Friedrich Gauss (1777-1855), meskipun diketahui oleh ahli matematika Cina sedini 179 AD.

Untuk melakukan pengurangan baris pada suatu matriks, seseorang menggunakan urutan operasi baris elementer untuk memodifikasi matriks sampai sudut kiri bawah dari matriks diisi dengan nol, sebanyak mungkin. Ada tiga jenis operasi baris dasar:

1) Menukar dua baris,
2) Mengalikan satu baris dengan angka bukan nol,
3) Menambahkan kelipatan satu baris ke baris lain.

Dengan menggunakan operasi ini, sebuah matriks selalu dapat ditransformasikan menjadi matriks segitiga atas, dan pada kenyataannya yang berbentuk eselon baris. Setelah semua koefisien terkemuka (entri bukan nol paling kiri di setiap baris) adalah 1, dan setiap kolom yang mengandung koefisien terkemuka memiliki nol di tempat lain, matriks dikatakan dalam bentuk eselon baris tereduksi. Bentuk akhir ini unik; dengan kata lain, itu tidak tergantung pada urutan operasi baris yang digunakan. Misalnya, dalam urutan operasi baris berikut (di mana beberapa operasi elementer dapat dilakukan pada setiap langkah), matriks ketiga dan keempat adalah yang dalam bentuk eselon baris, dan matriks terakhir adalah bentuk eselon baris tereduksi yang unik.

2.2 Metode Eliminasi Gauss-Jordan

Metode Eliminasi Gauss-Jordan adalah ekstensi dari metode eliminasi Gauss, dengan melanjutkan membuat segitiga bawah dari matriks tersebut, sehingga menghasilkan sebuah matriks identitas. Biasanya metode ini lazim digunakan untuk metode pemecahan sistem persamaan linear (SPL). Hal ini dilakukan dengan mengubah matriks augmented sistem menjadi bentuk eselon baris tereduksi melalui operasi baris.

2.3 Determinan Matriks

Dalam aljabar linier, determinan adalah nilai skalar yang dapat dihitung dari unsur-unsur matriks kuadrat dan mengkodekan sifat-sifat tertentu dari transformasi linear yang dijelaskan oleh matriks. Determinan matriks A dinotasikan dengan det (A), det A, atau | A |. Secara geometris, ini dapat dilihat sebagai faktor penskalaan volume dari transformasi linear yang dijelaskan oleh matriks. Ini juga volume yang ditandatangani dari n-dimensional parallelepiped spanned oleh kolom atau vektor baris dari matriks. Determinan positif atau negatif sesuai dengan apakah pemetaan linear mempertahankan atau membalikkan orientasi ruang-n.

Determinan terjadi sepanjang matematika. Sebagai contoh, sebuah matriks sering digunakan untuk merepresentasikan koefisien dalam sistem persamaan linear, dan determinan dapat digunakan untuk menyelesaikan persamaan tersebut, walaupun metode solusi lain jauh lebih efisien secara komputasi. Dalam aljabar linier, sebuah matriks (dengan entri dalam suatu bidang) tidak dapat dibalik jika dan hanya jika determinannya bukan nol, dan oleh karena itu matriksnya singular (tidak dapat dibalik) jika dan hanya jika determinannya adalah nol. Ini mengarah pada penggunaan determinan dalam mendefinisikan karakteristik polinomial dari matriks, yang akarnya adalah nilai eigen. Dalam geometri analitik, determinan menyatakan volume n-dimensi *parallelepiped* n-dimensi. Ini mengarah pada penggunaan determinan dalam kalkulus, determinan Jacobian dalam perubahan aturan variabel untuk integral fungsi dari beberapa variabel. Determinan sering muncul dalam identitas aljabar seperti identitas Vandermonde.

2.4 Matriks Balikan

Dalam aljabar linier, matriks ukuran n x n (sebut saja matriks A) disebut dapat dibalikkan (*invertible*, juga non-singular atau *non-degenerate*) jika terdapat matriks persegi nxn B sedemikian rupa sehingga:

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$$

dan I_n menunjukkan matriks identitas nxn dan perkalian yang digunakan adalah perkalian matriks biasa. Jika ini masalahnya, maka matriks B secara unik ditentukan oleh A dan disebut matriks kebalikan dari A, dilambangkan dengan A^{-1}.

Matriks kuadrat yang tidak dapat dibalik disebut singular atau *degenerate*. Matriks kuadrat adalah singular jika dan hanya jika determinannya adalah 0. Matriks singular jarang dalam arti bahwa matriks kuadrat yang dipilih secara acak dari distribusi seragam kontinu pada entri-entrinya hampir tidak pernah singular.

Matriks non-square (matriks $m \times n$ yang $m \neq n$) tidak memiliki invers. Namun, dalam beberapa kasus, matriks seperti itu mungkin memiliki invers kiri atau invers kanan. Jika A adalah $m \times n$ dan pangkat A adalah sama dengan n ($n \leq m$), maka A memiliki invers kiri: matriks $n \times m$ B sehingga BA = In. Jika A memiliki peringkat m ($m \leq n$), maka ia memiliki invers kanan: matriks $n \times m$ B sehingga AB = Im.

Pembalikan matriks adalah proses menemukan matriks B yang memenuhi persamaan sebelumnya untuk matriks A. Sementara kasus yang paling umum adalah bahwa matriks atas bilangan real atau kompleks, semua definisi ini dapat diberikan untuk matriks atas cincin apa pun. Namun, dalam kasus cincin yang komutatif, kondisi untuk matriks kuadrat menjadi tidak dapat dibalik adalah bahwa determinannya tidak dapat dibalik di dalam cincin, yang secara umum merupakan persyaratan yang lebih ketat daripada bukan nol. Untuk dering non-komutatif, penentu biasa tidak ditentukan. Kondisi untuk keberadaan invers kiri atau invers kanan lebih rumit karena gagasan peringkat tidak ada di atas cincin.

2.5 Matriks Kofaktor

Kofaktor adalah angka yang didapatkan ketika dilakukan penghapusan kolom dan baris elemen yang ditunjuk dalam sebuah matriks, yang hanya berupa kisi numerik dalam bentuk persegi panjang atau persegi. Kofaktor selalu didahului oleh tanda positif (+) atau negatif (-), tergantung apakah elemen tersebut dalam posisi + atau -.

Kofaktor ij dari suatu matriks ditentukan oleh:

$$C_{ij} = (-1)^{i+j}\left|A_{ij}\right|$$

dengan matriks A_{ij} adalah matriks minor dari matriks A dengan menghilangkan baris ke-i dan kolom ke-j.

Maka, matriks kofaktor dari matriks A adalah matriks berukuran A yang setiap elemen a_{ij} nya adalah kofaktor ij dari matriks A tersebut. Matriks kofaktor penting untuk perhitungan determinan dan adjoin sebuah matriks.

