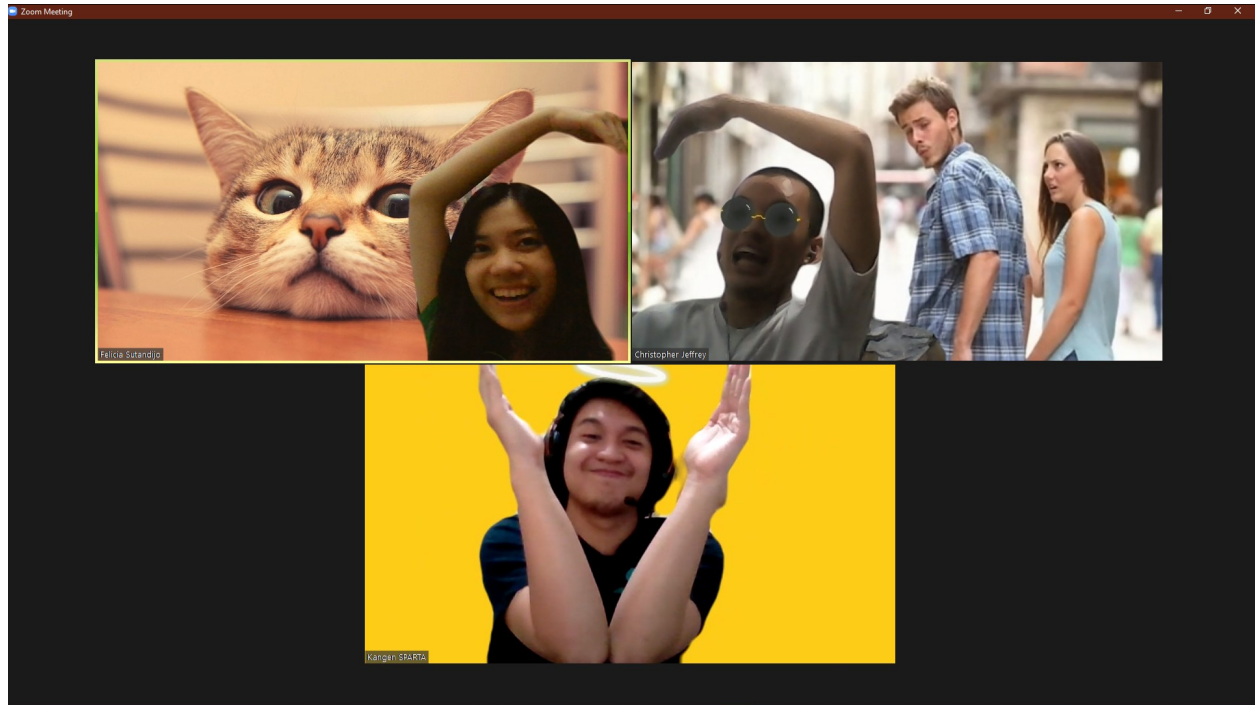


Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri

Sistem Persamaan Linier, Determinan, dan Aplikasinya



Anggota Kelompok "2 bois 1 girl":

Aditya Prawira N 13520049

Felicia Sutandijo 13520050

Christopher Jeffrey 13520055

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2021

Bab 1

Deskripsi masalah

A. Gambaran Umum

Peserta IF 2123 diminta untuk membuat sebuah 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. Spesifikasi Detail

Berikut spesifikasi yang lebih detail dari gambaran umum mengenai pustaka yang harus dibuat(nomor hanya untuk memudahkan pemahaman, tidak mengikuti ukuran tertentu):

1. Program dapat menerima masukan (input) baik dari *keyboard* maupun membaca masukan dari file text. Untuk SPL, masukan dari *keyboard* adalah m , n , koefisien a_{ij} , dan b_j . 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 a_{ij} . Masukan dari file berbentuk matriks, setiap elemen matriks dipisah oleh spasi.

Misalnya,

3	4.5	2.8	10
-3	7	8.3	11
0.5	-10	-9	12

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n , (x_0, y_0) , (x_1, y_1) , ..., (x_n, y_n) , 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), semua nilai-nilai $x_{1i}, x_{2i}, \dots, x_{ni}$, nilai y_i , dan nilai-nilai x_k 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 $x_4 = -2$, $x_3 = 2s - t$, $x_2 = s$, dan $x_1 = t$.)
6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing
7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.
8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.
9. Bahasa program yang digunakan adalah Java.
10. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).
11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing.

Misalnya, menu:

MENU

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

Teori singkat

A. Metode Eliminasi Gauss

Eliminasi Gauss adalah salah satu metode yang ditemukan oleh Carl Friedrich Gauss untuk menyelesaikan sistem persamaan linier banyak variabel. Metode ini mengubah persamaan linier menjadi bentuk baik matriks maupun matriks *augmented*. Setelah terbentuk matriks, dilakukan OBE atau Operasi Baris Elementer, yaitu sebuah operasi yang dilakukan untuk memanipulasi matriks dengan mengalikan sebuah baris dengan konstanta tidak nol, menukar dua buah baris, dan menambahkan sebuah baris dengan kelipatan baris lainnya. OBE dilakukan hingga membentuk matriks eselon baris, yaitu matriks yang memiliki *leading one* pada setiap baris, kecuali baris yang seluruhnya nol. Setelah menjadi matriks eselon, dilakukan sulih balik (substitusi balik) untuk mendapatkan solusi. Dari matriks eselon yang terbentuk, ada tiga kemungkinan solusi, yaitu memiliki tepat satu solusi, memiliki solusi banyak, dan tidak memiliki solusi sama sekali. Matriks dikatakan memiliki tepat satu solusi apabila dua kolom terakhir tidak bernilai nol. Jika ada baris yang semua elemennya bernilai nol kecuali elemen di kolom terakhir, maka matriks tidak memiliki solusi. Jika ada baris yang seluruh elemennya bernilai nol, dan jumlah baris kurang dari jumlah kolom dikurangi satu, maka solusinya tak hingga dan bentuk solusi akhir setelah dilakukan penyulihan balik berbentuk parametrik.

B. Metode Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan adalah metode mengubah isi dari suatu matriks dengan memanfaatkan *Operasi Baris Elementer* (OBE) hingga matriks menjadi berbentuk *eselon tereduksi*. Metode ini mirip dengan metode Gauss, dengan tambahan beberapa langkah akhir, bisa kita bayangkan sebagai extension atau step tambahan dari metode eliminasi Gauss. Bentuk eselon tereduksi sendiri itu mirip dengan eselon, perbedaannya adalah untuk eselon tereduksi, setiap kolom yang memiliki 1 utama memiliki nol di tempat lain. Maksudnya, dalam suatu kolom hanya ada satu baris yang berisi satu, sisanya berisi 0. Jadi, salah satu metode untuk mengubah suatu matriks sembarang hingga menjadi matriks eselon tereduksi dengan melakukan OBE adalah metode Eliminasi Gauss-Jordan. Metode ini (dan metode-metode lainnya di laporan ini) digunakan dalam pembuatan pustaka ini karena metode ini sistematis dan memiliki langkah yang pasti, sehingga dapat diubah menjadi suatu algoritma.

C. Determinan

Determinan adalah sebuah nilai skalar yang bisa didapatkan dari matriks berbentuk persegi, yaitu matriks yang memiliki jumlah baris dan kolom sama. Determinan matriks bisa dihitung dengan reduksi baris hingga semua elemen di atas atau di bawah diagonal utama matriks bernilai nol, kemudian semua elemen pada diagonal utama dikalikan.

Selain menghitung dengan reduksi baris, determinan juga dapat dihitung dengan ekspansi kofaktor. Dengan ekspansi kofaktor, determinan dihitung dengan menjumlahkan perkalian elemen-elemen kofaktor matriks yang bersesuaian. Determinan matriks dapat digunakan untuk menentukan matriks balikan suatu matriks.

D. Matriks Balikan

Matriks balikan adalah suatu matriks yang jika dikali dengan matriks lain dapat menghasilkan matriks identitas. Suatu matriks bisa saja tidak memiliki matriks balikan, yaitu ketika tidak ada konfigurasi matriks apapun yang jika dikali dengan matriks tersebut, dapat menghasilkan matriks identitas. Salah satu cara mengetahui ada tidaknya matriks balikan dari suatu matriks adalah dengan mencari tahu nilai determinan dari matriks tersebut. Jika determinan dari suatu matriks bernilai 0, matriks tersebut sudah pasti tidak memiliki matriks balikan. Ciri yang pasti dimiliki dari suatu matriks yang memiliki matriks balikan adalah, matriks tersebut pasti berukuran $n \times n$, maksudnya matriks tersebut pasti memiliki baris dan kolom yang sama. Jika baris dan kolom matriks tersebut berbeda, maka sudah pasti tidak memiliki matriks balikan.

