LAPORAN TUGAS BESAR 01

IF 2123 Aljabar Linier dan Geometri
Semester I Tahun 2020/2021

oleh :

Christian Tobing Alexandro (13519109)
Raihan Astrada Fathurrahman (13519113)
Muhammad Rifat Abiwardani (13519205)

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2020

Bab I

Deskripsi Masalah

Sistem persamaan linier (SPL) Ax = b dengan n peubah (variabel) dan m persamaan adalah berbentuk a11 x1 + a12 x2 + ... + a1n xn = b1 a21 x1 + a22 x2 + ... + a2n xn = b2 : : : : am1 x1 + am2 x2 + ... + amn xn = bm yang dalam hal ini xi adalah peubah, aij dan bi 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).
Sebuah matriks M berukuran n × n

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

determinannya adalah

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

Determinan matriks M berukuran n × n dapat dihitung dengan beberapa cara: reduksi baris dan ekspansi kofaktor.
SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

II. Interpolasi Polinom
Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x0, y0), (x1, y1), ..., (xn, yn). Tentukan polinom pn(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga yi = pn(xi) untuk i = 0, 1, 2, …, n. Setelah polinom interpolasi pn(x) ditemukan, pn(x) dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x0, xn]. Polinom interpolasi derajat n yang menginterpolasi titik-titik (x0, y0),(x1, y1), ..., (xn, yn). adalah berbentuk pn(x) = a0 + a1x + a2x 2 + … + anx n . Jika hanya ada dua titik, (x0, y0) dan (x1, y1), maka polinom yang menginterpolasi kedua titik tersebut adalah

p1(x) = a0 + a1x yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x0, y0), (x1, y1), dan (x2, y2), maka polinom yang menginterpolasi ketiga titik tersebut adalah p2(x) = a0 + a1x + a2x 2 atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x0, y0), (x1, y1), (x2, y2), dan (x3, y3), polinom yang menginterpolasi keempat titik tersebut adalah p3(x) = a0 + a1x + a2x 2 + a3x 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 (xi, yi) ke dalam persamaan polinom pn(x) = a0 + a1x + a2x 2 + … + anx n untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a0, a1, a2, …, an, a0 + a1x0 + a2x0 2 + ... + an x0 n = y0 a0 + a1x1 + a2x1 2 + ... + an x1 n = y1 ... ... a0 + a1xn + a2xn 2 + ... + an xn n = yn Solusi sistem persamaan lanjar ini, yaitu nilai a0, a1, …, an, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk p2(x) = a0 + a1x + a2x 2 . Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah
a0 + 8.0a1 + 64.00a2 = 2.0794
a0 + 9.0a1 + 81.00a2 = 2.1972
a0 + 9.5a1 + 90.25a2 = 2.2513
Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan a0 = 0.6762, a1 = 0.2266, dan a2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah p2(x) = 0.6762 + 0.2266x - 0.0064x 2 . Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: p2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

III. Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

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

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

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

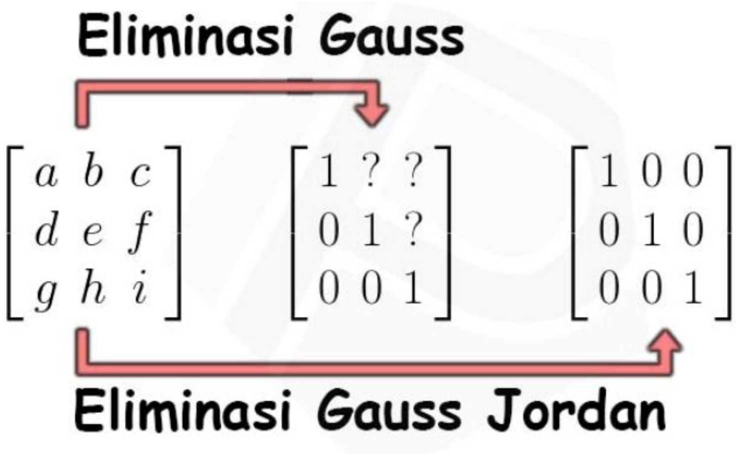
Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

Bab II

Teori Singkat

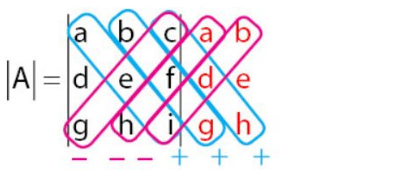
1. Metode Eliminasi Gauss-Jordan

   Dalam aljabar linear, eliminasi Gauss-Jordan merupakan penambahan beberapa prosedur dari metode eliminasi Gauss. Eliminasi Gauss-Jordan menggunakan representasi SPL dalam bentuk matriks augmented dan mengubahnya ke dalam bentuk matriks eselon tereduksi.



2. Determinan

   Determinan adalah nilai yang dapat dihitung dari unsur suatu matriks persegi. Misalkan determinan matriks A disimbolkan dengan det(A) / det A / |A|.



$$|A| = aei + bfg + cdh - ceg - afh - bdi$$

1. Tentukan determinan dari matriks di bawah ini!

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

Solusi:

Untuk menyelesaikan soal di atas, maka kita akan menggunakan aturan Sarrus.

|A| = aei + bfg + cdh – ceg – afh – bdi

|A| = (1x5x6) + (4x2x1) + (1x2x3) – (1x5x1) – (1x2x3) – (4x2x6)

|A| = 30 + 8 + 6 – 5 – 6 – 48

|A| = -15

3. Matriks balikan

   Jika A adalah suatu matriks kuadrat, dan jika kita dapat mencari matriks B sehingga AB = BA = I, maka A dikatakan dapat dibalik (invertible) dan B dinamakan invers dari A.

