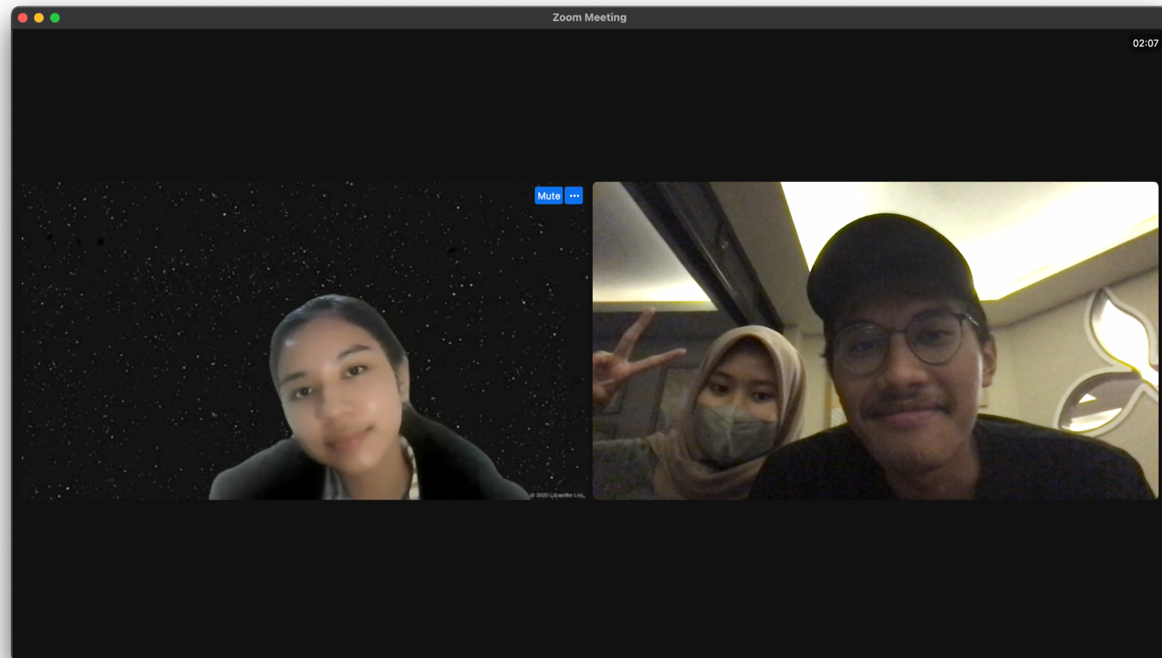
LAPORAN TUGAS BESAR I
“Sistem Persamaan Linier, Determinan, dan Aplikasinya”

Mata Kuliah Aljabar Linier dan Geometri (IF2123)

Dosen Pengampu : Dr. Ir. Rinaldi Munir, M.T.



DISUSUN OLEH :
Kelompok Moga Survive

Anggota :
1. Irsyad Nurwidianto Basuki (13521072)
2. Aulia Mey Diva Annandya (13521103)
3. Maggie ZR Simangunsong (13521117)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
SEMESTER I TAHUN 2022-2023

Daftar Isi

BAB I
Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. 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 (tidak berhingga), atau hanya satu (unik/tunggal).

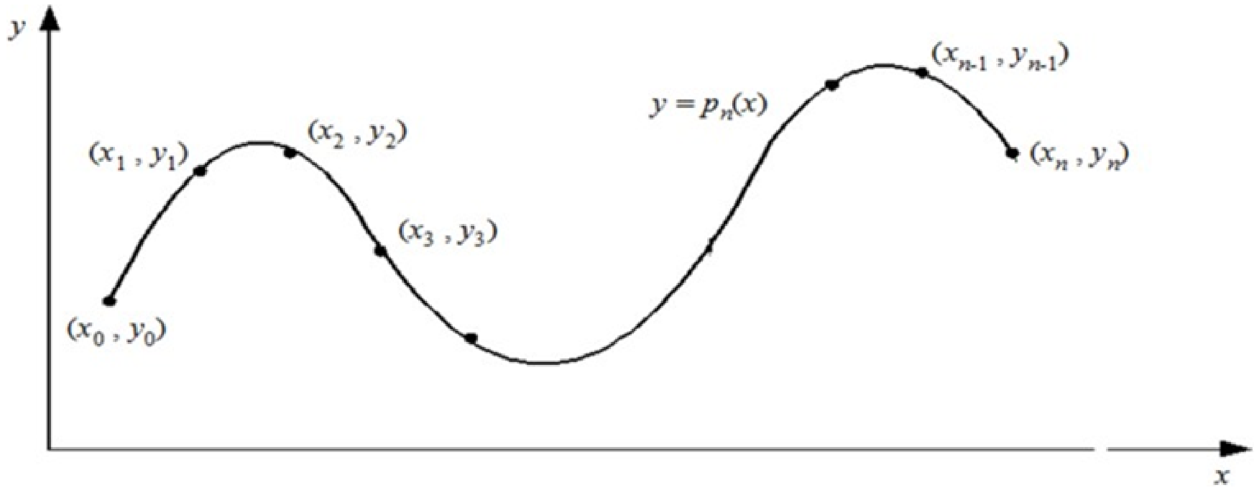
Di dalam Tugas Besar 1 ini, anda diminta membuat satu atau lebih *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, gunakan *library* tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi. Penjelasan tentang interpolasi dan regresi adalah seperti di bawah ini.

