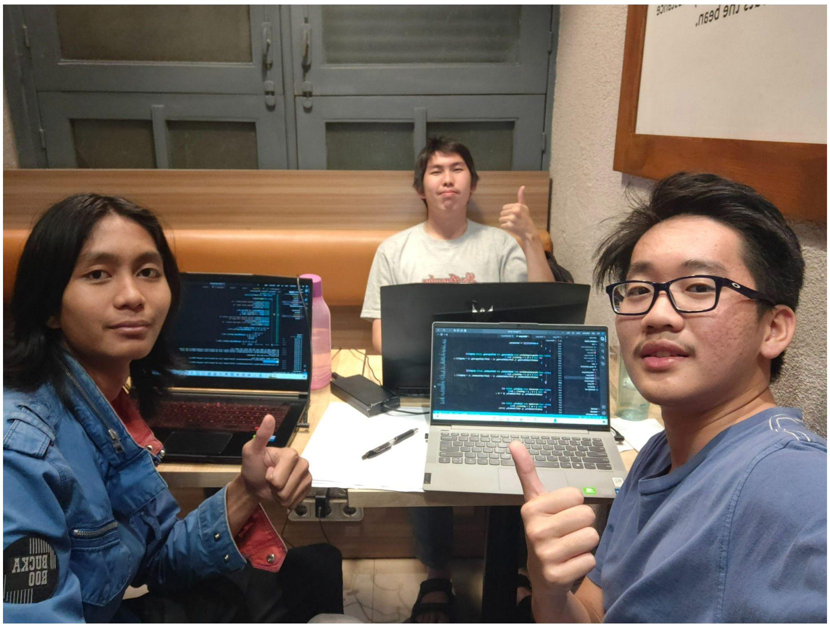
Laporan Tugas Besar 1 IF2123 Aljabar Linier dan Geometri

Sistem Persamaan Linier, Determinan, dan Aplikasinya

Semester I Tahun 2023/2024



oleh

1. Vanson Kurnialim 13522049
2. Muhamad Rafli Rasyiidin 13522088
3. Andhika Tantyo Anugrah 13522094

Kelompok kramer

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

Daftar Isi

BAB 1
Deskripsi Persoalan

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).

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

Gambar 1. Eliminasi Gauss dilakukan dengan matriks eselon baris dan eliminasi Gauss Jordan dengan matriks eselon baris tereduksi.

Di dalam Tugas Besar 1 ini, kami 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, *library* tersebut digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

BAB 2
Teori Singkat

Terdapat 2 metode eliminasi sistem persamaan linear yaitu metode eliminasi Gauss dan Gauss-Jordan. Metode eliminasi Gauss memanfaatkan operasi baris elementer untuk mengubah diagonal utama menjadi bernilai 1 sebagai *leading one* dan membuat elemen matriks di bawah diagonal utama menjadi bernilai 0. Metode eliminasi Gauss-Jordan merupakan lanjutan dari metode Gauss yaitu mereduksi elemen-elemen dalam matriks sehingga didapatkan matriks yang berisi *leading one* dan hasil dari SPL. Jika sudah didapatkan matriks hasil eliminasi Gauss maupun Gauss-Jordan, solusi dari sistem persamaan linear akan dapat dicari (solusi unik, solusi banyak, atau tidak ada solusi).

Determinan matriks merupakan selisih antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder. Namun, determinan tidak hanya dapat dicari dengan cara tersebut, melainkan beberapa cara lain seperti, mencari determinan melalui ekspansi kofaktor:

$$\det(A) = a_{11}C_{11} + a_{21}C_{21} + a_{31}C_{31}$$

Gambar 2. Definisi matematika dari determinan suatu matriks A.

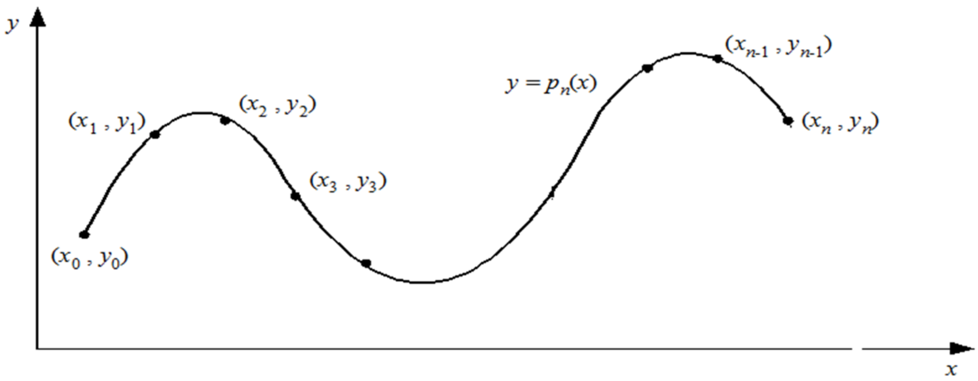
Determinan juga dapat dicari dengan mereduksi matriks sehingga memudahkan dalam perhitungan. Hal yang perlu diperhatikan adalah determinan hanya terdapat pada matriks persegi.

Sebuah matriks dikatakan *invertible* jika matriks tersebut memiliki balikan. Seperti konsep matriks balikan pada umumnya, jika suatu matriks dikalikan dengan balikannya maka akan didapatkan matriks identitas. Suatu matriks memiliki matriks balikan jika dan hanya jika determinannya tidak 0. Ada beberapa metode untuk mencari invers suatu matriks seperti membagi adjoin dengan determinan. Invers juga dapat dicari dengan metode identitas dimana suatu matriks digabungkan dengan matriks identitas di bagian kanannya. Lalu dilakukan operasi baris elementer yang sama pada kedua matriks agar matriks bagian kiri berubah menjadi matriks identitas, dengan demikian akan didapatkan matriks balikan pada bagian kanan matriks.

Kofaktor yang telah disebutkan tadi merupakan perkalian antara minor dan suatu angka yang besarnya menuruti suatu aturan, antara positif 1 atau minus 1. Sedangkan minor adalah determinan yang didapat setelah tidak mengikutsertakan baris dan kolom matriks pada elemen tersebut. Matriks kofaktor adalah sebuah matriks yang tiap elemennya merupakan besaran kofaktor dari elemen itu sendiri. Adjoin adalah transpose dari suatu matriks kofaktor yang mana transpose adalah menukarkan baris dan kolom pada matriks.

