Laporan Tugas Besar 1 IF2123 - Aljabar Linier dan Geometri

Disusun oleh:

"AldyDPP"

Renaldy Arief Susanto (13522022)

Abdul Rafi Radityo Hutomo (13522089)

INSTITUT TEKNOLOGI BANDUNG

TAHUN AJARAN 2023/2024

DAFTAR ISI

Halaman

BAB 1

DESKRIPSI MASALAH

1. Buatlah pustaka (library atau package) 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.

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

Spesifikasi program adalah sebagai berikut:

1. Program dapat menerima masukan (input) baik dari keyboard maupun membaca masukan dari file text. Untuk SPL, masukan dari keyboard adalah m, n, koefisien aij , dan bi. Masukan dari file berbentuk matriks augmented tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

```
3 4.5 2.8 10 12

-3 7 8.3 11 -4

0.5 -10 -9 12 0
```

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

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

luaran (output) dari determinan dan matriks balikan disesuaikan dengan persoalan masing-masing.

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0), (x1, y1), …, (xn, yn), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

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

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), m (jumlah sampel), semua nilai-nilai x1i, x2i, …, xni, nilai yi, dan nilai-nilai xk yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran 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 x4 = -2, x3 = 2s – t, x2 = s, dan x1 = t).

6. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan. Contoh luaran untuk interpolasi adalah sebagai berikut.

$$f(x) = -0.0064x^2 + 0.2266x + 0.6762,\ f(5) = \ldots$$

dan untuk regresi adalah

$$f(x) = -9.5872 + 1.0732x1,\ f(x_k) = \ldots$$

7. Untuk persoalan bicubic spline interpolation, masukan dari file text (.txt) yang berisi matriks berukuran 4 x 4 yang berisi konfigurasi nilai fungsi dan turunan berarah disekitarnya, diikuti dengan nilai a dan b untuk mencari nilai f(a, b).

   Misalnya jika nilai dari f(0, 0), f(1, 0), f(0, 1), f(1, 1), $f_x(0, 0)$, $f_x(1, 0)$, $f_x(0, 1)$, $f_x(1, 1)$, $f_y(0, 0)$, $f_y(1, 0)$, $f_y(0, 1)$, $f_y(1, 1)$, $f_{xy}(0, 0)$, $f_{xy}(1, 0)$, $f_{xy}(0, 1)$, $f_{xy}(1, 1)$ berturut-turut adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 serta nilai a dan b yang dicari berturut-turut adalah 0.5 dan 0.5 maka isi file text ditulis sebagai berikut:

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

8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.
9. Bahasa program yang digunakan adalah Java. Dibebaskan untuk menggunakan versi java apapun dengan catatan di atas java versi 8 (8/9/11/15/17/19/20).
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 Persamaaan 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

DASAR TEORI

1. Abstraksi

Matriks adalah sebuah tipe data yang dapat digunakan untuk menyimpan ($m \times n$) buah data, dengan penulisannya seabgai berikut.

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Gambar 2.1.1 Ilustrasi Matriks

a_{ij} adalah elemen baris ke-*i* kolom ke-*j*. Matriks kemudian mempunyai beberapa operasi-operasi yang dapat dilakukan, seperti penjumlahan, pengurangan, perkalian, dan transpose. Matriks dapat digunakan untuk merepresentasikan sebuah sistem persamaan linier. Secara umum, bentuknya sebagai berikut.

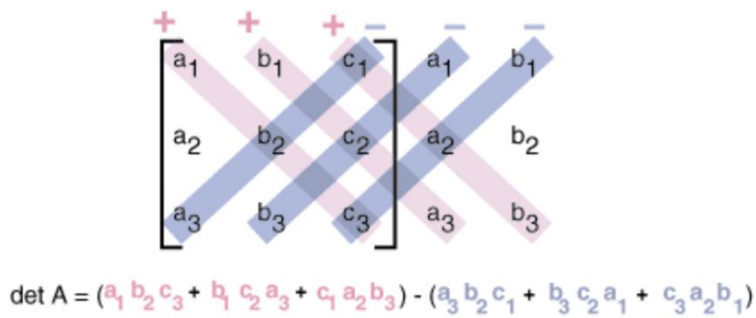
$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array}$$

Gambar 2.1.2 Sistem Persamaan Linear

x_i adalah variabel ke-*i*, dan b_i adalah hasil linier ke-*i*. Kemudian, dapat dilakukan operasi-operasi pada matriks ini untuk mencari nilai-nilai x_i. Ada beberapa cara untuk menentukan solusinya, di antaranya yang diimplementasikan dalam program ini adalah metode eliminasi Gauss, eliminasi Gauss-Jordan, metode matriks invers, serta metode Cramer.

2. Matriks Persegi, Determinan, dan Matriks Invers/Balikan

Determinan sebuah matriks hanya terdefinisikan apabila matriks berukuran persegi ($n \times n$). Determinan matriks mempunyai banyak sekali penggunaan dalam perhitungan/operasi matriks dan dapat dihitung dengan menggunakan cara sebagai berikut.



Gambar 2.2.1 Determinan Matriks

Determinan dapat digunakan untuk menghitung matriks invers, yaitu matriks yang apabila dikalikan dengan matriks aslinya akan menghasilkan matriks identitas. Apabila determinan sebuah matriks bernilai 0, matriks tersebut tidak mempunyai invers.

3. Operasi Baris Elementer (OBE), Metode Eliminasi Gauss, dan Metode Eliminasi Gauss-Jordan

Operasi baris elementer (OBE) merupakan operasi aritmatika (penjumlahan dan perkalian) yang dikenakan pada setiap unsur dalam suatu baris pada sebuah matriks.

Operasi baris elementer meliputi :

1. Pertukaran Baris
2. Perkalian suatu baris dengan konstanta tak nol

3. Penjumlahan hasil perkalian suatu baris dengan konstanta tak nol (seperti buti dengan baris yang lain.

Tujuan dilakukan operasi baris elementer pada suatu matriks adalah menghasilkan matriks yang memenuhi beberapa sifat berikut:

1. Pada baris tak nol maka unsur tak nol pertama adalah 1 (membuat satu utama).
2. Pada baris yang berurutan, baris yang lebih rendah memuat 1 utama yang lebih ke kanan.
3. Jika ada baris nol (baris yang semua unsurnya nol), maka ia diletakkan pada baris paling bawah.
4. Pada kolom yang memuat unsur 1 utama, maka unsur yang lainnya adalah nol.

Jika poin 1, 2, dan 3 dipenuhi, matriks hasil OBE dinamakan berbentuk eselon baris (hasil dari eliminasi Gauss). Sementara itu, jika semua poin dipenuhi matriks dinamakan berbentuk eselon baris tereduksi (dinamakan eliminasi Gauss-Jordan).

4. Invers Matriks untuk Mencari Solusi SPL

Solusi dari sistem persamaan linear dalam bentuk Ax = B dapat diselesaikan menggunakan invers matriks, dengan A adalah matriks persegi berisi koefisien x dan B matriks berisi hasil penjumlahan koefisien x_i * a_i. Matriks x yang berisi solusi x_1 hingga x_n didapat dengan cara mengalikan kedua ruas dengan A^{-1} sehingga didapat persamaan $x = A^{-1}B$.

5. Kaidah Cramer

Kaidah Cramer menyatakan bahwa jika Ax = b adalah sebuah sistem n persamaan linear dengan n variabel yang tidak diketahui sehingga $\det(A) \neq 0$, maka sistem ini memiliki solusi unik :

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

Dengan A_j didapat dengan mengganti kolom ke-j pada matriks A dengan matriks B.

6. Interpolasi Polinomial

Interpolasi polinomial merupakan persoalan untuk menentukan polinom $p_n(x)$ yang menginterpolasi(melewat) semua titik-titik tersebut. Polinomial yang didapat akan berderajat n + 1, dengan n adalah jumlah titik yang didapat. Persoalan interpolasi polinomial dapat diselesaikan dengan membentuk sistem persamaan linear yang dapat direpresentasikan dalam bentuk matriks sebagai berikut.

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & \dots & x_1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & x_n^{n-2} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Gambar 2.6.1 Ilustrasi SPL Interpolasi Polinomial dalam bentuk Matriks

Kemudian, Sistem Persamaan Linear tersebut dapat diselesaikan dengan metode yang sebelumnya sudah dijelaskan.

7. Regresi Linear Berganda

Regrese Linear Berganda merupakan persoalan untuk menentukan persamaan yang menghampiri nilai sebuah variabel y yang bergantung terhadap 1 atau lebih variabel x_i. Dengan menggunakan estimasi Ordinary Least Squares (OLS), persoalan ini dapat diselesaikan dengan membentuk sebuah sistem persamaan linear sebagai berikut

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

Gambar 2.7.1 Persamaan SPL OLS

atau dapat direpresentasikan dalam bentuk matriks seperti berikut

$$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)\hat{\boldsymbol{\beta}} = \mathbf{X}^{\mathrm{T}}\mathbf{y}\ .$$

Gambar 2.7.2 Persamaan Dalam Bentuk Matriks

dengan isi matriks

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

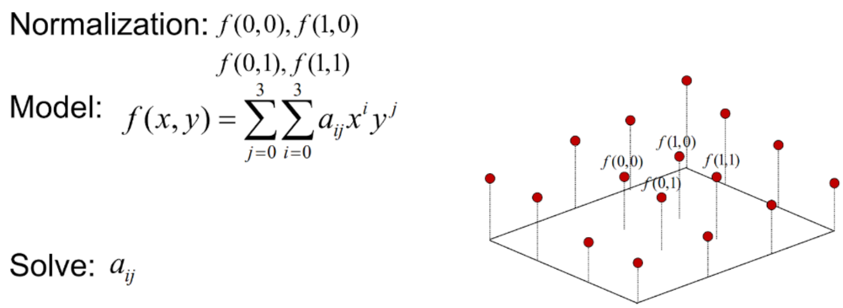
Gambar 2.7.3 Isi Matriks Pada Gambar 2.7.2

dengan p adalah jumlah variabel yang ada (regressor) dan p adalah jumlah sampel dan β adalah koefisien dari masing-masing x dengan kolom pertama pada matrix X berisi 1.