Beberapa tulisan cara membuat library di Java:

1. https://www.programcreek.com/2011/07/build-a-java-library-for-yourself/
2. https://developer.ibm.com/tutorials/j-javalibrary/
3. https://stackoverflow.com/questions/3612567/how-to-create-my-own-java-libraryapi

I. 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 $p_n(x)$ 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 $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x)$
$= a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0, y_0), (x_1, y_1), dan (x_2, y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0, y_0), (x_1, y_1), (x_2, y_2), dan (x_3, y_3), polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia $(n+1)$ buah titik data. Dengan menyulihkan (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk $i = 0, 1, 2, \ldots, n$, akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \ldots, a_n$,

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

Solusi sistem persamaan lanjar ini, yaitu nilai $a_0, a_1, \ldots, a_n$, 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 $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan *a0* = 0.6762, *a1* = 0.2266, dan *a2* = -0.0064. Polinom interpolasi yang melalui ketiga buah 2 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: $p2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

II. Bicubic Interpolation

Bicubic interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan dalam pembesaran citra yang merupakan pengembangan dari interpolasi linear dan cubic yang telah dipelajari pada kuliah metode numerik di aljabar geometri.

Diberikan sebuah matrix awal, misal M, kita akan mencari persamaan interpolasi *f(x,y)* dengan pemodelan sebagai berikut:

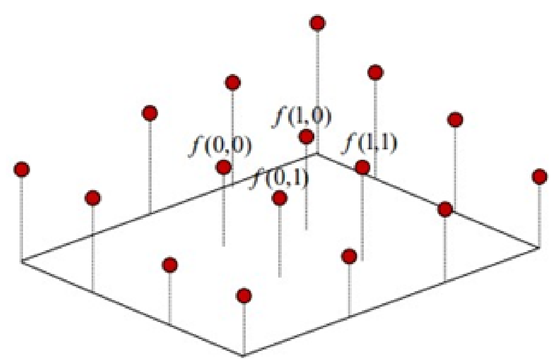
Normalization: $f(0,0), f(1,0)$

$f(0,1), f(1,1)$

Model: $$f(x,y) = \sum_{j=0}^{3} \sum_{i=0}^{3} a_{ij} x^i y^j$$

$$x = -1, 0, 1, 2$$

Solve: a_{ij}



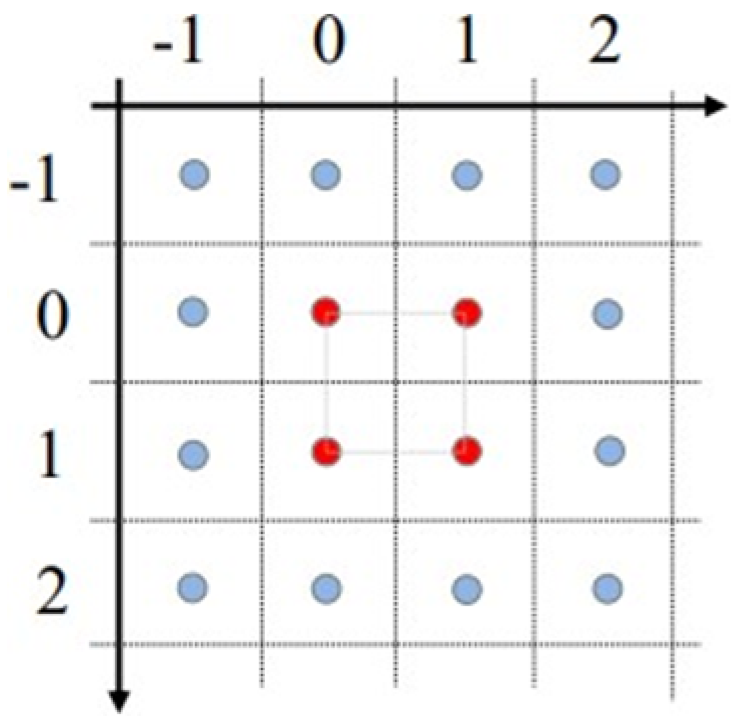
Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(x,y) akan menghasilkan sebuah matriks persamaan:

$$y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Elemen pada matrix X adalah koefisien a_{ij} yang diperoleh dari persamaan f(x,y) di atas. Sebagai contoh, elemen pada baris 4 kolom ke 10 adalah koefisien dari a_{12} dan diperoleh dari $2^1 * (-1)^2 = 2$, sesuai persamaan $x^i * y^j$.