E. Matriks Kofaktor

Matriks kofaktor adalah matriks yang berisi nilai-nilai minor dari matriks dan elemennya bernilai positif atau negatif sesuai dengan aturan yang sudah ditentukan. Untuk menentukan matriks kofaktor, kita harus menghitung minor dari baris dan kolom yang bersesuaian terlebih dahulu. Minor adalah determinan dari beberapa matriks persegi yang lebih kecil, yang didapatkan dengan memotong satu atau lebih baris dan kolom dari matriks awal. Setelah didapatkan minor dari masing-masing baris dan kolom, kita masukkan minor ke matriks kofaktor dengan nilai yang bersesuaian dengan baris dan kolomnya. Apabila baris bernilai genap, maka kolom pertamanya bernilai negatif, kolom kedua bernilai positif, dan seterusnya. Untuk baris yang bernilai ganjil, kolom pertamanya bernilai positif, kolom kedua bernilai negatif, dan seterusnya.

Matriks kofaktor dapat dimanfaatkan untuk mencari determinan. Metode mencari determinan menggunakan matriks kofaktor disebut metode ekspansi kofaktor. Metode ini memanfaatkan salah satu baris atau kolom dari matriks kofaktor, dan mengalikan semua elemen-elemennya dengan elemen pada matriks awal pada baris atau kolom yang bersesuaian kemudian dijumlahkan.

F. Matriks Adjoin

Matriks adjoin adalah matriks kofaktor yang ditranspose. Matriks adjoin digunakan untuk mencari matriks balikan. Untuk mencari matriks balikan dengan matriks adjoin, kita perlu mencari nilai determinan matriks terlebih dahulu. Kemudian, kalikan matriks adjoin dengan satu per determinan.

G. Kaidah Cramer

Kaidah Cramer adalah salah satu rumus untuk menyelesaikan sistem persamaan linier dengan variabel banyak yang berbentuk $Ax = b$, dan hanya berlaku ketika sistem tersebut memiliki solusi tunggal dan determinan matriks tidak sama dengan nol. Misalkan kita memiliki matriks A dengan ukuran $n \times n$, dengan begitu, kita akan mendapatkan nilai sebagai berikut:

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

Nilai $\det(A_n)$ didapatkan dengan mengganti entri pada kolom ke- j dari A dengan entri dari matriks b sebagai berikut:

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

H. Interpolasi Polinom

Interpolasi Polinom adalah salah satu metode yang bisa digunakan untuk memperkirakan suatu nilai. Persoalan interpolasi polinom adalah sebagai berikut, misal diberikan sejumlah $n + 1$ buah titik berbeda, $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$. Dari titik-titik tersebut ditentukan sebuah polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $i = 0, 1, 2, \dots, n$. Polinom yang melewati titik-titik tersebut berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$.

Dengan melakukan penyulihan balik titik (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ untuk $i = 0, 1, 2, \dots, n$ akan diperoleh $n + 1$ sebuah SPL. Kemudian dengan melakukan eliminasi Gauss, dapat ditentukan polinom interpolasi. Setelah kita temukan persamaan polinom interpolasinya, kita bisa memperkirakan nilai y di sembarang titik dengan x berada pada nilai $[x_0, x_n]$.

I. Regresi Linear Berganda

Selain menggunakan interpolasi polinom untuk memperkirakan suatu nilai dan fungsi, kita juga bisa menggunakan regresi linier berganda. Perbedaan dari metode interpolasi dan regresi linier berganda adalah interpolasi hanya menerima satu peubah x , sedangkan regresi linier berganda menerima lebih dari satu peubah x . Kita misalkan k

adalah banyak peubah x, maka bentuk umum hasil fungsi dari regresi linier berganda adalah $f(x) = B_0 + B_1x_1 + B_2x_2 + \dots + B_kx_k$.

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

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

Setelah didapatkan sistem persamaan linier di atas, ubah SPL tersebut ke dalam bentuk matriks *augmented* dan gunakan metode eliminasi Gauss untuk mendapatkan nilai B.

Bab 3

Implementasi pustaka dan program dalam Java

A. Struktur Utama Program

Program kami buat dengan kerangka utama:

Main.java

MenuSPL.java

MenuDeterminan.java

MenuInvers.java

MenuInterpolasi.java

MenuRLB.java

Matriks.java

Catatan: Indentasi pada menu-menu menunjukkan bahwa menu dipanggil dari program Main.java.

Main

Kerangka utama pertama pada program kami adalah Main.java atau kelas Main yang berisi satu atribut publik dan 3 metode publik. Atribut publik tersebut adalah scanner (sc) yang digunakan untuk menerima input dari pengguna sehingga pengguna bisa memilih program yang akan dijalankan.

Metode publik yang pertama adalah metode main, yang akan menjadi program utama yang dijalankan, di dalam metode main kami meminta masukkan dari pengguna dan dari masukkan tersebut kami memanggil metode dari kelas lain sesuai dengan masukkan yang diberikan pengguna.

Metode publik yang kedua adalah metode RobustIntInput, yaitu metode yang akan mengembalikan sebuah integer dan menerima dua buah integer sebagai batas bawah dan batas atas. Metode ini kami buat sebagai abstraksi tambahan agar kami tidak perlu membuat validasi input setiap kali kami meminta input.

Metode publik yang ketiga adalah metode RobustFilenameInput. Metode ini mirip dengan RobustIntInput. Perbedaannya adalah, RobustIntInput memastikan user memasukkan integer yang sesuai dengan range, RobustFilenameInput memastikan user memasukkan nama dari file(*filename*) yang benar-benar ada.

Setelah input pengguna valid, program akan lanjut berjalan sesuai dengan input yang diberikan, antara satu hingga enam.

- Jika pengguna memilih 1, akan dipanggil submenuSPL() dari kelas MenuSPL.
- Jika pengguna memilih 2, akan dipanggil SubmenuDeterminan() dari kelas MenuDeterminan.

- c. Jika pengguna memilih 3, akan dipanggil submenuInvers() dari kelas MenuInvers.
- d. Jika pengguna memilih 4, akan dipanggil Interpolasi() dari kelas MenuInterpolasi.
- e. Jika pengguna memilih 5, akan dipanggil submenuRLB() dari kelas MenuRLB.
- f. Pengguna akan keluar dari program apabila memilih 6.

Matriks

File utama kedua adalah Matriks.java. File ini berisi implementasi dari class Matriks, lengkap dengan atribut dan methodnya. Class Matriks memiliki tiga buah atribut, baris, kolom, dan isi. Baris adalah jumlah baris yang dapat digunakan di matriks tersebut, begitu juga dengan kolom. Keduanya bertipe integer. Isi adalah sebuah array dua dimensi bertipe double. Kami memilih double karena tipe double memiliki tingkat ketelitian angka yang paling tinggi, dan memori juga bukan masalah pada program ini.

Selanjutnya method, method kami bagi menjadi tiga bagian utama utama.

- a. Method Primitif
Method primitif berisi
Fungsi `Baris`, fungsi `Kolom`, fungsi `Isi`, untuk mengembalikan baris, kolom dan isi. Prosedur `ubahIsi`, `ubahBaris`, dan `ubahKolom` untuk mengubah isi, baris dan kolom. Fungsi `copyMatriks` untuk membuat salinan dari suatu matriks, prosedur `tukarBaris` dan prosedur `tukarKolom` untuk menukar dua buah baris atau kolom pada suatu matriks. Fungsi `semuaBarisNol` dan `semuaKolomNol` untuk mengembalikan nilai true jika semua baris di satu kolom bernilai nol atau semua kolom di satu baris bernilai nol. Fungsi `transpose` untuk mengembalikan matriks yang sudah di transpose, fungsi `PerkalianMatriks` untuk mengembalikan suatu matriks yang merupakan perkalian dari dua buah matriks lainnya. Fungsi `det` untuk menghasilkan nilai determinan dari suatu matriks, dan terakhir fungsi `isMatriksHomogen` yang akan menghasilkan true jika matriks homogen.
- b. Method Baca/Tulis
Di method baca/tulis, terdapat prosedur `isiMatriks` untuk mengisi suatu matriks dengan nilai input dari user. Prosedur `displayMatriks` untuk memberikan output berupa isi dari matriks dengan format yang, tiap data dalam satu baris dipisahkan dengan spasi dan tiap baris dipisahkan dengan newline. Untuk melakukan penulisan pada suatu file, kami membuat fungsi `FileRow` dan `FileColumn`, untuk memberikan banyak baris dan kolom di suatu file dengan nama file yang diberikan oleh user dari console. Lalu untuk mengisi matriks dengan inputan dari file, kami menggunakan prosedur `ReadMatriksFromFile`. Untuk output ke file, kami membuat prosedur `ouputToFile`, yang dipanggil oleh prosedur `konfirmOutputkeFile`.
- c. Method Tambahan
 1. Method OBE
Prosedur `OBEgauss` digunakan untuk melakukan OBE hingga menghasilkan matriks eselon pada matriks yang memanggil method. Prosedur `OBEgaussJordan` mirip dengan `OBEgauss`, perbedaannya, `OBEgaussJordan` melakukan OBE hingga terbentuk matriks eselon tereduksi

2. Method Invers

Terdapat prosedur `inversMatriksOBE`, digunakan untuk melakukan invers dengan cara OBE. Ada juga `inversMatriksAdj`, yang digunakan untuk mendapatkan invers dengan memanfaatkan determinan dan matriks Adjoin.