   Suatu matriks dapat dibalik jika dan hanya jika matriks tersebut adalah matriks persegi (matriks yang berukuran n x n) dan matriks tersebut non-singular (determinan != 0). Tidak semua matriks memiliki invers.

Contoh 1 :

Hitung invers matriks $A_{2\times2}$ berikut A = $\begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix}$.

Penyelesaian :

Jika kita punya matriks 2×2, misal A = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, maka invers matriks dapat dihitung menggunakan rumus

$$A^{-1} = B = \frac{1}{det \quad A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$= \frac{1}{3(2)-5(1)} \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix}$$

Cek, apakah AB = BA = I

$$AB = \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$$BA = \begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Karena AB = BA = I, maka berdasarkan **Definisi**, B adalah invers dari matriks A.

4. Minor Matriks
   Minor dari matriks A pada baris ke-i kolom ke-j, atau M_{ij}, adalah determinan dari bagian matriks A yang baris ke-i dan kolom ke-jnya dihilangkan. Sebagai contoh, untuk matriks A, minor dari baris 1 kolom 2 adalah:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$$

$$M_{12} = \begin{vmatrix} 2 & 3 \\ 1 & 8 \end{vmatrix}$$

5. Matriks kofaktor

   Kofaktor dari matriks A adalah matriks dengan nilai elemen pada baris ke-i kolom ke-j, atau C_{ij}, mengikuti rumus berikut:

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

   Sebagai contoh, untuk matriks A, kofaktor A adalah:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$$

$$C_A = \begin{bmatrix} M_{1,1} & -M_{1,2} & M_{1,3} \\ -M_{2,1} & M_{2,2} & -M_{2,3} \\ M_{3,1} & -M_{3,2} & M_{3,3} \end{bmatrix}$$

6. Matriks adjoin

   Matriks adjoin dari matriks A adalah transpose dari kofaktornya, yaitu elemen C_{ij} ditukar dengan elemen C_{ji}. Sebagai contoh, untuk matriks A, Adj_A adalah:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$$

$$Adj_A = \begin{bmatrix} M_{1,1} & -M_{1,2} & M_{1,3} \\ -M_{2,1} & M_{2,2} & -M_{2,3} \\ M_{3,1} & -M_{3,2} & M_{3,3} \end{bmatrix}$$

7. Sistem Persamaan Linier dengan Matriks Balikan

   Sebuah sistem persamaan linier dapat ditulis sebagai Ax = b, dimana A adalah matriks yang setiap barisnya menyimpan koefisien variabel x1...xn untuk persamaan 1...k dalam SPL, x adalah nilai variabel x1...xn, dan b adalah hasil setiap persamaan. Jika determinan A bukan 0, maka solusi variabel-variabel dalam SPL dapat ditulis sebagai $x = A^{-1}b$. Selebihnya, jika determinan A bukan 0, hubungan tersebut berlaku untuk matriks b yang berbentuk variabel bebas, karena matriks x bergantung pada matriks b.

   Namun jika det A = 0, A^{-1} tidak ada dan matriks x tidak dapat langsung dihitung. Akan tapi, jika SPL memiliki banyak solusi, matriks x dapat dihitung dengan mengganti beberapa baris dengan baris e (vektor basis) agar seolah

olah $0x_1 + 0x_2 + \ldots + x_k + \ldots + 0x_n = t$. Dalam kata lain, $x_k = t$ dengan t variabel bebas. Baris yang dijadikan variabel bebas dipilih sesuai karakteristik barisnya: jika penukaran baris tersebut dengan baris vektor basis mengakibatkan determinan matriks A berubah menjadi bukan 0, pilih baris itu. Jika tidak ada baris yang memenuhi kondisi itu, pilih baris yang elemennya 0 semua. Jika tidak ada baris yang memenuhi kondisi itu, pilih baris manapun. Baris yang terpilih lalu ditukar dengan vektor basis yang belum muncul dalam matriks A. Pada proses yg sama, ubah baris matriks b yang berkaitan menjadi variabel bebas. Ulang penukaran ini sehingga determinan matriks bukan 0. Sekarang, karena det A sudah bukan 0, A^{-1} ada dan $x = A^{-1}b$ bisa dihitung. Sebagai contoh:

$$x + y + 2z = 4$$
$$x + 2y + 3x = 6$$
$$2x + 2y + 4z = 8$$

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 2 & 4 \end{bmatrix}$$, dimana det(A) = 0.

$$b = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}$$

Mencoba menukar baris 1...

$$\det(A) = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 2 & 3 \\ 2 & 2 & 4 \end{vmatrix} = 2$$

Karena det(A) bukan 0 lagi, ubah b juga

$$b = \begin{vmatrix} t \\ 6 \\ 8 \end{vmatrix}$$

Mengali invers A dengan b

$$A^{-1}b = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & -1.5 \\ -1 & -1 & 1 \end{bmatrix} * \begin{bmatrix} t \\ 6 \\ 8 \end{bmatrix} = \begin{bmatrix} t \\ t \\ 2+t \end{bmatrix}$$

8. Kaidah Cramer

Aturan Cramer (*Cramer's Rule*) merupakan formula yang dipakai untuk menyelesaikan sistem persamaan linear dengan menggunakan determinan dari matriks yang terbentuk dari koefisien dan konstanta masing-masing persamaan di sistem tersebut.