8. Bicubic Spline Interpolation

Bicubic spline interpolation adalah metode interpolasi yang digunakan untuk mengaproksimasi fungsi di antara titik-titik data yang diketahui. Bicubic spline interpolation melibatkan konsep spline dan konstruksi serangkaian polinomial kubik di dalam setiap sel segi empat dari data yang diberikan. Pendekatan ini menciptakan permukaan yang halus dan kontinu, memungkinkan untuk perluasan data secara visual yang lebih akurat daripada metode interpolasi linear.

Dalam pemrosesan menggunakan interpolasi bicubic spline digunakan 16 buah titik, 4 titik referensi utama di bagian pusat, dan 12 titik di sekitarnya sebagai aproksimasi turunan dari keempat titik referensi untuk membagun permukaan bikubik. Bentuk pemodelannya adalah sebagai berikut.



Gambar 2.8.1 Illustrasi Bicubic Spline Interpolation

Selain melibatkan model dasar, juga digunakan model turunan berarah dari kedua sumbu, baik terhadap sumbu x, sumbu y, mapun keduanya. Persamaan polinomial yang digunakan adalah sebagai berikut.

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

$$f_x(x,y)=\sum_{j=0}^{3}\sum_{i=1}^{3} a_{ij}ix^{i-1} y^j$$

$$f_y(x,y)=\sum_{j=1}^{3}\sum_{i=0}^{3} a_{ij}jx^i y^{j-1}$$

$$f_{xy}(x,y)=\sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij}ijx^{i-1} y^{j-1}$$

Gambar 2.8.2 Persamaan untuk nilai f beserta turunan berarahnya

Dengan menggunakan nilai fungsi dan turunan berarah tersebut, dapat terbentuk sebuah matriks solusi X yang membentuk persamaan penyelesaian sebagai berikut.

$$y = Xa$$

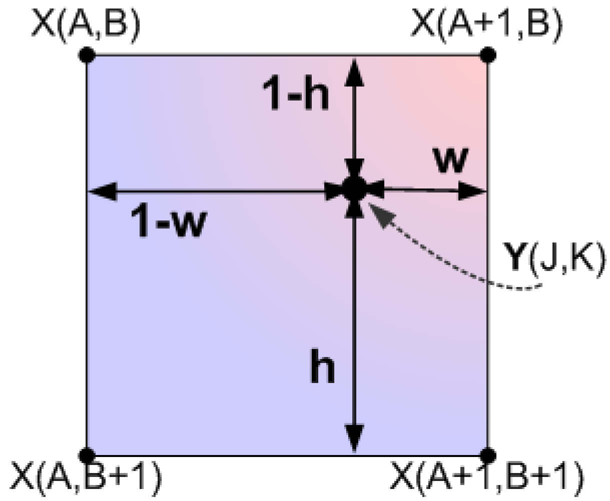
$$
\begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \\ f_x(0,0) \\ f_x(1,0) \\ f_x(0,1) \\ f_x(1,1) \\ f_y(0,0) \\ f_y(1,0) \\ f_y(0,1) \\ f_y(1,1) \\ f_{xy}(0,0) \\ f_{xy}(1,0) \\ f_{xy}(0,1) \\ f_{xy}(1,1) \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 4 & 6 & 0 & 3 & 6 & 9
\end{bmatrix}
\begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}
$$

Gambar 2.8.3 Persamaan bicubic spline interpolation dalam bentuk matriks

Nilai dari vektor a dapat dicari dari persamaan y = Xa, lalu vektor a tersebut digunakan sebagai nilai variabel dalam f(x, y), sehingga terbentuk fungsi interpolasi bicubic sesuai model.

9. Perbesaran Citra menggunakan Bicubic Spline Interpolation

Metode bicubic spline interpolation dapat digunakan untuk menghampiri nilai fungsi 2 dimensi. Salah satu aplikasinya adalah pada pemrosesan citra gambar 2D. Dengan menggunakan bicubic spline interpolation, dapat “diselipkan” pixel baru di antara pixel-pixel yang nilainya didapat dari hasil interpolasi, seperti pada illustrasi yang memperkirakan nilai pixel di Y(J, K) yang di dalam sebuah persegi yang terdiri atas 4 pixel lainnya.



Gambar 2.9.1 Ilustrasi Perkiraan Nilai Suatu Pixel

Dengan menggunakan nilai 12 pixel lain di sekitar 4 pixel utama pada gambar, nilai turunan berarah terhadap sumbu x, sumbu y, dan sumbu xy dapat diperkirakan melalui persamaan berikut.

$$f_x(A,B) = \frac{f(A+1,B) - f(A-1,B)}{2}$$

$$f_y(A,B) = \frac{f(A\ B+1) - f(A,B-1)}{2}$$

$$f_{xy}(A,B) = \frac{\big(f(A-1,B-1) + f(A+1,B+1)\big) - \big(f(A+1,B-1) + f(A-1,B+1)\big)}{4}$$

Persamaan tersebut dapat digabungkan dengan persamaan bicubic spline interpolation untuk mendapatkan persamaan f(x, y) untuk memperkirakan nilai pixel di dalam persegi yang dibentuk oleh 4 pixel.

BAB 3

IMPLEMENTASI PROGRAM DALAM JAVA

Pada tugas besar ini, program dibagi menjadi beberapa file dengan masing-masing file berisi sebuah kelas.

1. Matrix.java

File Matrix.java mendeklarakasikan kelas Matrix yang merupakan kelas yang merepresentasikan sebuah matriks secara umum dengan operasi-operasi dasar pada matriks.

Kelas Matrix dideklarasikan dengan atribut:

a) int rowCount : jumlah Baris pada Matriks
b) int columnCount : jumlah Kolom pada Matriks
c) double[][] contents : Isi dari matriks dengan setiap subarray merupakan sebuah baris

Dalam kelas Matriks diimplementasikan Class Method :

No	Method	Deskripsi
1	mutliplyMatrix(M1, M2)	Mengembalikan hasil perkalian antara M1 dengan M2

Dalam Kelas Matrix diimplementasikan Instance Method :

No	Method	Deskripsi
1	Matrix()	Konstruktor Matrix

2	initializeMatrix(i, j)	Mengassign atribut matriks dengan jumlah baris i dan jumlah kolom j dengan isi atribut yang sesuai
3	keyboardInputMatrix(sc)	Menerima input matrix melalui terminal
4	txtInputMatrix(path)	Membaca matrix dari txt file
5	printMatrix()	Men display isi matriks ke terminal
6	toString()	Mengembalikan isi matriks dalam bentuk string yang sudah diformat
7	transposeOf()	Mengembalikan matriks hasil transpose dari instance
8	transpose()	Mentranspose instance
9	scalarMultiply(x)	Mengalikan skalar diri sendiri dengan faktor x

2. SPLMatrix.java

File SPLMatrix.java mendeklarakasikan kelas SPLMatrix yang di extend dari kelas Matrix. Kelas ini merupakan kelas yang merepresentasikan sebuah matriks yang sudah terspesialisasi dalam menyelesaikan sebuah SPL.

Kelas SPLMatrix memiliki atribut yang sama dengan kelas matriks tanpa ada tambahan.

Dalam kelas SPLMatriks diimplementasikan Class Method :

No	Method	Deskripsi
1	augmentABMatrix(A, B)	Mengembalikan augmented SPLMatrix dari Matriks koefisien A dan Matrix hasil B

Dalam Kelas SPLMatrix diimplementasikan Instance Method :

No	Method	Deskripsi

1	SPLMatrix()	Konstruktor Matrix
2	multiplyERO(rowNum, factor)	Melakukan OBE pengalian baris ke rowNum dengan faktor factor
3	switchERO(row1, row2)	Melakukan OBE penukaran row1 dengan row2
4	addERO(row1, row2, factor)	Melakukan OBE penjumlahan elemen pada row1 dengan elemen pada row2 dikalikan dengan faktor
5	reduceToEchelon()	Mereduksi diri menjadi matrix eselon
6	echelonToReducedEchelon()	Mereduksi diri sendiri (yang telah direduksi menjadi matriks eselon) menjadi matriks eselon tereduksi.
7	AugmentedSols()	Mengembalikan array of SolutionType berupa solusi dari SPL dengan initial state matriks telah tereduksi menjadi matriks eselon atau eselon tereduksi.
8	getAugmentedSols()	Mengembalikan solusi dari SPL dalam bentuk String dengan initial state matriks telah tereduksi menjadi matriks eselon atau eselon tereduksi.
9	solveFromScratch()	Mengembalikan String solusi dari SPL menggunakan metode eliminasi gauss jordan dengan initial state matrix telah diisi
10	GaussSolve()	Mengembalikan String solusi dari SPL menggunakan metode eliminasi gauss dengan initial state matrix telah diisi
11	GaussJordanSolve()	Mengembalikan String solusi dari SPL menggunakan metode eliminasi gauss - jordan dengan initial state matrix telah diisi

12	inverseMatrixSolve()	Mengembalikan String solusi dari SPL menggunakan invers matriks dengan initial state matrix telah diisi dan diasumsikan matriks dapat diinvers
13	cramerSolve()	Mengembalikan String solusi dari SPL menggunakan kaidah cramer dengan initial state matrix telah diisi dan diasumsikan matriks dapat diselesaikan dengan kaidah cramer
14	getMatrixA()	Mengembalikan matriks koefisien dari augmented matriks
15	getMatrixB()	Mengembalikan matriks hasil penjumlahan dari augmented matriks

3. SolutionType.java

File SolutionType.java mendeklarakasikan kelas SolutionType. Kelas ini merupakan kelas yang merepresentasikan setiap komponen solusi dari sebuah SPL.