3. Method Determinan

Terdapat fungsi `pengubahOBE`, yaitu fungsi pendukung untuk melakukan satu kali OBE pada satu baris. Prosedur `detReduksiOBE` digunakan untuk melakukan OBE pada suatu matriks hingga membentuk 'segitiga atas'. Untuk prosedur `ditutupi`, menghasilkan suatu matriks yang merupakan hasil dari matriks lain yang sudah ditutupi (jika bingung maksud dari ditutupi, ingat kembali langkah pembuatan matriks kofaktor) pada baris dan kolom tertentu. Selanjutnya ada Prosedur `cofactor`, untuk menghasilkan matriks yang merupakan matriks kofaktor dari matriks lain.

4. Method Regresi Linear Berganda

Hanya terdapat satu prosedur, yaitu `regresiLinearBerganda`. Fungsi ini memiliki status akhir yaitu terbentuk solusi regresi linear dari matriks yang berbentuk persamaan linier.

5. Method SPL

Terdapat 4 fungsi sesuai dengan metode masing-masing, yaitu `splGauss`, `splGaussJordan`, `splInvers`, dan `splCramer`. Keempat fungsi tersebut mengeluarkan solusi SPL dalam tipe String untuk kemudian dapat dicetak dan disimpan ke dalam file.

B. Menu-Menu

Berikut penjelasan lebih lanjut dari tiap-tiap menu yang dapat diakses. Tiap file mewakili satu buah menu, dan tiap menu diakses dari Main

MenuSPL

Pada kelas MenuSPL, terdapat satu atribut publik dan satu metode publik. Satu atribut publik tersebut adalah scanner (sc) yang digunakan untuk menerima masukan dari pengguna yang dibutuhkan oleh program. Scanner tersebut juga digunakan untuk menerima pilihan metode dari pengguna, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer. Metode publik yang terdapat di dalam kelas MenuSPL adalah `submenuSPL()` yang akan dipanggil ke metode main di kelas Main. Metode ini akan menampilkan pilihan metode yang dapat digunakan oleh pengguna untuk menemukan solusi dari SPL yang dimasukkan. Kemudian terdapat pula pilihan metode yang ingin digunakan untuk menginput matriks, yaitu dari console atau dari file. Untuk menggunakan MenuSPL, pengguna harus memasukkan matriks dalam bentuk augmented, baik melalui console ataupun file. Kemudian setelah memasukkan matriks yang hendak diproses, akan dipanggil metode `splGauss()`, `splGaussJordan()`, `splInvers()`, atau `splCramer()` sesuai dengan pilihan pengguna. Kemudian hasil akan ditampilkan dan dapat disimpan ke dalam file bila pengguna menginginkannya.

MenuDeterminan

Pada kelas MenuDeterminan, terdapat satu atribut publik dan satu metode publik. Satu atribut publik tersebut adalah scanner (sc) yang digunakan untuk menerima masukan dari pengguna yang dibutuhkan oleh program. Scanner tersebut juga digunakan untuk menerima pilihan metode dari pengguna, yaitu metode reduksi atau metode kofaktor serta untuk memilih metode input matriks ke program. Metode publik yang dibuat di dalam kelas MenuDeterminan adalah SubmenuDeterminan() yang akan dipanggil ke metode main di kelas Main. Metode ini akan menampilkan pilihan metode yang bisa dipilih oleh pengguna untuk mendapatkan determinan dari matriks, serta metode yang ingin digunakan oleh pengguna untuk menginput matriks. Di dalam metode ini, akan dipanggil metode detReduksiOBE() dan cofactor() dari kelas Matriks untuk mengolah matriks.

MenuInvers

Pada kelas MenuInvers, terdapat satu atribut publik dan satu metode publik. Atribut publik yang terdapat pada kelas MenuInvers adalah scanner (sc) yang digunakan untuk menerima masukan pengguna sehingga pengguna bisa memilih ingin melakukan invers melalui metode OBE atau adjoin dan dapat memilih cara input matriks baik melalui konsol ataupun file. Metode publik yang terdapat pada kelas ini adalah submenuInvers() yang akan dipanggil ke metode main di kelas Main. Di dalam metode submenuInvers, pengguna akan memilih metode untuk mendapatkan matriks balikan terlebih dahulu. Setelah itu, pengguna memilih untuk memasukkan matriks melalui konsol atau file. Apabila pengguna memilih untuk memasukkan matriks melalui konsol, pengguna harus memberikan jumlah baris n dan kolom n , kemudian memasukkan elemen-elemen matriks berukuran $n \times n$. Jika pengguna memilih memasukkan matriks dari file, pengguna memberikan nama file melalui konsol dan metode ReadMatriksFromFile() dari kelas Matriks akan dipanggil. Setelah matriks terdefinisi baik melalui konsol maupun file, matriks diolah sesuai pilihan pengguna di awal, yaitu OBE atau adjoin. Setelah didapatkan matriks balikan, matriks balikan tersebut akan ditampilkan di konsol beserta langkah-langkahnya. Kemudian, pengguna akan diminta untuk memilih agar hasil tersebut ditulis ke sebuah file atau tidak, jika iya, maka pengguna memberikan masukkan nama file dan pesan bahwa matriks telah tersimpan akan dimunculkan. Jika pengguna memilih untuk tidak menulis hasil ke sebuah file, maka pengguna akan dikembalikan ke menu utama.

MenuInterpolasi

Pada kelas MenuInterpolasi, terdapat satu atribut publik dan satu metode publik. Atribut publik tersebut adalah scanner (sc) yang akan digunakan untuk membaca masukan dari pengguna. Atribut ini akan digunakan untuk membaca masukkan jumlah titik dan titik dari pengguna. Metode publik yang terdapat di dalam kelas ini adalah Interpolasi(). Metode ini akan mengolah matriks yang diberikan pengguna dan menampilkan hasil fungsi interpolasi polinom. Kemudian, metode ini juga akan menampilkan hasil nilai taksiran x yang dimasukkan pengguna. Setelah itu, pengguna akan diminta konfirmasi jika hasil akan ditulis ke file atau tidak.

MenuRLB

Pada kelas menuRLB terdapat satu atribut publik dan satu metode publik. Atribut publik tersebut adalah scanner yang digunakan untuk membaca masukan dari pengguna. Scanner tersebut akan digunakan untuk mengambil masukkan jumlah sampel dan jumlah peubah dari data pengguna. Metode publik yang terdapat pada kelas MenuRLB adalah submenuRLB(). Di dalam metode submenuRLB, program akan memanggil metode ReadMatriksFromFile dari kelas Matriks untuk mendapatkan matriks atau meminta pengguna memasukkan data melalui konsol, tergantung apa yang dipilih pengguna. Setelah itu, matriks akan diolah dengan memanggil metode regresiLinierBerganda dari kelas Matriks.

C. Garis Besar Program

Secara garis besar, program dijalankan mulai dari kelas Main yang akan memanggil metode dari kelas-kelas lain seperti MenuSPL, MenuDeterminan, MenuInvers, MenuRBL, MenuInterpolasi. Setelah kelas dipanggil, akan muncul menu utama pada konsol seperti gambar berikut:

```
===== MENU UTAMA =====  
1. Sistem Persamaan Linear  
2. Determinan  
3. Matriks Balikan  
4. Interpolasi Polinom  
5. Regresi Linear Berganda  
6. Keluar
```

Pengguna dapat memasukkan angka dari 1 hingga 6. Sesuai nomor dan nama yang tertera pada menu, jika pengguna memasukkan angka 1, maka pengguna akan memasuki submenu SPL (Sistem Persamaan Linier). Jika pengguna memasukkan angka 2, maka pengguna akan memasuki submenu determinan, dan seterusnya hingga pengguna memasukkan angka 6 untuk keluar dari program.

Pada submenu SPL, program main akan memanggil submenuSPL dari kelas MenuSPL dan akan muncul beberapa pilihan metode menyelesaikan SPL seperti:

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

Jika pengguna memilih angka 1, maka metode `splGauss()` dari kelas Matriks akan dipanggil. Untuk angka 2, metode `splGaussJordan()` dari kelas Matriks akan dipanggil. Jika pengguna memilih angka 3, metode `splInvers()` akan dipanggil. Terakhir, untuk pilihan 4, metode `splCramer()` akan dipanggil.

Pada submenu determinan, program main akan memanggil SubmenuDeterminan dari kelas MenuDeterminan dan akan menampilkan menu yang berisi metode menghitung determinan, yaitu:

1. Metode reduksi
2. Metode kofaktor
3. Cancel

Jika pengguna memilih angka 1, maka metode `detReduksiOBE()` akan dipanggil dari kelas Matriks. Jika pengguna memilih angka 2, maka metode `cofactor()` dari kelas Main akan dipanggil. Program akan kembali ke menu utama apabila pengguna memilih angka 3.