Kaidah Cramer merupakan salah satu cara lain yang dapat digunakan untuk menyelesaikan SPL. Cara kerjanya adalah dengan mengganti suatu kolom dengan hasil, lalu mencari determinan dari matriks yang telah diubah. Determinan tersebut merupakan determinan variabel kolom yang nantinya akan dibagi dengan determinan aktual dari matriks sehingga didapat nilai dari variabelnya.

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan $n+1$ buah titik berbeda, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $i = 0, 1, 2, \dots, n$.



Gambar 3. Ilustrasi beberapa titik yang diinterpolasi secara polinomial.

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

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

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

Regresi Linear Berganda merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada persamaan jadi untuk menghitung regresi linear sederhana, terdapat persamaan umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \varepsilon_i$$

Gambar 4. Persamaan umum regresi linear berganda.

Normal Estimation Equation for Multiple Linear Regression dapat digunakan untuk mendapatkan nilai dari setiap $\square_i$.

$$\begin{array}{ccccccccc}
n\hat{\beta}_0 & + & \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} & + & \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} & + \cdots + & \hat{\beta}_k \sum_{i=1}^{n} x_{ik} & = & \sum_{i=1}^{n} y_i \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} & + & \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 & + & \hat{\beta}_2 \sum_{i=1}^{n} x_{i1} x_{i2} & + \cdots + & \hat{\beta}_k \sum_{i=1}^{n} x_{i1} x_{ik} & = & \sum_{i=1}^{n} x_{i1} y_i \\
\vdots & & \vdots & & \vdots & \vdots & \vdots & & \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} & + & \hat{\beta}_1 \sum_{i=1}^{n} x_{ik} x_{i1} & + & \hat{\beta}_2 \sum_{i=1}^{n} x_{ik} x_{i2} & + \cdots + & \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 & = & \sum_{i=1}^{n} x_{ik} y_i
\end{array}$$

Gambar 5. *Normal Estimation Equation for Multiple Linear Regression*.

Setelah mendapatkan persamaan linear dari persamaan di atas, gunakan metode eliminasi Gauss untuk menyelesaikannya dan mendapatkan nilai $\square_i$.

Bicubic spline interpolation adalah penggabungan dari beberapa pendekatan sehingga didapatkan intrapolasi akhir. Basisnya merupakan persamaan $y = xa$ dalam bentuk 16x16 yang merepresentasikan nilai fungsi biasa, turunan ke arah x, turunan ke arah y, dan juga turunan pada keduanya. Berdasarkan matriks ini, akan didapatkan konstanta a yang dapat digunakan untuk mencari kembali pendekatan terhadap titik yang kita inginkan.

BAB 3
Implementasi Pustaka dan Program dalam Java

Kami membuat 5 file untuk menyelesaikan tugas besar ini. Ada *file* Interpolasi.java, Main. java, Matrix.java, Regresi.java, dan SPL.java yang terletak pada berkasnya masing-masing. Kami memilih untuk memisah fungsi-fungsi untuk memudahkan kami dalam mencerna kode yang dibuat dan mencegah kebingungan karena terlalu banyak fungsi di satu *file*.

File Main.java dikelola secara keseluruhan oleh Andhika. File Main ini berfungsi sebagai pusat dari tugas besar ini dimana terdapat menu dan pemilihan program mana yang mau dijalankan. Andhika juga mengurus bagian baca dari file yang diintegrasikan pada fungsi-fungsi lainnya. Pada file Main.java, ada dua kategori fungsi, ada fungsi dekorasi template dan fungsi I/O. Pada kategori dekorasi template, ada header, inputHere, wrongInput, menu, submenuSPL, submenuDet, submenuInverse, clear, inputPrompt, dan fileNotFound. Pada kategori fungsi I/O, ada getFileNameToInput dan getFileNameToOutput. Class Main kami usahakan bebas dari variabel bertipe Matrix untuk mempermudah kami dalam menulis kode di *class* tersebut. Fungsi main adalah fungsi utama yang akan selalu dijalankan saat program ini berjalan. Fungsi ini akan selalu berjalan dalam *infinite loop* sampai *user* memilih menu keluar. Dalam fungsi ini, *user* bisa mendapatkan hasil dari operasi matriks yang dapat dipilih sesuai kebutuhan *user*.

Secara garis besar, ketika keseluruhan program dijalankan, yang akan pertama dijalankan adalah Main.java dimana kita akan diminta untuk menginput angka sesuai dengan opsi yang kita inginkan. Lalu akan dipanggil fungsi dan atau prosedur dari file lain sesuai spesifikasi input sehingga mengeluarkan keluaran yang diinginkan. Permintaan masukan akan terus mengulang sampai user mengetikkan 7 yaitu keluar.

Matrix.java berisikan fungsi dan prosedur yang berkaitan dengan matrix. Kami membuat kelas Matrix yang berisikan array of array of double. Terdapat konstruktor dan selektor serta setter matrix seperti getRow, getCol, ELMT, setELMT, setRow, dll. Semua program di atas dibentuk oleh Andhika. Terdapat pula fungsi baca meliputi readMatrix dan readSPL (Vanson) serta readDet dan readMatrixAugmented (Rafli). DisplayMatrix merupakan fungsi untuk menampilkan matriks ke layar, dibuat oleh Rafli.

Berikut juga merupakan fungsi yang dibuat oleh Rafli, DetReduksiBaris, DetEkspansiKofaktor, InverseUsingAdjoint, InverseGaussJordan, Minor, Adjoint, Kofaktor, CopyMatrix, Transpose, isIdentity. Sedangkan Vanson membuat fungsi primitif pembantu terkait matrix seperti MultiplybyConst, nPenguranganMatrix, nPenjumlahanMatrix, barisKali, barisBagi, dan perkalianMatrix. Fungsi dan prosedur tersebut berfungsi sesuai dengan namanya.