2.6 Matriks Adjoin

Matriks adjoin dari sebuah matriks A adalah transpose dari matriks kofaktor A. Matriks adjoin digunakan untuk mencari matriks balikan menggunakan metode kofaktor, dengan matriks balikan (*inverse of* A atau A^{-1}) adalah 1/determinan(A) dikalikan dengan semua elemen di matriks adjoin dari A.

2.7 Kaidah Kramer

Kaidah atau aturan Kramer adalah metode yang dapat menyelesaikan sistem persamaan linear menggunakan determinan. Biasanya aturan Kramer digunakan untuk menyelesaikan sistem persamaan linear (SPL) yang dinyatakan dalam bentuk *augmented matrix*. Aturannya adalah:

1) Misalkan tujuannya adalah ingin mengetahui nilai x_i, maka buatlah *coefficient matrix* dari persamaan tersebut, namun elemen-elemen di kolom ke-i digantikan dengan setiap bi pada *augmented matrix*. Sebutlah matriks ini sebagai matriks numerator.
2) Pastikan *coefficient matrix* dari persamaan tersebut memiliki determinan tidak sama dengan nol. Jika memiliki determinan sama dengan nol, kemungkinan besar SPL memiliki banyak solusi atau tidak ada solusi.
3) Untuk mengetahui nilai x_i yang memenuhi SPL, bagi determinan dari matriks numerator dengan determinan dari *coefficient matrix*.

Untuk menggunakan kaidah Kramer, dibutuhkan kaidah-kaidah lain untuk mengetahui determinan sebuah matriks. Tentunya, kaidah ini sering digunakan untuk menyelesaikan SPL pada program-program di dunia nyata.

2.8 Interpolasi Polinom

Interpolasi polinomial adalah metode memperkirakan nilai antara titik data yang diketahui. Ketika data grafis mengandung celah, tetapi data tersedia di kedua sisi celah atau pada beberapa titik tertentu dalam celah, estimasi nilai dalam celah dapat dibuat dengan interpolasi.

Metode interpolasi yang paling sederhana adalah menggambar garis lurus antara titik data yang diketahui dan menganggap fungsinya sebagai kombinasi dari garis lurus tersebut. Metode ini, yang disebut interpolasi linier, biasanya menimbulkan kesalahan besar. Pendekatan yang lebih tepat menggunakan fungsi polinomial untuk menghubungkan titik-titik. Polinomial adalah ekspresi matematis yang terdiri dari sejumlah suku, masing-masing suku termasuk variabel atau variabel yang dipangkatkan dan dikalikan dengan suatu koefisien. Untuk menghitung interpolasi polinomial sendiri, salah satu cara yang dapat digunakan adalah dengan matriks.

BAB III
IMPLEMENTASI PROGRAM

3.1 Deskripsi Singkat

Program ditulis menggunakan bahasa Java. Dibuat class `Matriks` untuk memproses semua operasi yang melibatkan matriks dalam penyelesaian masalah di bab I. Serta class `DriverMatriks` untuk mengurus hal-hal yang tidak menyangkut operasi pada matriks. Dalam implementasi class `Matriks` digunakan beberapa atribut sebagai berikut:

1. Panjang efektif baris dan kolom `int NeffKol, NeffBar`
   Atribut NeffKol dan NeffBar adalah atribut bertipe integer yang menyatakan ukuran dari matriks ($NeffKol \times NeffBar$).
2. Array matriks bernama angka `BigDecimal[][] angka`
   Atribut angka adalah atribut bertipe array 2D dari BigDecimal yang menyatakan matriks.

3.2 Metode Matriks

Pada program yang kami buat, kami menggunakan beberapa metode sebagai berikut:

1) Prosedur `void InputDataMat()`
   Metode InputDataMat adalah prosedur yang digunakan untuk membaca inputan ukuran matriks dan elemen di dalamnya.
2) Fungsi `int InputDataMatFile(String fileName)`
   Metode InputDataMatFile adalah fungsi yang digunakan untuk membaca inputan ukuran matriks dan elemen di dalamnya dari suatu file return value dari fungsi ini bisa 0 jika pembacaan berhasil atau -1 jika gagal (ukuran matriks yang dibaca tidak seragam).
3) Fungsi `String OutputDataMat()`
   Fungsi yang mengembalikan string berupa matriks this.
4) Prosedur `void OBESwap(int a, int b)`
   Prosedur ini digunakan untuk men-*swap* atau menukar baris a dan b sesuai Operasi Baris Elementer (OBE).

5) Prosedur `void OBEScale(int a, BigDecimal k)`

   Prosedur ini digunakan untuk men-*scale* atau mengalikan baris a dengan dengan konstanta k sesuai Operasi Baris Elementer (OBE).

6) Prosedur `void OBEReplace(int a, int b, BigDecimal k)`

   Prosedur ini digunakan untuk men-*scale* atau menambahkan baris a dengan baris b yang dikalikan konstanta k sesuai Operasi Baris Elementer (OBE).

7) Prosedur `void KaliKons(BigDecimal k)`

   Prosedur ini mengalikan setiap elemen pada matriks dengan konstanta k.

8) Fungsi `public Matriks KaliMat(Matriks M1)`

   Fungsi ini mengembalikan sebuah matriks hasil perkalian matriks this dan matriks M1 sesuai aturan perkalian matriks.

9) Prosedur `void CopyMatriks(Matriks M)`

   Prosedur untuk menyalin matriks this ke matriks M.

10) Prosedur `void ReverseCopyMatriks(Matriks M)`

    Prosedur untuk menyalin matriks M ke matriks this.

11) Prosedur `void ClearMatriks()`

    Mengosongkan matriks.

12) Prosedur `void Augmented(Matriks M)`

    Membentuk matriks Augmented dari matriks this.

13) Prosedur `void unAugmented(Matriks M, int KSize)`

    Membentuk matriks Unaugmented dari matriks this, M akan berisi matriks yang "dipotong", dan KSize adalah panjang kolom yang "dipotong".

14) Fungsi `Matriks MatriksIdentitas()`

    Mengirimkan matriks identitas seukuran matriks this.

15) Fungsi `int toSegitigaAtas()`

    Membentuk matriks menjadi bentuk segitiga atasnya dan mengembalikan jumlah penggunaan OBESwap di prosesnya.

16) Prosedur `void toEchelon()`

    Membentuk matriks menjadi bentuk echelon yang belum tereduksi.

17) Prosedur `void toReducedEchelon()`

    Membentuk matriks menjadi bentuk echelon tereduksi.

18) Fungsi `BigDecimal DeterminanOBE()`

Menghitung determinan matriks this dan mengembalikan nilainya menggunakan metode OBE atau segitiga atas.