Vektor *a* dapat dicari dari persamaan tersebut (menggunakan inverse), lalu vektor *a* digunakan sebagai nilai variabel dalam *f(x,y)*. Sehingga terbentuk fungsi interpolasi bicubic sesuai model. Tugas Anda adalah menentukan persamaan *f(x,y)* lalu melakukan interpolasi berdasarkan *f(a,b)* dari masukan matriks 4 x 4. Nilai masukan *a* dan *b* dalam rentang [0,1] (Referensi gambar di bawah, nilai untuk diinterpolasi dalam kotak merah).



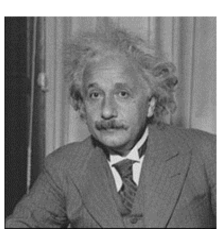
Untuk studi kasus ini, buatlah matriks X menggunakan rumus yang ada (tidak hardcode) serta carilah inverse matriks X dengan library kalian dalam penyelesaian masalah.

Referensi Bicubic Interpolation:

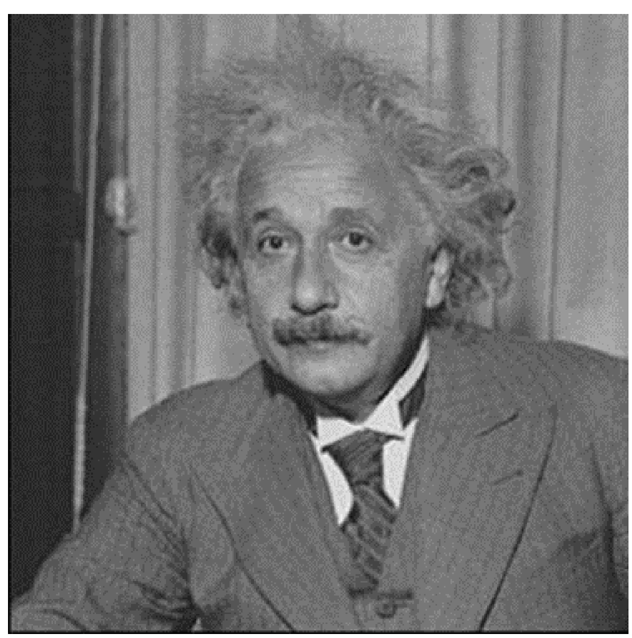
https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf

Bonus (Nilai 10): Gunakan interpolasi *bicubic* ini untuk melakukan perbesaran (*scaling*) pada citra (*image*) menjadi dua kali ukuran semula.

Contoh:



=====>



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:

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

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB II
Teori Singkat

I. Metode Eliminasi Gauss

Eliminasi Gauss adalah sebuah metode untuk menyederhanakan nilai-nilai di dalam matriks dengan melakukan operasi baris yang dapat digunakan sebagai salah satu metode penyelesaian sistem persamaan linear (SPL).. Metode ini menyatakan persamaan linear tersebut ke dalam bentuk matriks *augmented.* Kemudian menerapkan Operasi Baris Elementer (OBE) pada matriks *augmented* sampai terbentuk matriks eselon baris dan memecahkan persamaan yang berkoresponden pada matriks eselon baris dengan teknik penyulihan mundur (*backward substitution*).
SPL dalam bentuk matriks sebagai berikut.

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

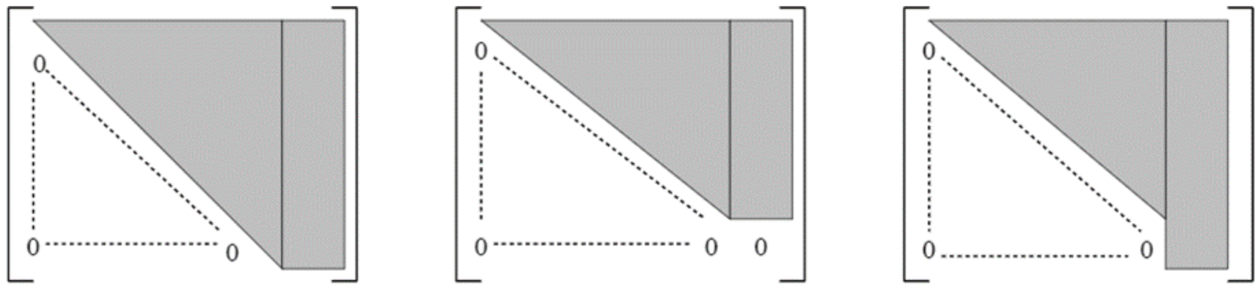
Setelah itu SPL dinyatakan secara ringkas dengan matriks *augmented* sebagai berikut.

$$[A \mid \mathbf{b}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right]$$

Terdapat tiga jenis operasi yang dapat dilakukan dalam metode ini:

1. Mengganti urutan dua baris
2. Mengalikan baris dengan angka yang bukan nol
3. Menambah suatu baris dengan baris yang lainnya

Metode eliminasi Gauss ini dilakukan jika terindikasi matriks *augmented* berakhir pada bentuk matriks eselon-baris. Untuk kemungkinan solusi dari SPL yang ada, dapat diindikasikan jika matriks *augmented* berakhir dengan pola sebagai berikut.