Pada submenu matriks balikan, program main akan memanggil submenuInvers dari kelas MenuInvers dan akan menampilkan beberapa pilihan untuk mendapatkan matriks balikan, yaitu:

1. Metode OBE
2. Metode Adjoin

Jika pengguna memasukkan angka 1, metode `inversMatriksOBE()` akan dipanggil dari kelas `Matriks` dan metode `inversMatriksAdj()` akan dipanggil dari kelas `Matriks` apabila pengguna memasukkan angka 2.

Pada submenu interpolasi polinom, program main akan memanggil metode `Interpolasi` dari kelas `MenuInterpolasi` dan akan meminta matriks dari pengguna. Pengguna bisa memilih metode input yang ingin digunakan untuk memberikan matriks kepada program. Detail pilihan metode input akan dibahas lebih lanjut.

Pada submenu regresi linier berganda, program main akan memanggil metode `submenuRBL` dari kelas `MenuRBL` dan akan meminta matriks dari pengguna. Pengguna bisa memilih metode input yang ingin digunakan untuk memberikan matriks kepada program. Detail pilihan metode input akan dibahas lebih lanjut.

Setelah pengguna memilih metode yang ingin digunakan pada submenu `SPL`, `determinan`, dan `matriks balikan`, atau pengguna memilih `interpolasi polinom` atau `regresi linier berganda`, masing-masing submenu akan menampilkan pilihan metode input matriks. Ada dua pilihan metode yang kami sediakan, yaitu input melalui konsol secara langsung atau melalui file dengan ekstensi `“.txt”`. Jika pengguna ingin memberi input matriks melalui konsol, pengguna akan diminta beberapa input tambahan sesuai dengan submenu dan metode yang dipilih.

Untuk submenu `SPL`, pengguna memasukkan matriks berbentuk *augmented*. Sebelum memasukkan matriks, pengguna diminta untuk memasukkan jumlah baris dan kolom matriks terlebih dahulu, kemudian pengguna memasukkan elemen-elemen matriks sesuai dengan baris dan kolomnya.

Untuk submenu `determinan` dan `matriks balikan (invers)`, pengguna diminta untuk memasukkan `n`, yaitu jumlah baris sekaligus kolom matriks karena matriks yang dimasukkan bukan berbentuk *augmented*. Setelah pengguna memasukkan `n`, pengguna memasukkan elemen matriks sesuai ukuran yang sudah dimasukkan di awal.

Untuk submenu `interpolasi polinom`, pengguna diminta untuk memasukkan jumlah titik yang akan digunakan untuk mencari interpolasi polinom. Kemudian, pengguna memasukkan nilai `x` yang akan ditaksir dengan fungsi yang didapatkan dari interpolasi polinom. Setelah itu, pengguna memasukkan titik dengan format `(x, y)` sebanyak jumlah titik yang sudah dimasukkan oleh pengguna.

Untuk submenu `regresi linier berganda`, pengguna diminta untuk memasukkan `k`, yaitu jumlah sampel data dan `n`, yaitu jumlah peubah `x` dari data. Kemudian, pengguna memasukkan nilai dari `x` yang akan ditaksir menggunakan fungsi hasil regresi. Bentuk matriks yang dimasukkan oleh pengguna berbentuk *augmented*.

Sedangkan untuk input melalui file memiliki garis besar yang sama untuk semua submenu, yaitu pengguna memasukkan nama file matriks yang ingin diolah. Akan tetapi, pengguna harus memperhatikan bahwa untuk submenu SPL, interpolasi polinom, dan regresi linier berganda bentuk matriks yang akan dibaca adalah matriks *augmented* dan untuk submenu determinan dan matriks balikan matriks berbentuk matriks biasa.

Setelah pengguna memasukkan matriks, baik melalui metode input dari konsol maupun metode input dari file, masing-masing submenu akan menampilkan output yang berbeda-beda bergantung pada persoalan yang diselesaikan. Untuk persoalan SPL, output yang akan ditampilkan adalah langkah-langkah singkat dari metode yang dipilih dan solusi dari persoalan. Jika solusi unik, maka program akan menampilkan nilai dari x_1 hingga x_n . Jika persoalan memiliki banyak solusi, maka program akan menampilkan solusi dalam bentuk parametrik seperti $x_1 = 3 + t$ dengan $x_2 = t$. Jika persoalan tidak memiliki solusi, maka program akan mengeluarkan pesan bahwa persoalan tidak memiliki solusi.

Untuk persoalan determinan, program akan menampilkan langkah-langkah singkat dari metode yang dipilih dan hasil determinan yang sudah dihitung. Sedangkan untuk persoalan matriks balikan, program akan menampilkan langkah-langkah singkat dan hasil matriks balikan yang terbentuk. Namun, jika matriks tidak memiliki balikan, program akan mengeluarkan pesan bahwa matriks tidak memiliki balikan dan alasan mengapa matriks tidak memiliki balikan seperti determinan bernilai nol.

Untuk persoalan interpolasi polinom dan regresi linier berganda, program akan mengeluarkan langkah-langkah singkat, hasil fungsi yang didapatkan dari interpolasi polinom atau regresi linier berganda, dan nilai setelah x ditaksir menggunakan fungsi yang dihasilkan.

Setelah itu, program akan memanggil metode `confirmOutputToFile()` dari kelas `Matriks`. Metode tersebut meminta konfirmasi dari pengguna untuk menyimpan hasil keluaran program kedalam sebuah file. Jika pengguna memasukkan karakter "y", maka pengguna akan diminta untuk memasukkan nama file untuk menyimpan hasil keluaran tersebut. Program akan menampilkan pesan bahwa file berhasil disimpan dengan nama yang dimasukkan pengguna. Jika pengguna memasukkan karakter "n", maka program akan langsung kembali ke menu utama.

Bab 4

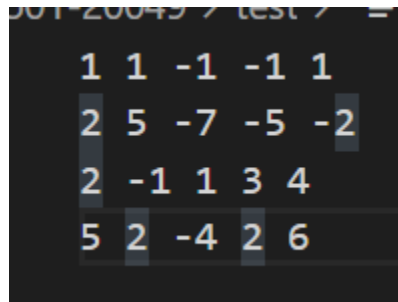
Eksperimen

Bab ini berisi hasil eksekusi dari *Studi Kasus* yang telah diberikan di file spek tugas yang diberikan oleh asisten IF2123. Studi kasus dibagi menjadi tujuh soal. Berikut hasil dari eksekusi program kami terhadap studi kasus tersebut (nomor dibawah mungkin tidak sesuai dengan nomor studi kasus. Untuk memastikan, terdapat keterangan dimasing-masing studi kasus dan bisa juga dilihat dari judul)

A. Studi Kasus SPL dengan bentuk $Ax = b$

Input kepada program kami berikan dalam bentuk file. Pada spek tugas, bagian ini berkorespondensi dengan nomor satu.

1. Berikut ini isi dari file input pada studi kasus SPL nomor satu bagian a.



The image shows a terminal window with a dark background. At the top, there is a header line that reads "001-20043 / test / 1". Below this, there is a 4x5 grid of numbers. The numbers are arranged as follows:

1	1	-1	-1	1
2	5	-7	-5	-2
2	-1	1	3	4
5	2	-4	2	6

Gambar 4.1.1.1 isi file input 1a

Pada studi kasus tidak dispesifikasikan cara untuk menyelesaikan persoalan. Kami menggunakan cara Gauss-Jordan. Pernyataan ini berlaku untuk uji berikutnya. Berikut hasil eksekusi dari program


```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
../test/1a_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.667 0.000
0.000 1.000 0.000 -2.667 0.000
0.000 0.000 1.000 -1.000 0.000
0.000 0.000 0.000 0.000 1.000
SPL tidak memiliki solusi
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.1.1.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

1.000 0.000 0.000 0.667 0.000
0.000 1.000 0.000 -2.667 0.000
0.000 0.000 1.000 -1.000 0.000
0.000 0.000 0.000 0.000 1.000
SPL tidak memiliki solusi

```

Gambar 4.1.1.3 isi file output dari hasil eksekusi

2. Berikut ini isi dari file input pada studi kasus SPL nomor satu bagian b.

```
1 -1 0 0 1 3
1 1 0 -3 0 6
2 -1 0 1 -1 5
-1 2 0 -2 -1 -1
```

Gambar 4.1.2.1 isi file input 1b

Berikut hasil eksekusi dari program

```
===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/1b_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.000 -1.000 3.000
0.000 1.000 0.000 0.000 -2.000 0.000
0.000 0.000 0.000 1.000 -1.000 -1.000
0.000 0.000 0.000 0.000 0.000 0.000
SPL memiliki tak hingga solusi:
Misalkan:
x3 = c
x5 = e
Maka:
x1 = 1.000e + 3.000
x2 = 2.000e + 0.000
x4 = 1.000e + -1.000
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt
```

Gambar 4.1.2.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```
1.000 0.000 0.000 0.000 -1.000 3.000
0.000 1.000 0.000 0.000 -2.000 0.000
0.000 0.000 0.000 1.000 -1.000 -1.000
0.000 0.000 0.000 0.000 0.000 0.000
SPL memiliki tak hingga solusi:
Misalkan:
x3 = c
x5 = e
Maka:
x1 = 1.000e + 3.000
x2 = 2.000e + 0.000
x4 = 1.000e + -1.000
```

Gambar 4.1.2.3 isi file output dari hasil eksekusi

3. Berikut ini isi dari file input pada studi kasus SPL nomor satu bagian c.

