TUGAS BESAR 1 IF 2123 ALJABAR LINIER DAN GEOMETRI
SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

Oleh

Kelompok 39 (AminA)

Naufal Syifa Firdaus	13521050
Puti Nabilla Aidira	13521088
Muhammad Dhiwaul Akbar	13521158

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

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-1b), 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.

Spesifikasi Tugas:

A. Buatlah pustaka dalam **Bahasa Java** untuk menemukan solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan *n* peubah dan *n* persamaan), menghitung determinan matriks dengan reduksi baris dan dengan ekspansi kofaktor, dan menghitung balikan matriks.

B. Gunakan pustaka diatas untuk membuat program penyelesaian berbagai persoalan dalam bentuk SPL, menyelesaikan persoalan interpolasi dan regresi linier, menghitung matriks balikan, menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

   Spesifikasi program adalah sebagai berikut:

   1. Program dapat menerima masukan (input) baik dari *keyboard* maupun membaca masukan dari file text. Untuk SPL, masukan dari

keyboard adalah *m*, *n*, koefisien *aij* , dan *bi*. Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

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

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

```
 3 4.5 2.8
-3 7 8.3
 0.5 -10 -9
```

3. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah *n*, (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*, *yn*), dan nilai *x* yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

```
8.0 2.0794
9.0 2.1972
9.5 2.2513
```

4. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah *n* (jumlah peubah *x*), m (jumlah sampel), semua nilai-nilai *x*1*i*, *x*2*i*, ..., *xni*, nilai *yi*, dan nilai-nilai *xk* yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
5. Untuk persoalan SPL, luaran (*output*) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya *x*4 = -2, *x*3 = 2*s* − *t*, *x*2 = *s*, dan *x*1 = *t*.)

6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing

```
 -3 7 8.3 11 -4
3 4.5 2.8 10 12
0.5 -10 -9 12 0
0.1 0.2
```

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada *x* yang diberikan. **Contoh** luaran untuk interpolasi adalah $f(x)$= -0.0064x2+ 0.2266x + 0.6762 , f(5) =... dan untuk regresi adalah *f(x)* = -9.5872 + 1.0732x1, $f(xk)$= ...
8. Untuk persoalan interpolasi bicubic, masukan dari file text (.txt) yang berisi matriks berukuran 4x4 yang berisi nilai f(i,j) dengan i dan j adalah indeks matriks diikuti dengan nilai a dan b untuk mencari nilai f(a,b). misalnya jika nilai dari f(-1,-1), f(-1,0), f(-1,1), f(-1,2),f(0,-1), f(0,0), f(0,1), f(0,2), f(1,-1), f(1,0), f(1,1), f(1,2), f(2,-1), f(2,0), f(2,1), f(2,2) berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9 ,10, 11, 12, 13, 14, 15 ,16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi file text ditulis sebagai berikut:

   ```
   1 2 3 4
   5 6 7 8
   9 10 11 12
   13 14 15 16
   0.5 0.5
   ```

   luaran yang dihasilkan adalah nilai dari f(0.5,0.5).

   Masukannya adalah matriks 4 x 4, diikuti oleh nilai a dan b, maka luarannya adalah nilai *f(a,b)*.
9. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam file**.
10. Bahasa program yang digunakan adalah Java.

11. Program **tidak harus** berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas *Eclipse* misalnya).
12. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

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

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

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

Begitu juga untuk pilihan menu nomor 2 dan 3.

Bab II

Teori Singkat

A. Metode Eliminasi Gauss

Metode eliminasi Gauss adalah metode eliminasi yang menggunakan operasi baris elementer (OBE) pada matriks *augmented* untuk membentuk suatu matriks eselon baris, dengan tujuan penyelesaian sistem persamaan linear (SPL). Matriks eselon baris (Gambar 1) adalah matriks yang memiliki *leading one* (satu utama) pada setiap baris yang tidak seluruhnya nol, dengan setiap *leading one* baris lebih rendah terdapat lebih jauh ke kanan daripada *leading one* baris di atasnya. Matriks *augmented* (Gambar 2) adalah matriks yang secara ringkas merepresentasikan SPL, dengan kolom-kolom pada bagian kiri sebagai koefisien (a_{ij}) dari variabel (x_j) dan kolom terakhir sebagai ruas kanan SPL (b_i). Operasi baris elementer (OBE) adalah operasi yang dapat dilakukan terhadap matriks augmented, diantaranya mengalikan sebuah baris dengan konstanta tidak nol, mempertukarkan dua buah baris, dan menambahkan sebuah baris dengan kelipatan baris lainnya.

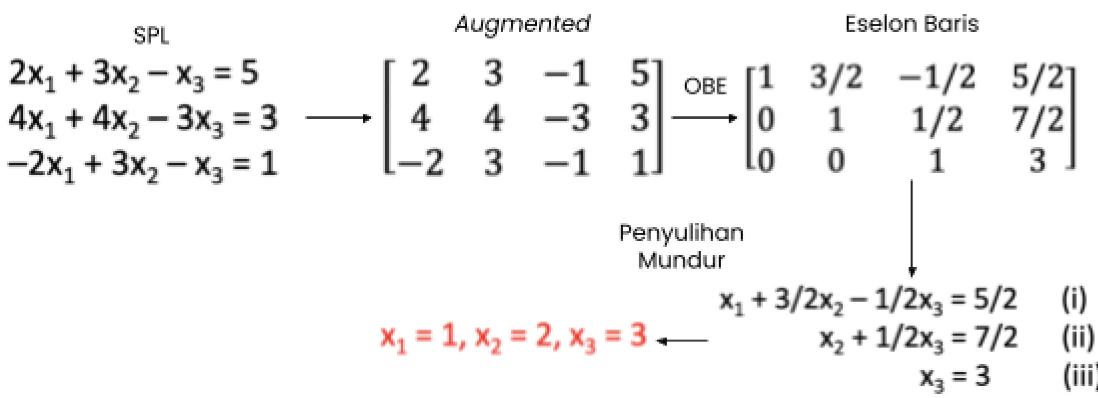
Untuk menyelesaikan suatu SPL dengan metode eliminasi Gauss, suatu SPL harus terlebih dahulu diubah kedalam bentuk matriks *augmented*. Selanjutnya, dilakukan OBE hingga membentuk matriks eselon baris. Terakhir, dilakukan teknik penyulihan mundur pada matriks eselon baris tersebut. Contoh proses penyelesaian SPL dapat dilihat pada Gambar 3.