Solusi unik Solusi banyak Tidak ada solusi

Selanjutnya, solusi dari SPL didapatkan dengan terlebih dahulu menyelesaikan salah satu persamaan untuk satu variabel dan kemudian mensubstitusi ekspresi ini ke dalam persamaan yang tersisa, proses ini dilanjutkan sampai semua variabel asli telah dievaluasi.

II. Metode Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah pengembangan dari eliminasi Gauss yang hasilnya lebih sederhana dengan menghasilkan matriks eselon baris tereduksi.

Pada metode eliminasi Gauss-Jordan tidak diperlukan lagi substitusi secara mundur untuk memperoleh nilai-nilai variabel. Nilai variabel langsung diperoleh dari matriks augmented akhir.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix} \sim\text{OBE}\sim \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & * \\ 0 & 1 & 0 & \cdots & 0 & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & * \end{bmatrix}$$

Metode eliminasi Gauss-Jordan terdiri dari dua fase:

1. Fase maju (*forward phase*) atau fase eliminasi Gauss

   Fase ini akan menghasilkan nilai-nilai 0 di bawah 1 utama

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \sim \overset{\text{OBE}}{\cdots} \sim \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

2. Fase mundur (backward phase)

   Fase ini akan menghasilkan nilai-nilai 0 di atas 1 utama

$$\begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \overset{R1-(3/2)R2}{\sim} \begin{bmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \underset{R2-(1/2)R3}{\overset{R1+(5/4)R3}{\sim}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Matriks eselon baris tereduksi

Metode ini juga dapat digunakan untuk mencari invers dari sebuah matriks.

III. Determinan

Determinan suatu matriks hanya dapat dicari untuk matriks persegi, yaitu jika banyak baris sama dengan banyak kolom. Determinan suatu matriks didefinisikan sebagai selisih antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder. Determinan dari matriks A dapat dituliskan det(A) atau |A|.
Untuk matriks A berukuran 2x2 :

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

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

Untuk matriks A berukuran 3x3 :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{matrix}$$

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

Untuk matriks A berukuran nxn :
Determinan Matriks Segitiga

1. Matriks segitiga atas (upper triangular) : semua elemen di bawah diagonal utama adalah nol

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

2. Matriks segitiga bawah (lower triangular) : semua elemen di atas diagonal utama adalah nol

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

Secara umum, untuk matriks segitiga A berukuran nxn,
$\det(A) = a_{11}a_{22}a_{33}\ldots a_{nn}$

IV. Matriks Balikan

Matriks balikan adalah matriks yang jika dilakukan operasi perkalian dengan matriks asalnya akan menghasilkan matriks identitas. Suatu matriks memiliki balikan (invers) jika determinan dari matriks tersebut tidak sama dengan nol. Untuk sebuah matriks A, matriks balikannya ditulis sebagai A^{-1} di mana $A.A^{-1} = I$ (matriks identitas).
Untuk matriks A yang berukuran n x n, A^{-1} dapat dicari dengan cara berikut,

$$[A|I] \overset{\text{G-J}}{\sim} [I|A^{-1}]$$

Jika pada akhir operasi didapatkan matriks sebagai berikut,

$$\left(\begin{array}{ccc|ccc} 1 & 6 & 4 & 1 & 0 & 0 \\ 0 & 1 & 9/8 & 2/8 & -1/8 & 0 \\ \boxed{0 \quad 0 \quad 0} & & & -1 & 1 & 1 \end{array}\right)$$

Ada baris bernilai 0

maka dapat disimpulkan bahwa matriks *A* tidak memiliki balikan.

V. Matriks Kofaktor

Matriks Kofaktor adalah sebuah hasil operasi matriks persegi yang melibatkan determinan matriks minor-nya. Misal, A adalah matriks *n* x *n* dan *Cij* adalah kofaktor entri a_{ij}, maka matriks kofaktor dari A adalah

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

Di mana C_{ij} berkoresponden dengan minor entri M_{ij} (determinan minor), yang tanda positif/negatif-nya dibalik jika nilai $i + j$ adalah bilangan ganjil. Ekspansi kofaktor dapat menghasilkan determinan, baik secara kolom maupun baris. Selain itu, matriks kofaktor yang ditranspose akan menghasilkan matriks adjoin.

VI. Matriks Adjoin

Adjoin adalah transpose dari suatu matriks kofaktor. Adjoin matriks digunakan untuk mencari invers dari sebuah matriks dengan cara mengalikannya dengan 1/determinan.