Pada SPL.java terdapat 4 program utama yaitu SPLGauss, SPLGaussJordan, SPLInverse dibuat oleh Vanson dan SPLCramer dibuat oleh Rafli. Selain 4 program utama tersebut, terdapat program pembantu seperti Gauss dan parameter (Vanson) serta GaussianElimination, Switch, dan MatrixFileReader(Andhika tetapi gagal). Gauss dan GaussianElimination merupakan prosedur yang sama yaitu OBE sampai mencapai kondisi Gauss atau baris eselon. Namun pada saat pengerjaan, kedua program selesai pada waktu yang berdeketan (bisa dibilang berbarengan setelah asistensi selesai di Upnormal dengan Kak Leon). Sehingga kedua program tersebut disimpan dan akhirnya keduanya digunakan pada fungsi yang berbeda. SPLGauss dan SPLGaussJordan serta RegresiLinear menggunakan GaussianElimination sedangkan InterpolasiPolinom dan BicubicInterpolation menggunakan Gauss.

Regresi.java berisikan prosedur RegresiLinear sebagai program utama dan displayRegresi sebagai prosedur penampil hasil regresi. Kedua prosedur ini dibuat oleh Rafli.

Interpolasi.java berisikan 2 fungsi utama yaitu InterpolasiPolinom (Andhika) dan Bicubic Interpolation (Vanson). Sedangkan program pembantu meliputi powerOf (Andhika) yang mengembalikan pangkat integer dari double dan fungsi bernama fungsi (Vanson) yang mengembalikan $x^i * y^j$ untuk membantu dalam menyusun matrix 16x16 pada BicubicInterpolation.

Penjelasan lebih lanjut terkait fungsi-fungsi utama:

- public static void SPLGauss()

SPLGauss merupakan prosedur untuk mencari solusi SPL menggunakan operasi baris elementer model Gauss (segitiga bawah bernilai 0). Tidak ada parameter. Output akan berupa matrix yang telah diinput, matrix hasil OBE, dan hasil solusi SPL nya.Kami memasukkan matrix pada tampilan dengan konsiderasi terkait kejelasan proses dan pemeriksaan bagi user maupun kami sebagai pemrogram.

- public static void SPLGaussJordan()

SPLGaussJordan merupakan prosedur untuk mencari solusi SPL menggunakan OBE seperti SPLGauss tapi sampai kondisi atas dan bawah leading one bernilai 0. Parameter dari fungsi ini sama seperti parameter pada SPLGauss.

- public static void SPLInverse()

SPLInverse merupakan prosedur yang mencari solusi SPL menggunakan inverse. Parameter sama seperti di atas. Menggunakan detEkspansiKofaktor dan inverseUsingAdjoint.

- public static void SPLCramer()

SPLCramer merupakan prosedur yang berfungsi untuk mencari solusi SPL menggunakan kaidah cramer. Parameter sama seperti di atas. Prosedur ini memanfaatkan fungsi detEkspansiKofaktor untuk mencari determinan SPL. Prosedur ini hanya dapat menerima *input* n baris dan n+1 kolom (kolom ditambah 1 untuk pengisian hasil persamaan linier). Jika *input* baris dan kolom tidak sesuai format, maka program akan langsung berhenti tanpa perlu menerima *input* SPL dan mengeluarkan pesan bahwa Kaidah Cramer tidak dapat dilakukan.

- public static void InterpolasiPolinom()

InterpolasiPolinom merupakan prosedur dengan output hasil dari interpolasi sesuai spek. Parameter sama seperti di atas. Menggunakan prosedur OBE Gauss dan fungsi powerOf.

- public static void BicubicInterpolation()

BicubicInterpolation merupakan prosedur dengan output hasil dari interpolasi sesuai spek. Parameter sama seperti di atas. Memanfaatkan prosedur OBE Gauss dan fungsi fungsi.

- public static void RegresiLinear()

RegresiLinear merupakan prosedur untuk mencari persamaan regresi linear dari data yang diketahui dan juga menampilkan hasil dari nilai yang ingin ditaksir. Prosedur ini memanfaatkan GaussianElimination() untuk mencari koefisien dan konstanta dari persamaan regresi. Prosedur ini juga memanfaatkan displayRegresi() untuk menampilkan hasil perhitungan ke layar.

Semua prosedur di atas dapat dipanggil dengan matrix.<*prosedur*> dimana matrix adalah variabel bertipe bentukan Matrix. Contoh : matrix.SPLGauss() dan matrix.BicubicInterpolation().

Penjelasan terkait parameter solusi banyak pada SPLGauss dan SPLGaussJordan :

Variabel yang dimisalkan akan dimisalkan menggunakan nama variabel itu sendiri. Contoh : X4 = X4, sehingga nanti pada pengaplikasiannya contoh tampillannya adalah seperti ini :

X1 = 1.0 - 2.0 X2 - 3.0 X3 - 4.0 X4
X2 = X2
X3 = X3
X4 = X4

Permintaan input matrix atau SPL pada beberapa fungsi berupa matrix augmented, namun pada beberapa program tetap ditampilkan panduan seperti “Masukkan baris 1 kolom 1 : ” dengan alasan *clarity* atau kejelasan program. Namun pada pengaplikasiannya, walaupun terdapat tampilan seperti itu, user tidak diwajibkan memasukkan input satu per satu sesuai perintah. User dapat langsung men-paste keseluruhan input walau pesan perintah akan tertumpuk di atas.

Dikarenakan suatu bug dan permasalahan menjelang deadline, kami memutuskan untuk tidak mengimplementasikan *readFile* dan *saveFile* dengan kekhawatiran kode yang tidak beres akan mengganggu kode lainnya. Dalam hasil akhir yang kami kumpulkan, masih ada sisa tulang belulang readfile yang tentunya tidak akan mengganggu performa program, dan hanyalah sebagai tampilan saja. Bekas-bekas kode tersebut tidak dihapus karena faktor waktu dan sudah terintegrasi dengan banyak fungsi lainnya sehingga ditakutkan akan berdampak bagi kode lainnya.