19) Fungsi `boolean IsInvertible()`

    Mengirimkan nilai boolean yang berarti matriks ada balikannya, menggunakan pengecekan determinan.

20) Prosedur `void InverseOBE()`

    Meng-*inverse* atau membalikkan matriks menggunakan metode OBE.

21) Fungsi `Matriks MinorMatriks(int idxRow, int idxCol)`

    Mengembalikan matriks yang berupa matriks minor dari matriks this pada kolom idxCol dan baris idxRow.

22) Fungsi `BigDecimal DeterminanKofaktor()`

    Menghitung determinan matriks this dan mengembalikan nilainya menggunakan metode ekspansi kofaktor.

23) Fungsi `Matriks Kofaktor()`

    Mengembalikan matriks kofaktor dari matriks this.

24) Fungsi `Matriks Transpose()`

    Mengembalikan matriks yang di-*transpose* dari matriks this.

25) Fungsi `Matriks Adjoin()`

    Mengembalikan matriks adjoin yaitu *transpose* dari matriks kofaktor yang terbuat dari matriks this.

26) Fungsi `Matriks InverseAdjoin()`

    Mengembalikan matriks berupa balikan dari matriks this dengan metode matriks adjoin.

27) Fungsi `BigDecimal SolveSPLKramer(int valNum)`

    Mengembalikan nilai solusi berupa BigDecimal dari x ke-valNum dari sebuah matriks dengan metode Kramer.

28) Fungsi `boolean isNoSol()`

    Mengembalikan nilai keberadaan solusi dari matriks this untuk penyelesaian SPL dengan Gauss atau Gauss-Jordan.

29) Fungsi `boolean isNoSol2()`

    Mengembalikan nilai keberadaan solusi dari matriks this untuk penyelesaian SPL menggunakan Kramer atau matriks balikan dengan pengecekan nilai determinan

30) Fungsi `boolean isFreeVar(int valNum)`

Mengembalikan nilai apakah x ke-valNum adalah variabel bebas atau bukan.

31) Fungsi `String SolveSPLgJordan(int valNum)`

Mengembalikan nilai berupa sebuah string, solusi dari matriks augmented this untuk x ke-valNum menggunakan metode Gauss-Jordan, nilai yang dikembalikan berupa string untuk mengakomodasi adanya variabel bebas.

32) Fungsi `String SolveSPLgauss(int valNum)`

Mengembalikan nilai berupa sebuah string, solusi dari matriks augmented this untuk x ke-valNum menggunakan metode Gauss, nilai yang dikembalikan berupa string untuk mengakomodasi adanya variabel bebas.

33) Fungsi `BigDecimal SolveSPLinverse(int valNum)`

Mengembalikan nilai solusi berupa BigDecimal dari x ke-valNum dari sebuah matriks dengan metode matriks balikan.

3.3 Metode DriverMatriks

Dalam implementasi class `DriverMatriks` digunakan beberapa metode sebagai berikut:

1) Prosedur `public static void main()`

Sebagai program utama.

2) Prosedur `public static void PrintMenu()`

3) Prosedur `public static void MainMenu(Matriks M)`

Menyediakan pilihan operasi-operasi pada matriks M seperti determinan, balikkan, interpolasi, penyelesaian spl, dll.

4) Prosedur `public static void SPLMenu(Matriks M)`

Menyediakan menu penyelesaian SPL pada matriks M. Berisi penyelesaian dengan metode Gauss, Gauss-Jordan, matriks balikan, dan aturan Kramer.

5) Prosedur `public static void DETMenu(Matriks M)`

Menyediakan menu perhitungan determinan pada matriks M. Berisi penyelesaian dengan metode OBE atau segitiga atas dan ekspansi kofaktor.

6) Prosedur `public static void INVMenu(Matriks M)`

Menyediakan menu perhitungan balikan untuk matriks M. Berisi penyelesaian dengan metode OBE dan matriks adjoin.

7) Prosedur `public static void MintaInput(Matriks M)`

Meminta input dari pengguna bisa dari keyboard maupun file, terdapat pengecekan apakah file valid atau tidak jika pilihan dari file

8) Prosedur `public static void MintaOutput(String out)`

   Memberikan output dengan 2 metode yaitu ke layar atau ke file.

9) Prosedur `public static void InputInterpolasi(Matriks X, Matriks Y)`

   Meminta input secara khusus untuk persoalan interpolasi, hasil x-y interpolasi akan dimasukkan pada matriks X dan Y.

10) Prosedur `public static void Interpolasi()`

    Mengurus masalah interpolasi, termasuk input, output, dan taksiran.

11) Fungsi `public static boolean FileCheck(String fileName)`

    Mengembalikan boolean apakah file dengan nama fileName ada atau tidak.

3.4 Garis Besar Program

Garis besar program utama dapat dijelaskan sebagai berikut:

1) Pertama pengguna akan diberikan menu utama seperti berikut:

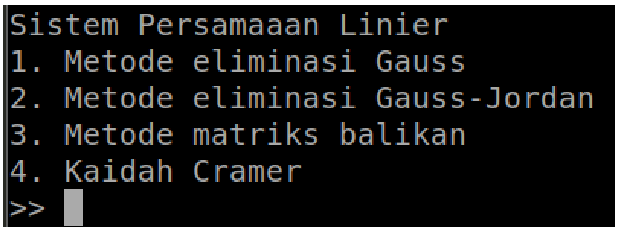
```
MENU
1. Sistem Persamaaan Linier
2. Determinan
3. Matriks balikan
4. Matriks kofaktor
5. Adjoin
6. Interpolasi Polinom
7. Keluar
>>
```

2) Pengguna bisa memilih SPL, determinan, matriks balikan, matriks kofaktor, matriks adjoin, interpolasi polinom, dan keluar dari program.

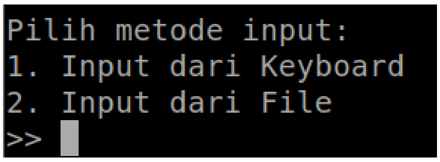
3) Jika pengguna memilih SPL, determinan, atau matriks balikan, pengguna akan diberikan opsi untuk metode penghitungannya.