```
1-20049 / test / = 1c
0 1 0 0 1 0 2
0 0 0 1 1 0 -1
0 1 0 0 0 1 1
```

Gambar 4.1.3.1 isi file input 1c

Berikut hasil eksekusi dari program

```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/1c_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
0.000 1.000 0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 1.000 0.000 1.000 -2.000
0.000 0.000 0.000 0.000 1.000 -1.000 1.000
SPL memiliki tak hingga solusi:
Misalkan:
x1 = a
x3 = c
x6 = f
Maka:
x2 = -1.000f + 1.000
x4 = -1.000f + -2.000
x5 = 1.000f + 1.000
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.1.3.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

0.000 1.000 0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 1.000 0.000 1.000 -2.000
0.000 0.000 0.000 0.000 1.000 -1.000 1.000
SPL memiliki tak hingga solusi:
Misalkan:
x1 = a
x3 = c
x6 = f
Maka:
x2 = -1.000f + 1.000
x4 = -1.000f + -2.000
x5 = 1.000f + 1.000

```

Gambar 4.1.3.3 isi file output dari hasil eksekusi

4. Pada bagian d, terdapat dua macam kasus, pertama ketika n bernilai 6 dan kedua ketika n bernilai 10. Berikut ini isi dari file input pada studi kasus SPL nomor satu bagian d dengan n bernilai 6.

```

1 0.5 0.333333 0.25 0.2 0.166667 1
0.5 0.333333 0.25 0.2 0.166667 0.142857 0
0.333333 0.25 0.2 0.166667 0.142857 0.125 0
0.25 0.2 0.166667 0.142857 0.125 0.111111 0
0.2 0.166667 0.142857 0.125 0.111111 0.1 0
0.166667 0.142857 0.125 0.111111 0.1 0.0909091 0

```

Gambar 4.1.4.1 isi file input 1d dengan $n = 6$

Berikut hasil eksekusi dari program

```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/1d6_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.000 0.000 0.000 11.060
0.000 1.000 0.000 0.000 0.000 0.000 58.563
0.000 0.000 1.000 0.000 0.000 0.000 -1203.880
0.000 0.000 0.000 1.000 0.000 0.000 4144.573
0.000 0.000 0.000 0.000 1.000 0.000 -5226.013
0.000 0.000 0.000 0.000 0.000 1.000 2226.061
SPL memiliki solusi tunggal:
x1 = 11.060; x2 = 58.563; x3 = -1203.880; x4 = 4144.573; x5 = -5226.013; x6 = 2226.061;

Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.1.4.2 hasil eksekusi program dengan $n = 6$

Berikut isi dari file tempat output disimpan

```

1.000 0.000 0.000 0.000 0.000 0.000 11.060
0.000 1.000 0.000 0.000 0.000 0.000 58.563
0.000 0.000 1.000 0.000 0.000 0.000 -1203.880
0.000 0.000 0.000 1.000 0.000 0.000 4144.573
0.000 0.000 0.000 0.000 1.000 0.000 -5226.013
0.000 0.000 0.000 0.000 0.000 1.000 2226.061
SPL memiliki solusi tunggal:
x1 = 11.060; x2 = 58.563; x3 = -1203.880; x4 = 4144.573; x5 = -5226.013; x6 = 2226.061;

```

Gambar 4.1.4.3 isi file output dari hasil eksekusi dengan $n = 6$

Berikut ini isi dari file input pada studi kasus SPL nomor satu bagian d dengan n bernilai 10.

```

1 0.5 0.333333 0.25 0.2 0.166667 0.142857 0.125 0.111111 0.1 1
0.5 0.333333 0.25 0.2 0.166667 0.142857 0.125 0.111111 0.1 0.0909091 0
0.333333 0.25 0.2 0.166667 0.142857 0.125 0.111111 0.1 0.0909091 0.0833333 0
0.25 0.2 0.166667 0.142857 0.125 0.111111 0.1 0.0909091 0.0833333 0.0769231 0
0.2 0.166667 0.142857 0.125 0.111111 0.1 0.0909091 0.0833333 0.0769231 0.0714286 0
0.166667 0.142857 0.125 0.111111 0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0
0.142857 0.125 0.111111 0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0.0625 0
0.125 0.111111 0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0.0625 0.0588235 0
0.111111 0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0.0625 0.0588235 0.0555556 0
0.1 0.0909091 0.0833333 0.0769231 0.0714286 0.0666667 0.0625 0.0588235 0.0555556 0.0526316 0

```

Gambar 4.1.4.1 isi file input 1d dengan $n = 10$

Berikut hasil eksekusi dari program

```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/1d10_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 57.327
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -1142.353
0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 5821.692
0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 -8644.016
0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 -3710.110
0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 7498.255
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 27493.041
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 -47662.394
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 20504.042
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 -203.969
SPL memiliki solusi tunggal:
x1 = 57.327; x2 = -1142.353; x3 = 5821.692; x4 = -8644.016; x5 = -3710.110; x6 = 7498.255; x7 = 27493.041;
x8 = -47662.394; x9 = 20504.042; x10 = -203.969;

Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.1.3.2 hasil eksekusi program dengan $n = 10$

Berikut isi dari file tempat output disimpan

```

1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 57.327
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 -1142.353
0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 5821.692
0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 -8644.016
0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 -3710.110
0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 7498.255
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 27493.041
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 -47662.394
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 20504.042
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 -203.969
SPL memiliki solusi tunggal:
x1 = 57.327; x2 = -1142.353; x3 = 5821.692; x4 = -8644.016; x5 = -3710.110; x6 = 7498.255; x7 =
27493.041; x8 = -47662.394; x9 = 20504.042; x10 = -203.969;

```

Gambar 4.1.3.3 isi file output dari hasil eksekusi dengan $n = 10$

B. Studi Kasus Interpolasi

Input kepada program kami berikan dalam bentuk file. Pada spek tugas, bagian ini berkorespondensi dengan nomor enam.

1. Berikut isi dari file input pada Studi Kasus Interpolasi Nomor 6 bagian a.

```

Algeo01-20049 > test > 6a_input.txt
1 0.1 0.003
2 0.3 0.067
3 0.5 0.148
4 0.7 0.248
5 0.9 0.370
6 1.1 0.518
7 1.3 0.697

```

Gambar 4.2.1.1 isi file input 6a

Terdapat empat buah x yang akan diuji,

Untuk x bernilai 0.2,

Berikut hasil eksekusi dari program


```

===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->../test/6a_input.txt
masukkan nilai x yang akan ditaksir nilainya
->0.2
hasil taksirannya adalah:
p(x) = -0.02 + 0.24*x^(1) + 0.20*x^(2) + 0.00*x^(3) + 0.03*x^(4) + 0.00*x^(5) + -0.00*x^(6)= 0.03
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes

```

Gambar 4.2.1.2 hasil eksekusi program dengan x bernilai 0.2

Berikut isi dari file tempat output disimpan

```

Algeo01-20049 > src > test.txt
1  dengan nilai x = 0.2,
2  p(x) = -0.022976562500000138 + 0.2400000000000024*x^1 + 0.1973958333333196*x^2 + 3.
    4409974869475946E-14*x^3 + 0.026041666666624073*x^4 + 2.550133859867565E-14*x^5 + -5.
    910407676398067E-15*x^6 = 0.032960937500000016
3

```

Gambar 4.2.1.3 isi file output dari hasil eksekusi dengan x bernilai 0.2

Jika anda sadar, pada gambar 4.2.1.2, terdapat nilai 0.00 dan -0.00. Hal itu terjadi bukan karena program tidak meng-cover kasus tersebut, namun karena ada pembulatan pada saat output menjadi 2 angka dibelakang koma. Ini kami lakukan agar output lebih mudah dilihat, dan hal itu tidak kami lakukan di output ke file(seperti gambar 4.1.3). Untuk program-program berikutnya, hal ini mungkin saja terjadi lagi, dan penjelasan yang kami berikan tetap sama.

Untuk x bernilai 0.55,

Berikut hasil eksekusi dari program

```

===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->../test/6a_input.txt
masukkan nilai x yang akan ditaksir nilainya
->0.55
hasil taksirannya adalah:
p(x) = -0.02 + 0.24*x^(1) + 0.20*x^(2) + 0.00*x^(3) + 0.03*x^(4) + 0.00*x^(5) + -0.00*x^(6)= 0.17
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y

```

Gambar 4.2.1.4 hasil eksekusi program dengan x bernilai 0.55

Berikut isi dari file tempat output disimpan