$$\begin{bmatrix} 1 & * & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Gambar 1. Contoh matriks eselon baris

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

Gambar 2. Matriks *augmented*



Gambar 3. Proses penyelesaian SPL dengan metode eliminasi Gauss

B. Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan adalah lanjutan dari metode eliminasi Gauss yang melakukan OBE sampai terbentuk matriks eselon baris tereduksi. Matriks eselon baris tereduksi (Gambar 4) adalah matriks yang memenuhi sifat-sifat matriks eselon baris dan pada setiap kolom yang memiliki *leading one*, *leading one* tersebut adalah satu-satunya elemen tak nol. Metode eliminasi Gauss-Jordan terdiri dari dua fase: fase maju (eliminasi Gauss) untuk membentuk nol-nol di bawah *leading one* dan fase mundur untuk membentuk nol-nol di atas *leading one.* Dengan metode eliminasi Gauss-Jordan, solusi SPL dapat langsung didaptkan dari matriks *augmented* akhir tanpa perlu melakukan penyulihan mundur.

$$
\begin{bmatrix}
0 & 1 & * & 0 & 0 & 0 & * & * & 0 & * \\
0 & 0 & 0 & 1 & 0 & 0 & * & * & 0 & * \\
0 & 0 & 0 & 0 & 1 & 0 & * & * & 0 & * \\
0 & 0 & 0 & 0 & 0 & 1 & * & * & 0 & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & *
\end{bmatrix}
$$

Gambar 4. Contoh matriks eselon baris tereduksi

C. Determinan

Determinan adalah nilai khusus yang dapat dihitung dari sebuah matriks persegi. Untuk matriks 2x2, determinan dapat dihitung dengan rumus sederhana: det(A) = (a_{11} x a_{22}) - (a_{12} x a_{21}), dengan a_{ij} elemen-elemen matriks A, i indeks baris, dan j indeks kolom. Untuk matriks segita, baik segitiga atas atau segitiga bawah, (Gambar 5) determinan matriks dapat dihitung dari perkalian semua elemen pada diagonal utamanya. Untuk matriks yang memiliki baris nol (baris yang semua elemennya nol) determinannya adalah nol. Determinan matriks identitas (matriks yang elemen pada diagonal utamanya satu, sedangkan elemen lain nol) adalah satu.

Terdapat beberapa aturan perubahan nilai determinan ketika suatu matriks dioperasikan dengan OBE. Pertama, apabila suatu baris dikalikan konstanta (k), maka det(A’) = k det(A). Kedua, apabila dua buah baris dipertukarkan maka det(A’) = - det(A). Ketiga, apabila sebuah baris ditambahkan perkalian baris yang lain det(A’) = det(A).

Terdapat pula beberapa teorema dasar pada determinan. Pertama, determinan matriks transpose sama dengan determinan matriks awalnya, det(A^T) = det(A). Kedua, jika sebuah matriks merupakan perkalian dari dari dua buah matriks, maka deteminannya juga merupakan perkalian dari determinan kedua matriks tersebut, A = BC → det(A) = det(B)det(C). Ketiga, determinan suatu matriks balikan merupakan satu per determinan matriks awalnya, det(A^{-1}) = 1/det(A).

Secara umum, determinan suatu matriks dapat dihitung dengan dua cara: reduksi baris dan ekspansi kofaktor. Metode reduksi baris dilakukan dengan mengaplikasikan OBE pada matriks hingga terbentuk matriks segitiga kemudian menghitung determinan matriks tersebut dengan mengalikan elemen-elemen diagonal utamanya (dengan memperhatikan efek OBE pada perubahan nilai determinan). Metode ekspansi kofaktor dilakukan dengan menghitung kofaktor (C_{ij}) dari setiap elemen matriks (a_{ij}), kemudian menghitung determinannya menggunakan salah satu persaman pada Gambar 6. Kofaktor (C_{ij}) didapat dari minor elemen a_{ij} (M_{ij}) dengan perubahan tanda, $C_{ij} = (-1)^{i+j} M_{ij}$. Minor elemen a_{ij} (M_{ij}) didapat dengan menghitung determinan matriks yang dibentuk dari elemen-elemen matriks A yang tidak sekolom dan tidak sebaris dengan elemen a_{ij} . Perhitungan M_{ij} dilakukan sampai mencapai matriks 2x2 yang perhitungan determinannya dapat menggunakan rumus sederhana.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Gambar 5. Matriks segitiga atas (kanan) dan segitiga bawah (kiri)

$$\det(A) = a_{11}C_{11} + a_{12}C_{12} + \dots + a_{1n}C_{1n}$$
$$\det(A) = a_{21}C_{21} + a_{22}C_{22} + \dots + a_{2n}C_{2n}$$
$$\vdots$$
$$\det(A) = a_{n1}C_{n1} + a_{n2}C_{n2} + \dots + a_{nn}C_{nn}$$

Secara baris

$$\det(A) = a_{11}C_{11} + a_{21}C_{21} + \dots + a_{n1}C_{n1}$$
$$\det(A) = a_{12}C_{12} + a_{22}C_{22} + \dots + a_{n2}C_{n2}$$
$$\vdots$$
$$\det(A) = a_{1n}C_{1n} + a_{2n}C_{2n} + \dots + a_{nn}C_{nn}$$

Secara kolom

Gambar 6. Persamaan untuk menghitung determinan dengan ekspansi kofaktor

D. Matriks Balikan

Matriks balikan adalah suatu matriks yang jika dikalikan dengan matriks awalnya menghasikan matriks identitas, dengan kata lain $AA^{-1} = I$. Terdapat dua metode untuk mencari matriks balikan dari suatu matriks: metode eliminasi Gauss-Jordan dan matriks *adjoin*.