BAB 4
Eksperimen

1. Studi Kasus 1: SPL

   1.a

   Menggunakan SPLGauss :

```
SPL yang anda input :
1.000000 1.000000 -1.000000 -1.000000 1.000000
2.000000 5.000000 -7.000000 -5.000000 -2.000000
2.000000 -1.000000 1.000000 3.000000 4.000000
5.000000 2.000000 -4.000000 2.000000 6.000000
Hasil OBE Gauss :
1.000000 1.000000 -1.000000 -1.000000 1.000000
0.000000 1.000000 -0.333333 -2.333333 -0.333333
0.000000 0.000000 1.000000 -1.000000 0.750000
0.000000 0.000000 0.000000 0.000000 -0.500000
SPL tidak memiliki solusi!
```

   Menggunakan SPLGaussJordan :

```
SPL yang anda input :
1.000000 1.000000 -1.000000 -1.000000 1.000000
2.000000 5.000000 -7.000000 -5.000000 -2.000000
2.000000 -1.000000 1.000000 3.000000 4.000000
5.000000 2.000000 -4.000000 2.000000 6.000000
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 0.666667 1.833333
0.000000 1.000000 0.000000 -2.666667 -0.083333
0.000000 0.000000 1.000000 -1.000000 0.750000
0.000000 0.000000 0.000000 0.000000 -0.500000
SPL tidak memiliki solusi!
```

   Menggunakan SPLInverse :

```
SPL tidak bisa diselesaikan dengan metode inverse karena determinan = 0!
```

Menggunakan SPLCramer :

```
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Baris 1:1 1 -1 -1 1
Baris 2:2 5 -7 -5 -2
Baris 3:2 -1 1 3 4
Baris 4:5 2 -4 2 6

Kaidah Cramer gagal karena determinan matriks bernilai 0
```

Kaidah Cramer tidak bisa dilakukan karena determinan SPL tersebut bernilai 0. Begitu juga dengan SPLInverse.

1.b

Menggunakan SPLGauss :

```
SPL yang anda input :
1.000000 -1.000000 0.000000 0.000000 1.000000 3.000000
1.000000 1.000000 0.000000 -3.000000 0.000000 6.000000
2.000000 -1.000000 0.000000 1.000000 -1.000000 5.000000
-1.000000 2.000000 0.000000 -2.000000 -1.000000 -1.000000
Hasil OBE Gauss :
1.000000 -2.000000 0.000000 2.000000 1.000000 1.000000
0.000000 1.000000 0.000000 -1.666667 -0.333333 1.666667
0.000000 0.000000 0.000000 1.000000 -1.000000 -1.000000
0.000000 0.000000 0.000000 0.000000 1.000000 -0.000000
SPL memiliki solusi banyak.
X1 = 3.0
X2 = 0.0
X3 = X3
X4 = -1.0
X5 = 0.0
```

Menggunakan SPLGaussJordan :

```
SPL yang anda input :
1.000000 -1.000000 0.000000 0.000000 1.000000 3.000000
1.000000 1.000000 0.000000 -3.000000 0.000000 6.000000
2.000000 -1.000000 0.000000 1.000000 -1.000000 5.000000
-1.000000 2.000000 0.000000 -2.000000 -1.000000 -1.000000
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 0.000000 0.000000 3.000000
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 1.000000 0.000000 -1.000000
0.000000 0.000000 0.000000 0.000000 1.000000 -0.000000
SPL memiliki solusi banyak.
X1 = 3.0
X2 = 0.0
X3 = X3
X4 = -1.0
X5 = 0.0
```

Menggunakan SPLInverse :

```
SPL tidak bisa diselesaikan dengan metode inverse karena bukan matrix persegi!
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 6

Kaidah Cramer gagal karena matriks tidak memiliki determinan
Pastikan jumlah kolom = jumlah baris + 1
```

Program tidak dapat memproses SPL menggunakan kaidah Cramer karena ukuran SPL 4x6. Dengan ukuran tersebut, SPL tidak akan memiliki determinan sehingga kaidah Cramer tidak bisa dilakukan. Begitu juga dengan SPLInverse.

1.c

Menggunakan SPLGauss :

```
SPL yang anda input :
0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 2.000000
0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 -1.000000
0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000
Hasil OBE Gauss :
0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 2.000000
0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 -1.000000
0.000000 0.000000 0.000000 0.000000 1.000000 -1.000000 1.000000
SPL memiliki solusi banyak.
X1 = 0.0
X2 = 1.0 - X6
X3 = X3
X4 = -2.0 - X6
X5 = 1.0 + X6
X6 = X6
```

Menggunakan SPLGaussJordan :

```
SPL yang anda input :
0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 2.000000
0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 -1.000000
0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000
Hasil OBE Gauss-Jordan :
0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 1.000000 0.000000 1.000000 -2.000000
0.000000 0.000000 0.000000 0.000000 1.000000 -1.000000 1.000000
SPL memiliki solusi banyak.
X1 = 0.0
X2 = 1.0 - X6
X3 = X3
X4 = -2.0 - X6
X5 = 1.0 + X6
X6 = X6
```

Menggunakan SPLInverse :

```
SPL tidak bisa diselesaikan dengan metode inverse karena bukan matrix persegi!
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 3
Masukkan jumlah kolom: 7

Kaidah Cramer gagal karena matriks tidak memiliki determinan
Pastikan jumlah kolom = jumlah baris + 1
```

Program tidak dapat memproses SPL menggunakan kaidah Cramer karena ukuran SPL 3x7. Dengan ukuran tersebut, SPL tidak akan memiliki determinan sehingga kaidah Cramer tidak bisa dilakukan. Begitu juga dengan SPLInverse.

1.d
Untuk n = 6 :
Menggunakan SPLGauss :

```
SPL yang anda input :
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 1.000000
0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0.000000
0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.000000
0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.000000
0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 0.000000
0.166667 0.142857 0.125000 0.111111 0.100000 0.090909 0.000000
Hasil OBE Gauss :
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 1.000000
0.000000 1.000000 1.000000 0.900000 0.800000 0.714286 -6.000000
0.000000 0.000000 1.000000 1.500000 1.714286 1.785714 30.000000
0.000000 0.000000 0.000000 1.000000 2.000000 2.777778 -140.000000
0.000000 0.000000 0.000000 0.000000 1.000000 2.500000 630.000000
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 -2772.000000
X1 = 36.000
X2 = -630.000
X3 = 3360.000
X4 = -7560.000
X5 = 7560.000
X6 = -2772.000
```

Menggunakan SPLGaussJordan:

```
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 36.000000
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 -630.000000
0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 3360.000000
0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 -7560.000001
0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 7560.000001
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 -2772.000000
X1 = 36.000
X2 = -630.000
X3 = 3360.000
X4 = -7560.000
X5 = 7560.000
X6 = -2772.000
```

Menggunakan SPLInverse:

```
X1 = 36.000
X2 = -630.000
X3 = 3360.000
X4 = -7560.000
X5 = 7560.000
X6 = -2772.000
```