Misalkan suatu sistem persamaan linear 3 × 3

$$\begin{cases} a_{11}x + a_{12}y + a_{13}z = c_1 \\ a_{21}x + a_{22}y + a_{23}z = c_2 \\ a_{31}x + a_{32}y + a_{33}z = c_3 \end{cases}$$

Solusi dari sistem tersebut adalah (*x*, *y*, *z*), dimana

$$x = \frac{D_x}{D} = \frac{\begin{vmatrix} c_1 & a_{12} & a_{13} \\ c_2 & a_{22} & a_{23} \\ c_3 & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}, y = \frac{D_y}{D} = \frac{\begin{vmatrix} a_{11} & c_1 & a_{13} \\ a_{21} & c_2 & a_{23} \\ a_{31} & c_3 & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}, z = \frac{D_z}{D} = \frac{\begin{vmatrix} a_{11} & a_{12} & c_1 \\ a_{21} & a_{22} & c_2 \\ a_{31} & a_{32} & c_3 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}}$$

dengan syarat *D* ≠ 0.

9. Interpolasi Polinom

Interpolasi polinomial digunakan untuk mencari titik-titik antara dari n buah titik P1 (x1, y1), P2 (x2, y2), P3 (x3, y3), …, PN (xN, yN) dengan menggunakan pendekatan fungsi polinomial pangkat n-1 yang berbentuk seperti persamaan berikut.

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

Masukan nilai dari setiap titik ke dalam persamaan polinomial di atas dan diperoleh persamaan simultan dengan n persamaan dan n variabel bebas:

$$y_1 = a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + \ldots + a_{n-1}x_1^{n-1}$$
$$y_2 = a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3 + \ldots + a_{n-1}x_2^{n-1}$$
$$y_3 = a_0 + a_1x_3 + a_2x_3^2 + a_3x_3^3 + \ldots + a_{n-1}x_3^{n-1}$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$y_n = a_0 + a_1x_n + a_2x_n^2 + a_3x_n^3 + \ldots + a_{n-1}x_n^{n-1}$$

Langkah pengerjaan polinom yang pertama adalah menentukan jumlah titik N yang diketahui. Setelah mendapat N kemudian kita memasukkan titik-titik yang diketahui Pi=(xi,yi) untuk i=1,2,3,…..,N. Setelah itu kita menyusun augment matrik dari titik-titik yang diketahui seperti gambar berikut.

$$J = \left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} & y_1 \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} & y_2 \\ 1 & x_3 & x_3^2 & \ldots & x_3^{n-1} & y_3 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1} & y_n \end{array}\right]$$

Augment matriks tersebut kemudian kita selesaikan dengan menggunakan metode eliminasi gauss/jordan. Setelah itu, kita susun koefisien fungsi polinomial berdasarkan penyelesaian SPL. Kemudian kita masukkan nilai x yang ingin kita ketahui/taksir. Kemudian hitung nilai y dari fungsi polinomial dengan rumus seperti gambar berikut.

$$y = \sum_{i=0}^{N-1} a_i x^i$$

10. Regresi Linier Berganda

Analisis Regresi adalah analisis yang mengukur pengaruh variabel bebas terhadap variabel terikat. Persamaan umum dari Regresi Linier Berganda adalah:

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

Dengan Y adalah variabel bebas, dan X adalah variabel-variabel bebas, a adalah konstanta (intersep) dan b adalah koefisien regresi pada masing-masing variabel bebas. Ada berbagai macam cara untuk menentukan regresi linier berganda, tetapi pada program ini penentuan regresi linier berganda menggunakan Normal Estimation Equation for Multiple Linear Regression yang memiliki bentuk sebagai berikut:

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

Sistem persamaan linier tersebut kemudian diselesaikan dengan menggunakan metode eliminasi Gauss sehingga didapatkan konstanta dan koefisien dari persamaan.

Bab III

Implementasi Program dalam Java

Class - class yang digunakan :

Class Matriks

Atribut

NBrsEff : integer { Jumlah baris efisien dari matriks }
NKolEFF : integer { Jumlah kolom efisien dari matriks }
Mat : double[][] { Array 2 dimensi berisi double dengan NMax = 100 }

Method

procedure MakeMatriks(input NB: int, input NK: int)
{ I.S. : Jumlah baris dan jumlah kolom terdefinisi }
{ F.S. : Terbentuk matriks dengan NBrsEff = NB, dan NKolEFF = NK }

procedure InputMatriks()
{ I.S. : Terdapat matriks yang akan diisi }
{ F.S. : Mengisi matriks dengan input dari *keyboard* }

procedure InputMatriksFile()
{ I.S. : Terdapat matriks yang akan diisi }
{ F.S. : Mengisi matriks dengan input dari File Eksternal }

procedure Deaugment(input/output M: Matriks, output b: Matriks)
{ I.S. : Matriks M terdefinisi berukuran MxN; M matriks SPL augmented }
{ F.S. : Matriks b adalah kolom terakhir matriks input M, dan matriks M adalah matriks input M dengan kolom terakhirnya dihapus }