Untuk mencari matriks balikan dengan metode eliminasi Gauss-Jordan, terlebih dahulu dibentuk matriks *augmented* dengan kolom-kolom pada bagian kiri berisi matriks A dan kolom-kolom pada bagian kanan berisi matriks identitas I, [A | I]. Selanjutnya, lakukan eliminasi Gauss-Jordan dengan mengaplikasikan OBE hingga mendapat matriks *augmented* dengan kolom-kolom pada bagian kiri berisi matriks identitas I, maka kolom-kolom pada bagian kanan akan berisi matriks balikan dari A, $[I \mid A^{-1}]$. Contoh perhitungan matriks balikan dengan metode eliminasi Gauss-Jordan dapat dilihat pada Gambar 7.

Untuk mencari matriks balikan dengan matriks *adjoin*, terlebih dulu dilakukan perhitungan terhadap determinan dan *adjoin* matriks (definisi matriks *adjoin* dijelaskan pada bagian F). Selanjutnya gunakan rumus: $A^{-1} = \frac{1}{det(A)} adj(A)$. Contoh perhitungan matriks balikan dengan matriks *adjoin* dapat dilihat pada Gambar 8.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix} \rightarrow \overset{\textit{Augmented}}{\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & \underbrace{0}_{A} & 8 & 0 & \underbrace{0}_{I} & 1 \end{array}\right)} \xrightarrow{\text{OBE}} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & \underbrace{0}_{I} & 1 & 5 & \underbrace{-2}_{A} & -1 \end{array}\right)$$

Gambar 7. Contoh perhitungan matriks balikan dengan metode eliminasi Gauss-Jordan

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A) = \frac{1}{64} \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & -10 & 16 \end{bmatrix} = \begin{bmatrix} 12/64 & 4/64 & 12/64 \\ 6/64 & 2/64 & -10/64 \\ -16/64 & -10/64 & 16/64 \end{bmatrix}$$

Gambar 8. Contoh perhitungan matriks balikan dengan matriks *adjoin*

E. Matriks Kofaktor

Matriks kofaktor (Gambar 9) dari matriks A adalah matriks yang elemen-elemennya merupakan kofaktor (C_{ij}) dari elemen-elemen matriks A (a_{ij}).

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

Gambar 9. Matriks kofaktor

F. Matriks *Adjoin*

Matriks *Adjoin* dari matriks A adalah matriks *transpose* dari matriks kofaktor. Contoh matriks *adjoin* dapat dilihat pada Gambar 10.

$$\text{matriks kofaktor: } \begin{bmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{bmatrix} \longrightarrow \text{adj(A)} = \begin{bmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & -10 & 16 \end{bmatrix}$$

Gambar 9. Contoh matriks *adjoin*

G. Kaidah Cramer

Kaidah Cramer adalah salah satu metode yang dapat digunakan untuk menyelesaikan SPL dengan memanfaatkan nilai determinan. Kaidah Cramer menyatakan jika Ax = b merupakan SPL yang terdiri dari n persamaan linier dengan n peubah sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi yang unik yaitu $x_1 = \frac{det(A1)}{det(A)}, x_2 = \frac{det(A2)}{det(A)}, \ldots\ x_n = \frac{det(An)}{det(A)}$. Dengan A_j merupakan matriks yang diperoleh dengan mengganti elemen-elemen pada kolom ke-j matriks A dengan elemen-elemen pada matriks b. Contoh perhitungan solusi SPL dapat dilihat pada Gambar 10.

Penyelesaian: A**x** = **b**

$$A = \begin{bmatrix} -1 & 2 & -3 \\ 2 & 0 & 1 \\ 3 & -4 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \qquad \det(A) = \begin{vmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{vmatrix} = 10$$

$$A_1 = \begin{bmatrix} \mathbf{1} & 2 & -3 \\ \mathbf{0} & 0 & 1 \\ \mathbf{2} & -4 & 4 \end{bmatrix} \qquad A_2 = \begin{bmatrix} -1 & \mathbf{1} & -3 \\ 2 & \mathbf{0} & 1 \\ 3 & \mathbf{2} & 4 \end{bmatrix} \qquad A_3 = \begin{bmatrix} -1 & 2 & \mathbf{1} \\ 2 & 0 & \mathbf{0} \\ 3 & -4 & \mathbf{2} \end{bmatrix}$$

$$x_1 = \frac{\det(A_1)}{\det(A)} = \frac{8}{10} = \frac{4}{5} \qquad x_2 = \frac{\det(A_2)}{\det(A)} = \frac{-15}{10} = \frac{-3}{2} \qquad x_3 = \frac{\det(A_3)}{\det(A)} = \frac{-16}{10} = \frac{-8}{5}$$

Gambar 10. Contoh perhitungan solusi SPL dengan kaidah Cramer

H. Interpolasi Polinom

Interpolasi polinom adalah salah satu metode numerik yang digunakan untuk menaksir sebuah kurva rumit. Interpolasi polinom merupakan interpolasi dari kumpulan data dengan menggunakan polinom berderajat terendah yang dapat melewati titik-titik kumpulan data tersebut. Teorema interpolasi polinom menyatakan, untuk n+1 buah titik berbeda, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n) dapat ditentukan suatu polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, ..., n. Selanjutnya, persamaan $y_i = p_n(x_i)$ dapat digunakan untuk menghitung taksiran nilai y di sembarang titik di dalam selang $[x_0..x_n]$ dengan menyulihkan (x_i, y_i) kedalam persamaan tersebut hingga terbentuk n buah sistem persamaan linier dengan peubah a_0, a_1, ... a_n (Gambar 11). Selanjutnya, selesaikan persamaan linier tersebut dengan metide eliminasi Gauss untuk mendapatkan koefisien-koefisien a_i . Setelah mendapatkan koefisien-koefisien a_i taksiran nilai y di sembarang titik di dalam selang $[x_0..x_n]$ dapat dicari dengan $p_n(x)$.

$$
\begin{aligned}
a_0 + a_1x_0 + a_2{x_0}^2 + \ldots + a_n\, {x_0}^n &= y_0 \\
a_0 + a_1x_1 + a_2{x_1}^2 + \ldots + a_n\, {x_1}^n &= y_1 \\
\ldots \qquad\qquad\qquad & \ldots \\
a_0 + a_1x_n + a_2{x_n}^2 + \ldots + a_n\, {x_n}^n &= y_n
\end{aligned}
$$