Menggunakan SPLCramer:

```
Solusi:
x1 = 35.999998
x2 = -629.999950
x3 = 3359.999710
x4 = -7559.999411
x5 = 7559.999516
x6 = -2771.999864
```

Untuk n = 10:

Menggunakan SPLGauss:

```
Hasil OBE Gauss :
1.000000 0.500000 0.333333 0.250000 0.200000 0.166667 0.142857 0.125000 0.111111 0.100000 1.000000
0.000000 1.000000 1.000000 0.900000 0.800000 0.714286 0.642857 0.583333 0.533333 0.490909 -6.000000
0.000000 0.000000 1.000000 1.500000 1.714286 1.785714 1.785714 1.750000 1.696970 1.636364 30.000000
0.000000 0.000000 0.000000 1.000000 2.000000 2.777778 3.333333 3.712121 3.959596 4.111888 -140.000000
0.000000 0.000000 0.000000 0.000000 1.000000 2.500000 4.090909 5.568182 6.853147 7.930070 630.000000
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 3.000000 5.653846 8.615385 11.630769 -2772.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 3.500000 7.466667 12.600000 12012.000054
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 4.000000 9.529412 -51480.001129
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 4.499994 218789.127760
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 -923630.451075
X1 = 99.996
X2 = -4949.683
X3 = 79193.220
X4 = -600538.194
X5 = 2522224.572
X6 = -6305486.499
X7 = 9608263.465
X8 = -8750306.935
X9 = 4375120.512
X10 = -923630.451
```

Menggunakan SPLGaussJordan:

```
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 99.996347
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 -4949.682803
0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 79193.219783
0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 -600538.193870
0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 2522224.572291
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 -6305486.499451
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 9608263.465044
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 -8750306.935234
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 4375120.511596
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 -923630.451075
X1 = 99.996
X2 = -4949.683
X3 = 79193.220
X4 = -600538.194
X5 = 2522224.572
X6 = -6305486.499
X7 = 9608263.465
X8 = -8750306.935
X9 = 4375120.512
X10 = -923630.451
```

Menggunakan SPLInverse:

```
X1 = 99.997
X2 = -4949.750
X3 = 79194.641
X4 = -600551.105
X5 = 2522286.119
X6 = -6305655.592
X7 = 9608540.728
X8 = -8750574.721
X9 = 4375261.017
X10 = -923661.334
```

Menggunakan SPLCramer:

```
Solusi:
x1 = 34.590178
x2 = -381.540248
x3 = 0.375622
x4 = 5698.242961
x5 = -9412.043303
x6 = -3176.675595
x7 = 7577.607165
x8 = 5393.857417
x9 = -2806.710061
x10 = -3005.422213
```

Semua jawaban sudah benar dan paramater juga sudah tepat.

2. Studi Kasus 2: SPL

2.a :

Menggunakan SPLGauss :

```
SPL yang anda input :
1.000000 -1.000000 2.000000 -1.000000 -1.000000
2.000000 1.000000 -2.000000 -2.000000 -2.000000
-1.000000 2.000000 -4.000000 1.000000 1.000000
3.000000 0.000000 0.000000 -3.000000 -3.000000
Hasil OBE Gauss :
1.000000 -2.000000 4.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
SPL memiliki solusi banyak.
X1 = -1.0 + X4
X2 = 2.0 X3
X3 = X3
X4 = X4
```

Menggunakan SPLGaussJordan :

```
SPL yang anda input :
1.000000 -1.000000 2.000000 -1.000000 -1.000000
2.000000 1.000000 -2.000000 -2.000000 -2.000000
-1.000000 2.000000 -4.000000 1.000000 1.000000
3.000000 0.000000 0.000000 -3.000000 -3.000000
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 -1.000000 -1.000000
0.000000 1.000000 -2.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
SPL memiliki solusi banyak.
X1 = -1.0 + X4
X2 = 2.0 X3
X3 = X3
X4 = X4
```

Menggunakan SPLInverse :

```
SPL tidak bisa diselesaikan dengan metode inverse karena determinan = 0!
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Baris 1:1 -1 2 -1 -1
Baris 2:2 1 -2 -2 -2
Baris 3:-1 2 -4 1 1
Baris 4:3 0 0 -3 -3

Kaidah Cramer gagal karena determinan matriks bernilai 0
```

Kaidah Cramer tidak bisa dilakukan karena determinan SPL tersebut bernilai 0. Begitu juga dengan SPLInverse.

2.b

Menggunakan SPLGauss :

```
SPL yang anda input :
2.000000 0.000000 8.000000 0.000000 8.000000
0.000000 1.000000 0.000000 4.000000 6.000000
-4.000000 0.000000 6.000000 0.000000 6.000000
0.000000 -2.000000 0.000000 3.000000 -1.000000
2.000000 0.000000 -4.000000 0.000000 -4.000000
0.000000 1.000000 0.000000 -2.000000 0.000000
Hasil OBE Gauss :
1.000000 0.000000 -1.500000 0.000000 -1.500000
0.000000 1.000000 0.000000 -1.500000 0.500000
0.000000 0.000000 1.000000 0.000000 1.000000
0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
X1 = 0.000
X2 = 2.000
X3 = 1.000
X4 = 1.000
```

Menggunakan SPLGaussJordan :

```
SPL yang anda input :
2.000000 0.000000 8.000000 0.000000 8.000000
0.000000 1.000000 0.000000 4.000000 6.000000
-4.000000 0.000000 6.000000 0.000000 6.000000
0.000000 -2.000000 0.000000 3.000000 -1.000000
2.000000 0.000000 -4.000000 0.000000 -4.000000
0.000000 1.000000 0.000000 -2.000000 0.000000
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 1.000000 0.000000 0.000000 2.000000
0.000000 0.000000 1.000000 0.000000 1.000000
0.000000 0.000000 0.000000 1.000000 1.000000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000
X1 = 0.000
X2 = 2.000
X3 = 1.000
X4 = 1.000
```

Menggunakan SPLInverse :

```
SPL tidak bisa diselesaikan dengan metode inverse karena bukan matrix persegi!
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 6
Masukkan jumlah kolom: 5

Kaidah Cramer gagal karena matriks tidak memiliki determinan
Pastikan jumlah kolom = jumlah baris + 1
```

Program tidak dapat memproses SPL menggunakan kaidah Cramer karena ukuran SPL 6 x5. Dengan ukuran tersebut, SPL tidak akan memiliki determinan sehingga kaidah Cramer tidak bisa dilakukan. Begitu juga dengan SPLInverse.