function Augment(Matriks M, b) -> Matriks
{ I.S. Matriks M terdefinisi berukuran MxN dan matriks b terdefinisi berukuran Mx1 }
{ F.S. Program mengembalikan matriks berukuran Mx(N+1) yaitu matriks M di-augment dengan b }

function TulisMatriks(Matriks M) -> array of string
{ I.S. : Matriks M terdefinisi }
{ F.S. : Program mengembalikan array string setiap elemen matriks dengan ketentuan: elemen dalam satu baris terpisah dengan spasi, setiap baris terpisah dengan newline }

function KaliMatriks (Matriks M1, M2) -> Matriks
{ I.S. Matriks M1 terdefinisi berukuran IxJ dan Matriks M2 terdefinisi berukuran JxK }
{ F.S. Program mengembalikan hasil perkalian matriks dari M1 x M2 }

function Determinan(Matriks M) -> double
{ I.S. : Matriks terdefinisi berukuran NxN }
{ F.S. : Program mengembalikan determinan dari matriks dengan menggunakan metode ekspansi kofaktor }

function Determinan2(Matriks M) -> double
{ I.S. : Matriks terdefinisi berukuran NxN }
{ F.S. : Program mengembalikan determinan dari matriks dengan menggunakan metode reduksi baris }

function Inverse(Matriks M) -> Matriks
{ I.S. : Matriks M berukuran NxN dan determinan M != 0 }
{ F.S. : Program mengembalikan matriks balikan }

function Crammer(Matriks MHsl) -> String[]
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mencetak solusi dari SPL dengan kaidah Crammer }

function Gauss(Matriks Mkonstan) -> Matriks
{ I.S. : Matriks Mkonstan terdefinisi berukuran MxN; Mkonstan matriks SPL augmented }
{ F.S. : Program mengubah matriks menjadi eselon baris }

function Gauss1(Matriks M) -> Matriks
{ I.S. : Matriks M terdefinisi berukuran MxN; M matriks SPL augmented }
{ F.S. : Program mengubah matriks menjadi eselon baris }

function GaussJordan(Matriks M) -> Matriks
{ I.S. : Matriks M terdefinisi berukuran MxN; M matriks SPL augmented }
{ F.S. : Program mengubah matriks menjadi eselon baris tereduksi }

function TulisSPLUnik(Matriks M) -> String[]
{ I.S. : Matriks M terdefinisi berukuran NxN }
{ F.S. : Program mengembalikan array string hasil solusi SPL }

function TulisSPLInvParametrik(Matriks A, M1, b1, int nvar) -> String[]
{ I.S. : Matriks M1 terdefinisi berukuran NxN dan memiliki solusi SPL parametrik; Matriks A terdefinisi berukuran NxN; Matriks b1 terdefinisi berukuran Nx1; int nvar adalah banyak variabel dalam SPL }

{ F.S. : Program mengembalikan array string hasil solusi SPL dari matriks dengan metode invers }

function TukerKolom(Matriks k, int j) -> Matriks
{ I.S. : Parameter input terdefinisi }
{ F.S. : Program menukar kolom ke-j matriks dengan matriks kolom k }

function TukerBaris(Matriks b, int i) -> Matriks
{ I.S. : Parameter input terdefinisi }
{ F.S. : Program menukar baris ke-i matriks dengan matriks baris b }

function AdaBasis(int Loc) -> boolean
{ I.S. : Parameter input & matriks terdefinisi }
{ F.S. : Program mengembalikan true jika terdapat baris basis dengan elemen 1 di kolom ke-Loc }

function AmbilBaris(int i) -> Matriks
{ I.S. : Parameter input & matriks terdefinisi }
{ F.S. : Program menyalin baris ke-i dalam matriks }

function AmbilKolom(int i) -> Matriks
{ I.S. : Parameter input & matriks terdefinisi }
{ F.S. : Program menyalin kolom ke-j dalam matriks }

function Square() -> Matriks
{ I.S. : Matriks terdefinisi }
{ F.S. : Program memperbesar matriks dengan baris/kolom berisi 0 }

function IsEmpty() -> boolean
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mengembalikan true jika seluruh elemen matriks berisi 0 }

function AdaBarisKosong() -> boolean
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mengembalikan true jika terdapat baris pada matriks yang seluruh elemennya berisi 0 }

function AdaKolomKosong() -> boolean
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mengembalikan true jika terdapat kolom pada matriks yang seluruh elemennya berisi 0 }

function Copy() -> Matriks
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mengembalikan matriks salinan }

function JadiBasis(Matriks M) -> Matriks
{ I.S. Matriks M terdefinisi }
{ F.S. Program mengembalikan matriks M dengan satu baris dijadikan baris basis (yaitu baris yang elemennya 0 kecuali 1 yang bernilai 1, seperti [0,0,0,1,0]) }

function Identitas(Matriks M) -> Matriks
{ I.S. Matriks M terdefinisi berukuran NxN }
{ F.S. Program mengembalikan matriks identitas berukuran NxN, yaitu matriks yang tiap elemennya 0 kecuali pada M_{ij} dimana i = j, elemennya bernilai 1 }

function ExtSolusi(Matriks M, b) -> Matriks
{ I.S. Matriks M terdefinisi berukuran NxN, matriks b terdefinisi berukuran Mx1 dimana M <= N }
{ F.S. Matriks b berukuran Nx1, dengan baris yang lebih besar dari M diisi dengan 0 }