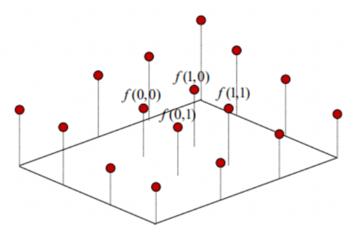
Gambar 11. n buah sistem persamaan linier hasil penyulihan (x_i, y_i) ke $y_i = p_n(x_i)$

I. Interpolasi *Bicubic*

Interpolasi *bicubic* adalah teknik interpolasi pada data 2D. Teknik ini merupakan pengembangan dari interpolasi linear dan *cubic* dan biasa digunakan dalam pembesaran citra. Untuk menghitung persamaan interpolasi f(x, y) dari sebuah matriks M, lakukan penyulihan nilai-nilai yang diketahui pada matriks M hingga terbentuk sebuah matriks persamaan y = Xa. Contoh pemodelan f(x, y) untuk matriks 4x4 dapat dilihat pada Gambar 12. Contoh matriks persamaan y = Xa dapat dilihat Gambar 13. Selanjutnya, nilai-nilai koefisien a_{ij} dicari dengan menyelesaikan persamaan $y^{-1}x = a$.

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

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



Gambar 12. Contoh pemodelan f(x, y) untuk matriks 4x4

$$y = Xa$$

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

Gambar 13. Contoh matriks persamaan y = Xa

J. Regresi Linier Berganda

Regresi linier merupakan salah satu metode numerik yang digunakan untuk memprediksi/menaksir nilai. Regresi linier berganda merupakan regresi linier dengan banyak peubah x lebih dari 1. Rumus umum regresi linier berganda adalah $y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_k x_{ki} + \epsilon_i$. Dengan, nilai $b_0 ... b_k$ dicari menggunakan penyulihan data-data (x, y) yang diketahui ke persamaan *Normal Estimation Equation for Multiple Linear Regression* (Gambar 14), kemudian menyelesaikannya dengan metode eliminasi Gauss.

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

Gambar 14. *Normal Estimation Equation for Multiple Linear Regression*

Bab III
Implementasi Pustaka dan Program Java

A. Class MAIN

Program utama yang menjalankan semua method operasi matriks

Nama	Parameter	Tipe & Return Value	Keterangan
main	-	Public void	Menjalankan program secara keseluruhan dengan memanggil fungsi yang telah dibuat sesuai input user.

B. Class Menu

Berisi method untuk mencetak menu pada layar.

Nama	Parameter	Tipe & Return Value	Keterangan
menuMain	-	Public void	Mencetak menu awal.
menuSpl	-	Public void	Mencetak menu SPL.
menuDeterminan	-	Public void	Mencetak menu determinan.
menuBalikan	-	Public void	Mencetak menu balikan.
menuInput	-	Public void	Mencetak menu pilihan input.
menuWritetoFile	-	Public void	Mencetak menu pilihan penyimpanan

			hasil ke file.
input	Scanner read	Public int	Menerima input pilihan 1 atau 2 dari keyboard

C. Class IOFile

Berisi method untuk meng-*handle* file yang bisa dipakai method lain dalam program.

Nama	Parameter	Tipe & Return Value	Keterangan
InputFileName	Scanner read	Public String	Menerima input nama file
RowCounter	String filename	Public int	Menghitung baris matriks dalam file
ColCounter	String filename	Public int	Menghitung kolom matriks dalam file
readFile	String filename, int row, int col	Public double[][]	Mengembalikan matriks yang dibaca dari file.
createFile	String filename	Public boolean	Membuat file dalam directory dan mengembalikan true jika berhasil
writeFile_1	String filename, val	Public void	Menulis string masukan kedalam file.
writeRegresi	String filename, double[][] res, double restaksir	Public void	Menulis hasil regresi dan nilai taksirannya ke dalam file

writeSolGauss	String filename, String[] res	Public void	Menulis hasil perhitungan eliminasi gauss ke dalam file
writeMatrix	String filename, double[][] m	Public void	Menulis matriks kedalam file

D. Class IOKeyboard

Berisi method untuk meng-*handle* masukan pengguna dari keyboard.

Nama	Parameter	Tipe & Return Value	Keterangan
InputOption	Scanner read	Public int	Menerima pilihan pengguna untuk terima input dari keyboard atau file.
WriteFileOption	Scanner read	Public int	Menerima pilihan pengguna untuk menulis hasil di file atau tidak.
readMatrix	Scanner read	Public double[][]	Mengembalikan matriks yang di masukan dari keyboard.
readArrTaksir	Scanner read, int n	Public double[]	Menerima nilai yang ingin ditaksir dari pengguna
readMatrixSPL	Scanner read	Public double[][]	Menerima input matriks a dan b dari SPL.
readMatrixRegresi	Scanner read	Pubilc double[][]	Menerima data matriks untuk regresi.

printMatrix	Double[][] mat	Public void	Menampilkan matriks ke layar.

E. Class MatrixOp

Berisi operasi-operasi primitif matriks.

Nama	Parameter	Tipe & Return Value	Keterangan
getRow	double[][] m	Public int	Mengembalikan jumlah baris matriks.
getCol	double[][] m	Public int	Mengembalikan jumlah kolom matriks.
copyRow	double[] rcopy, double[] rori, int n	Public void	Menyalin baris matriks.
timesVal	double[][] m , double val	Public void	Mengalikan matriks dengan skalar.
transpose	double[][] m	Public void	Men-transpose matriks.
ftranspose	double[][] m	Public double[][]	Mengembalikan matriks baru hasil transpose matriks.
isColZero2	double[][] mat, int col	Public boolean	Mengembalikan apakah elemen-elemen suatu kolom bernilai nol semua.
isRowZero	double[][] mat	Public boolean	Mengecek apakah ada baris yang elemen-elemennya

			bernilai nol semua.
isColZero	double[][] mat	Public boolean	Mengecek apakah ada kolom yang elemen-elemennya bernilai nol semua.
matrixTimes	double[][] m1, double[][] m2	Public double[][]	Perkalian matriks versi 1.
matrixTimes2	double[][] m1, double[][] m2	Public double[][]	Perkalian matriks versi 2.
isSquare	double[][] m	Public boolean	Mengembalikan apakah matriks merupakan matriks persegi.

F. Class gauss

Berisi method untuk menghitung solusi SPL dengan metode Gauss.