VII. Kaidah Cramer

Jika A**x** = **b** adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variable) sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi unik yaitu

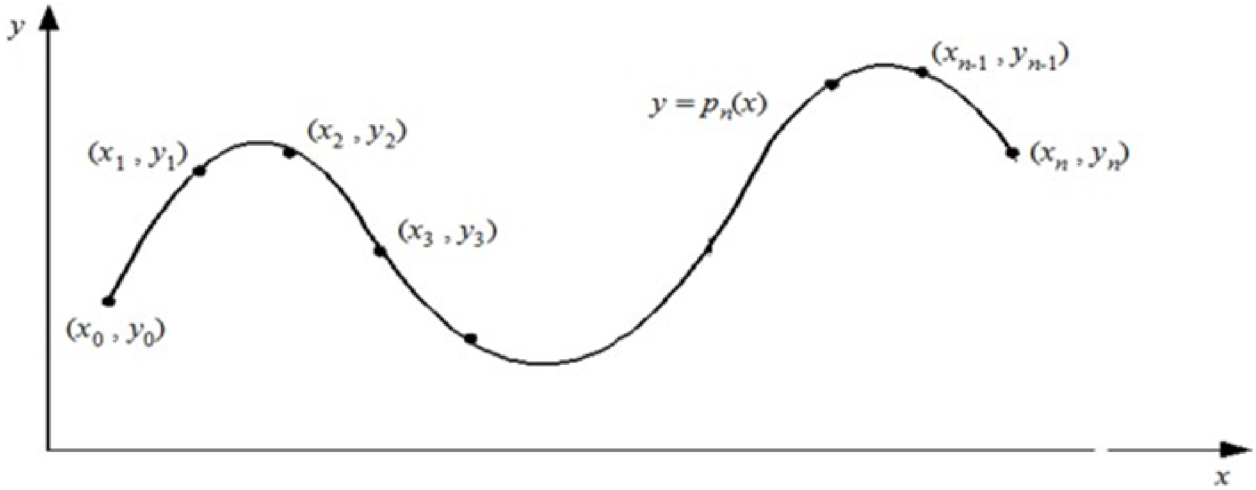
$$x_1 = \frac{\det(A_1)}{\det(A)}, \quad x_2 = \frac{\det(A_2)}{\det(A)}, \quad \ldots, \quad x_n = \frac{\det(A_n)}{\det(A)}$$

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

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

VIII. Interpolasi Polinom

Bentuk polinom interpolasi derajat n yang menginterpolasi (x_0, y_0), (x_1, y_1), … , (x_n, y_n) adalah $P_n(x) = a_0 + a_1x + a_2ax_2 + \cdots + a_nax_n$.



Polinom berderajat n dibuat dengan (n + 1) buah titik data. Setelah itu, (x_i, y_i) disulihkan ke dalam persamaan polinom sehingga didapatkan sistem persamaan lanjar. Kemudian untuk mendapatkan solusi SPL yang akan menjadi nilai untuk persamaan polinom interpolasi tersebut digunakan metode eliminasi gauss.

IX. Regresi Linier Berganda

Regresi Linear Berganda merupakan salah satu metode untuk memprediksi suatu nilai dengan bentuk rumus umum:

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

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

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

Untuk mendapatkan hasil dari SPL tersebut digunakan metode eliminasi gauss yang nantinya dapat digunakan untuk menaksir suatu nilai.

X. Bicubic Interpolation

Interpolasi Bikubik merupakan sebuah metode interpolasi pengembangan dari interpolasi linear dan kubik yang digunakan dalam pembesaran citra. Metode ini digunakan untuk memperbaiki kualitas citra hasil penskalaan. Metode interpolasi bikubik ini dapat membuat tepi-tepi citra hasil menjadi lebih halus.

Diberikan sebuah matrix awal, misal M, kita akan mencari persamaan interpolasi *f(x,y)* dengan pemodelan sebagai berikut:

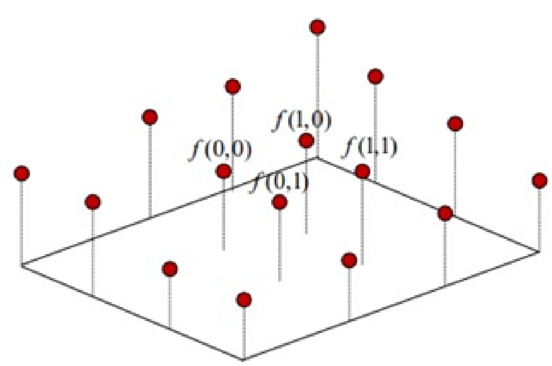
Normalization: $f(0,0), f(1,0)$

$f(0,1), f(1,1)$

Model: $$f(x,y) = \sum_{j=0}^{3} \sum_{i=0}^{3} a_{ij} x^i y^j$$

$$x = -1, 0, 1, 2$$

Solve: a_{ij}



Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(x,y) akan menghasilkan sebuah matriks persamaan:

$$y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Elemen pada matrix X adalah koefisien a_{ij} yang diperoleh dari persamaan f(x,y) di atas. Vektor *a* dapat dicari dari persamaan tersebut (menggunakan inverse), lalu vektor *a* digunakan sebagai nilai variabel dalam *f(x,y)*. Sehingga terbentuk fungsi interpolasi bicubic sesuai model.