```

dengan nilai x = 0.55,
p(x) = -0.022976562500000138 + 0.2400000000000024*x^1 + 0.1973958333333196*x^2 + 3.
    4409974869475946E-14*x^3 + 0.026041666666624073*x^4 + 2.550133859867565E-14*x^5 + -5.
    910407676398067E-15*x^6 = 0.17111865234374998

```

Gambar 4.2.1.5 isi file output dari hasil eksekusi dengan x bernilai 0.55

Untuk x bernilai 0.85,

Berikut hasil eksekusi dari program

```
===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->../test/6a_input.txt
masukkan nilai x yang akan ditaksir nilainya
->0.85
hasil taksirannya adalah:
p(x) = -0.02 + 0.24*x^(1) + 0.20*x^(2) + 0.00*x^(3) + 0.03*x^(4) + 0.00*x^(5) + -0.00*x^(6)= 0.34
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
```

Gambar 4.2.1.6 hasil eksekusi program dengan x bernilai 0.85

Berikut isi dari file tempat output disimpan

```
dengan nilai x = 0.85,
p(x) = -0.022976562500000138 + 0.2400000000000024*x^1 + 0.1973958333333196*x^2 + 3.
4409974869475946E-14*x^3 + 0.026041666666624073*x^4 + 2.550133859867565E-14*x^5 + -5.
910407676398067E-15*x^6 = 0.33723583984375
```

Gambar 4.2.1.7 isi file output dari hasil eksekusi dengan x bernilai 0.85

Untuk x bernilai 1.28,

Berikut hasil eksekusi dari program

```
===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->../test/6a_input.txt
masukkan nilai x yang akan ditaksir nilainya
->1.28
hasil taksirannya adalah:
p(x) = -0.02 + 0.24*x^(1) + 0.20*x^(2) + 0.00*x^(3) + 0.03*x^(4) + 0.00*x^(5) + -0.00*x^(6)= 0.68
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
```

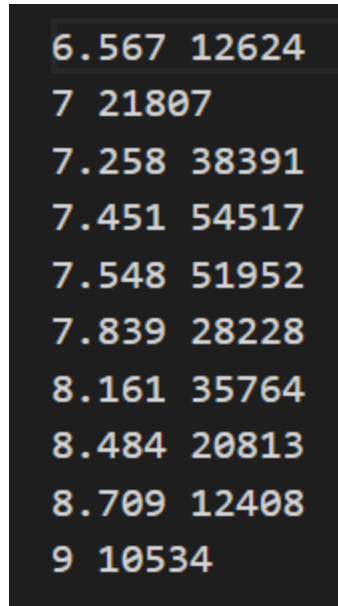
Gambar 4.2.1.8 hasil eksekusi program dengan x bernilai 1.28

Berikut isi dari file tempat output disimpan

```
dengan nilai x = 1.28,
p(x) = -0.022976562500000138 + 0.2400000000000024*x^1 + 0.1973958333333196*x^2 + 3.
4409974869475946E-14*x^3 + 0.026041666666624073*x^4 + 2.550133859867565E-14*x^5 + -5.
910407676398067E-15*x^6 = 0.6775418374999999
```

Gambar 4.2.1.9 isi file output dari hasil eksekusi dengan x bernilai 1.28

2. Berikut isi dari file input pada Studi Kasus Interpolasi Nomor 6 bagian b.



6.567	12624
7	21807
7.258	38391
7.451	54517
7.548	51952
7.839	28228
8.161	35764
8.484	20813
8.709	12408
9	10534

Gambar 4.2.2.1 isi file input 6b

Terdapat empat buah tanggal yang akan diuji,

Untuk tanggal 16/07/2021

Jika diubah menjadi desimal sesuai dengan aturan yang diberikan, akan menjadi 7 + (16/31) dan bernilai 7.516

Berikut hasil eksekusi dari program

```
===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->./test/6b_input.txt
masukkan nilai x yang akan ditaksir nilainya
->7.516
hasil taksirannya adalah:
p(x) = 7187066071657.87 + -9346993079172.33*x^(1) + 5334203055240.20*x^(2) + -1756810186361.38*x^(3) + 368550807175.53*x^(4) + -51131876760.13*x^(5) + 4695806315.43*x^(6) + -275474539.42*x^(7) + 9372849.24*x^(8) + -140993.71*x^(9)= 53538.00
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt
```

Gambar 4.2.2.2 hasil eksekusi program dengan tanggal 16/07/2021

Berikut isi dari file tempat output disimpan

```
1 dengan nilai x = 7.516,
2 p(x) = 7.187066071657867E12 + -9.346993079172328E12*x^1 + 5.334203055240195E12*x^2 + -1.7568101863613809E12*x^3 + 3.6855080717553394E11*x^4 + -5.113187676013275E10*x^5 + 4.695806315428793E9*x^6 + -2.754745394206693E8*x^7 + 9372849.23910132*x^8 + -140993.71224863594*x^9 = 53537.99609375
3
```

Gambar 4.2.2.3 isi file output dari hasil eksekusi dengan tanggal 16/07/2021

Untuk tanggal 10/08/2021

Jika diubah menjadi desimal sesuai dengan aturan yang diberikan, akan menjadi 8+ (10/31) dan bernilai 8.323

Berikut hasil eksekusi dari program

```
===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->./test/6b_input.txt
masukkan nilai x yang akan ditaksir nilainya
->8.323
hasil taksirannya adalah:
p(x) = 7187066071657.87 + -9346993079172.33*x^(1) + 5334203055240.20*x^(2) + -1756810186361.38*x^(3) + 368550807175.53*x^(4) + -51131876760.13*x^(5) + 4695806315.43*x^(6) + -275474539.42*x^(7) + 9372849.24*x^(8) + -140993.71*x^(9)= 36294.85
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt
```

Gambar 4.2.2.4 hasil eksekusi program dengan tanggal 10/08/2021

Berikut isi dari file tempat output disimpan

```
dengan nilai x = 8.323,
p(x) = 7187066071657867E12 + -9.346993079172328E12*x^1 + 5.334203055240195E12*x^2 + -1.7568101863613809E12*x^3 + 3.6855080717553394E11*x^4 + -5.113187676013275E10*x^5 + 4.695806315428793E9*x^6 + -2.754745394206693E8*x^7 + 9372849.23910132*x^8 + -140993.71224863594*x^9 = 36294.84765625
```

Gambar 4.2.2.5 isi file output dari hasil eksekusi dengan tanggal 10/08/2021

Untuk tanggal 05/09/2021

Jika diubah menjadi desimal sesuai dengan aturan yang diberikan, akan menjadi 9 + (5/30) dan bernilai 9.167

Berikut hasil eksekusi dari program

```
===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->./test/6b_input.txt
masukkan nilai x yang akan ditaksir nilainya
->9.167
hasil taksirannya adalah:
p(x) = 7187066071657.87 + -9346993079172.33*x^(1) + 5334203055240.20*x^(2) + -1756810186361.38*x^(3) + 368550807175.53*x^(4) + -51131876760.13*x^(5) + 4695806315.43*x^(6) + -275474539.42*x^(7) + 9372849.24*x^(8) + -140993.71*x^(9)= -667693.22
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt
```

Gambar 4.2.2.6 hasil eksekusi program dengan tanggal 05/09/2021

Berikut isi dari file tempat output disimpan

```
dengan nilai x = 9.167,
p(x) = 7187066071657867E12 + -9.346993079172328E12*x^1 + 5.334203055240195E12*x^2 + -1.7568101863613809E12*x^3 + 3.6855080717553394E11*x^4 + -5.113187676013275E10*x^5 + 4.695806315428793E9*x^6 + -2.754745394206693E8*x^7 + 9372849.23910132*x^8 + -140993.71224863594*x^9 = -667693.21875
```

Gambar 4.2.2.7 isi file output dari hasil eksekusi dengan tanggal 05/09/2021

Jika kita lihat, hasilnya negatif. Untuk kasus jumlah covid-19, angka ini memang tidak mungkin. Namun memang ini hasil dari interpolasi yang seharusnya, dan jika kita lihat

data yang diberikan dan kita perkirakan secara intuitif, memang akan menghasilkan negatif karena ada tren penurunan jumlah. Karena program dibuat untuk menghitung interpolasi secara umum dan bukan untuk kasus covid-19, jadi program memperbolehkan hasil negatif.

Untuk tanggal bebas, kami memilih 31/07/2021

Jika diubah menjadi desimal sesuai dengan aturan yang diberikan, akan menjadi 7 + (31/31) dan bernilai 8

Berikut hasil eksekusi dari program

```
===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->../test/6b_input.txt
masukkan nilai x yang akan ditaksir nilainya
->8
hasil taksirannya adalah:
p(x) = 7187066071657.87 + -9346993079172.33*x^(1) + 5334203055240.20*x^(2) + -1756810186361.38*x^(3) + 368550807175.53*x^(4) + -51131876760.13*x^(5) + 4695806315.43*x^(6) + -275474539.42*x^(7) + 9372849.24*x^(8) + -140993.71*x^(9)= 27752.56
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt
```

Gambar 4.2.2.8 hasil eksekusi program dengan tanggal 31/07/2021

Berikut isi dari file tempat output disimpan

```
dengan nilai x = 8.0,
p(x) = 7.187066071657867E12 + -9.346993079172328E12*x^1 + 5.334203055240195E12*x^2 + -1.7568101863613809E12*x^3 + 3.6855080717553394E11*x^4 + -5.113187676013275E10*x^5 + 4.695806315428793E9*x^6 + -2.754745394206693E8*x^7 + 9372849.23910132*x^8 + -140993.71224863594*x^9 = 27752.5625
```

Gambar 4.2.2.9 isi file output dari hasil eksekusi dengan tanggal 31/07/2021

3. Pada studi kasus ini, kami dibebaskan untuk memilih nilai n, yaitu banyak data. Kami memilih n bernilai 10. Berikut isi dari file input pada Studi Kasus Interpolasi Nomor 6 bagian c.