Semua solusi sudah benar dan parameter sudah tepat.

3. Studi Kasus 3: SPL

   3.a

   Menggunakan SPLGauss :

```
SPL yang anda input :
8.000000 1.000000 3.000000 2.000000 0.000000
2.000000 9.000000 -1.000000 -2.000000 1.000000
1.000000 3.000000 2.000000 -1.000000 2.000000
1.000000 0.000000 6.000000 4.000000 3.000000
Hasil OBE Gauss :
1.000000 0.125000 0.375000 0.250000 0.000000
0.000000 1.000000 -45.000000 -30.000000 -24.000000
0.000000 0.000000 1.000000 0.648855 0.541985
0.000000 0.000000 0.000000 1.000000 -0.258108
X1 = -0.224
X2 = 0.182
X3 = 0.709
X4 = -0.258
```

Menggunakan SPLGaussJordan :

```
SPL yang anda input :
8.000000 1.000000 3.000000 2.000000 0.000000
2.000000 9.000000 -1.000000 -2.000000 1.000000
1.000000 3.000000 2.000000 -1.000000 2.000000
1.000000 0.000000 6.000000 4.000000 3.000000
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 0.000000 -0.224324
0.000000 1.000000 0.000000 0.000000 0.182432
0.000000 0.000000 1.000000 0.000000 0.709459
0.000000 0.000000 0.000000 1.000000 -0.258108
X1 = -0.224
X2 = 0.182
X3 = 0.709
X4 = -0.258
```

Menggunakan SPLInverse :

```
X1 = -0.224
X2 = 0.182
X3 = 0.709
X4 = -0.258
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 4
Masukkan jumlah kolom: 5
Baris 1:8 1 3 2 0
Baris 2:2 9 -1 -2 1
Baris 3:1 3 2 -1 2
Baris 4:1 0 6 4 3
Solusi:
x1 = -0.224324
x2 = 0.182432
x3 = 0.709459
x4 = -0.258108
```

3.b

Menggunakan SPLGauss :

```
Hasil OBE Gauss :
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 13.000000
1.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 -1.000000 -1.000000 -7.000000
1.000000 0.000000 0.000000 17.486594 1.000000 0.000000 0.000000 0.000000 -17.486594 -44.153882
0.000000 0.000000 1.000000 0.000000 1.000000 17.486594 0.000000 0.000000 0.000000 224.438396
0.000000 1.000000 3.656840 1.000000 3.656840 1.000000 0.000000 0.000000 0.000000 0.000000
1.000000 1.221578 0.069858 1.221578 0.069858 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 0.000000 0.000000 0.000000 0.000000
1.000000 17.486594 14.314759 0.000000 1.000000 17.486594 0.000000 0.000000 0.000000 0.000000
1.000000 0.273460 0.000000 0.273460 1.000000 0.273460 0.000000 0.000000 0.000000 0.000000
1.000000 1.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN
X5 = NaN
X6 = NaN
X7 = NaN
X8 = NaN
X9 = NaN
```

Menggunakan SPLGaussJordan :

```
Hasil OBE Gauss-Jordan :
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 13.000000
0.000000 -18761369254933147000000000.000000 0.000000 0.000000 0.000000 -11658231046227561000000000.000000 0.0000
00 -1.000000 23951762835327.710000 -16960671270141466000000000.000000
0.000000 1369721477342.997800 0.000000 0.000000 0.000000 8511388073472.315000 0.000000 0.000000 -17.486594 1238
25694133484.450000
0.000000 0.000000 0.000000 0.000000 1.000000 16.486594 0.000000 0.000000 0.000000 224.438396
0.000000 -1170350.506042 0.000000 0.000000 471408.377471 499412.580736 0.000000 0.000000 0.000000 0.000000
0.000000 -508.738765 0.000000 0.000000 204.912732 217.090290 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 -47.419004 -50.453222 0.000000 19.227704 -30.191301 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 0.000000 0.000000 0.000000 0.000000
0.000000 28.464761 29.801352 -4.508426 -15.486594 12.978167 0.000000 0.000000 0.000000 0.000000
0.000000 -0.726540 -1.000000 0.273460 1.000000 0.273460 0.000000 0.000000 0.000000 0.000000
1.000000 1.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
X1 = NaN
X2 = NaN
X3 = NaN
X4 = NaN
X5 = NaN
X6 = NaN
X7 = NaN
X8 = NaN
X9 = NaN
```

Mengguanakan SPLInverse :

```
SPL tidak bisa diselesaikan dengan metode inverse karena bukan matrix persegi!
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 12
Masukkan jumlah kolom: 10

Kaidah Cramer gagal karena matriks tidak memiliki determinan
Pastikan jumlah kolom = jumlah baris + 1
```

Program tidak dapat memproses SPL menggunakan kaidah Cramer karena ukuran SPL 12x10. Dengan ukuran tersebut, SPL tidak akan memiliki determinan sehingga kaidah Cramer tidak bisa dilakukan. Begitu juga dengan SPLInverse.

4. Studi Kasus 4: SPL
   Persamaan yang terdapat dalam testcase tidaklah sesuai untuk langsung digunakan pada program. Maka persamaan tersebut harus dikondisikan agar sesuai untuk program.