BAB III
Implementasi Pustaka dan Program dalam Java

1. Matrix.java

Class ini berisi representasi matriks yang akan dibuat. Pembuatan matriks menggunakan array dua dimensi dengan isi elemen bertipe double.

a. Atribut

Atribut	Deskripsi
int row	Berisi data integer sebagai baris efektif dari matriks
int col	Berisi data integer sebagai kolom efektif dari matrik
double[][] m	Kontainer untuk menampung angka double dan representasi isi matriks.

b. Konstruktor

Konstruktor(Parameter)	Deskripsi
public Matrix(int i, int j)	Akan membuat suatu matriks dengan baris i dan kolom j dengan isi angka ‘0’
public double getELMT(int i, int j)	Fungsi yang mengambil nilai elemen pada baris i dan kolom j tertentu.

c. Metode

Metode(Parameter)	Deskripsi
public int getRow()	Mengembalikan banyak baris dari matriks
public int getColumn()	Mengembalikan banyak kolom dari matriks
public void readMatrix()	Mengambil input dari keyboard
public void setELMT(int i, int j, double k)	Mengubah atau menentukan nilai pada baris i dan kolom j dengan

	nilai k
public void displayMatrix(Matrix m)	Menampilkan matriks ke layar
public void copyMatrix(Matrix m)	Menerima parameter berupa matriks dan membuat duplikasi dari matriks tersebut.
public int countELMT()	Mengembalikan ukuran matriks, yaitu baris i x kolom j
public void MatrixIdentitas()	Mengembalikan matriks dengan tipe matriks identitas
public Matrix kaliMatrix(Matrix m1, Matrix m2)	Mengalikan dua matriks, yaitu matriks m1 dengan m2
public void divRow(int i, double k)	Membagi suatu baris pada matriks dengan k
public void swapRow(int i1, int i2)	Menukar 2 baris, i1 dengan i2
public void otherKRow(int i1, int i2, double k)	Mengurangi suatu baris i1 dengan k kelipatan baris i2
public static boolean isRowEmpty(Matrix m, int i)	Mengecek apakah baris i pada matriks m kosong atau tidak
public static Matrix swapEmptyRow(Matrix m)	Menukar baris kosong menjadi baris paling bawah di matriks
public static Matrix coeffMat(Matrix m)	Fungsi untuk mengembalikan matriks koefisien
public static Matrix constMat(Matrix m)	Fungsi untuk mengembalikan matriks konstanta

2. SPL.java

Class ini berisi metode - metode yang berhubungan dengan penyelesaian SPL

a. Metode

public static Matrix ubahEselon(Matrix m)	Fungsi yang membuat matriks menjadi matriks eselon dengan

	metode Eliminasi Gauss
public static Matrix ubahEselonReduksi(Matrix m)	Fungsi yang membuat matriks menjadi matriks eselon baris dengan metode Eliminasi Gauss-Jordan
public static Matrix Cramer(Matrix m)	Fungsi untuk melakukan metode Cramer
public static Matrix inverse(Matrix m)	Fungsi untuk melakukan metode
public static Matrix Solusi Gauss(Matrix m)	Fungsi untuk menghitung solusi gauss
public static Matrix SolusiInversCramer(Matrix m)	Fungsi untuk mengeluarkan hasil dari spl cramer dan inverse

3. Determinan.java

Class ini berisi metode yang berhubungan dengan Determinan.

a. Metode

public static double detRedBar(Matrix m)	Fungsi untuk mendapatkan determinan dengan metode Reduksi Baris
public static double detKofaktor(Matrix m)	Fungsi untuk mendapatkan determinan dengan metode Kofaktor

4. Invers.java

Class ini berisi metode yang berhubungan dengan Invers.

a. Metode

public static Matrix InversGaussJordan(Matrix m)	Fungsi untuk mendapatkan invers dengan metode Gauss-Jordan
public static Matrix getKofMatrix ()	Fungsi untuk mendapatkan invers dengan metode Kofaktor
public static Matrix getAdj()	Fungsi untuk mendapatkan adjoin

	dari suatu matriks
public static Matrix adjInv()	Fungsi untuk mendapatkan invers dengan metode Adjoin

5. RegresiInterpolasi.java

Class ini berisi metode untuk mengkalkulasi Interpolasi Polinom.

a. Metode

Metode	Deskripsi
public static void InterpolasiPolinom (Matrix m, double x)	Fungsi untuk menghitung interpolasi polinom dan mendapatkan nilai taksiran dari titik - titik data
public static void InterpolasiBikubik(Matrix m, double x, double y)	Fungsi untuk menghitung interpolasi bikubik dari suatu titik dalam range -1 .. 2
public static void RegresiLinierGanda(Matrix m, Matrix n)	Fungsi untuk menghitung Regresi Linier Berganda

6. Main.java

Class ini sebagai program utama yang menggerakkan seluruh program yang ada.

a. Metode

Metode	Deskripsi
public static void dispMenu()	Fungsi untuk menampilkan menu utama
public static void dispExit()	Fungsi untuk menampilkan menu keluar
public static void main(String[] args) throws Exception	Program utama untuk menggerakkan seluruh program

7. IOFile.java

Class ini berisi konstruktor dan metode - metode untuk masukan dan keluaran file.

a. Konstruktor

IOFile (int rows, int cols)	Sebagai konstruktor IOFile

b. Metode

public static int readBaris(String s)	Membaca baris dalam file
public static int readKol(String s)	Menghitung jumlah kolom dalam file
public static Matrix bacaM(String s)	Membaca matriks dalam file
public static void simpanMatrix(String namaFile, Matrix m)	Menyimpan matrix awal dalam file
public static void simpanMatrix(String namaFile, Matrix m)	Menyimpan matrix yang telah terubah dalam file

BAB IV
Eksperimen

1. Baca File

a.