Nama	Parameter	Tipe & Return Value	Keterangan
gaussel	double mat[][]	Public double[][]	Melakukan eliminasi gauss, mengembalikan matriks eselon baris hasil eliminasi gauss.
splsolver	double mat[][]	Public String[]	Memanggil method-method penyelesaiaan SPL lain berdasarkan kategori: tidak ada solusi, solusi unik, dan solusi banyak. Mengembalikan solusi SPL sekaligus mencetaknya pada

			layar.
isnosol	double mat[][]	Public boolean	Mengecek apakah SPL tidak memiliki solusi (matriks memiliki baris yang elemennya nol kecuali kolom terakhir).
cutmat	double mat[][]	Public double[][]	Memotong baris-baris yang elemen-elemennya nol semua.
uniqsolver	double mat[][]	Public String[]	Menyelesaikan SPL untuk solusi unik dengan melakukan penyulihan mundur. Mengembalikan string solusi.
idxlo	double mat[][], int col	Public int	Mengembalikan indeks baris *leading one* pada suatu kolom, -1 jika kolom tidak memiliki *leading one.*
paramsolver	double mat[][]	Public String[]	Menyelesaikan SPL untuk solusi banyak (dengan parameter) dengan melakukan penyulihan mundur. Mengembalikan string solusi.
makeparam	int idx, int[] paramidx, int count	Private char	Mengenerate parameter unik (a…b), apabila

			dibtuhkan parameter untuk indeks baris tersebut.
gaussdriver	Scanner read	Public void	Meng-*handle* input/output khusus yang diperlukan untuk class gauss, dipanggil langsung dari class MAIN.

G. Class gaussjordan

Berisi method untuk menghitung solusi SPL dengan metode Gauss-Jordan.

Nama	Parameter	Tipe & Return Value	Keterangan
gaussel	double mat[][]	Public double[][]	Melakukan eliminasi gauss jordan, mengembalikan matriks eselon baris tereduksi hasil eliminasi gauss-jordan.
splsolver	double mat[][]	Public String[]	Memanggil method-method penyelesaiaan SPL lain berdasarkan kategori: tidak ada solusi, solusi unik, dan solusi banyak. Mengembalikan solusi SPL sekaligus mencetaknya pada layar.
isnosol	double mat[][]	Public boolean	Mengecek apakah SPL tidak memiliki

			solusi (matriks memiliki baris yang elemennya nol kecuali kolom terakhir).
cutmat	double mat[][]	Public double[][]	Memotong baris-baris yang elemen-elemennya nol semua.
uniqsolver	double mat[][]	Public String[]	Menyelesaikan SPL untuk solusi unik dengan melakukan penyulihan mundur. Mengembalikan string solusi.
idxlo	double mat[][], int col	Public int	Mengembalikan indeks baris *leading one* pada suatu kolom, -1 jika kolom tidak memiliki *leading one.*
paramsolver	double mat[][]	Public String[]	Menyelesaikan SPL untuk solusi banyak (dengan parameter) dengan melakukan penyulihan mundur. Mengembalikan string solusi.
makeparam	int idx, int[] paramidx, int count	Private char	Mengenerate parameter unik (a…b), apabila dibtuhkan parameter untuk indeks baris tersebut.

gaussjordandriver	Scanner read	Public void	Meng-*handle* input/output khusus yang diperlukan untuk class gaussjordan, dipanggil langsung dari class MAIN.

H. Class InverseAdjoin

Kumpulan Method untuk menghasilkan inverse dari sebuah matriks.

Nama	Parameter	Tipe & Return Value	Keterangan
inverseAdjoin	Double[][] m	Public double[][]	Menghasilkan matriks balikan dari masukan matriks
adjoinDriver	Scanner read	Public void	Menjalankan method inverseAdjoin, menerima input matriks, dan menulis hasil ke file

I. Class InverseGauss

Berisi method untuk menghasilkan inverse dengan metode eliminasi gauss.

Nama	Parameter	Tipe & Return Value	Keterangan
createIdentity	Int row, int col	Public double[][]	Menghasilkan matriks identitas dengan baris dan kolom yang diberikan.
addIdentity	double[][]	Public double[][]	Menggabungkan matriks dengan matriks identitasnya.

gaussJordan	double[][]	Public void	Menerapkan eliminasi gauss jordan pada matriks
inverseGauss	double[][]	Public double[][]	Menghasilkan matriks balikan dari matriks yang dimasukan.

J. Class regresi

Berisi method untuk membentuk persamaan regresi dan menaksir nilai.

Nama	Parameter	Tipe & Return Value	Keterangan
regresisolver	double mat[][]	Public double[][]	Menghasilkan matriks eselon baris tereduksi berisi koefisien b persamaan regresi pada kolom terakhir dengan menghitung NEE dan mengeliminasinya dengan metode eliminasi gauss-jordan.
printpers	double res[][]	Public void	Mencetak persamaan regresi ke layar.
taksir	double[] xtaksir, double[][] res	Public double	Menaksir nilai fungsi pada data tertentu menggunakan persamaan regresi.
regresidriver	Scanner read	Public void	Meng-*handle* input/output khusus yang diperlukan untuk class regresi,

			dipanggil langsung dari class MAIN.

K. Class Cofactor

Class untuk memproses matriks dan menghasilkan determinannya dengan metode ekspansi kofaktor.

Nama	Parameter	Tipe & Return Value	Keterangan
minorEntry	double[][] m, int a, int b	Public double[][]	Menghasilkan minor entry dari elemen dengan indeks a dan b.
matrixCofactor	Double[][] m	Public double[][]	Menghasilkan matriks kofaktor dari matriks yang diberi.
determinantKofaktor	double[][]	Public double	Menghasilkan determinan sebuah matriks
cofactorDriver	Scanner read	Public void	Menjalankan fungsi diatas, menerima input matriks, dan menulis hasil dalam file.

L. Class DeterminantReduction

Berisi method untuk menghitung determinan matriks dengan metode reduksi baris.

Nama	Parameter	Tipe & Return Value	Keterangan

correctionPos	double[][] m	Public int	Menukar baris agar valid ketika dilakukan OBE dan menghasilkan int untuk menentukan positif/negatif determinannya.
determinanUt	double[][] m	Public double	Menghasilkan determinan sebuah matriks
reductionDriver	Scanner read	public void	Menjalankan method diatas, menerima input matriks dan menulis hasil ke file.