Kelas SolutionType dideklarasikan dengan atribut :

a) double realPart : bagian dari solusi yang bernilai real
b) HashMap<Character, Double> parametricParts : bagian dari solusi yang bersifat parametrik direpresentasikan sebagai hashmap dengan character dimap ke nilai koefisiennya
c) int length : panjang persamaan

Dalam kelas SolutionType diimplementasikan Class Method :

No	Method	Deskripsi
1	add(x1, x2, factor)	Mengembalikan solution type hasil penjumlahan antara x1 dengan x2*factor

Dalam Kelas SolutionType diimplementasikan Instance Method :

No	Method	Deskripsi
1	SolutionType(n)	Konstruktor Matrix, menginisialisasi atribut
2	setRealPart(realPart)	Menjadikan nilai atribut realPart menjadi argumen realPart
3	getRealPart()	Getter atribut realPart
4	getParametricParts()	Getter atribut parametricParts
5	toString()	Mengembalikan representasi diri dalam bentuk String

4. SquareMatrix.java

File SquareMatrix.java mendeklarakasikan kelas SquareMatrix yang diextend dari kelas SPLMatrix. Kelas ini merupakan kelas yang merepresentasikan sebuah matriks berbentuk persegi dan berisi method-method untuk sebuah matriks persegi.

Kelas SquareMatrix dideklarasikan dengan atribut yang sama dengan SPLMatrix dengan tambahan :

a) SPLMatrix MatrixI : Matrix Identitas untuk membantu perhitungan invers Matrix
b) Double Determinant : Determinan matriks yang dinisialisasi dengan nilai 1
c) boolean isInvertible : Predikat invertibilitas dari matriks

Dalam Kelas SquareMatrix diimplementasikan Instance Method :

No	Method	Deskripsi
1	SquareMatrix()	Konstruktor SquareMatrix
2	inverseByERO()	Mengembalikan invers matriks dari diri sendiri dengan metode OBE

3	solsByInverse(Matrix B)	Mengembalikan Matrix Solusi dari persamaan Ax = B, dengan A adalah diri sendiri menggunakan invers matriks
4	getSolsByInverse(Matrix B)	Mengembalikan solusi dari persamaan AX = B, dengan A adalah diri sendiri dalam bentuk string menggunakan invers matriks
5	determinantByERO()	Mengembalikan determinan diri sendiri dengan metode OBE
6	determinantByCofactor()	Mengembalikan determinan diri sendiri dengan ekspansi Kofaktor menggunakan baris pertama sebagai acuan
7	InverseByCofactor()	Mengembalikan Invers Matriks diri sendiri dengan menggunakan Kofaktor
8	solsByCramer(Matrix B)	Mengembalikan Matrix Solusi dari persamaan Ax = B, dengan A adalah diri sendiri menggunakan kaidah Cramer
9	getSolsByCramer(Matrix B)	Mengembalikan solusi dari persamaan AX = B, dengan A adalah diri sendiri dalam bentuk string dengan menggunakan kaidah cramer
10	transposeOf()	Mengembalikan SquareMatrix hasil transpos dari diri sendiri
11	checkInvertibility()	Menentukan predikat dari atribut isInvertible

5. InterpolationSolver.java

File InterpolationSolver.java mendeklarakasikan kelas InterpolationSolver. Kelas ini merupakan kelas yang dideklarasikan untuk menyelesaikan permasalahan terkait.

Kelas InterpolationSolver dideklarasikan dengan atribut:

a) double[][] xyValue : kumpulan pasangan nilai (x, y) yang disimpan dalam array 2 dimensi
b) int degree : Derajat dari polinomial hasil interpolasi
c) SPLMatrix coefficientSPL : SPLMatrix berisi persamaan SPL interpolasi
d) SolutionType[] coefficients : Array berisi SolutionType yang merupakan koefisien dari x^n
e) Double aprox : berisi nilai x yang ingin didekati nilai y nya

Dalam Kelas InterpolationSolver diimplementasikan Instance Method :

No	Method	Deskripsi
1	InterpolationSolver()	Konstruktor InterpolationSolver
2	keyboardInput(Scanner sc)	Melakukan input titik-titik yang ingin diinterpolasi dari terminal
3	textInput(String path)	Membaca format titik-titik yang ingin diinterpolasi melalui file txt sesuai dengan path yang diberikan
4	generateMatrix()	Membuat SPL dari nilai xy yang telah diinput
5	generateSolution()	Menghitung koefisien hasil interpolasi
6	solveAndPrint()	Menyelesaikan masalah interpolasi dan mengoutput hasilnya pada terminal dalam bentuk fungsi
7	solveAndReturnString()	Menyelesaikan masalah interpolasi dan mengembalikan hasilnya dalam bentuk String

6. MultivariateLinearRegressionSolver.java

File MultivariateLinearRegressionSolver.java mendeklarakasikan kelas MultivariateLinearRegressionSolver. Kelas ini merupakan kelas yang dideklarasikan untuk menyelesaikan permasalahan terkait.

Kelas MultivariateLinearRegressionSolver dideklarasikan dengan atribut:

a) int n : Jumlah regressor
b) int m : Jumlah sampel
c) SPLMatrix xyMatrix : Matriks berisi nilai-nilai x dan y
d) SPLMatrix OLSEquationMatrix : Matrix berisi SPL persamaan OLS

Dalam Kelas InterpolationSolver diimplementasikan Instance Method :

No	Method	Deskripsi
1	InterpolationSolver()	Konstruktor InterpolationSolver
2	keyboardInput(Scanner sc)	Melakukan input titik-titik yang ingin diregresikan dari terminal
3	textInput(String path)	Membaca format titik-titik yang ingin diregresikan melalui file txt sesuai dengan path yang diberikan
4	constructOLSMatrix()	Membangun OLSEquationMatrix dari nilai-nilai x dan y yang sudah diinput
5	getBetaMatrix()	Mengembalikan Array of SolutionType berisi koefisien-koefisien dari hasil Regresi Linear Berganda
6	calculateAndPrint()	Melakukan regresi linear berganda dan mengoutput hasilnya melalui terminal
7	calculateAndGetString()	Melakukan regresi linear berganda dan mengembalikan hasilnya sebagai String

7. BicubicSpline.java

File BicubicSpline.java mendeklarakasikan kelas BicubicSpline. Kelas ini merupakan kelas yang dideklarasikan untuk menyelesaikan permasalahan terkait.

Kelas BicubicSpline dideklarasikan dengan atribut:

a) SquareMatrix X : Matriks 16x16 yang didefinisikan dalam persamaan bicubic spline
b) Matrix Y : Matriks berisi nilai-nilai f(a, b) dengan kombinasi a, b = 0 atau 1 beserta turunan berarahnya
c) Matrix a : Matriks berisi nilai koefisien aij yang didefinisikan pada persamaan bicubic spline
d) double[] target : Array berisi titik yang ingin diperkirakan nilainya

Dalam Kelas InterpolationSolver diimplementasikan Instance Method :

No	Method	Deskripsi
1	BicubicSpline()	Konstruktor BicubicSpline
2	keyboardInput(Scanner sc)	Melakukan input titik-titik yang ingin diinterpolasi dari terminal
3	textInput(String path)	Membaca format titik-titik yang ingin diinterpolasi melalui file txt sesuai dengan path yang diberikan
4	createXMatrix()	Membangun X Matrix yang didefinisikan pada persamaan Bicubic Spline Interpolation
5	evaluateA()	Menghitung array a yang berisi koefisien aij yang didefinisikan pada persamaan bicubic spline
6	calculateVal(x, y)	Menghitung nilai hasil interpolasi pada titik x, y
7	splineAndPrint()	Melakukan output hasil interpolasi terhadap target pada terminal

8	splineAndReturnString()	Mengembalikan output interpolasi terhadap target sebagai String

7. Bonus.java dan BonusRGB.java

File Bonus.java dan BonusRGB.java mendeklarakasikan kelas Bonus dan BonusRGB. Kedua kelas ini merupakan kelas yang dideklarasikan untuk menyelesaikan permasalahan terkait. Kedua kelas yang dideklarasikan nyaris identik dengan beberapa penyesuaian dalam mengolah data warna. Pada bagian ini akan dibahas untuk kelas BonusRGB saja dan kelas Bonus tidak dibahas karena sangat mirip.

Kelas BonusRGB dideklarasikan dengan atribut:

a) SquareMatrix D : Matriks 16x16 yang didapat dari nilai pixel dan persamaan perkiraan turunannya
b) SquareMatrix X : Matrix 16x16 yang didefinisikan dari persamaan bicubic spline
c) double zoomFactor : Faktor perbesaran gambar
d) double[][][] originPicture : Matrix berisi Array[0..2] of double yang berisi nilai warna pada masing-masing pixel pada gambar yang ingin diproses
e) int originHeight : tinggi gambar yang ingin diproses
f) int originWidth : lebar gambar yang ingin diproses
g) double[][][] enlargedPicture : Matrix berisi Array[0..2] of double yang berisi nilai warna pada masing-masing pixel pada gambar yang telah diperbesar
h) double xBuffer : variabel untuk membantu alur pemrosesan pixel
i) double yBuffer : variabel untuk membantu alur pemrosesan pixel

Dalam Kelas BonusRGB diimplementasikan Instance Method :

No	Method	Deskripsi

1	loadPixelsToGrayscale()	Mendapatkan data warna-warna pixel dari gambar
2	constructDMatrix()	Membangun Matrix 16x16 D yang didefinisikan pada persamaan.
3	createXMatrix()	Membangun Matrix 16x16 X yang didefinisikan pada persamaan bicubic spline interpolation
4	createResArray()	Menginisialisasi matriks hasil perbesaran gambar
5	workOnFourPixels(x, y, k)	Memproses 4 titik sebagai acuan untuk interpolasi dengan titik (x, y) berada di kiri atas atau berperan sebagai titik (0, 0)
6	workOnAllPixels()	Menggilir pemrosesan 4 titik hingga seluruh titik telah diproses
7	writeImage(String Name)	Merubah nilai warna pixel pada array enlargedPicture ke dalam file png
8	getImageAndZoom(Path, factor, resPath)	Memperbesar image pada Path sebesar factor dan menghasilkan output berupa file png yang telah diperbesar pada resPath.