`---------------MENU(^_^)---------------`
`1. Sistem Persamaan Linier`
`2. Determinan`
`3. Matriks balikan`
`4. Interpolasi Polinom`
`5. Interpolasi Bicubic`
`6. Regresi linier berganda`
`7. Keluar`
 	
`Input menu :`
	Tampilan menu

b.

`Input menu : 1`
`Pilih cara baca :`
`1. Baca file :`
`2. Baca input keyboard :`
	Memilih bentuk input

2. SPL berbentuk matriks *augmented*

a.

`1 1 1 0`
`2 3 1 1`
`3 1 2 1`
	Matriks untuk SPL Gauss

b.

Input menu : 1 ********Sistem Persamaan Linier******** 1. Metode eliminasi Gauss 2. Metode eliminasi Gauss-Jordan 3. Metode matriks balikan 4. Kaidah Cramer 99.Kembali ke Menu Utama Input menu : 1 Pilih cara baca : 1. Baca file : 2. Baca input keyboard : 1 Nama File : test.txt x1 = 0.0, x2 = 1.0, x3 = -1.0 ---------------(^_^)--------------- Keluar program? 1. Ya 99. Kembali ke menu utama Input :	**SPL eliminasi gauss**
1 1 2 4 2 -1 1 2 1 2 3 7 ********Sistem Persamaan Linier******** 1. Metode eliminasi Gauss 2. Metode eliminasi Gauss-Jordan 3. Metode matriks balikan 4. Kaidah Cramer 99.Kembali ke Menu Utama Input menu : 2 Pilih cara baca : 1. Baca file : 2. Baca input keyboard : 1 Nama File : test.txt Tidak ada solusi untuk SPL ini. ---------------(^_^)--------------- Keluar program? 1. Ya 99. Kembali ke menu utama Input :	**SPL Metode eliminasi gauss jordan**

Input menu : 1 ********Sistem Persamaan Linier******** 1. Metode eliminasi Gauss 2. Metode eliminasi Gauss-Jordan 3. Metode matriks balikan 4. Kaidah Cramer 99.Kembali ke Menu Utama Input menu : 3 Pilih cara baca : 1. Baca file : 2. Baca input keyboard : 1 Nama File : test.txt x1 = 20.0x2 = 13.0x3 = 29.0	**SPL metode matriks balikan**
3 2 1 2 4 -1 ********Sistem Persamaan Linier******** 1. Metode eliminasi Gauss 2. Metode eliminasi Gauss-Jordan 3. Metode matriks balikan 4. Kaidah Cramer 99.Kembali ke Menu Utama Input menu : 4 Pilih cara baca : 1. Baca file : 2. Baca input keyboard : 1 Nama File : test.txt x1 = 0.75x2 = -0.625	**SPL Metode Cramer**

3.

a.

<table>
<tr>
<td>
<pre>
Input menu : 2
***************Determinan**************
1. Metode Ekspansi Kofaktor
2. Metode Reduksi Baris
99.Kembali ke Menu Utama

Input sub menu : 1
Pilih cara baca :
1. Baca file :
2. Baca input keyboard :
1
Nama File :
test.txt
Determinan Kofaktor = -1356.0
</pre>
</td>
<td>Determinan metode ekspansi kofaktor</td>
</tr>
<tr>
<td>
<pre>
Input menu : 2
***************Determinan**************
1. Metode Ekspansi Kofaktor
2. Metode Reduksi Baris
99.Kembali ke Menu Utama

Input sub menu : 2
Pilih cara baca :
1. Baca file :
2. Baca input keyboard :
1
Nama File :
test.txt
Determinan Reduksi = -1356.0
</pre>
</td>
<td>Determinan metode Reduksi Baris</td>
</tr>
</table>

4.

<table>
<tr>
<td>
<pre>
7 0 2 0
-2 1 -4 1
9 5 -2 -7
4 5 -3 8
</pre>
<pre>
Input menu : 3
************Matriks Balikan************
1. Metode eliminasi Gauss-Jordan
2. Metode Adjoint
99.Kembali ke Menu Utama

Input menu : 1
Pilih cara baca :
1. Baca file :
2. Baca input keyboard :
1
Nama File :
test.txt

Hasil matriks balikan (gauss-jordan) :
0.19026548672566368 0.11061946902654864 -0.004424778761061946 -0.017699115044247784
-0.3355457227138641 -0.33849557522123863 0.12020648967551623 0.14749262536873153
-0.16592920353982293 -0.38716814159292023 0.01548672566371681 0.06194690265486724
0.052359882005899736 0.011061946902654893 -0.06710914454277286 0.06489675516224189
</pre>
</td>
<td>Matriks balikan metode gauss Jordan</td>
</tr>
</table>

<table>
<tr>
<td>
<pre>
Input menu : 3
************Matriks Balikan************
1. Metode eliminasi Gauss-Jordan
2. Metode Adjoint
99.Kembali ke Menu Utama