M. Class SplBalikan

Berisi method untuk menyelesaikan SPL matriks dengan metode matriks inverse.

Nama	Parameter	Tipe & Return Value	Keterangan
splInverse	double[][] m, Scanner read	Public void	Menghitung solusi matriks dan menuliskan hasilnya ke file.
splbalikanDriver	Scanner read	Public void	Menerima input matriks dan menjalankan method splInverse.

N. Class SplCramer

Kumpulan method untuk menyelesaian spl dengan kaidah kramer.

Nama	Parameter	Tipe & Return Value	Keterangan
solveCramer	double[][] m, Scanner read	Public void	Menghitung solusi matriks dan menuliskan hasilnya ke file.
splcramerDriver	Scanner read	Public void	Menerima input matriks dan menjalankan method splInverse.

O. Class Interpolasipolinom

Terdiri atas beberapa method untuk menyelesaikan persoalan interpolasi polinom.

Nama	Parameter	Tipe & Return Value	Keterangan
approach	Double[] koefisien, double x	Public double	Menghitung perkiraan nilai y di sembarang titik di dalam selang [x0 , xn].
matrixinterpol	double[][] firstmatrix	Public double[][]	Menerima input matriks dan menghasilkan maktriks interpolasi
polynomEq	Double[][] initmatrix	Public double []	Menerima input matrix dengan baris n (sembarang) dan kolom 2 , lalu menghasilkan matriks yang berisi koefisien dari hasil

			persamaan interpolasi dengan jumlah kolom 1
equation	Double[] koefisien , double result	public string	Menghasilkan string yang berbentuk persamaan n kuadrat dari matriks koefisien dan result dari approksimasi nilai yang kita input sebelumnya
interpolasipolinom driver	Scanner read	Public void	Merupakan method yang menjalankan semua perintah menuju hasil dari interpolasi polinom dan perintah yang berisi pilihan untuk menyimpan file atau tidak , yang mana method ini terhubung ke class MAIN

P. Class bicubicinterpolation

Terdiri atas beberapa method untuk menyelesaikan persoalan interpolasi bikubik .

Nama	Parameter	Tipe & Return Value	Keterangan
estimateres	double[] koefisien, double a, doouble b	Public double	Menghasilkan perkiraan nilai dari f(a,b) koefisien persamaannya

			adalah matriks koefisien
matriksinit		Public double[][]	Menghasilkan matriks inisiasi dari bicubic interpolasi yang memiliki 16 baris dan 16 kolom
bicubiceq	double[][] startmatrix	Public double[]	Menghasilkan matriks koefisien yang berisi koefisien-koefisien Dari persamaan bikubik interpolasi
bicubicinterpolationdriver	Scanner read	Public void	Merupakan method yang menjalankan semua perintah menuju hasil dari bikubik interpolasi dan perintah yang berisi pilihan untuk menyimpan file atau tidak , yang mana method ini terhubung ke class MAIN

Bab IV
Eksperimen

1. Soal SPL Ax = b

a.

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 1a
Membaca file '../test/1a.txt'
SPL tidak memiliki solusi
```

- Metode eliminasi Gauss-Jordan

```
Masukan nama file tanpa extension: 1a
Membaca file '../test/1a.txt'
SPL tidak memiliki solusi
```

- Metode Matriks Inverse

```
Masukan nama file tanpa extension: 1a
Membaca file '../test/1a.txt'
-----  Matriks Tidak Mempunyai Balikan!  -----
```

- Metode Cramer

```
Masukan nama file tanpa extension: 1a
Membaca file '../test/1a.txt'
-----  Determinan Matriks Sama Dengan Nol!  -----
```

b.

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 1b
Membaca file '../test/1b.txt'
x1 = 3.0 + (1.0)b
x2 = 0.0 + (2.0)b
x3 = a
x4 = -1.0 + (1.0)b
x5 = b
```

- Metode eliminasi Gauss-Jordan

```
Masukan nama file tanpa extension: 1b
Membaca file '../test/1b.txt'
x1 = 3.0 + (1.0)b
x2 = 0.0 + (2.0)b
x3 = a
x4 = -1.0 + (1.0)b
x5 = b
```

- Metode Matriks Inverse

```
Masukan nama file tanpa extension: 1b
Membaca file '../test/1b.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

- Metode Kramer

```
Masukan nama file tanpa extension: 1b
Membaca file '../test/1b.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

c.

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 1c
Membaca file '../test/1c.txt'
x1 = a
x2 = 1.0 + (-1.0)c
x3 = b
x4 = -1.0 + (-1.0)a
x5 = 1.0 + (1.0)c
x6 = c
```

- Metode eliminasi Gauss-Jordan

```
Masukan nama file tanpa extension: 1c
Membaca file '../test/1c.txt'
x1 = a
x2 = 1.0 + (-1.0)c
x3 = b
x4 = -1.0 + (-1.0)a
x5 = 1.0 + (1.0)c
x6 = c
```

- Metode Matriks Inverse

```
Masukan nama file tanpa extension: 1c
Membaca file '../test/1c.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

- Metode Cramer

```
Masukan nama file tanpa extension: 1c
Membaca file '../test/1c.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

d.

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

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

- Metode eliminasi Gauss

n = 6

```
Masukan nama file tanpa extension: 1d1
Membaca file '../test/1d1.txt'
x1 = 36.00000000098021
x2 = -630.0000000292657
x3 = 3360.000000203484
x4 = -7560.000000539233
x5 = 7560.000000603351
x6 = -2772.000000240222
```

n = 10

```
Masukan nama file tanpa extension: 1d2
Membaca file '../test/1d2.txt'
x1 = 99.99634656462877
x2 = -4949.682659993443
x3 = 79193.21483186202
x4 = -600538.1369470647
x5 = 2522224.258680418
x6 = -6305485.547186602
x7 = 9608261.78322886
x8 = -8750305.214203862
x9 = 4375119.565502384
x10 = -923630.2349573893
```

- Metode eliminasi Gauss-Jordan

n = 6

```
Masukan nama file tanpa extension: 1d1
Membaca file '../test/1d1.txt'
x1 = 36.00000000098021
x2 = -630.0000000292657
x3 = 3360.000000203484
x4 = -7560.000000539233
x5 = 7560.000000603351
x6 = -2772.000000240222
```

n = 10