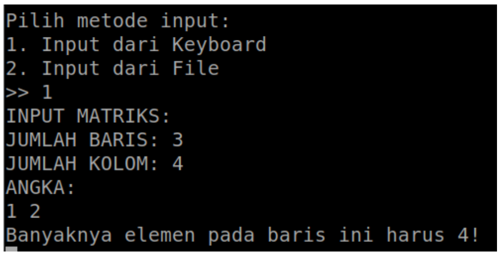
```
Determinan
1. Metode Segitiga Atas
2. Metode Ekspansi Kofaktor
>>
```

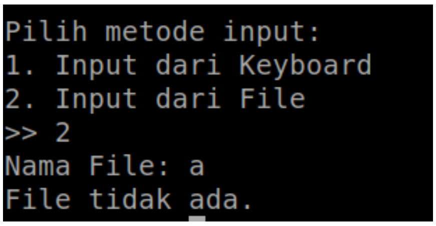
```
Matriks Balikan
1. Metode eliminasi Gauss-Jordan
2. Metode Matriks Adjoin
>>
```

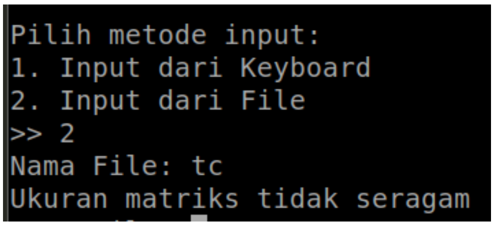


4) Setiap kali program membutuhkan masukan matriks, *input* bisa dimasukkan melalui 2 opsi, *keyboard* atau *file*, jika ada kesalahan dalam input, maka pengguna akan diminta *input* ulang.

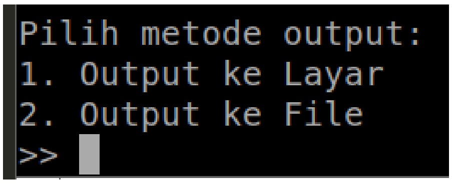








5) Setiap kali program mengeluarkan luaran, *output* bisa dikeluarkan dengan 2 cara, melalui layar atau ke *file*.



6) Setelah melakukan suatu aksi, pengguna akan diberikan menu utama kembali.

BAB IV
EKSPERIMEN

Untuk eksperimen, kami melakukan uji *test case* berdasarkan studi kasus modul spesifikasi tugas besar, yaitu sebagai berikut:

4.1 Temukan solusi SPL Ax = b, berikut:

- SPL 1

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

Menurut program dengan metode Gauss-Jordan, SPL tidak memiliki solusi.

```
INPUT MATRIKS:
JUMLAH BARIS: 4
JUMLAH KOLOM: 5
ANGKA:
1 1 -1 -1 1
2 5 -7 -5 -2
2 -1 1 3 4
5 2 -4 2 6

SPL tidak memiliki solusi!
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.

$$\begin{cases} x_1 + x_2 - x_3 - x_4 = 1 \\ 3 \times x_2 - 5 \times x_3 - 3 \times x_4 = -4 \\ -2 \times x_3 + 2 \times x_4 = -2 \\ 0 = 1 \end{cases}$$

There are no solutions.

- SPL 2

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

Menurut program dengan metode Gauss-Jordan, SPL memiliki solusi sebagai berikut:

```
INPUT MATRIKS:
JUMLAH BARIS: 4
JUMLAH KOLOM: 6
ANGKA:
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1
```

```
Pilih metode output:
1. Output ke Layar
2. Output ke File
1
x1 = 3.000 + 1.000s5
x2 = 2.000s5
x3 = s3
x4 = - 1.000 + 1.000s5
x5 = s5
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.

Answer:

- $x_1 = 3 + x_5$
- $x_2 = 2 \times x_5$
- $x_3 = x_3$
- $x_4 = -1 + x_5$
- $x_5 = x_5$

- SPL 3

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

Menurut program dengan metode Gauss, SPL memiliki solusi sebagai berikut:

```
INPUT MATRIKS:
JUMLAH BARIS: 3
JUMLAH KOLOM: 7
ANGKA:
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1
```

```
Pilih metode output:
1. Output ke Layar
2. Output ke File
1
x1 = s1
x2 = 1.000 - 1.000s6
x3 = s3
x4 = - 2.000 - 1.000s6
x5 = 1.000 + 1.000s6
x6 = s6
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.

Answer:

- $x_1 = x_1$
- $x_2 = 1 - x_6$
- $x_3 = x_3$
- $x_4 = -2 - x_6$
- $x_5 = 1 + x_6$
- $x_6 = x_6$

- SPL 4

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

Matriks Hilbert n=6

Menurut program dengan metode Kramer, SPL memiliki solusi sebagai berikut:

➔ Isi file tc (dibulatkan untuk tampilan):

```
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 1
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0
```

➔ Output:

```
Pilih metode input:
1. Input dari Keyboard
2. Input dari File
2
Nama File:
tc

Pilih metode output:
1. Output ke Layar
2. Output ke File
1
x1 = 36.000
x2 = - 630.000
x3 = 3360.000
x4 = - 7560.000
x5 = 7560.000
x6 = - 2772.000
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.

Answer:
- $x_1 = 36$
- $x_2 = -630$
- $x_3 = 3360$
- $x_4 = -7560$
- $x_5 = 7560$
- $x_6 = -2772$

- SPL 5

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

Matriks Hilbert n=10

Menurut program dengan metode Inverse, SPL memiliki solusi sebagai berikut:

➔ Isi file tc (dibulatkan untuk tampilan):

```
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 1
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0
0.142857 0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0
0.125000 0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0.058824 0
0.111111 0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0.058824 0.055556 0
0.100000 0.090909 0.083333 0.076923 0.071429 0.066667 0.062500 0.058824 0.055556 0.052632 0
```

➔ Output:

```
Nama File:
tc

Pilih metode output:
1. Output ke Layar
2. Output ke File
1
x1 = 100.000
x2 = - 4950.000
x3 = 79200.000
x4 = - 600600.000
x5 = 2522520.000
x6 = - 6306300.000
x7 = 9609600.000
x8 = - 8751600.000
x9 = 4375800.000
x10 = - 923780.000
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama.

Reduced Row Echelon Form of a Matrix (RREF)

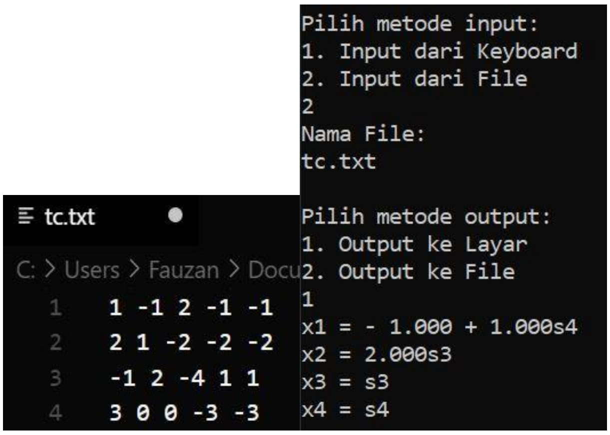
$$\left[\begin{array}{cccccccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4950 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 79200 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -600600 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2522520 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -6306299 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 9609599 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -8751599 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4375800 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -923780 \end{array}\right]$$

4.2 SPL berbentuk:

- SPL 1 matriks augmented

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

menggunakan metode Gauss-Jordan dari masukan file tc.txt, didapatkan:



```
tc.txt
C: > Users > Fauzan > Docu
1    1 -1 2 -1 -1
2    2 1 -2 -2 -2
3    -1 2 -4 1 1
4    3 0 0 -3 -3
```

```
Pilih metode input:
1. Input dari Keyboard
2. Input dari File
2
Nama File:
tc.txt

Pilih metode output:
1. Output ke Layar
2. Output ke File
1
x1 = - 1.000 + 1.000s4
x2 = 2.000s3
x3 = s3
x4 = s4
```

Dari kalkulator *online*, didapat hasil yang serupa:

$$X = \begin{pmatrix} -1 + x_4 \\ 2 \times x_3 \\ x_3 \\ x_4 \end{pmatrix}$$

- SPL 2 matriks augmented

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

Menggunakan metode Gauss-Jordan, didapatkan:

```
Pilih metode output:
1. Output ke Layar
2. Output ke File
1
x1 = 0.000
x2 = 2.000
x3 = 1.000
x4 = 1.000
```

Saat dicoba di kalkulator *online*, didapat hasil yang sama:

$$X = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 1 \end{pmatrix}$$

- SPL 1 persamaan

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

Diubah menjadi bentuk augmented matriks, menjadi:

```
tc.txt
C: > Users > Fauzan > D
8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3
```

Dan dengan menggunakan metode Kramer, didapatkan solusinya:

```
x1 = -0.224
x2 = 0.182
x3 = 0.709
x4 = -0.258
```

yang sama dengan solusi yang di kalkulator *online*:

$$x_1 = \frac{-83}{370}$$
$$x_2 = \frac{27}{148}$$
$$x_3 = \frac{105}{148}$$
$$x_4 = \frac{-191}{740}$$

- SPL 2 persamaan

Reduced Row Echelon Form of a Matrix (RREF)

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

```
0 0 0 0 0 0 1 1 1 13
0 0 0 1 1 1 0 0 0 15
1 1 1 0 0 0 0 0 0 8
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.04289 0 0.04289 0.75 0.04289 0.75 0 3.81
0 0 1 0 0 1 0 0 1 18
0 1 0 0 1 0 0 1 0 12
1 0 0 1 0 0 1 0 0 6
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04
```

```
Pilih metode input:
1. Input dari Keyboard
2. Input dari File
2
Nama File:
tc

SPL tidak memiliki solusi!
```

4.3 Untuk persoalan determinan, matriks balikan, dan matriks kofaktor, silahkan cari masing-masing dua buah matriks yang berukuran 5 x 5 dan 10 x 10.

- Untuk matriks 5x5 digunakan matriks berikut:

$$\begin{pmatrix} -3 & -5 & 4 & -3 & -3 \\ 10 & -1 & 1 & -10 & 8 \\ -9 & 8 & 2 & 6 & -5 \\ 6 & 1 & 9 & -5 & 5 \\ -6 & -4 & -3 & 0 & 4 \end{pmatrix}$$

Determinannya menurut program yang dibuat adalah 48547.

```
tc

-3 -5 4 -3 -3
10 -1 1 -10 8
-9 8 2 6 -5
6 1 9 -5 5
-6 -4 -3 0 4
```

```
Determinan
1. Metode Segitiga Atas
2. Metode Ekspansi Kofaktor
>> 2

Pilih metode input:
1. Input dari Keyboard
2. Input dari File
2
Nama File:
tc

Determinannya: 48547
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama

Input:

$$\begin{pmatrix} -3 & -5 & 4 & -3 & -3 \\ 10 & -1 & 1 & -10 & 8 \\ -9 & 8 & 2 & 6 & -5 \\ 6 & 1 & 9 & -5 & 5 \\ -6 & -4 & -3 & 0 & 4 \end{pmatrix}$$

Determinant:

48547

Untuk balikannya, dengan program yang dibuat didapatkan:

```
Pilih metode input:
1. Input dari Keyboard
2. Input dari File
2
Nama File:
tc

Inversenya:
Pilih metode output:
1. Output ke Layar
2. Output ke File
1
-0.050 -0.040 -0.076 0.018 -0.076
-0.030 0.088 0.107 -0.029 -0.028
0.022 -0.067 -0.008 0.111 0.001
-0.116 -0.173 -0.099 0.097 0.015
-0.089 -0.022 -0.014 0.081 0.109
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama (perbedaan pada pembulatan):

Inverse:

$$\begin{pmatrix} -0.049828 & -0.0399613 & -0.076256 & 0.0183122 & -0.0756586 \\ -0.0301975 & 0.0878118 & 0.106742 & -0.0294148 & -0.0280759 \\ 0.0216285 & -0.0669866 & -0.00823944 & 0.111109 & 0.00100933 \\ -0.11562 & -0.173337 & -0.0988115 & 0.0974726 & 0.0146044 \\ -0.0887181 & -0.0223701 & -0.0138217 & 0.081385 & 0.109193 \end{pmatrix}$$

- Untuk matriks 10x10 digunakan matriks berikut:

$$\begin{bmatrix} 21 & 34 & -1 & -42 & -36 & 37 & -32 & 29 & 31 & -48 \\ 46 & 45 & 19 & -49 & -13 & 35 & -48 & -2 & -44 & -40 \\ 6 & 9 & 46 & -23 & 27 & -44 & -13 & -33 & 19 & -15 \\ 12 & 27 & -34 & 6 & 0 & 14 & -35 & -23 & 22 & -14 \\ -47 & 25 & -17 & 36 & 7 & -10 & -11 & -16 & 11 & 31 \\ 47 & 13 & -33 & -32 & -14 & 18 & -49 & 26 & 24 & 2 \\ 38 & 31 & -43 & 36 & 35 & 41 & 16 & 16 & -20 & -4 \\ 11 & -37 & 5 & 33 & -20 & -40 & -28 & -41 & -32 & 10 \\ -21 & 2 & 11 & -44 & 29 & -28 & -47 & 38 & -10 & 27 \\ -17 & -31 & -6 & 4 & 17 & -24 & 49 & -25 & 15 & 10 \end{bmatrix}$$

Determinannya menurut program yang dibuat adalah .