8. Konsep GUI

Pada folder src, terdapat sebuah folder bernama gui yang berisi semua class yang menampilkan gui program. Class tidak akan dibahas secara mendetail satu per satu, tetapi akan dibahas secara garis besar alur flow program dan macam-macam class yang ada. Berikut adalah list *bytecode* java yang telah dibuat untuk GUI dan penjelasannya.

8.1. GUI.java

Class ini serupa dengan program “main”. Di dalamnya hanya terdapat perintah untuk membuat LaunchPage baru, yaitu hal pertama yang dilakukan ketika memulai aplikasi.

8.2. LaunchPage.java

Ini adalah class yang memuat halaman utama program yang muncul pertama ketika menjalankan aplikasi. Pada halam ini, pengguna dapat memilih kalkulator mana yang akan digunakan (SPL, Inverse, dsb.).

8.3. TemplateFrame.java, TemplateButton.java, dan TemplateField.java

TemplateFrame adalah halaman dasar yang digunakan untuk membangun halaman-halaman lain. **Semua class halaman berupa ekstensi dari halaman ini.** Begitu rupanya dengan TemplateButton.java dan TemplateField.java, yaitu secara berurutan tombol dan kolom input dasar yang digunakan sebagai basis ketika meng-construct tombol/kolom input. Ketiga *class* template ini berisi ciri ekstrinsik yang akan berlaku untuk semua *class* ekstensinya, seperti visibility, ukuran, resizeability, dsb.

8.4. ErrorPage.java

Ini adalah class halaman error, parameternya berupa pesan error dan dipanggil apabila pengguna perlu dilempar ke halaman error dan akan ditampilkan pesan error yang sesuai.

8.5. TxtInputPage.java, TxtOutputPage.java, dan TxtOutputPage2.java

Ketiga class ini adalah halaman untuk menerima dan menulis file txt.Pengguna akan diminta untuk memasukkan *path* file lengkap untuk input, tetapi hanya nama file saja untuk output. Output txt file akan diletakkan dalam folder “res” yang akan dibuat secara otomatis pada folder yang sama. TxtInputPage akan dipanggil ketika pengguna memilih “Input with txt file instead” pada tiap-tiap halaman kalkulator.

8.6. HeaderText.java, NextTemplateButton.java, dan ExitTemplateButton.java

HeaderText adalah class yang berupa objek JLabel yang memuat judul halaman. Metode diri nya dipanggil pada sebagian besar class untuk menempelkan judul pada halaman. NextTemplateButton.java, dan ExitTemplateButton.java adalah ekstensi dari TemplateButton.java untuk secara berturut-turut mempermudah konstruksi exit button dan next button pada setiap page.

8.7. Inverse1.java - Inverse4.java.

Keempat halaman ini menagani input-output untuk kalkulasi matriks balikan. Apabila pengguna memasukkan matriks yang tidak mempunyai invers, pengguna dilempar ke halaman determinan dengan hasil 0 dengan pesan "matriks ini tidak punya invers"

8.8. Determimant1.java - Determimant4.java.

Keempat halaman ini menagani input-output untuk kalkulasi determinan.

8.9. SPL1.java - SPL4.java.

Keempat halaman ini menangani input-output untuk kalkulasi sistem persamaan linier. Apabila sistem persamaan linier dipilih untuk diselesaikan menggunakan metode Cramer atau matriks invers, sementara matriks non-*augemented* nya tidak berukuran persegi, pengguna akan dilempar ke halaman error dengan pesan yang sesuai.

8.10. PolinomeInterpolation1.java - PolinomeInterpolation3.java

Ketiga halaman ini menangani input-output untuk kalkulasi sistem persamaan linier

8.11. BicubicSpline1.java - BicubicSpline3.java.

Ketiga halaman ini menangani input-output untuk kalkulasi interpolasi Bicubic Spline

8.12. LinReg1.java - LinReg3.java.

Ketiga halaman ini menangani input-output untuk kalkulasi regresi linier multivariabel

8.13. **BONUS.java, BONUS2.java, BONUS3.java**

Ketiga halaman ini menangani input-output untuk perbesaran *image*. Pengguna akan diminta memasukkan *input path* dari *image* yang dipilih, *output path*, *zoom factor*, dan juga nama file output.

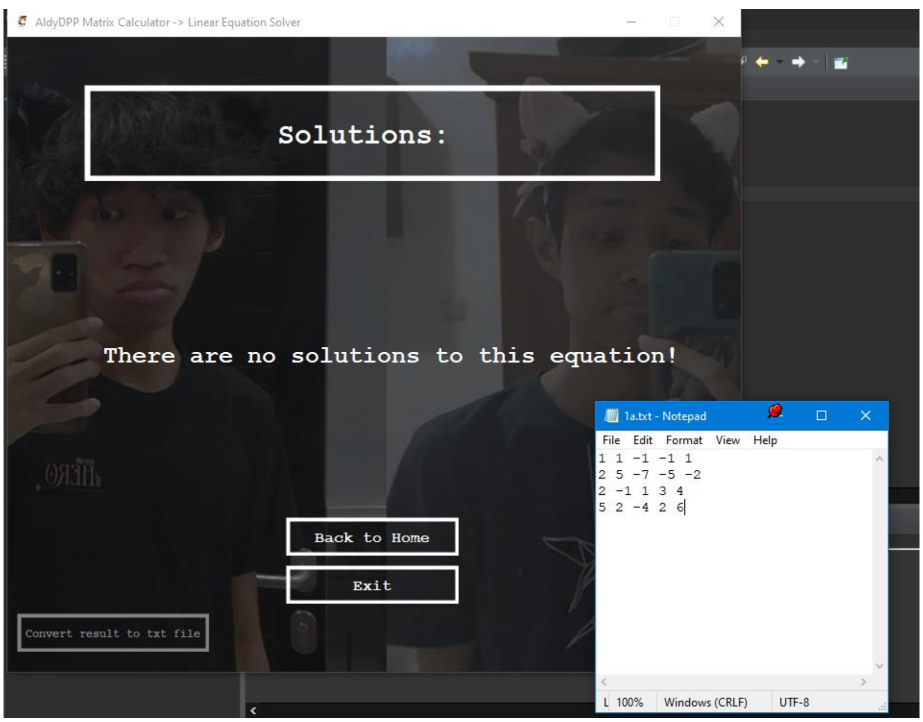
BAB 4

EKSPERIMEN

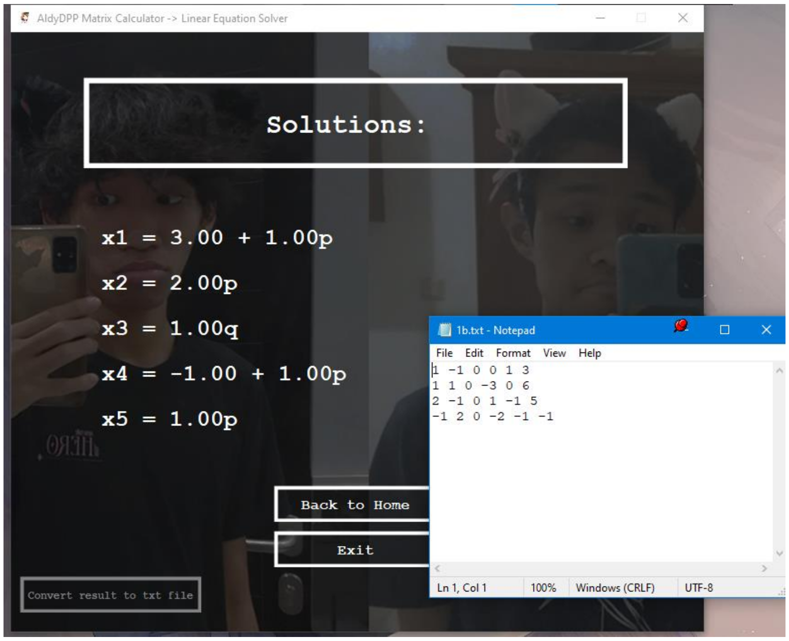
Pada bab ini, dipaparkan luaran program setelah memasukkan *test case* yang terdapat pada bab studi kasus pada file spesifikasi tugas besar.

1. Solusi SPL *Ax* = *b*

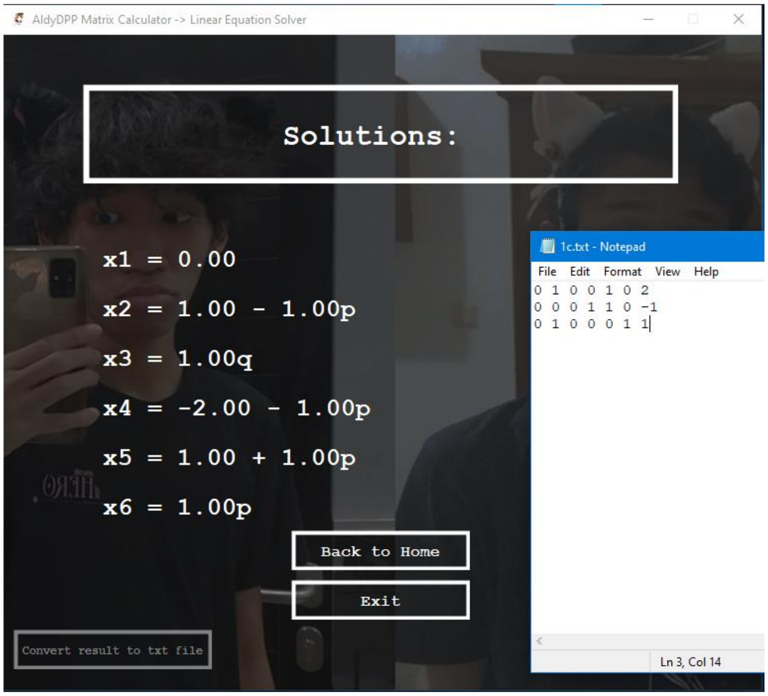
Pada studi kasus nomor ini, diberikan sistem persamaan linear dalam bentuk perkalian matriks ***Ax* = *b***. Solusi persoalan dapat dilihat pada gambar berikut. Pada gambar 1, dapat dilihat bahwa matriks SPL 1a tidak mempunyai solusi. Dapat dilihat pula pada gambar 2 dan 3 bahwa matriks SPL 1b dan 1c mempunyai penyelesaian parametrik, sementara pada gambar 4 dan 5 tertera solusi nomor 1d, yaitu penyelesaian matriks hilbert.