```
Masukan nama file tanpa extension: 1d2
Membaca file '../test/1d2.txt'
x1 = 99.99634656462877
x2 = -4949.682659993443
x3 = 79193.21483186202
x4 = -600538.1369470647
x5 = 2522224.258680418
x6 = -6305485.547186602
x7 = 9608261.78322886
x8 = -8750305.214203862
x9 = 4375119.565502384
x10 = -923630.2349573893
```

- Metode Matriks Inverse

n = 6

```
-----  Hasil Perhitungan  -----
X1 = 36.000000000980656
X2 = -630.0000000292673
X3 = 3360.000000203484
X4 = -7560.000000539232
X5 = 7560.000000603351
X6 = -2772.000000240222
```

n = 10

```
-----  Hasil Perhitungan  -----
X1 = 99.99634656426257
X2 = -4949.682659994306
X3 = 79193.21483186251
X4 = -600538.1369470842
X5 = 2522224.25868042
X6 = -6305485.547186624
X7 = 9608261.78322886
X8 = -8750305.214203862
X9 = 4375119.565502384
X10 = -923630.2349573893
```

- Metode Cramer

n = 6

```
Masukan nama file tanpa extension: 1d1
Membaca file '../test/1d1.txt'
-----  Hasil Perhitungan  -----
X1 = 35.99999849275147
X2 = -629.9999497041402
X3 = 3359.9997099521784
X4 = -7559.999411026779
X5 = 7559.999515511824
X6 = -2771.9998638224647
```

n = 10

```
Masukan nama file tanpa extension: 1d2
Membaca file '../test/1d2.txt'
-----  Hasil Perhitungan  -----
X1 = 34.59017767772993
X2 = -381.54024790908835
X3 = 0.37562228166023426
X4 = 5698.242961320487
X5 = -9412.043303205875
X6 = -3176.6755953264324
X7 = 7577.607165479928
X8 = 5393.857417158976
X9 = -2806.7100611943056
X10 = -3005.422213180605
```

2. SPL berbentuk matriks *augmented*

a.

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 2a
Membaca file '../test/2a.txt'
x1 = -1.0 + (1.0)b
x2 = 0.0 + (2.0)a
x3 = a
x4 = b
```

- Metode eliminasi Gauss-Jordan

```
Masukan nama file tanpa extension: 2a
Membaca file '../test/2a.txt'
x1 = -1.0 + (1.0)b
x2 = 0.0 + (2.0)a
x3 = a
x4 = b
```

- Metode Matriks Inverse

```
Masukan nama file tanpa extension: 2a
Membaca file '../test/2a.txt'
-----  Matriks Tidak Mempunyai Balikan!  -----
```

- Metode Cramer

```
Masukan nama file tanpa extension: 2a
Membaca file '../test/2a.txt'
-----  Determinan Matriks Sama Dengan Nol!  -----
```

b.

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 2b
Membaca file '../test/2b.txt'
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

- Metode eliminasi Gauss-Jordan

```
Masukan nama file tanpa extension: 2b
Membaca file '../test/2b.txt'
x1 = 0.0
x2 = 2.0
x3 = 1.0
x4 = 1.0
```

- Metode Matriks Inverse

```
Masukan nama file tanpa extension: 2b
Membaca file '../test/2b.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

- Metode Cramer

```
Masukan nama file tanpa extension: 2b
Membaca file '../test/2b.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

3. SPL Berbentuk

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 3a
Membaca file '../test/3a.txt'
x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

- Metode elimminasi Gauss Jordan

```
Masukan nama file tanpa extension: 3a
Membaca file '../test/3a.txt'
x1 = -0.2243243243243243
x2 = 0.18243243243243246
x3 = 0.7094594594594594
x4 = -0.25810810810810797
```

- Metode Matriks Inverse

```
-----  Hasil Perhitungan  -----
X1 = -0.2243243243243243
X2 = 0.1824324324324243
X3 = 0.7094594594594593
X4 = -0.2581081081081081 4
```

- Metode Cramer

```
Masukan nama file tanpa extension: 3a
Membaca file '../test/3a.txt'
-----  Hasil Perhitungan  -----
X1 = -0.22432432432432434
X2 = 0.18243243243243243
X3 = 0.7094594594594594
X4 = -0.2581081081081081
```

b.

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

- Metode eliminasi Gauss

```
Masukan nama file tanpa extension: 3b
Membaca file '../test/3b.txt'
SPL tidak memiliki solusi
```

- Metode eliminasi Gauss Jordan

```
Masukan nama file tanpa extension: 3b
Membaca file '../test/3b.txt'
SPL tidak memiliki solusi
```

- Metode Matriks Inverse

```
Masukan nama file tanpa extension: 3b
Membaca file '../test/3b.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

- Metode Cramer

```
Masukan nama file tanpa extension: 3b
Membaca file '../test/3b.txt'
-----  Matriks A Tidak Berbentuk Square!  -----
```

4. Interpolasi Polinom

a. Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. 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

- x=0.2

```
Persamaan polinomial yang diperoleh
y = -4212.434532 x^6 + 7102.399162 x^5 -4346.313951 x^4 + 1220.854891 x^3 -163.915663 x^2 + 10.276384 x^1 -0.184559
```

```
f(0.200000)=0.130000
```

- x=0.55

```
Persamaan polinomial yang dihasilkan
y = -4212.434532 x^6 + 7102.399162 x^5 -4346.313951 x^4 + 1220.854891 x^3 -163.915663 x^2 + 10.276384 x^1 -0.184559
```

```
f(0.550000)=2.137572
```

- x=0.85

```
Persamaan polinomial yang dihasilkan
y = -4212.434532 x^6 + 7102.399162 x^5 -4346.313951 x^4 + 1220.854891 x^3 -163.915663 x^2 + 10.276384 x^1 -0.184559
```

```
f(0.850000)=-66.269639
```

- x = 1.28

```
Persamaan polinomial yang dihasilkan
y = -4212.434532 x^6 + 7102.399162 x^5 -4346.313951 x^4 + 1220.854891 x^3 -163.915663 x^2 + 10.276384 x^1 -0.184559
```