function VarMask(Matriks b, M) -> Matriks
{ I.S. Matriks b terdefinisi berukuran Nx1 dan matriks M terdefinisi berukuran NxN }
{ F.S. mengembalikan matriks bvar berukuran Nx1 dengan ketentuan: jika M[i][j] bernilai 0 untuk i dari 0 hingga N-1, isi bvar[j][0] dengan k dimana k adalah satu ditambah banyaknya elemen sebelum bvar[j][0] yang nilainya bukan 0; jika kolom ke-j matriks M tidak kosong, isi bvar[j][0] dengan 0 }

function UnMask(Matriks b, bvar) -> Matriks
{ I.S. Matriks b dan bvar terdefinisi berukuran Nx1 }
{ F.S. mengembalikan matriks b1 berukuran Nx1 dengan ketentuan: jika bvar[i][0] adalah 0, isi b1[i][0] dengan b[i][0]; jika bvar[i][0] bukan 0, isi b1[i][0] dengan 0 }

function ConjMatriks(Matriks M, M1) -> Matriks
{ I.S. Matriks M dan M1 terdefinisi berukuran NxN }
{ F.S. mengembalikan matriks M2 dengan ketentuan: jika baris ke-i matriks M beda dengan baris ke-i matriks M1, isi baris ke-i matriks M2 dengan baris kosong; jika baris ke-i matriks M sama dengan baris ke-i matriks M1, isi baris ke-i matriks M2 dengan baris ke-i matriks M }

function SPLInverse(Matriks M, b) -> array of string
{ I.S. Matriks M terdefinisi berukuran MxN dan matriks b terdefinisi berukuran Mx1 }
{ F.S. mengembalikan array string hasil pencarian solusi SPL dengan metode inverse }

function IsUnik(Matriks M) -> boolean
{ I.S. Matriks M terdefinisi berukuran Nx(N+1); M matriks SPL augmented }
{ F.S. mengembalikan true jika determinan matriks yang menempati ukuran NxN matriks M bukan 0, dan false jika 0 }

function NoSolution(Matriks M) -> boolean
{ I.S. Matriks M terdefinisi berukuran MxN; M matriks SPL augmented }
{ F.S. mengembalikan true jika baris terakhir matriks M berisi 0 semua kecuali elemen paling kanan (elemen di M[NB-1][NK-1], dan false jika tidak }

function TulisSPL(Matriks M) -> array of string
{ I.S. Matriks M terdefinisi berukuran MxN; M matriks SPL augmented }
{ F.S. mengembalikan array string persamaan solusi SPL }

Class Interpolasi

Atribut
NBrs : integer { Jumlah baris efisien }
NKol : integer { Jumlah kolom efisien }
Mat : Matriks { Persamaan Interpolasi Polinom dalam bentuk Matriks Augmented }

Method
function MakePolinom(int N)
{ I.S. : Parameter fungsi terdefinisi }
{ F.S. : Program membuat interpolasi polinom dengan bentuk Matriks Augmented }

procedure InputPolinom()
{ I.S. : Atribut Matriks siap diisi }
{ F.S. : Polinom berbentuk Augmented matriks terisi dengan input dari *keyboard* }

procedure toPolinom()
{ I.S. : Matriks input terdefinisi }
{ F.S. : Matriks input dirubah menjadi bentuk polinom }

function Fungsi(double X) -> double
{ I.S. : Augmented matriks dari polinom & nilai yang akan ditaksir terdefinisi }
{ F.S. : Program mengembalikan hasil taksiran dari X }

Class Regresi
Atribut
X : Matriks { Seluruh nilai X yang dibentuk menjadi matriks }
Y : Matriks { Seluruh nilai Y yang dibentuk menjadi matriks }
Augmented : Matriks { Matriks Augmented yang menggabungkan Matriks X dan Matriks Y }
TabTaksiran : array of double { Array yang menyimpan input Xo sampai Xn untuk menghitung nilai taksiran }

Method
function MakeRegresi(Matriks X, Matriks Y)
{ I.S. : Matriks X & Matriks Y terdefinisi }
{ F.S. : Program meng-*assign* atribut X & Y dengan Matriks X & Matriks Y }

function SigmaX(int j) -> double
{ I.S. : Matriks & parameter fungsi terdefinisi }
{ F.S. : Program mengembalikan hasil penjumlahan baris untuk setiap elemen baris pada kolom ke-j (Sigma Xki) }

function SigmaY() -> double
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mengembalikan hasil penjumlahan baris untuk setiap baris dari Matriks Y (Sigma Yi) }

function SigmaXX(int i, int j) -> double
{ I.S. : Matriks & parameter fungsi terdefinisi }
{ F.S. : Program mengembalikan hasil penjumlahan dari setiap baris Matriks X pada kolom i dikali Matriks X pada kolom j (Sigma Xbi * Xki) }

function SigmaXY(int i) -> double
{ I.S. : Matriks & parameter fungsi terdefinisi }
{ F.S. : Program mengembalikan hasil penjumlahan dari setiap baris Matriks X pada kolom i dikali Matriks Y pada baris i (Sigma Xki * Yi) }

procedure NormEq()
{ I.S. : Matriks terdefinisi }
{ F.S. : Program mengembalikan *Normal Estimation Equation for Multiple Linear Regression* dan meng-*assign* pada atribut Augmented }

procedure InputTaksiran()
{ I.S. : TabTaksiran kosong }
{ F.S. : Program menyimpan input Xo sampai Xn pada TabTaksiran }

function Fungsi() -> double
{ I.S. : Augmented matriks berbentuk eselon baris tereduksi, SPL memiliki solusi unik, dan TabTaksiran terdefinisi }
{ F.S. : Program mengembalikan estimasi nilai dari input pada TabTaksiran }