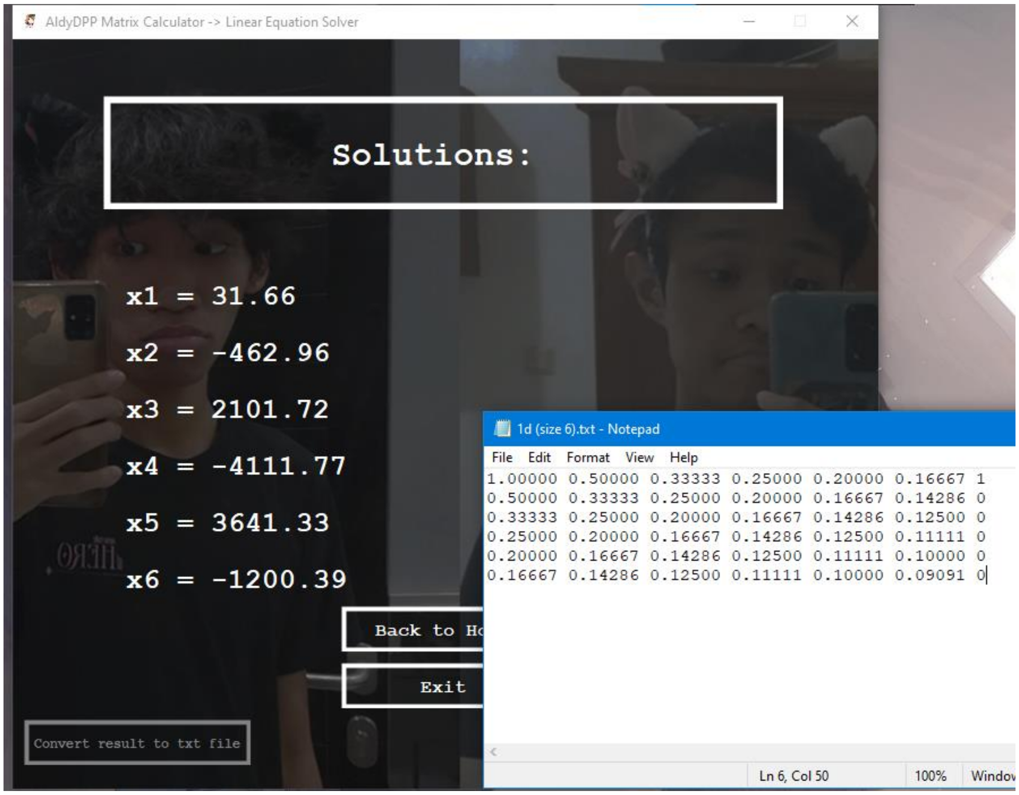
Gambar 5.1. Hasil uji test case studi kasus nomor 1a



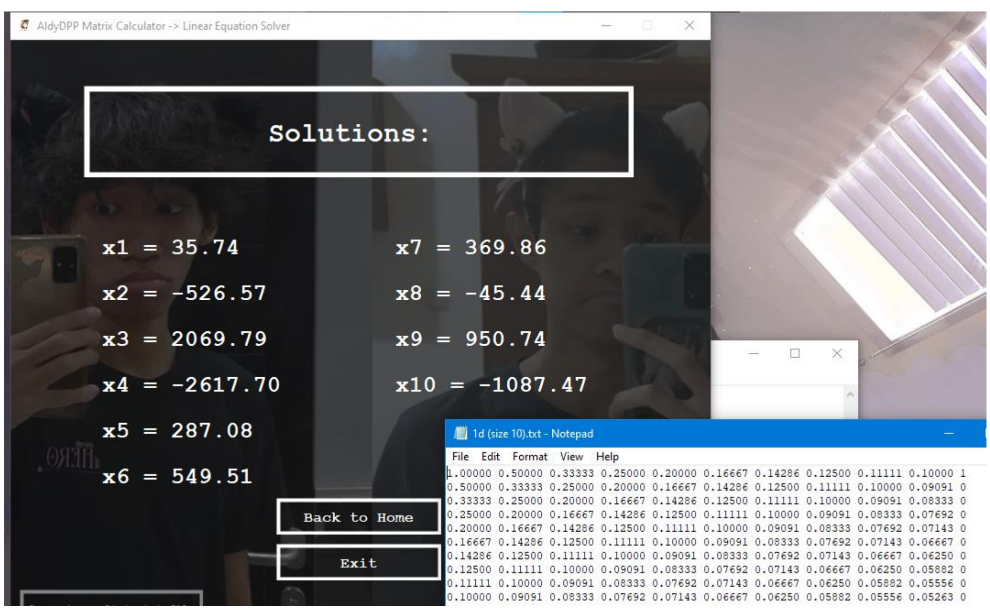
Gambar 5.2. Hasil uji test case studi kasus nomor 1b



Gambar 5.3. Hasil uji test case studi kasus nomor 1c



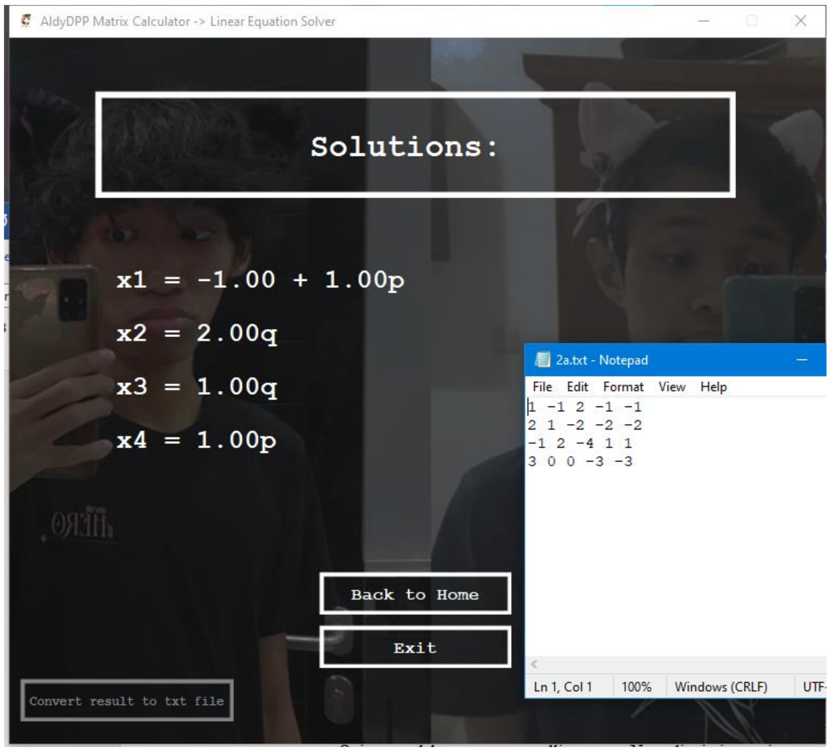
Gambar 5.4. Hasil uji test case studi kasus nomor 1d (matriks hilbert ukuran 6)



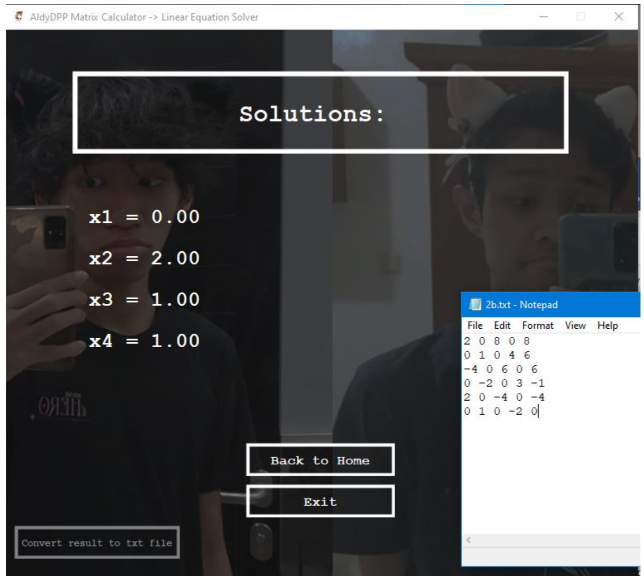
Gambar 5.5. Hasil uji test case studi kasus nomor 1d (matriks hilbert ukuran 10)

2. Solusi SPL berbentuk *augmented*

Studi kasus nomor ini mirip dengan persoalan nomor 1, yakni SPL. Penyelesaian matriks SPL solusi persoalan sudah cukup jelas dari tampilan program. Dapat dilihat bahwa matriks SPL *augmented* 2a mempunyai solusi parametrik dan matriks SPL *augmented* 2b mempunyai solusi integer.