```

0 0
0.2 0.342769558243
0.4 0.418884230141
0.6 0.468431469612
0.8 0.50715796853
1 0.53788284274
1.2 0.560924674815
1.4 0.576187119762
1.6 0.583685661287
1.8 0.583674720078
2 0.576651529752

```

Gambar 4.2.3.1 isi file input 6c

Pada saat eksekusi, program selalu meminta pengguna untuk memasukkan sebuah nilai x yang akan dicari nilai p(x) nya (sesuai dengan spek tugas). Kami memilih untuk memasukkan data bernilai 0. Berikut hasil eksekusi dari program

```

===== MENU Interpolasi Polinom =====
Input dari terminal(1) atau file(2)?
->2
masukkan nama file
->./test/6c_input_n=10.txt
masukkan nilai x yang akan ditaksir nilainya
->0
hasil taksirannya adalah:
p(x) = 3.88*x^(1) + -18.81*x^(2) + 57.32*x^(3) + -111.51*x^(4) + 143.28*x^(5) + -123.11*x^(6) + 69.96*x^(7) + -25.22*x^(8)
+ 5.22*x^(9) + -0.47*x^(10) = 0.00
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.2.3.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

dengan nilai x = 0.0,
p(x) = 3.881607235791409*x^1 + -18.8127807104637*x^2 + 57.32056797867952*x^3 + -111.
50502425643381*x^4 + 143.27638041471994*x^5 + -123.1121709250711*x^6 + 69.96195270909666*x^7 + -25.
221073136268718*x^8 + 5.220714357904088*x^9 + -0.4722908252142895*x^10 = 0.0

```

Gambar 4.2.3.2 isi file output dari hasil eksekusi

Berikut fungsi pengganti f(x),

$$\begin{aligned}
 p(x) = & 3.881607235791409*x^1 + -18.8127807104637*x^2 + \\
 & 57.32056797867952*x^3 + -111.50502425643381*x^4 + \\
 & 143.27638041471994*x^5 + -123.1121709250711*x^6 + \\
 & 69.96195270909666*x^7 + -25.221073136268718*x^8 + \\
 & 5.220714357904088*x^9 + -0.4722908252142895*x^{10}
 \end{aligned}$$

C. Studi Kasus Regresi Linear Berganda

Bagian ini berkorepondensi dengan studi kasus nomor tujuh. Diberikan 20 data yang akan dibuat regresi linear bergandanya. Data x_1 merepresentasikan humidity, x_2 merepresentasikan temperatur, dan data x_3 merepresentasikan tekanan udara. Berikut input yang diberikan

```
72.4 76.3 29.18 0.90
41.6 70.3 29.35 0.91
34.3 77.1 29.24 0.96
35.1 68.0 29.27 0.89
10.7 79.0 29.78 1.00
12.9 67.4 29.39 1.10
8.3 66.8 29.69 1.15
20.1 76.9 29.48 1.03
72.2 77.7 29.09 0.77
24.0 67.7 29.60 1.07
23.2 76.8 29.38 1.07
47.4 86.6 29.35 0.94
31.5 76.9 29.63 1.10
10.6 86.3 29.56 1.10
11.2 86.0 29.48 1.10
73.3 76.3 29.40 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
```

Gambar 4.3.1 isi file input 7

Berikut hasil eksekusi dari program

```

===== MENU Regresi Linear Berganda =====
Input dari terminal(1) atau file(2)?
->2
Masukkan nama file yang akan dibaca:
../test/7_input.txt
masukkan nilai x sejumlah 3 yang akan ditaksir nilainya
->50 76 29.3
hasil taksirannya adalah:
y = -3.51 + -0.00*x1 + 0.00*x2 + 0.15*x3
jika dimasukkan nilai x pada rumus tersebut, akan didapat hasilnya yaitu = 0.94
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.3.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

dengan nilai
x1 = 50.0
x2 = 76.0
x3 = 29.3

y = -3.5077781408835103 + -0.002624990745878327*x1 + 7.989410472218274E-4*x2 + 0.15415503019830143*x3 = 0.9384342262216645

```

Gambar 4.3.3 isi file output dari hasil eksekusi

D. Studi Kasus Matriks Augmented

Bagian ini berkorespondensi dengan studi kasus nomor dua. Diberikan dua buah input data berbentuk matriks augmented.

Berikut input pertama yang diberikan

```

1  -1  2  -1  -1
2  1  -2  -2  -2
-1  2  -4  1  1
3  0  0  -3  -3

```

Gambar 4.4.1 isi file input pertama

Berikut hasil eksekusi dari program


```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/2a_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 -1.000 -1.000
0.000 1.000 -2.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
SPL memiliki tak hingga solusi:
Misalkan:
x3 = c
x4 = d
Maka:
x1 = 1.000d + -1.000
x2 = 2.000c + 0.000
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.4.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

1.000 0.000 0.000 -1.000 -1.000
0.000 1.000 -2.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
SPL memiliki tak hingga solusi:
Misalkan:
x3 = c
x4 = d
Maka:
x1 = 1.000d + -1.000
x2 = 2.000c + 0.000

```

Gambar 4.4.3 isi file output dari hasil eksekusi

Berikut input kedua yang diberikan

```

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

```

Gambar 4.4.4 isi file input kedua

Berikut hasil eksekusi dari program

```

===== MENU UTAMA =====
1. Sistem Persamaan Linear
2. Determinan
3. Matriks Balikan
4. Interpolasi Polinom
5. Regresi Linear Berganda
6. Keluar
masukan pilihanmu
->1
===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/2b_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 2.000
0.000 0.000 1.000 0.000 1.000
0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
SPL memiliki solusi tunggal:
x1 = 0.000; x2 = 2.000; x3 = 1.000; x4 = 1.000;

Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.4.5 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

I-20049 / src / = tes.txt
1.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 2.000
0.000 0.000 1.000 0.000 1.000
0.000 0.000 0.000 1.000 1.000
0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000
SPL memiliki solusi tunggal:
x1 = 0.000; x2 = 2.000; x3 = 1.000; x4 = 1.000;

```

Gambar 4.4.6 isi file output dari hasil eksekusi

E. Studi Kasus SPL

Bagian ini berkorespondensi dengan studi kasus nomor tiga. Diberikan dua buah input data berbentuk persamaan. Data telah kami ubah terlebih dahulu input menjadi data yang dapat dibaca oleh komputer.

Berikut input pertama yang diberikan

```

8 1 3 2 0
2 9 -1 -2 1
1 3 2 -1 2
1 0 6 4 3

```

Gambar 4.5.1 isi file input pertama

Berikut hasil eksekusi dari program

```

71
===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/3a_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.000 -0.224
0.000 1.000 0.000 0.000 0.182
0.000 0.000 1.000 0.000 0.709
0.000 0.000 0.000 1.000 -0.258
SPL memiliki solusi tunggal:
x1 = -0.224; x2 = 0.182; x3 = 0.709; x4 = -0.258;

Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.5.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

1.000 0.000 0.000 0.000 -0.224
0.000 1.000 0.000 0.000 0.182
0.000 0.000 1.000 0.000 0.709
0.000 0.000 0.000 1.000 -0.258
SPL memiliki solusi tunggal:
x1 = -0.224; x2 = 0.182; x3 = 0.709; x4 = -0.258;

```

Gambar 4.5.3 isi file output dari hasil eksekusi

Berikut input kedua yang diberikan

```
0 0 0 0 0 0 1 1 1 13.00
0 0 0 1 1 1 0 0 0 15.00
1 1 1 0 0 0 0 0 0 8.00
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.04289 0.75 0.04289 0 0.04289 0 0 3.81
0 0 1 0 0 1 0 0 1 18.00
0 1 0 0 1 0 0 1 0 12.00
1 0 0 1 0 0 1 0 0 6.00
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04
```

Gambar 4.5.4 isi file input kedua

Berikut hasil eksekusi dari program

```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/3b_input.txt
Menukar baris 2 dengan 0
Menukar baris 4 dengan 1
Menukar baris 3 dengan 2
Menukar baris 4 dengan 3
Menukar baris 5 dengan 4
Menukar baris 6 dengan 5
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SPL tidak memiliki solusi
Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.5.5 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