Input menu : 2
Pilih cara baca :
1. Baca file :
2. Baca input keyboard :
1
Nama File :
test.txt

Hasil matriks balikan (adjoint) :
0.005162241887905605 0.0 0.0014749262536873156 0.0
-0.0014749262536873156 7.374631268436578E-4 -0.0029498525073746312 7.374631268436578E-4
0.00663716814159292 0.003687315634218289 -0.0014749262536873156 -0.005162241887905605
0.0029498525073746312 0.003687315634218289 -0.0022123893805309734 0.0058997050147492625
</pre>
</td>
<td>Matriks balikan metode adjoin</td>
</tr>
</table>

5. **Studi Kasus Interpolasi**

a. Program menerima masukan nilai *x* yang akan dicari nilai fungsi *f*(*x*).

x	0.4	0.7	0.11	0.14	0.17	0.2	0.23
f(*x*)	0.043	0.005	0. 058	0.072	0.1	0.13	0.147

```
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.147 0.23
```

Tampilan program

```
Input menu : 4
************Interpolasi Polinom************
Pilih cara baca :
1. Baca file :
2. Baca input keyboard :
1
Nama File :
test.txt
Masukkan x :
9.2
Hasil taksir dengan persamaan polinom p6(X) = 505.6406 + (-15692.908X^1) + 192609.8032X^2 + (-1185257.3014X^3) + 3804315.9153X^4 + (-5927220.4873X^5) + 3445975.29
01X^6 adalah 1.7251792653523818E12
```

Dari pengujian di atas, hasil pengujian adalah sebagai berikut.

x = 9.2 f(x) = 1.72517926….

Hasil Taksiran dari Interpolasi Polinom adalah f(x) = 1.72517927

6. **Studi Kasus Interpolasi Bicubic**

Diberikan matriks input :

$$\begin{bmatrix} 153 & 59 & 210 & 96 \\ 125 & 161 & 72 & 81 \\ 98 & 101 & 42 & 12 \\ 21 & 51 & 0 & 16 \end{bmatrix}$$

Tentukan nilai :
f(0.5, 0.5) =

```
Input menu : 5
************Interpolasi Bicubic************
Pilih cara baca :
1. Baca file :
2. Baca input keyboard :
1
Nama File :
test.txt
Masukkan titik x :
0.5
Masukkan titik y :
0.5
Hasil dari interpolasi bikubik dengan titik f(0.5,0.5) = 97.7266
```

7. **Studi Kasus Regresi Linier Berganda**

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.

Input
Hasil

BAB V
Kesimpulan, Saran, dan Refleksi

5.1 Kesimpulan

Pada Tugas Besar 1 IF2123 Aljabar Linier dan Geometri ini, kami menggunakan bahasa pemrograman Java untuk membuat suatu program komputer untuk mengoptimalkan aplikasi matriks. Program ini dapat menyelesaikan berbagai permasalahan linier seperti mencari solusi Sistem Persamaan Linier (SPL) dengan menggunakan metode eliminasi Gauss, eliminasi Gauss-Jordan, matriks balikan, dan kaidah Cramer. Program ini juga dapat membuat matriks balikan (invers) dan menghitung determinan sebuah matriks dengan menggunakan metode reduksi baris dan ekspansi kofaktor. Program ini dapat melakukan regresi linier berganda menggunakan *Normal Estimation Equation* dan menghitung hasil fungsi menggunakan interpolasi polinom dan interpolasi bikubik. Dengan menggunakan Java, dibuat suatu konstruktor yang membentuk objek matriks beserta *library* yang berisi beberapa *class* dan *method* yang dapat digunakan untuk menyelesaikan studi-studi kasus.

5.2 Saran

Dalam pengerjaan tugas besar ini, kami menyadari segala kendala dan keterbatasan pada program yang kami buat. Kami menyadari bahwa program yang kami buat masih kurang maksimal dan masih terdapat banyak kekurangan. Tentunya dengan adanya kekurangan ini, untuk selanjutnya masih dapat dikembangkan untuk optimalisasi lebih lanjut.

5.3 Refleksi

Masalah pertama, dari awal penurunan tugas kurang koordinasi dalam pembuatan tugas ini sehingga terjadi keterlambatan yang mengakibatkan ketidakmaksimalan pembuatan tugas. Masalah kedua, kurangnya komunikasi antar anggota juga mengakibatkan terhambatnya pembuatan tugas ini. Masalah ketiga, bahasa Java mengharuskan kita untuk belajar terlebih dahulu yang lumayan memakan waktu.

Referensi

https://iseng-project.id/materi-matematika/sma/determinan-kofaktor/
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-02-Matriks-Eselon.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-03-Sistem-Persamaan-Linier.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-04-Tiga-Kemungkinan-Solusi-SPL.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-05-Sistem-Persamaan-Linier-2.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-08-Determinan-bagian1.pdf
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-09-Determinan-bagian2.pdf

Link Repository Github Kelompok Moga Survive:
https://github.com/irsyadnb/Algeo01-21072.git