```
tc                    Matriks.ja
21 34 -1 -42 -36 37 -32 29 31 -48
46 45 19 -49 -13 35 -48 -2 -44 -40
6 9 46 -23 27 -44 -13 -33 19 -15
12 27 -34 6 0 14 -35 -23 22 -14
-47 25 -17 36 7 -10 -11 -16 11 31
47 13 -33 -32 -14 18 -49 26 24 2
38 31 -43 36 35 41 16 16 -20 -4
11 -37 5 33 -20 -40 -28 -41 -32 10
-21 2 11 -44 29 -28 -47 38 -10 27
-17 -31 -6 4 17 -24 49 -25 15 10
```

```
Determinan
1. Metode Segitiga Atas
2. Metode Ekspansi Kofaktor
>> 1

Pilih metode input:
1. Input dari Keyboard
2. Input dari File
2
Nama File:
tc

Determinannya: -225642673261163966.000
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang tidak berbeda jauh, perbedaan ini mungkin terjadi karena operasi pada floating point yang tidak presisi dalam angka yang besar.

Determinant:

```
-225642673261164000.000
```

Untuk balikannya, dengan program yang dibuat didapatkan:

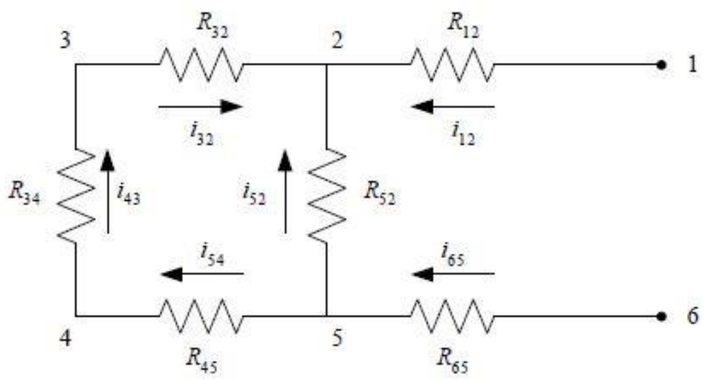
```
Nama File:
tc

Pilih metode output:
1. Output ke Layar
2. Output ke File
1
-0.002 0.002 0.009 -0.015 0.003 0.016 0.007 0.004 -0.008 0.001
0.028 0.014 0.010 -0.038 0.039 0.012 0.018 0.013 -0.002 0.027
-0.026 -0.007 0.007 0.009 -0.016 -0.003 -0.016 -0.018 -0.011 -0.038
-0.001 -0.012 0.004 0.003 -0.004 -0.006 0.005 0.003 -0.004 -0.020
-0.022 -0.008 0.002 0.030 -0.024 -0.012 -0.003 -0.015 0.007 -0.019
-0.051 -0.006 -0.012 0.047 -0.037 -0.009 -0.030 -0.036 -0.011 -0.049
0.016 0.008 0.003 -0.028 0.018 0.007 0.011 0.006 -0.004 0.027
0.022 -0.007 0.001 -0.020 0.006 -0.001 0.014 0.010 0.009 0.003
-0.011 -0.012 0.006 0.013 -0.009 0.005 -0.009 -0.013 -0.007 -0.017
-0.029 0.005 0.000 -0.007 0.007 0.021 -0.013 -0.015 -0.012 -0.012
```

Jika dibandingkan dengan kalkulator *online*, didapat hasil yang sama:

Matrix Inverse:

```
-0.002  0.002  0.009 -0.015  0.003  0.016  0.007  0.004 -0.008  0.001
 0.028  0.014  0.010 -0.038  0.039  0.012  0.018  0.013 -0.002  0.027
-0.026 -0.007  0.007  0.009 -0.016 -0.003 -0.016 -0.018 -0.011 -0.038
-0.001 -0.012  0.004  0.003 -0.004 -0.006  0.005  0.003 -0.004 -0.020
-0.022 -0.008  0.002  0.030 -0.024 -0.012 -0.003 -0.015  0.007 -0.019
-0.051 -0.006 -0.012  0.047 -0.037 -0.009 -0.030 -0.036 -0.011 -0.049
 0.016  0.008  0.003 -0.028  0.018  0.007  0.011  0.006 -0.004  0.027
 0.022 -0.007  0.001 -0.020  0.006 -0.001  0.014  0.010  0.009  0.003
-0.011 -0.012  0.006  0.013 -0.009  0.005 -0.009 -0.013 -0.007 -0.017
-0.029  0.005  0.000 -0.007  0.007  0.021 -0.013 -0.015 -0.012 -0.012
```

4.4 Dalam sebuah rangkaian listrik berlaku hukum-hukum arus Kirchoff menyatakan bahwa jumlah aljabar dari semua arus yang memasuki suatu simpul haruslah nol: Diberikan sebuah rangkaian:



dan persamaan dari rangkaian menggunakan hukum Kirchoff :

$$
\begin{array}{llll}
i_{12} & + i_{52} & + i_{32} & = 0 \\
i_{65} & - i_{52} & - i_{54} & = 0 \\
i_{43} & - i_{32} & & = 0 \\
i_{54} & - i_{43} & & = 0
\end{array}
$$

dan persamaan hukum Ohm:

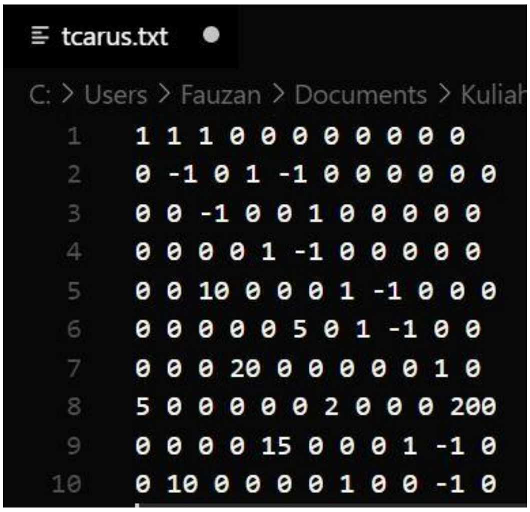
$$
\begin{array}{llll}
i_{32} R_{32} & - V_3 & + V_2 & = 0 \\
i_{43} R_{43} & - V_4 & + V_3 & = 0 \\
i_{65} R_{65} & & + V_5 & = 0 \\
i_{12} R_{12} & & + V_2 & = 0 \\
i_{54} R_{54} & - V_5 & + V_4 & = 0 \\
i_{52} R_{52} & - V_5 & + V_2 & = 0
\end{array}
$$

Yang ditanya adalah variabel i_{12}, i_{52}, i_{32}, i_{65}, i_{54}, i_{43}, V_2, V_3, V_4, V_5.