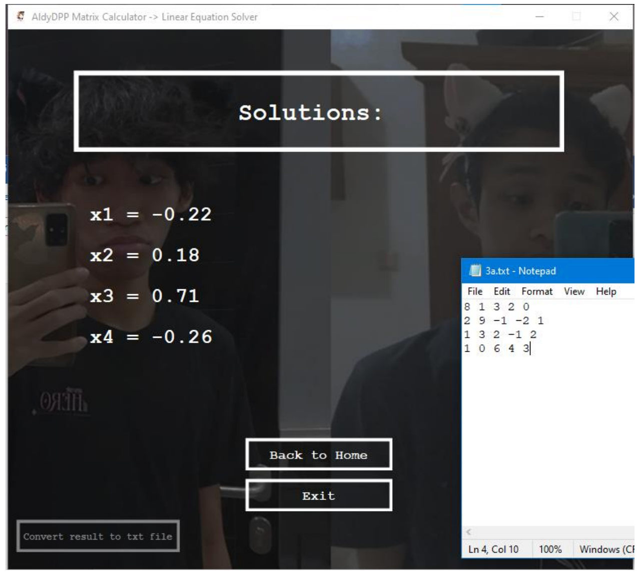
Gambar 5.6 hasil uji test case studi kasus nomor 2a



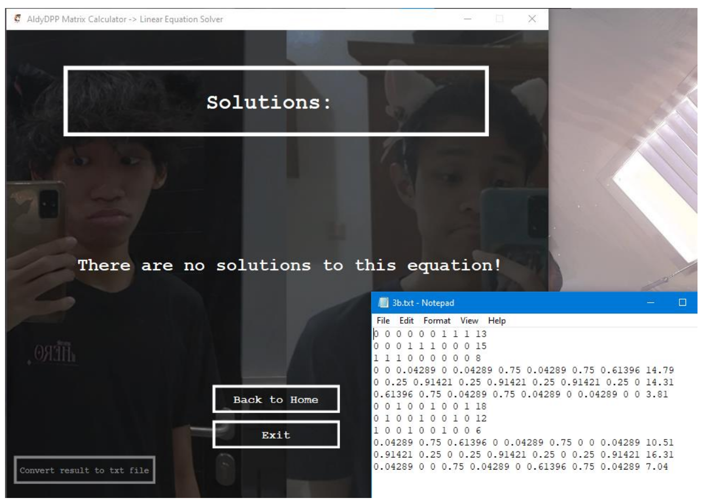
Gambar 5.7. Hasil uji test case studi kasus nomor 2b

3. Solusi SPL Berbentuk

Lagi-lagi, studi kasus ini pun mirip dengan nomor 1 dan 2. Diberikan matriks dalam bentuk penjumlahan aljabar beberapa variabel. Solusi sistem persamaan nomor 3a sudah disajikan dengan jelas pada gambar 8, dan untuk sistem persamaan nomor 3b, tidak ada solusi seperti yang terlihat pada output program pada gambar 9.



Gambar 5.8. Hasil uji test case studi kasus nomor 3a



Gambar 5.9. Hasil uji test case studi kasus nomor 3b

4. Analisis Hukum Konservasi Massa pada Sistem Reaktor

Studi kasus ini memberikan persoalan hukum kekekalan massa, yaitu *mass flow rate* yang masuk ke sebuah reaktor harus sama dengan *mass flow rate* yang keluar dari reaktor (bersifat konstan). *Mass flow rate* adalah besarnya massa yang keluar/masuk per satuan waktu. *Mass flow rate* dapat dihitung dengan debit (Q =

m^3/s) dikali dengan massa jenis (ρ = kg/m^3). Pada soal ini, sudah diketahui Q_{masuk} dan Q_{keluar} dari setiap reaktor dan ingin dicari massa jenis dari zat yang sedang masuk/keluar tersebut. Diketahui pula beberapa konstanta m_{Ain} dan m_{Cin} , yaitu berturut-turut *mass flow rate* yang masuk pada reaktor A dan turut *mass flow rate* yang masuk pada reaktor C.

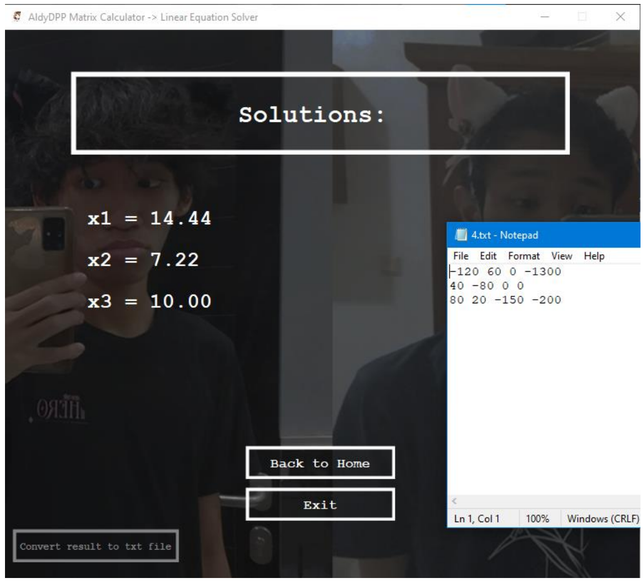
Terdapat tiga reaktor sehingga dibentuk tiga persamaan linier berdasarkan *mass flow rate* pada reaktor A, B, dan C. Berikut adalah matriks persamaannya (sudah disederhanakan dari bentuk soal).

```
-120 60 0 -1300

40 -80 0 0

80 20 -150 -200
```

Setelah matriks di atas dimasukkan ke dalam kalkulator SPL program, ditampilkan luaran **X_1 = 14.44, X_2 = 7.22,** dan **X_3 = 10**. Satuan dari ketiga peubah ini bergantung pada satuan konstanta m_{Ain} dan m_{Cin} yang mempunyai satuan mg/s sehingga jika dibagi dengan debit m^3/s, satuan peubah menjadi mg/m^3. Secara formal, berikut adalah solusi persoalan studi kasus ini.

- Massa jenis dari zat yang keluar dari reaktor A adalah 14.44 mg/m^3
- Massa jenis dari zat yang keluar dari reaktor B adalah 7.22 mg/m^3
- Massa jenis dari zat yang keluar dari reaktor C adalah 10 mg/m^3

Solusi ini kemudian dapat digunakan untuk mencari informasi lebih lanjut terkait zat-zat pada sistem reaktor ini. Sebagai contoh, dapat diketahui molaritas atau konsentrasi zat apabila diketahui massa molar, dan sebagainya.



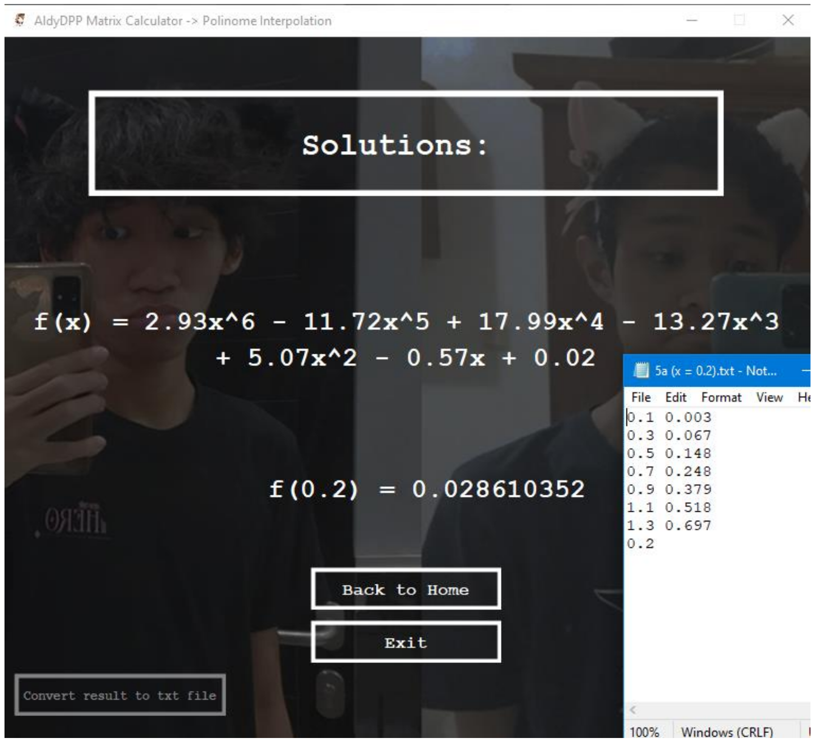
Gambar 5.10. Hasil uji test case studi kasus nomor 4

5. Studi Kasus Interpoasi

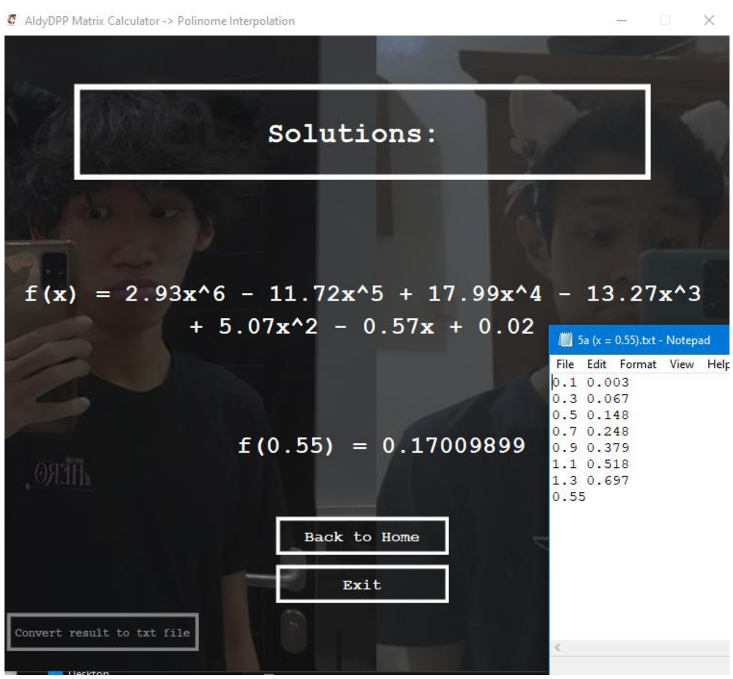
Studi kasus nomor ini mempersambahkan tiga persoalan berbeda yang dapat diselesaikan menggunakan interpolasi titik.