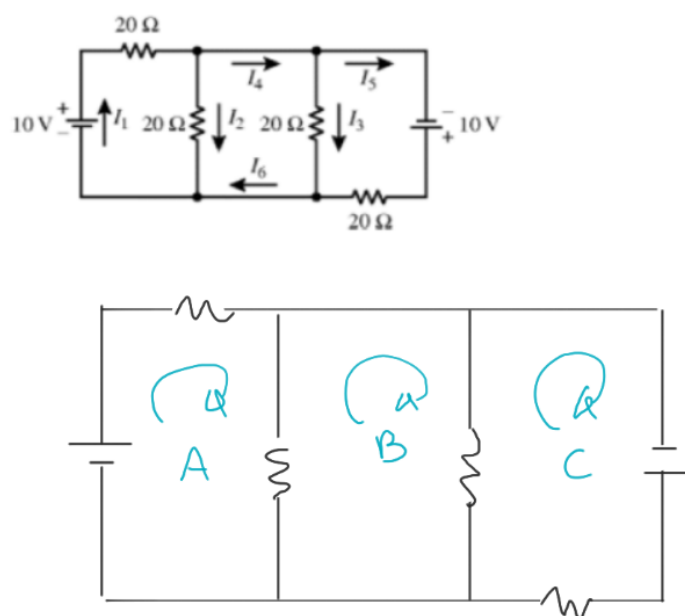
1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SPL tidak memiliki solusi

```

Gambar 4.5.6 isi file output dari hasil eksekusi

F. Studi Kasus Rangkaian Listrik

Bagian ini berkorespondensi dengan studi kasus nomor empat. Diberikan diagram rangkaian listrik. Data telah kami ubah terlebih dahulu input menjadi data yang dapat dibaca oleh komputer.



Gambar 4.6.1 ilustrasi penamaan Arus A, B, dan C

Kami cari terlebih dahulu persamaannya,

Pada Loop A,

$$40I_A - 20I_B = 10$$

Pada Loop B,

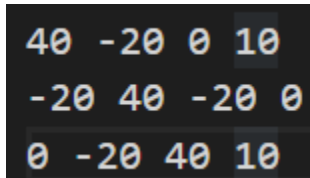
$$-20I_A + 40I_B - 20I_C = 0$$

Pada Loop C,

$$-20I_B + 40I_C = 10$$

Dari tiga persamaan itu, dibentuk file input.

Berikut input pertama yang diberikan



```
40 -20 0 10
-20 40 -20 0
0 -20 40 10
```

Gambar 4.6.2 isi file input pertama

Berikut hasil eksekusi dari program

```
===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/4_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 0.500
0.000 1.000 0.000 0.500
0.000 0.000 1.000 0.500
SPL memiliki solusi tunggal:
x1 = 0.500; x2 = 0.500; x3 = 0.500;

Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt
```

Gambar 4.6.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```
1.000 0.000 0.000 0.500
0.000 1.000 0.000 0.500
0.000 0.000 1.000 0.500
SPL memiliki solusi tunggal:
x1 = 0.500; x2 = 0.500; x3 = 0.500;
```

Gambar 4.6.3 isi file output dari hasil eksekusi

Didapat bahwa x_1 yaitu arus A bernilai 0.5 Ampere, x_2 yaitu arus B bernilai 0.5 Ampere, dan x_3 yaitu arus C bernilai 0.5 Ampere. Jika kita sesuaikan dengan yang diminta pada soal, akan didapat:

Arus Satu sebesar -0.5 Ampere,

Arus Dua sebesar 0 Ampere,

Arus Tiga sebesar 0 Ampere,

Arus Empat sebesar 0.5 Ampere,

Arus Lima sebesar 0.5 Ampere, terakhir

Arus Enam sebesar 0.5 Ampere.

G. Studi Kasus Sistem Reaktor

Bagian ini berkorespondensi dengan studi kasus nomor lima. Diberikan berbagai persamaan dengan variabel, dan juga nilai dari variabelnya. Data telah kami ubah terlebih dahulu input menjadi data yang dapat dibaca oleh komputer.

Berikut input pertama yang diberikan

```
-120 60 0 -1300
40 -80 0 0
80 20 -150 -200
```

Gambar 4.7.1 isi file input

Berikut hasil eksekusi dari program

```

===== MENU SPL =====
Pilih metode penyelesaian SPL:
1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer
->2
Pilih cara input matriks:
1. Dari console
2. Dari file
->2
Masukkan nama file yang akan dibaca:
->../test/5_input.txt
Matriks setelah dilakukan eliminasi Gauss-Jordan:
1.000 0.000 0.000 14.444
0.000 1.000 0.000 7.222
0.000 0.000 1.000 10.000
SPL memiliki solusi tunggal:
x1 = 14.444; x2 = 7.222; x3 = 10.000;

Apakah Anda ingin menyimpan hasil ke sebuah file? (y/n)
->y
Masukkan nama file (tanpa .txt dan jangan gunakan spasi): tes
File berhasil disimpan di tes.txt

```

Gambar 4.7.2 hasil eksekusi program

Berikut isi dari file tempat output disimpan

```

1.000 0.000 0.000 14.444
0.000 1.000 0.000 7.222
0.000 0.000 1.000 10.000
SPL memiliki solusi tunggal:
x1 = 14.444; x2 = 7.222; x3 = 10.000;

```

Gambar 4.7.3 isi file output dari hasil eksekusi

Pada output didapat X_1 , X_2 , dan X_3 , sedangkan soal menggunakan X_a , X_b , dan X_c . X_1 pada output merupakan X_a , X_2 merupakan X_b , X_3 merupakan X_c .

Bab 5

Kesimpulan, saran, dan refleksi

A. Kesimpulan

Hasil yang dicapai dari pengerjaan tugas besar ini adalah pembuatan program pemrosesan matriks yang mampu menyelesaikan 5 persoalan utama, yaitu sistem persamaan linier, determinan, matriks balikan, interpolasi polinom, dan regresi linier berganda. Penyelesaian persoalan sistem persamaan linier dapat menggunakan empat buah metode berbeda, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer. Untuk persoalan determinan, metode-metode yang digunakan antara lain metode reduksi dan metode kofaktor. Persoalan matriks balikan dapat diselesaikan menggunakan metode OBE dan metode adjoin. Melalui pembuatan program ini, kami banyak belajar tentang pemrograman khususnya menggunakan bahasa Java yang bersifat pemrograman berorientasi objek.

B. Saran

Dalam pengerjaan tugas besar ini, ada beberapa saran yang didapatkan. Pertama, sebaiknya file spesifikasi tugas besar dikelompokkan berdasarkan kelompok-kelompok fitur utama yang perlu dibuat untuk memudahkan pembacaan. Misalnya, untuk kasus tugas besar ini, spesifikasi dapat dibagi ke dalam 5 bagian utama, yaitu sistem persamaan linier, determinan, matriks balikan, interpolasi polinom, dan regresi linier berganda yang masing-masing berisi spesifikasi yang lebih detail. Selain itu, deadline pengerjaan tugas besar juga dapat dibuat lebih lama.

C. Refleksi

Setelah menyelesaikan tugas besar ini, kami memperoleh banyak pelajaran yang akan sangat berguna bagi kehidupan dan karir kami. Yang pertama pastilah pembelajaran bahasa Java. Melalui tugas besar ini, kami bertiga mendapatkan pengalaman berharga yaitu langsung terjun memprogram menggunakan bahasa Java yang sama sekali belum pernah kami lakukan sebelumnya. Selain dari segi pemrograman, kami juga banyak belajar mengenai matriks dan operasi-operasi yang kami buat dalam tugas besar ini. Setelah mengimplementasikan hal-hal yang kami pelajari di kelas aljabar linier dan geometri tentang matriks, kami menjadi lebih paham dengan cara kerjanya serta dapat mengotomasi proses-proses tersebut. Terakhir, kami juga belajar bagaimana caranya berkoordinasi dan bekerja dalam tim secara efektif. *Soft skill* yang kami dapatkan antara lain kerja sama, komunikasi, *leadership*, bekerja dengan deadline, dan masih banyak lagi. Kemudian, untuk *hard skill*, kami juga belajar bagaimana cara menggunakan berbagai macam alat seperti Github, Visual Studio Code, hingga linux dan cara memecahkan masalah pada alat-alat tersebut.

Referensi

Belajar Statistik. 2019. *Multiple Linear Regression: Assumptions, Coefficients, and Problem Solving using Matrix*. Diakses pada 26 September 2021, dari <https://www.youtube.com/watch?v=kHBdNdINs8k&t=421s>

Iver Mølgaard Ottosen. 2011. *The Row Reduction Algorithm*. Diakses pada 20 September 2021, dari <https://people.math.aau.dk/~ottosen/MMA2011/rralg.html>

Rinaldi Munir. 2021. *Slide Seri Bahan Kuliah Algeo*. Diakses pada 23 - 29 September 2021, dari <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/algeo21-22.htm>