Garis Besar Program Driver

Program akan membuat suatu variabel bertipe boolean yang diberi nama **exit** dan diassign dengan nilai **<u>false</u>**. Kemudian program akan masuk ke **while-loop** yang memiliki EOP ketika variabel exit bernilai **<u>true</u>**. Dalam **while-loop** program akan menampilkan **2 metode input data**, dengan ***keyboard*** dan dengan **file eksternal**, dan akan menerima input angka untuk **memilih metode input data**. Input angka akan **disimpan** dalam variabel bernama **PilihanMet**. Kemudian, program akan menampilkan **Menu Awal** yang terdiri atas,

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

Jika pengguna memasukkan **input angka ‘1’** maka program akan menampilkan **pilihan metode SPL** yang terdiri dari Eliminasi Gauss, Eliminasi Gauss Jordan, Matriks Balikan, dan Kaidah Cramer. Program kemudian meminta **input untuk pilihan metode** kemudian menjalankan metode yang dipilih dan juga dengan metode input data yang dipilih di awal. Jika pengguna memasukkan **input angka ‘2’** maka yang berbeda program akan menampilkan **pilihan metode determinan**, yaitu ekspansi kofaktor dan reduksi baris, kemudian seperti biasa menjalankan metode yang dipilih dan juga dengan metode input data yang dipilih di awal. Untuk **input angka ‘3’, ‘4’, ‘5’ tidak terdapat tampilan SubMenu** dan program akan menjalankan prosedur yang sesuai input pengguna.

Jika pengguna memasukkan **input angka ‘6’** maka program akan menerima **konfirmasi dari pengguna** apakah memang ingin keluar dari program. Jika **‘Y’** maka program akan mengubah variabel **exit** menjadi **<u>true</u>** dan program akan **keluar dari while-loop**. Jika **‘N’** maka program akan **kembali ke Menu Awal**. **(*Setiap input angka dari pengguna akan divalidasi terlebih dahulu)**

Bab IV
Eksperimen

I. Sistem Persamaan Linear (SPL)

		Output				
No.		Metode Gauss (1)	Metode Gauss-Jordan (2)	Metode Invers (3)	Metode Kaidah Cramer (4)	Analisis
1.	a.	1.0X1 + 0.4X2 -0.8X3 + 0.4X4 = 1.2 + 1.0X2 -1.2857142857142858X3 -1.380952380952381X4 = -1.0476190476190477 + 1.0X3 -0.9999999999999998X4 = 0.7499999999999996 + 1.0X4 = -1.2867427506772848E15	TIDAK ADA SOLUSI	SPL tidak memiliki solusi	SPL tidak memiliki solusi	Metode 1 kurang tepat, metode 3 & 4 lebih pelan daripada metode 2.
	b.	SOLUSI BANYAK 1.0X1 -0.5X2 + 0.5X4 -0.5X5 = 2.5 + 1.0X2 -2.3333333333333335X4 + 0.3333333333333333X5 = 2.3333333333333335 -1.6666666666666665X4 + 1.6666666666666667X5 = 1.6666666666666665 + 1.0X4 -1.0X5 = -1.0	SOLUSI BANYAK 1.0X1 -1.0X5 = 3.0 + 1.0X2 -2.0X5 = 0.0 + 1.0X4 -1.0X5 = -1.0 = 0.0	Solusi dari X1 adalah = 3.0 + k1 Solusi dari X2 adalah = 2.0k1 Solusi dari X3 adalah = k2 Solusi dari X4 adalah = -1.0 + k1 Solusi dari X5 adalah = k1	–	Program berhasil; tidak terdefinisi untuk metode 4, sesuai spesifikasi tugas
	c.	+ 1.0X2 + 1.0X5 = 2.0 + 1.0X2 +	+ 1.0X2 + 1.0X5 = 2.0 + 1.0X4 +	Solusi dari X1 adalah = k3 Solusi dari X2	–	Program berhasil; tidak terdefinisi untuk