```
f(1.280000)=-3485.144902
```

b. Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6,567	12.624
30/06/2022	7	21.807
08/07/2022	7,258	38.391
14/07/2022	7,451	54.517
17/07/2022	7,548	51.952
26/07/2022	7,839	28.228
05/08/2022	8,161	35.764
15/08/2022	8,484	20.813
22/08/2022	8,709	12.408
31/08/2022	9	10.534

A. 16/07/2022

```
Persamaan polinomial yang dihasilkan
y = -134834.449095 x^9 + 8937024.068705 x^8 -261784399.489609 x^7 + 4445246728.153746 x^6 -48187349775.590460 x^5 + 345509254362.330570 x^4 -1636750
946034.985600 x^3 + 4932533737229.896000 x^2 -8564052148298.946000 x^1 + 6509622848897.870000
```

```
f(7.533000)=52804.267578
```

B. 10/08/2022

```
Persamaan polinomial yang dihasilkan
y = -134834.449095 x^9 + 8937024.068705 x^8 -261784399.489609 x^7 + 4445246728.153746 x^6 -48187349775.590460 x^5 + 345509254362.330570 x^4 -1636750
946034.985600 x^3 + 4932533737229.896000 x^2 -8564052148298.946000 x^1 + 6509622848897.870000
```

```
f(8333.000000)=-2591536238994634000000000000000000000000.000000
```

C. 05/09/2022

```
Persamaan polinomial yang dihasilkan
y = -134834.449095 x^9 + 8937024.068705 x^8 -261784399.489609 x^7 + 4445246728.153746 x^6 -48187349775.590460 x^5 + 345509254362.330570 x^4 -1636750
946034.985600 x^3 + 4932533737229.896000 x^2 -8564052148298.946000 x^1 + 6509622848897.870000
```

```
f(9.167000)=-660922.335938
```

c. 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 $h = (2 - 0)/5 = 0.4$.

5. Interpolasi *Bicubic*

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

Tentukan nilai:

f(0,0) = ?

f(0.5, 0.5) = ?

f(0.25, 0.75) = ?

f(0.1, 0.9) = ?

- f(0,0)

```
-----  Berhasil menuliskan pada 5.1  -----
f(0.0,0.0)= 153.0
```

- f(0.5,0.5)

```
-----  Berhasil menuliskan pada 5.2  -----
f(0.5,0.5)= -49.0
```

- f(0.25,0.75)

```
-----  Berhasil menuliskan pada 5.3  -----
f(0.25,0.75)= -640.4609375
```

- f(0.1,0.9)

```
-----  Berhasil menuliskan pada 5.4  -----
f(0.1,0.9)= -1176.1098559999998
```

6. Regresi liner 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.

```
Masukan nama file tanpa extension: 6
Membaca file '../test/6.txt'
y = -3.454805 + (-0.002658)x1 + (0.000965)x2 + (0.151936)x3
y = 0.930020
```

Bab V
Penutup

A. Kesimpulan

Program kelompok kami dapat memproses matriks baik dari masukan pengguna melalui keyboard atau melalui file yang telah disiapkan sebelumnya. Semua metode dan cara penyelesaian yang diajarkan di mata kuliah Aljabar Geometri ini kami implementasikan dalam program dengan bentuk algoritma penyelesaian.

Sistem Persamaan Linear kami implementasikan menggunakan 4 metode penyelesaian diantaranya Eliminasi Gauss, Eliminasi Gauss-Jordan, Matriks Balikan, dan Kaidah Cramer. Keempatnya dapat dipilih sebagai metode penyelesaian dalam program tapi untuk matriks yang tidak mempunyai balikan dan determinannya nol tidak bisa menggunakan metode Matriks Balikan dan Kaidah Cramer sehingga akan ditampilkan pesan error untuk kasus tersebut.

Untuk menghitung determinan, terdapat dua metode dalam program yaitu dengan Reduksi Baris dan Ekspansi Kofaktor. Keduanya mempunyai syarat matriks harus berbentuk *square* agar bisa diproses program jika tidak, program akan menampilkan pesan eror. Lalu, kami implementasikan juga metode Matriks Adjoin dan Eliminasi Gauss-Jordan untuk menghasilkan inverse dari sebuah matriks.

Sesuai spesifikasi yang diberi, interpolasi polinom dan bicubic kami implementasikan dalam program dengan masukan dari keyboard pengguna maupun dari file d dan nilai yang ditaksir terkandung dalam file. Untuk regresi linear berganda kami implementasikan dengan masukan yang serupa teknisnya dengan interpolasi.

Hasil dari tiap-tiap metode dalam program kami ditampilkan di layar sekaligus bisa juga dituliskan kedalam file baru yang namanya ditentukan oleh pengguna. Soal tes yang terdapat dalam ketentuan kami simpan dalam bentuk file .txt untuk dipakai dalam program dengan output yang kami taksir sudah tepat.

B. Saran

Persoalan pada lembar ketentuan akan lebih tepat jika diberi contoh yang terdapat jawaban benarnya sehingga memudahkan kami untuk mengetes program sesuai ketentuan. Ketentuan masukan dan keluaran dalam bentuk paragraf menurut kami kurang menjelaskan secaa detail dan terdapat kalimat yang dapat diartikan berbeda-beda. Untuk rekan-rekan yang akan atau sedang mengerjakan tugas besar, saran kami adalah untuk tidak ditunda dan dipahami spekifikasi yang diminta sesegera mungkin.

C. Refleksi

Dalam pengerjaan tugas besar ini kami tidak langsung menganggapi dan membagikan tugas individu sesegera mungkin yang adalah sebuah kesalahan dari kami. Waktu yang tersisa untuk kami mengerjakan program utama, laporan, dan menyempurnakan program menjadi lebih sedikit dan terburu-buru. Walaupun pada akhirnya dapat terselesaikan, pengerjaan ini menjadi pembelajaran bagi kami untuk selalu mengerjakan sesuatu sesegera mungkin untuk antisipasi kejadian yang tidak terduga.

Link Repository

https://github.com/nomsf/Algeo01-21050

Daftar Referensi

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-07-Aplikasi-SPL-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
https://en.wikipedia.org/wiki/Linear_regression#cite_note-Freedman09-1
https://en.wikipedia.org/wiki/Bicubic_interpolation
https://en.wikipedia.org/wiki/Polynomial_interpolation