5.1. Interpolasi untuk Menafsir Nilai Fungsi

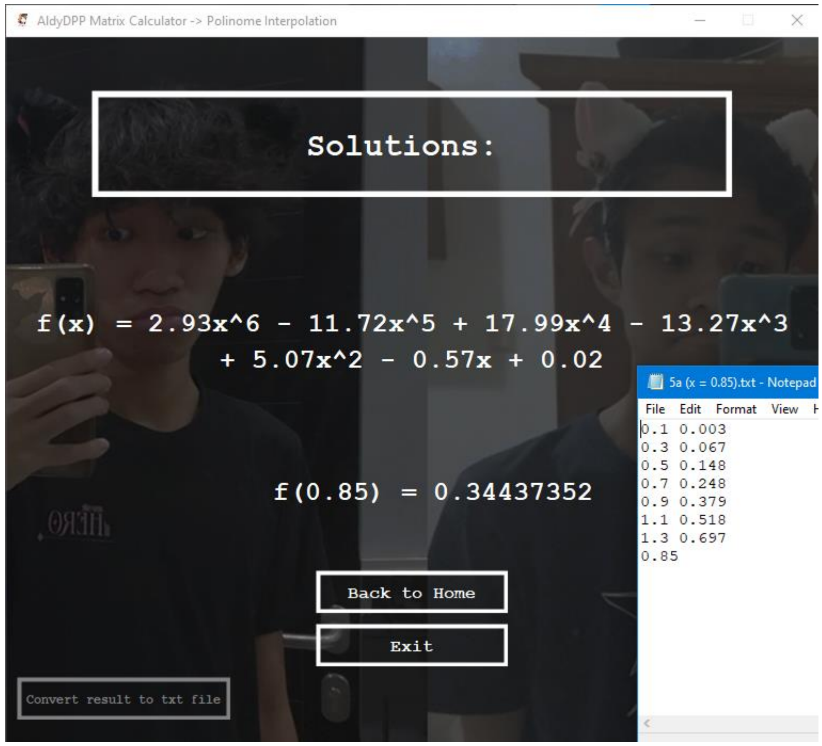
Persoalan ini adalah kasus penggunaan konsep interpolasi yang paling mendasar. Diberikan sebuah fungsi f dan sejumlah nilai x beserta hasil pemetaan x oleh f, yaitu $f(x)$. Dalam kata lain, diberi sejumlah titik $(x, f(x))$ dan ingin diketahui nilai $f(x)$ untuk nilai x tertentu. Berikut disajikan gambar-gambar tampilan luaran program untuk empat absis yang diberikan pada soal, yakni $x = 0.2$, $x = 0.55$, $x = 0.85$, dan $x = 1.28$.



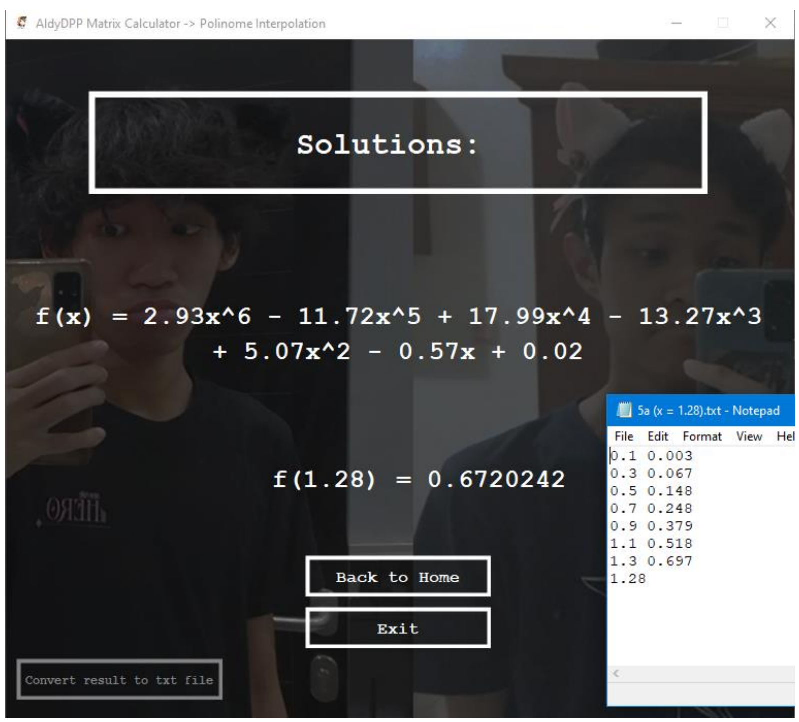
Gambar 5.11. Hasil uji test case studi kasus nomor 5a dengan absis x = 0.2



Gambar 5.12. Hasil uji test case studi kasus nomor 5a dengan absis x = 0.55



Gambar 5.13. Hasil uji test case studi kasus nomor 5a dengan absis x = 0.85



Gambar 5.14. Hasil uji test case studi kasus nomor 5a dengan absis x = 1.28

Dapat dilihat sesuai gambar 11-14, solusi sistem persamaan $f(x)$ adalah sebagai berikut.

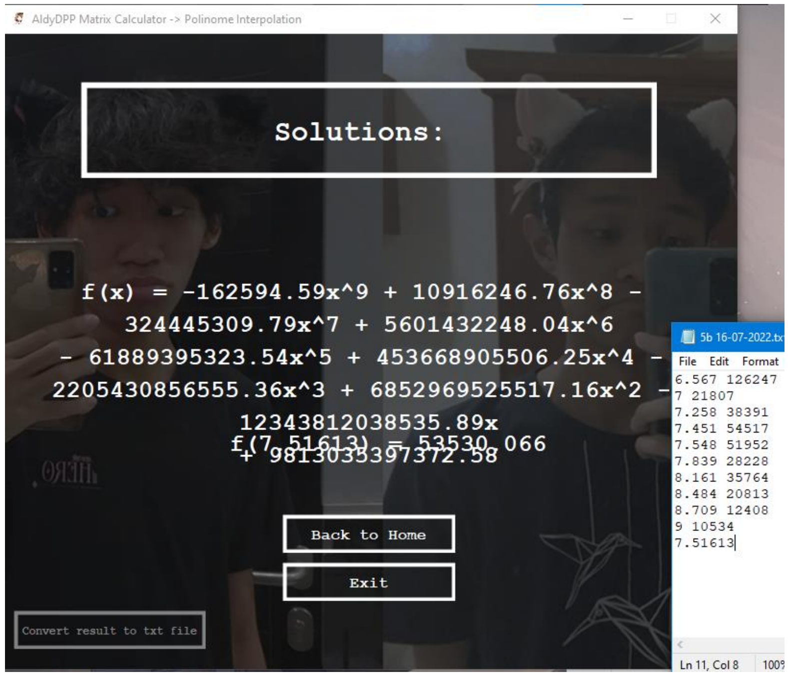
$$f(x) = 2.93\,x^6 - 11.72\,x^5 + 17.99\,x^4 - 13.27\,x^3 + 5.07\,x^2 - 0.57\,x + 0.02$$

Nilai $f(x)$ untuk empat nilai yang diberikan soal adalah sebagai berikut.

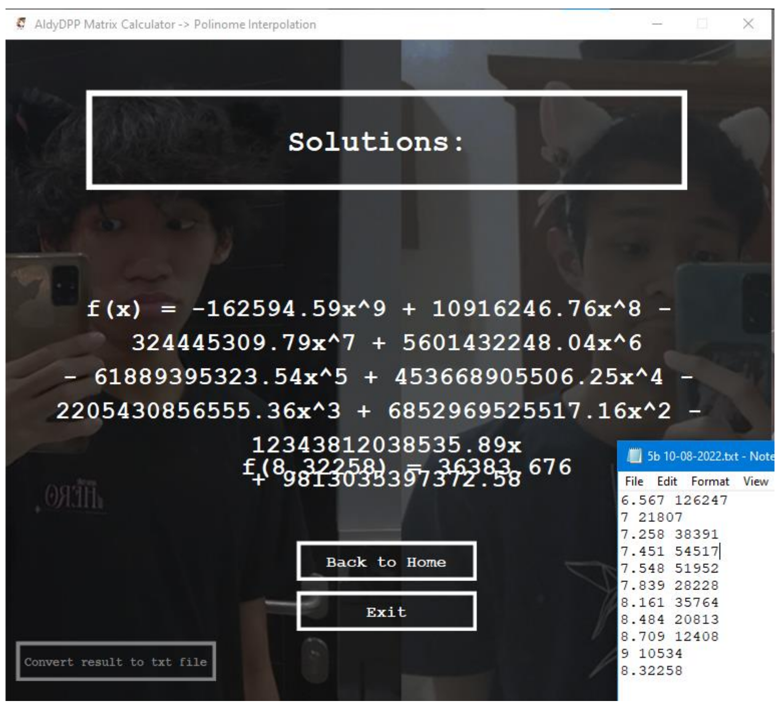
- $f(0.2) = 0.0286$
- $f(0.55) = 0.170$
- $f(0.85) = 0.344$
- $f(1.28) = 0.672$

5.2. Interpolasi untuk Menafsir Jumlah Bertambahnya Penderita Covid-19 pada Tahun 2022

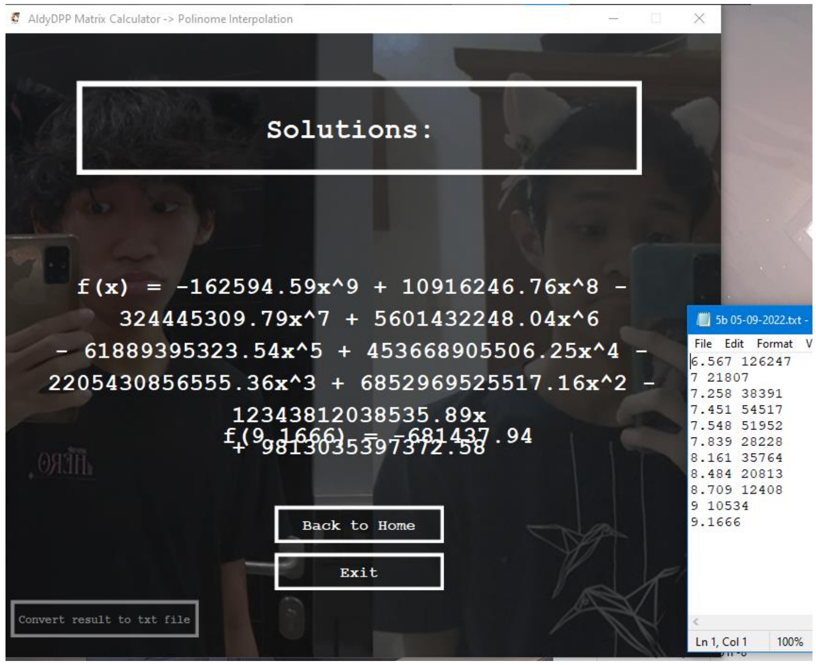
Subnomor ini menggunakan konsep interpolasi titik dengan sumbu-y atau nilai $f(x)$ adalah jumlah kasus, sementara sumbu-x atau nilai x adalah tanggal. Tanggal dikonversi menjadi bentuk angka menggunakan rumus yang sudah tertera pada soal. Berikut adalah tampilan output program untuk ketiga tanggal yang diberikan serta satu masukan user lain.