		1.0X6 = 1.0 + 1.0X4 + 1.0X5 = -1.0	1.0X5 = -1.0 + 1.0X2 + 1.0X6 = 1.0	adalah = 1.0 - k1 Solusi dari X3 adalah = k2 Solusi dari X4 adalah = -2.0 - k1 Solusi dari X5 adalah = 1.0 + k1 Solusi dari X6 adalah = k1		metode 4, sesuai spesifikasi tugas
	d.	1.0X1 + 0.5X2 + 0.33333333X3 + 0.25X4 + 0.2X5 + 0.16666667X6 = 1.0 + 1.0X2 + 1.06666667199 99998X3 + 1.00000002999 99992X4 + 0.91428566971 42865X5 + 0.83333331000 00005X6 = -3.9999998800 000016 + 1.0X3 + 1.66666569333 39987X4 + 2.05714251977 1598X5 + 2.27272629090 9701X6 = 11.9999872320 08493 + 1.0X4 + 2.05716897496 584X5 + 2.92210008339 8741X6 = -108.00162606 891433 + 1.0X5 + 2.55679130668 49317X6 = 472.352353037 00574 + 1.0X6 = -2907.4290902 622597	Solusi dari X1 adalah = 36.665857040 674645 Solusi dari X2 adalah = -648.4982119 651226 Solusi dari X3 adalah = 3483.0393118 560332 Solusi dari X4 adalah = -7876.266695 864389 Solusi dari X5 adalah = 7906.0417768 41276 Solusi dari X6 adalah = -2907.429090 6643255	Solusi dari X1 adalah = 36.6659118697 7761 Solusi dari X2 adalah = -648.499352000 154 Solusi dari X3 adalah = 3483.04549221 00666 Solusi dari X4 adalah = -7876.28054680 1672 Solusi dari X5 adalah = 7906.05566555 2771 Solusi dari X6 adalah = -2907.43425028 9559	Solusi dari X1 adalah = 36.6658473569 01985 Solusi dari X2 adalah = -648.498210979 6282 Solusi dari X3 adalah = 3483.03936386 71676 Solusi dari X4 adalah = -7876.26668865 7107 Solusi dari X5 adalah = 7906.04175501 953 Solusi dari X6 adalah = -2907.42913472 21388	Program berhasil namun metode 3 jauh lebih pelan daripada metode 1, 2 & 4
2 .	a.	SOLUSI BANYAK 1.0X1 -1.0X4 =	SOLUSI BANYAK 1.0X1 -1.0X4	Solusi dari X1 adalah = -1.0 + k1	SPL memiliki banyak solusi	Program berhasil

		-1.0 + 1.0X2 -2.0X3 = 0.0	= -1.0 + 1.0X2 -2.0X3 = 0.0	Solusi dari X2 adalah = 2.0k2 Solusi dari X3 adalah = k2 Solusi dari X4 adalah = k1		
	b.	SOLUSI BANYAK 1.0X1 + 4.0X3 = 4.0 + 1.0X2 + 4.0X4 = 6.0 + 1.0X3 = 1.0 + 1.0X4 = 1.0	SOLUSI BANYAK 1.0X1 = -0.0 + 1.0X2 = 2.0 + 1.0X3 = 1.0 + 1.0X4 = 1.0	Solusi dari X1 adalah = 0.0 Solusi dari X2 adalah = 2.0 Solusi dari X3 adalah = 1.0 Solusi dari X4 adalah = 0.99999999999 99999	–	Program berhasil; tidak terdefinisi untuk metode 4, sesuai spesifikasi tugas
3 .	a.	TIDAK ADA SOLUSI	Solusi dari X1 adalah = -0.224324324 32432425 Solusi dari X2 adalah = 0.1824324324 324323 Solusi dari X3 adalah = 0.7094594594 594595 Solusi dari X4 adalah = -0.258108108 10810825	Solusi dari X1 adalah = -0.22432432432 432434 Solusi dari X2 adalah = 0.18243243243 243243 Solusi dari X3 adalah = 0.70945945945 94594 Solusi dari X4 adalah = -0.25810810810 810814	Solusi dari X1 adalah = -0.22432432432 432434 Solusi dari X2 adalah = 0.18243243243 243243 Solusi dari X3 adalah = 0.70945945945 94594 Solusi dari X4 adalah = -0.25810810810 81081	Metode 1 kurang tepat, metode 2-4 berhasil
	b.					Hasil belum didapatkan karena saat percobaan prosesnya lumayan lama

II. Interpolasi

Eksperimen II.1. berikut dilakukan terhadap test case 5 pada spesifikasi tugas:

	Output				
Taksiran	Metode Gauss (1)	Metode Gauss-Jordan (2)	Metode Invers (3)	Metode Kaidah Cramer (4)	Analisis
0.2	–	Nilai taksiran dari X=0.2 adalah: 0.0329609375 00000086	Nilai taksiran dari X=0.2 adalah: 0.0329609375 000589	Nilai taksiran dari X=0.2 adalah: 0.0329609375 00006095	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan

					metode 2; belum diimplementasi metode 1
0.55	–	Nilai taksiran dari X=0.55 adalah: 0.17111865234375	Nilai taksiran dari X=0.55 adalah: 0.17111865234352644	Nilai taksiran dari X=0.55 adalah: 0.171118652343611	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementasi metode 1
0.85	–	Nilai taksiran dari X=0.85 adalah: 0.3372358398437499	Nilai taksiran dari X=0.85 adalah: 0.3372358398439909	Nilai taksiran dari X=0.85 adalah: 0.3372358398419484	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementasi metode 1
1.28	–	Nilai taksiran dari X=1.28 adalah: 0.6775418375	Nilai taksiran dari X=1.28 adalah: 0.6775418374952457	Nilai taksiran dari X=1.28 adalah: 0.6775418374766189	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementasi metode 1

Tabel 4.1. Ekspe

Eksperimen II.2. berikut dilakukan terhadap test case 6 pada spesifikasi tugas:

	Output				
Taksiran	Metode Gauss (1)	Metode Gauss-Jordan (2)	Metode Invers (3)	Metode Kaidah Cramer (4)	Analisis
5.806	–	Nilai taksiran dari X=5.806 adalah: 22794.691448509693	Nilai taksiran dari X=5.806 adalah: 22794.6914484375	Nilai taksiran dari X=5.806 adalah: 22794.691448443237	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementas

					i metode 1
8.968	–	Nilai taksiran dari X=8.968 adalah: 175769.76058197021	Nilai taksiran dari X=8.968 adalah: 175769.7605819687	Nilai taksiran dari X=8.968 adalah: 175769.76058197012	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementasi metode 1
9.5	–	Nilai taksiran dari X=9.5 adalah: 68216.42839050293	Nilai taksiran dari X=9.5 adalah: 68216.4283904875	Nilai taksiran dari X=9.5 adalah: 68216.42839049702	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementasi metode 1
02/11/20 (11,067)	–	Nilai taksiran dari X=11.067 adalah: -2.20051175410614E7	Nilai taksiran dari X=11.067 adalah: -2.20051175410614E7	Nilai taksiran dari X=11.067 adalah: -2.20051175410614E7	Metode 2-4 berhasil, namun metode 3 & 4 lebih pelan dibandingkan metode 2; belum diimplementasi metode 1

III. Regresi Linier Berganda

Eksperimen berikut dilakukan terhadap test case 8 pada spesifikasi tugas:

	Output				
Nilai yang ditaksir	Metode Gauss (1)	Metode Gauss-Jordan (2)	Metode Invers (3)	Metode Kaidah Cramer (4)	Analisis
Humidity = 50 Temp. = 76 Pressure = 29.3	–	Nilai taksiran dari input adalah: 0.9384342262216596	Nilai taksiran dari input adalah: 0.938434228347357	Nilai taksiran dari input adalah: 0.9384342262058163	Metode 2-4 berhasil; belum diimplementasi metode 1

Bab V
Kesimpulan, Saran, dan Refleksi

Kesimpulan

Dari Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Semester I Tahun 2020/2021 ini, dapat disimpulkan bahwa kita dapat menyelesaikan suatu Sistem Persamaan Linear (SPL) dengan menggunakan berbagai metode seperti metode Gauss, Gauss-Jordan, Matriks Balikan, dan Kaidah Cramer. Kemudian, solusi yang didapatkan dari suatu SPL dapat berupa solusi unik, solusi parametrik, dan tidak memiliki suatu solusi. Selain itu, perhitungan determinan dan balikan (*inverse*) dari suatu matriks juga dapat dihitung dengan berbagai metode seperti menggunakan ekspansi kofaktor dan reduksi baris. Yang terakhir, kita dapat menerapkan suatu SPL untuk menaksir suatu nilai pada permasalah Interpolasi Polinom dan juga Regresi Linear Berganda. Meskipun demikian, dari algoritma yang digunakan pada program, metode Gauss dan Gauss-Jordan lebih cepat dibandingkan metode invers dan kaidah Cramer sehingga tidak efektif untuk menggunakan metode invers dan kaidah Cramer untuk matriks berukuran besar. Maka dapat disimpulkan bahwa algoritma metode Gauss dan Gauss-Jordan adalah yang paling efisien.

Saran

Pada Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Semester I Tahun 2020/2021, materi yang digunakan adalah mengenai Sistem Persamaan Linier, Determinan, dan Aplikasinya. Mayoritas materi tersebut akan menggunakan suatu matriks sehingga akan lebih mudah untuk membuat suatu tipe data bentukan atau class Matriks dengan atribut yang berisi panjang baris efisien, panjang kolom efisien, dan nilai matriks yang disimpan pada array dua dimensi. Selain itu, karena pembuatan program ini menggunakan bahasa pemrograman Java, maka penting bagi kita untuk memahami tentang class-based dan object-oriented programming (OOP) karena kedua hal tersebut akan digunakan ketika kita membuat suatu kode program yang menggunakan bahasa Java.

Refleksi

Refleksi yang penulis dapatkan dalam pengerjaan Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Semester I Tahun 2020/2021 adalah, dalam menggunakan platform *github*, sebaiknya kita lebih teliti dalam melihat apakah kode yang berada dalam laptop/komputer kita sudah merupakan update-an terbaru di repository *github*. Hal tersebut harus kita lakukan agar mengantisipasi terjadinya stuck pada saat kita ingin mendorong file dari laptop/komputer kita ke repository *github*.

Daftar Referensi

Anton, Howard dan Chris Rorres. 2013. Elementary Linear Algebra: Applications Version, 11th Edition. Kanada: Wiley.

http://mezeylab.cb.bscb.cornell.edu/labmembers/documents/supplement%205%20-%20multiple%20regression.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/MetNum/2010-2011/Interpolasi%20Polinom.pdf

https://mathcyber1997.com/materi-soal-dan-pembahasan-aturan-cramer/#:~:text=Dalam%20bidang%20aljabar%20linear%2C%20Aturan,masing%20persamaan%20di%20sistem%20tersebut

https://www.dosenpendidikan.co.id/rumus-interpolasi/

https://www.konsultanstatistik.com/2009/03/regresi-linear.html

https://www.kelaspintar.id/blog/tips-pintar/cara-mencari-determinan-matriks-yang-mudah-5484/

https://yos3prens.wordpress.com/2014/12/16/menyelesaikan-sistem-persamaan-linear-menggunakan-determinan-dan-aturan-cramer/3/