   Berikut adalah hasil penyesuaian input terurut dari Xa, Xb, Xc, dan solusi :
   -120 60 0 -1300
   40 -80 0 0
   80 20 -150 -200

   Menggunakan SPLGauss :

```
SPL yang anda input :
-120.000000 60.000000 0.000000 -1300.000000
40.000000 -80.000000 0.000000 0.000000
80.000000 20.000000 -150.000000 -200.000000
Hasil OBE Gauss :
1.000000 -0.500000 0.000000 10.833333
0.000000 1.000000 0.000000 7.222222
0.000000 0.000000 1.000000 10.000000
X1 = 14.444
X2 = 7.222
X3 = 10.000
```

Menggunakan SPLGaussJordan :

```
SPL yang anda input :
-120.000000 60.000000 0.000000 -1300.000000
40.000000 -80.000000 0.000000 0.000000
80.000000 20.000000 -150.000000 -200.000000
Hasil OBE Gauss-Jordan :
1.000000 0.000000 0.000000 14.444444
0.000000 1.000000 0.000000 7.222222
0.000000 0.000000 1.000000 10.000000
X1 = 14.444
X2 = 7.222
X3 = 10.000
```

Menggunakan SPLInverse :

```
X1 = 14.444
X2 = 7.222
X3 = 10.000
```

Menggunakan SPLCramer:

```
Masukkan jumlah baris: 3
Masukkan jumlah kolom: 4
Baris 1:-120 60 0 -1300
Baris 2:40 -80 0 0
Baris 3:80 20 -150 -200
Solusi:
x1 = 14.444444
x2 = 7.222222
x3 = 10.000000
```

Jadi didapatkan nilai X1, X2, dan X3 sesuai yang tertera di atas.

5. Studi Kasus 5: Interpolasi Polinomial

a. Mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel (orde 6)

```
Masukkan orde: 6
Baris 1:0.1 0.003
Baris 2:0.3 0.067
Baris 3:0.5 0.148
Baris 4:0.7 0.248
Baris 5:0.9 0.370
Baris 6:1.1 0.518
Baris 7:1.3 0.697
Masukkan nilai x yang ingin ditaksir nilainya: 0.2

f(x) = -0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x -0.0230, f(0.2000) = 0.0330
```

```
Masukkan orde: 6
Baris 1:0.1 0.003
Baris 2:0.3 0.067
Baris 3:0.5 0.148
Baris 4:0.7 0.248
Baris 5:0.9 0.370
Baris 6:1.1 0.518
Baris 7:1.3 0.697
Masukkan nilai x yang ingin ditaksir nilainya: 0.55

f(x) = -0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x -0.0230, f(0.5500) = 0.1711
```

```
Masukkan orde: 6
Baris 1:0.1 0.003
Baris 2:0.3 0.067
Baris 3:0.5 0.148
Baris 4:0.7 0.248
Baris 5:0.9 0.370
Baris 6:1.1 0.518
Baris 7:1.3 0.697
Masukkan nilai x yang ingin ditaksir nilainya: 0.85

f(x) = -0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x -0.0230, f(0.8500) = 0.3372
```

```
Masukkan orde: 6
Baris 1:0.1 0.003
Baris 2:0.3 0.067
Baris 3:0.5 0.148
Baris 4:0.7 0.248
Baris 5:0.9 0.370
Baris 6:1.1 0.518
Baris 7:1.3 0.697
Masukkan nilai x yang ingin ditaksir nilainya: 1.28

f(x) = -0.0000x^6 + 0.0000x^5 + 0.0260x^4 + 0.0000x^3 + 0.1974x^2 + 0.2400x -0.0230, f(1.2800) = 0.6775
```

b. Jumlah kasus positif baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022

```
Masukkan orde: 9
Baris 1:6.567 12.624
Baris 2:7 21.807
Baris 3:7.258 38.391
Baris 4:7.451 54.517
Baris 5:7.548 51.952
Baris 6:7.839 28.228
Baris 7:8.161 35.764
Baris 8:8.484 20.813
Baris 9:8.709 12.408
Baris 10:9 10.534
Masukkan nilai x yang ingin ditaksir nilainya: 7.5161290323

f(x) = -141.1156x^9 + 9381.4246x^8 -275742.3669x^7 + 4700679.8027x^6 -51188814.4540x^5 + 368993729.1796x^4 -1759104325.7967x^3 +
 5341832292.4416x^2 -9361774336.2444x + 7199777932.7774, f(7.5161) = 53.5327
```

```
Masukkan orde: 9
Baris 1:6.567 12.624
Baris 2:7 21.807
Baris 3:7.258 38.391
Baris 4:7.451 54.517
Baris 5:7.548 51.952
Baris 6:7.839 28.228
Baris 7:8.161 35.764
Baris 8:8.484 20.813
Baris 9:8.709 12.408
Baris 10:9 10.534
Masukkan nilai x yang ingin ditaksir nilainya: 8.3225806452

f(x) = -141.1156x^9 + 9381.4246x^8 -275742.3669x^7 + 4700679.8027x^6 -51188814.4540x^5 + 368993729.1796x^4 -1759104325.7967x^3 +
 5341832292.4416x^2 -9361774336.2444x + 7199777932.7774, f(8.3226) = 36.3162
```

```
Masukkan orde: 9
Baris 1:6.567 12.624
Baris 2:7 21.807
Baris 3:7.258 38.391
Baris 4:7.451 54.517
Baris 5:7.548 51.952
Baris 6:7.839 28.228
Baris 7:8.161 35.764
Baris 8:8.484 20.813
Baris 9:8.709 12.408
Baris 10:


9 10.534
Masukkan nilai x yang ingin ditaksir nilainya: 9.1666666667

f(x) = -141.1156x^9 + 9381.4246x^8 -275742.3669x^7 + 4700679.8027x^6 -51188814.4540x^5 + 368993729.1796x^4 -1759104325.7967x^3 +
 5341832292.4416x^2 -9361774336.2444x + 7199777932.7774, f(9.1667) = -664.9637
```

```
Masukkan orde: 9
Baris 1:6.567 12.624
Baris 2:7 21.807
Baris 3:7.258 38.391
Baris 4:7.451 54.517
Baris 5:7.548 51.952
Baris 6:7.839 28.228
Baris 7:8.161 35.764
Baris 8:8.484 20.813
Baris 9:8.709 12.408
Baris 10:9 10.534
Masukkan nilai x yang ingin ditaksir nilainya: 13

f(x) = -141.1156x^9 + 9381.4246x^8 -275742.3669x^7 + 4700679.8027x^6 -51188814.4540x^5 + 368993729.1796x^4 -1759104325.7967x^3 +
 5341832292.4416x^2 -9361774336.2444x + 7199777932.7774, f(13.0000) = -589417575.9429
```

(Tanggal 12/31/2022)

c. Selesaikan,

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

dengan polinom interpolasi derajat n di dalam selang [0, 2].
Jika n = 4, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak h = (2 – 0)/4 = 0.5.

```
Masukkan orde: 4
Baris 1:0 2.41421356
Baris 2:0.5 1.12355827
Baris 3:1 0.6492820269553
Baris 4:1.5 0.4036006442302
Baris 5:2 0.2571305929941
Masukkan nilai x yang ingin ditaksir nilainya: 0.5

f(x) = 0.3056x^4 -1.7005x^3 + 3.6487x^2 -4.0187x + 2.4142, f(0.5000) = 1.1236
```