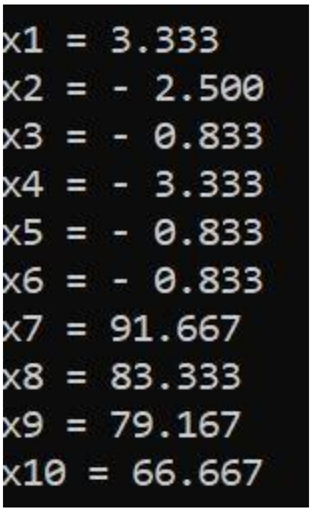
Pertama, asumsikan bahwa:

- $x_1 = i_{12}$
- $x_2 = i_{52}$
- $x_3 = i_{32}$
- $x_4 = i_{65}$
- $x_5 = i_{54}$
- $x_6 = i_{43}$
- $x_7 = V_2$
- $x_8 = V_3$
- $x_9 = V_4$
- $x_{10} = V_5$

Dengan menggabungkan persamaan di hukum Ohm dan Kirchoff, dan dimasukkan ke file input file tcarus.txt dan metode Gauss, didapat:



```
tcarus.txt
C: > Users > Fauzan > Documents > Kuliah
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 5 0 1 -1 0 0
0 0 0 20 0 0 0 0 0 1 0
5 0 0 0 0 0 2 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
```



```
x1 = 3.333
x2 = - 2.500
x3 = - 0.833
x4 = - 3.333
x5 = - 0.833
x6 = - 0.833
x7 = 91.667
x8 = 83.333
x9 = 79.167
x10 = 66.667
```

Kesimpulannya adalah solusi-solusi untuk persamaan-persamaan fisis di atas adalah:

- i_{12}= 3.333 A
- i_{52}= -2.500 A
- i_{32}= -0.833 A
- i_{65}= 3.333 A
- i_{54}= -0.833 A
- i_{43}= -0.833 A
- V_2= 91.667 V
- V_3= 83.333 V
- V_4= 79.167 V
- V_5= 66.667 V

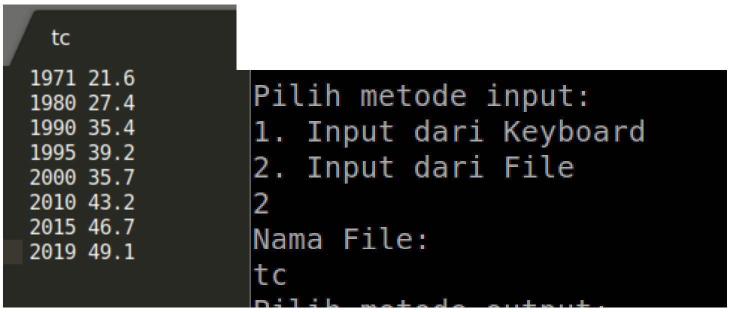
Dengan tanda minus (-) menunjukkan arah yang berlawanan dengan asumsi di diagram rangkaian di atas.

4.5 (Interpolasi) Jumlah penduduk Jawa Barat dari tahun 1971 hingga 2019 (dibulatkan ke juta) adalah sebagai berikut:

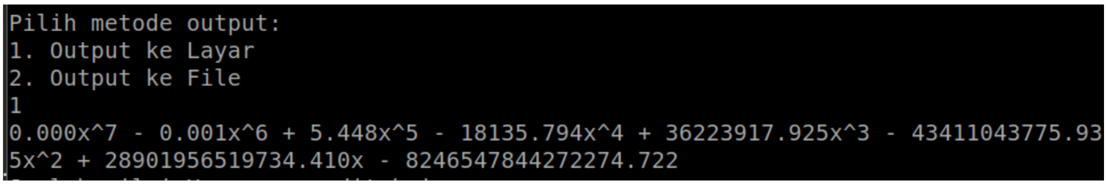
Tahun	Jumlah x 10^6)
1971	21,6
1980	27,4
1990	35,4
1995	39,2
2000	35,7
2010	43,2
2015	46,7
2019	49.1

Berdasarkan data tersebut prediksilah jumlah penduduk Jawa Barat pada tahun 1975, 1983, 1992, 2005, 2012 (atau nilai lain sesuai masukan user) dengan menggunakan polinom interpolasi.

Berikut masukan program:



keluaran program:



Jika dibandingkan dengan kalkulator *online* didapatkan fungsi yang serupa:

Input interpretation:

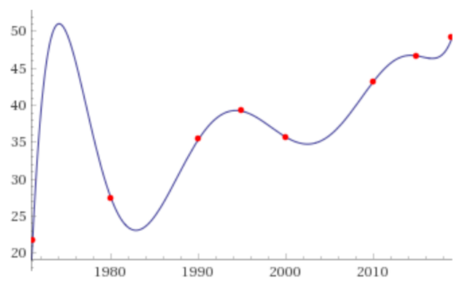
interpolating polynomial	{{1971, 21.6}, {1980, 27.4}, {1990, 35.4}, {1995, 39.2}, {2000, 35.7}, {2010, 43.2}, {2015, 46.7}, {2019, 49.1}}

Enlarge | Data | Customize | Plaintext

Interpolating polynomial:

$$6.5022 \times 10^{-8} x^7 - 0.000909145 x^6 + 5.44783 x^5 - 18\,135.8 x^4 + 3.62239 \times 10^7 x^3 - 4.3411 \times 10^{10} x^2 + 2.8902 \times 10^{13} x - 8.24655 \times 10^{15}$$

Plot of the interpolating polynomial:



untuk masalah prediksi atau taksiran:

```
Jumlah nilai X yang mau ditaksir:
5
Masukkan nilai X yang akan ditaksir: 1975
Taksiran nilai Y: 49.493
Masukkan nilai X yang akan ditaksir: 1983
Taksiran nilai Y: 23.095
Masukkan nilai X yang akan ditaksir: 1992
Taksiran nilai Y: 38.214
Masukkan nilai X yang akan ditaksir: 2005
Taksiran nilai Y: 35.932
Masukkan nilai X yang akan ditaksir: 2012
Taksiran nilai Y: 45.727
```

4.6 Sederhanakan fungsi:

$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

dengan polinom interpolasi derajat n di dalam selang [0, 2]. Sebagai contoh, jika n = 5, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak h = (2 – 0)/5 = 0.4.

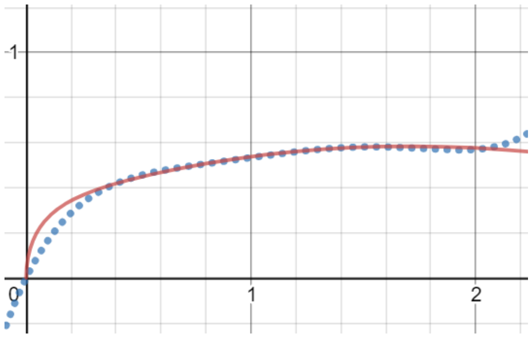
- Untuk n=5, titik-titik pasangan x-y dapat dilihat sebagai berikut:

```
tc
0.0 0.0
0.4 0.418884
0.8 0.507158
1.2 0.560925
1.6 0.583686
2.0 0.576652
```

Fungsi interpolasinya adalah:

```
Nama File:
tc
Pilih metode output:
1. Output ke Layar
2. Output ke File
1
0.236x^5 - 1.421x^4 + 3.237x^3 - 3.553x^2 + 2.035x
```

Dalam graphing calculator dibandingkan dengan fungsi awalnya:



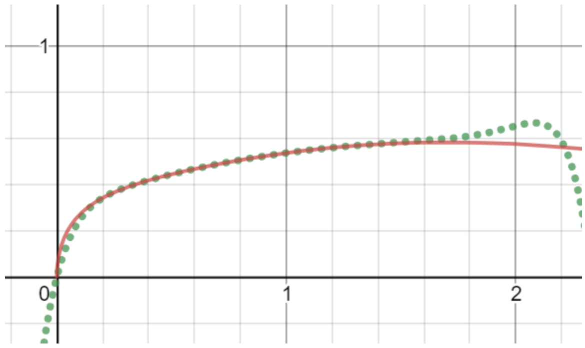
- Untuk n=10, titik-titik pasangan x-y dapat dilihat sebagai berikut:

```
tc
0.0 0.0
0.2 0.34277
0.4 0.418884
0.6 0.468431
0.8 0.507158
1.0 0.537883
1.2 0.560925
1.4 0.576187
1.6 0.583686
1.8 0.583675
2.0 0.576652
```

Fungsi interpolasi hasilnya adalah:

```
Nama File:
tc
Pilih metode output:
1. Output ke Layar
2. Output ke File
1
-0.472x^10 + 5.218x^9 - 25.209x^8 + 69.934x^7 - 123.072x^6 + 143.239x^5 - 111.48
4x^4 + 57.313x^3 - 18.811x^2 + 3.882x
```

Dalam graphing calculator dibandingkan dengan fungsi awalnya:



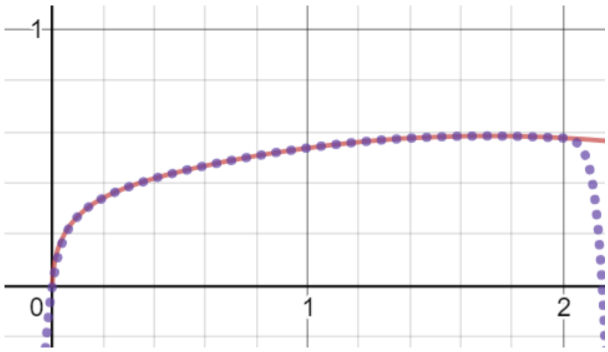
- Untuk n=20, titik-titik pasangan x-y dapat dilihat sebagai berikut:

```
tc
0.0 0.0
0.1 0.27069
0.2 0.34277
0.3 0.386532
0.4 0.418884
0.5 0.445431
0.6 0.468431
0.7 0.488865
0.8 0.507158
0.9 0.523479
1.0 0.537883
1.1 0.55037
1.2 0.560925
1.3 0.569532
1.4 0.576187
1.5 0.580897
1.6 0.583686
1.7 0.584593
1.8 0.583675
1.9 0.581
2.0 0.576652
```

Fungsi interpolasi hasilnya adalah (digunakan angka desimal lebih banyak):

```
Pilih metode output:
1. Output ke Layar
2. Output ke File
>> 1
-4.1840176406x^20 + 87.9394103013x^19 - 864.1163794220x^18 + 5274.2634562848x^17
 - 22407.9149052024x^16 + 70375.3780566506x^15 - 169362.6965630013x^14 + 319462.
1939758963x^13 - 479113.9420037688x^12 + 576188.0613261680x^11 - 557869.56949417
10x^10 + 434922.6851852206x^9 - 272077.1761739571x^8 + 135587.4569668257x^7 - 53
208.4860340712x^6 + 16170.4060770157x^5 - 3718.9399464404x^4 + 627.9051403017x^3
 - 75.1953368143x^2 + 6.4691426670x
```

Dalam graphing calculator dibandingkan dengan fungsi awalnya:



BAB V
KESIMPULAN

5.1 Kesimpulan

Banyak permasalahan-permasalahan matematis yang dapat diaplikasikan di dunia nyata, seperti menyelesaikan Sistem Persamaan Linear (SPL) dari persamaan *nodal* rangkaian listrik atau diagram gaya di sistem mekanika, atau melakukan interpolasi dari set data yang didapat untuk mendapatkan fungsi yang paling mendekati data tersebut, yang dapat diselesaikan menggunakan metode matriks. SPL dapat dimodelkan sebagai sebuah *augmented matrix*, atau dapat diselesaikan menggunakan interpolasi juga dapat dimodelkan sebagai sebuah matriks yang dapat digunakan untuk mendekati fungsi tersebut. Dengan menggunakan aplikasi aljabar linier (model matriks untuk SPL), banyak permasalahan dunia nyata dapat diselesaikan menggunakan pemodelan matriks.

5.2 Refleksi

Kami sebagai penulis merasa makin bertambah pengetahuan kami mengenai pengaplikasian aljabar linier di permasalahan sehari-hari. Kami telah dibekali teori-teori mengenai aljabar linier serta cara memprogram suatu program, sehingga kami makin paham tentang aplikasi penyelesaian masalah secara nyata (pengaplikasian konsep untuk menyelesaikan masalah menggunakan program). Kami juga meningkatkan *softskill* mengenai kerjasama di tim dan pembagian kerja dalam menyelesaikan tugas ini. Kami pun juga belajar untuk mendokumentasikan hasil pekerjaan kami dengan baik lewat laporan ini.

5.3 Saran

Pada pengerjaan program, ada beberapa kekeliruan dalam membuat program, salah satunya mengenai tipe elemen dari matriks, dimana harus menggunakan tipe variabel lain untuk menyelesaikan operasi untuk ukuran matriks yang lebih besar . Seharusnya kami melakukan riset dan pengecekan terlebih dahulu agar tidak mengganti-ganti kode program di akhir, sehingga lebih sistematis.

DAFTAR REFERENSI

https://en.wikipedia.org/wiki/Adjugate_matrix#Definition (diakses dan diterjemahkan pada 20.05 pada 19 September 2019)

https://en.wikipedia.org/wiki/Minor_(linear_algebra) (diakses dan diterjemahkan pada 20.08 pada 19 September 2019)

https://en.wikipedia.org/wiki/Determinant (diakses dan diterjemahkan pada 20.19 pada 19 September 2019)

https://en.wikipedia.org/wiki/Gaussian_Elimination (diakses dan diterjemahkan pada 20.21 pada 19 September 2019)

https://whatis.techtarget.com/definition/polynomial-interpolation (diakses dan diterjemahkan pada 09.53 pada 20 September 2019)