6. Studi Kasus 6: Regresi Linear Berganda

```
Masukkan jumlah peubah x: 3
Masukkan jumlah sampel: 20
Silakan masukkan sampel di kolom terakhir!
Masukkan peubah x dan sampel:
72.4 76.3 29.18 0.9
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68 29.27 0.89
10.7 79 29.78 1
12.9 67.4 29.39 1.1
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24 67.7 29.6 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.1
10.6 86.3 29.56 1.1
11.2 86 29.48 1.1
73.3 76.3 29.4 0.91
75.4 77.9 29.28 0.87
96.6 78.7 29.29 0.78
107.4 86.8 29.03 0.82
54.9 70.9 29.37 0.95

Masukkan nilai x yang ingin ditaksir: 50 76 29.30

f(x1,x2,x3) = -3.507778 - 0.002625 X1 + 0.000799 X2 + 0.154155 X3
f(50.000000,76.000000,29.300000) = 0.938434
```

Pada studi kasus regresi linear berganda, ada 3 buah nilai x sehingga kita *input* 3 pada jumlah peubah. Kemudian, kita *input* jumlah sampel y sebanyak 20 buah sesuai dengan soal. Setelah itu, akan terbentuk sebuah matriks dengan ukuran kolom 4 (kolom berjumlah 4 untuk menampung 3 peubah dan 1 sampel) dan ukuran baris 20. Selanjutnya, program akan menerima *input* sebuah matriks augmented yang merepresentasikan data peubah dan sampel dengan urutan x_1, x_2, x_3, y. Setelah selesai menerima *input* data peubah dan sampel, program dilanjutkan dengan menerima nilai x yang ingin ditaksir. Nilai tersebut ditulis secara berurutan (x_1, x_2, x_3) dalam 1 baris. Selanjutnya, program akan menampilkan fungsi hasil regresi dan hasil dari nilai yang ingin ditaksir.

7. Studi Kasus 7: Interpolasi Bicubic Spline
   Berikut adalah hasil dari test case setelah digabung :

```
f(0.00,0.00) = 21.0000
f(0.50,0.50) = 87.7969
f(0.25,0.75) = 117.7322
f(0.10,0.90) = 128.5752
```

Matriks 16x16 yang dibuat di dalam program (tidak hardcode) sudah sama persis dengan spesifikasi. Metode yang digunakan untuk mendapatkan matriks a adalah dengan Gauss x dengan input sebagai hasilnya. Matriks a juga sudah dicek dengan kalkulator Gauss dengan hasil yang sama. Berikutnya, pendekatan nilai *f*(x,y) dilakukan dengan iterasi sederhana memanfaatkan matriks a yang sudah didapat. Asumsi iterasi tersebut benar, maka jawaban di atas sudah benar. Hasil yang dikeluarkan ke layar dibatasi hanya 4 angka di belakang koma untuk alasan readability. Mungkin terdapat ketidakakuratan hasil namun akan sangat kecil.

BAB 5
Kesimpulan, Saran, Komentar, dan Refleksi

Kesimpulan dari tugas besar ini adalah kami dapat mengaplikasikan operasi matrix dan cara manipulasi matrix yang sudah diajarkan di kelas Aljabar Linier dan Geometri. Kami berhasil membuat sebuah pustaka pada bahasa Java yang bisa digunakan pada program utama. Kami juga berhasil membungkus fungsi-fungsi dan program utama yang telah kami buat ke dalam *file* JAR. Selain memanfaatkan pustaka ini untuk menyelesaikan permasalahan sistem persamaan linear dengan matriks, pustaka ini juga dapat menyelesaikan persoalan interpolasi polinomial, regresi linear berganda, dan interpolasi spline bikubik.

Kami dari kelompok kramer memiliki beberapa saran untuk tugas besar ini dan juga untuk tugas besar kami di masa yang akan datang. Kami memberi saran agar kami diberikan lebih banyak sumber belajar untuk materi yang belum pernah kami pelajari. Berdasarkan beberapa faktor, memiliki sumber referensi yang jelas untuk mempelajari materi-materi baru ini tentunya akan sangat membantu kami. Selain itu, kami merasa kasus uji perlu diperbanyak dengan alasan kuantitas.

Komentar yang ingin kami berikan tidaklah banyak. Kami ingin mengapresiasi asisten kami, Kak Leon yang selalu siap sedia dalam menjawab pertanyaan. Sistem asistensi pada tubes ini juga bersifat santai, sehingga kami juga tidak merasa terbebani.

Dari tugas besar ini, kami juga dapat belajar bahasa Java dan tidak terpaku pada bahasa yang selama ini kami gunakan. Kami dapat menemukan batas kemampuan masing-masing dan melewatinya bersama-sama dengan berbagi perspektif dan ide-ide unik yang muncul saat kami berdialog dengan rekan-rekan seangkatans. Kami juga dipaksa untuk mencari sumber belajar lain dan belajar secara mandiri karena ada materi atau persoalan yang belum pernah kami pelajari sebelumnya. Di tugas besar ini, pengaturan waktu juga diuji karena di setiap mata kuliah prodi IF, ada tugas-tugas, kuis, praktikum, dan ada juga kegiatan-kegiatan di dalam maupun luar kampus yang sangat menyita waktu kami. Kami juga belajar pengorbanan dalam bentuk waktu, tenaga, dan uang.

Daftar Pustaka

1. Java Syntax (w3schools.com)
2. Spesifikasi Tugas Besar 1 IF2123 Algeo 2023/2024
3. Slide Bahan Kuliah IF2123 Aljabar Geometri - Semester I Tahun 2023/2024
4. Create a JAR File Containing the Class File (The Java™ Tutorials > Security Features in Java SE > Signing Code and Granting It Permissions) (oracle.com)
5. Working with Manifest Files: The Basics (The Java™ Tutorials > Deployment > Packaging Programs in JAR Files) (oracle.com)
6. Regular Expression (Regex) Tutorial (ntu.edu.sg)
7. Regex expressions in Java, \\s vs. \\s+ - Stack Overflow