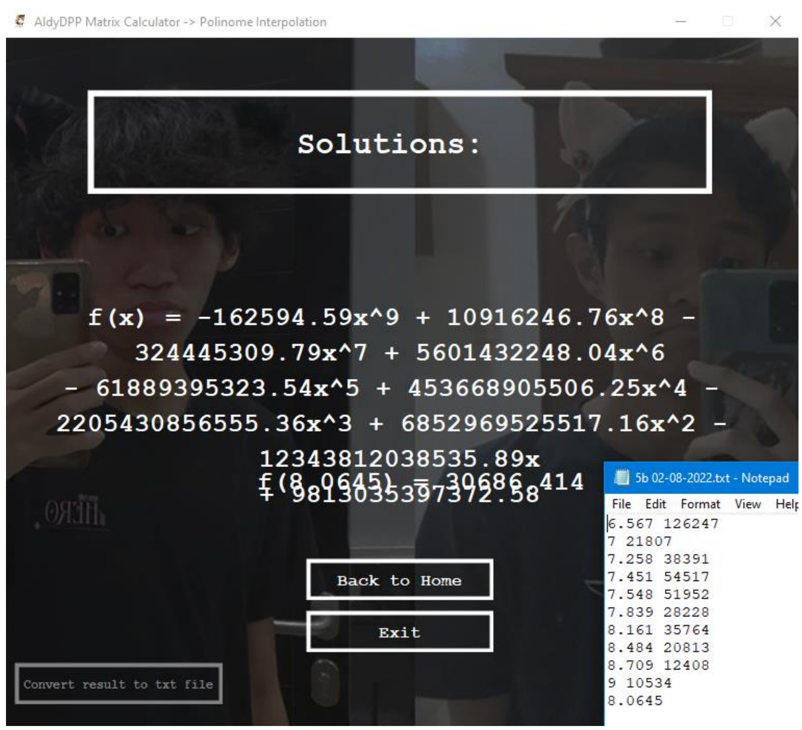
Gambar 5.15. Hasil uji test case studi kasus nomor 5b dengan masukan tanggal 16 Juli 2022



Gambar 5.16. Hasil uji test case studi kasus nomor 5b, masukan tanggal 10 Agustus 2022



Gambar 5.17. Hasil uji test case studi kasus nomor 5b, masukan tanggal 9 September 2022



Gambar 5.18. Hasil uji test case studi kasus nomor 5b, masukan tanggal 2 Agustus 2022

Pada gambar, solusi kurang terlihat dengan jelas. Berikut disajikan solusinya.

$$\boldsymbol{f(x) = -162594.59x^9 + 10916246.76x^8}$$

$$\mathbf{- 324445309.79x^7 + 5601432248.04x^6}$$

$$\mathbf{- 61889395323.54x^5 + 453668905506.25x^4}$$

$$\mathbf{- 2205430856555.36x^3 + 6852969525517.16x^2}$$

$$\mathbf{- 12343812038535.89x^1 + 9813035397372.58}$$

Dan berikut adalah solusi untuk tiga tanggal yang diberikan soal serta satu masukan lainnya.

- Tanggal 16 Juli 2022: $f(7.51613) = 53530.066$
- Tanggal 10 Agustus 2022: $f(8.32258) = 36383.676$
- Tanggal 9 September 2022: $f(9.1666) = -681437.94$
- Tanggal 2 Agustus 2022: $f(8.0645) = 30686.414$

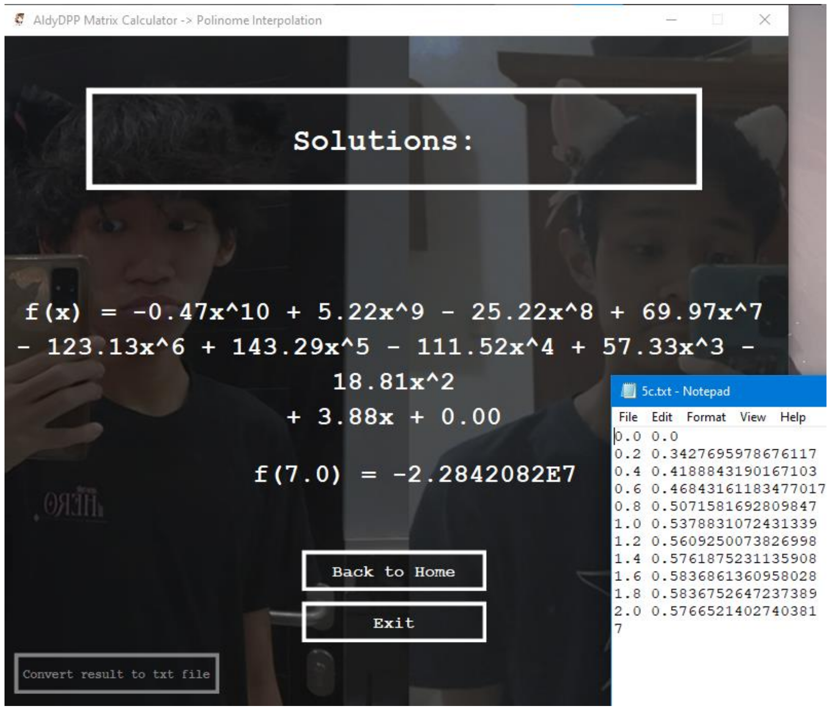
Perlu diingat bahwa nilai fungsi masih harus dibulatkan karena jumlah jiwa bersifat bilangan bulat. Di samping itu, aproksimasi untuk tanggal 9 September bernilai negatif. Hal tersebut jelas salah, tetapi wajar. Ini terjadi karena interpolasi dilakukan untuk range nilai x = [7,9], sementara nilai yang ditafsir bernilai lebih dari 9 sehingga estimasi tidak akurat. Sedangkan untuk nilai $f(x)$ untuk tanggal 16 Juli dan 10 Agustus sudah benar.

5.3. Interpolasi Untuk Penyederhanaan Fungsi

Pada subnomor ini, ingin dicari sebuah fungsi polinom yang mendekati fungsi eksponen yang diberikan soal. Hal tersebut dapat dilakukan menggunakan interpolasi titik pada range yang diinginkan, dalam hal ini [0, 2]. Diambil nilai n = 10, kemudian sebelas titik ditinjau dengan jarak antartitik adalah 0.2. Berikut adalah tabel nilai-nilai x yang disampel pada range [0,2]. Nilai $f(x)$ telah dihitung menggunakan kalkulator dan dibulatkan (nilai utuh tetap digunakan ketika menginput di program).

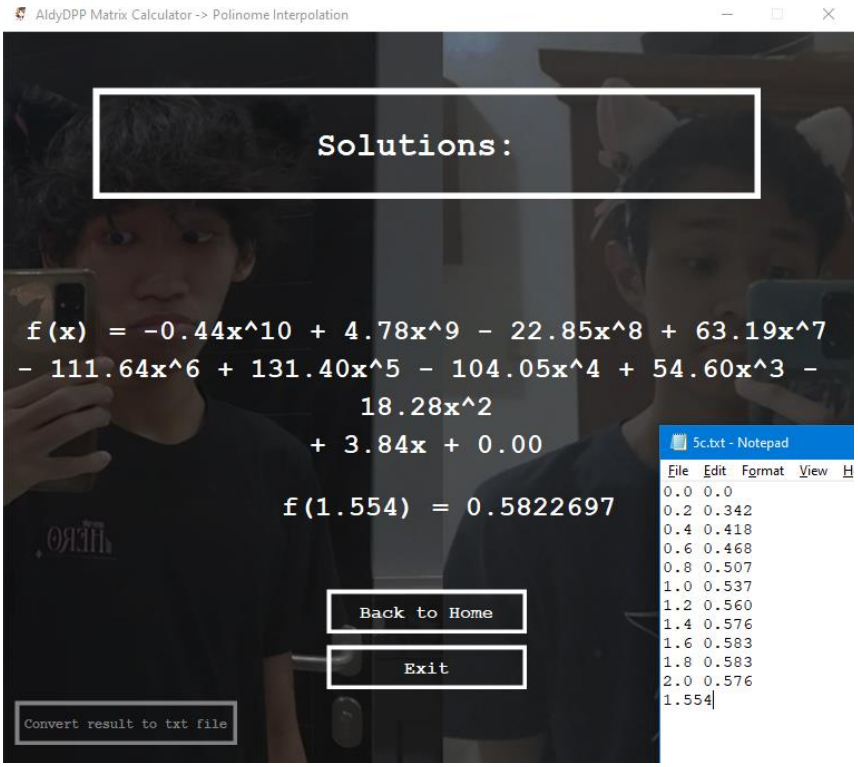
x	$f(x)$
0	0
0.2	0.342
0.4	0.418
0.6	0.468
0.8	0.507
1.0	0.537
1.2	0.560
1.4	0.576
1.6	0.583
1.8	0.583
2.0	0.576

Kemudian didapatkan fungsi derajat 10 yang dapat dilihat pada gambar 18 berikut.



Gambar 5.19. Hasil uji test case studi kasus nomor 5c dengan contoh absis x = 7.

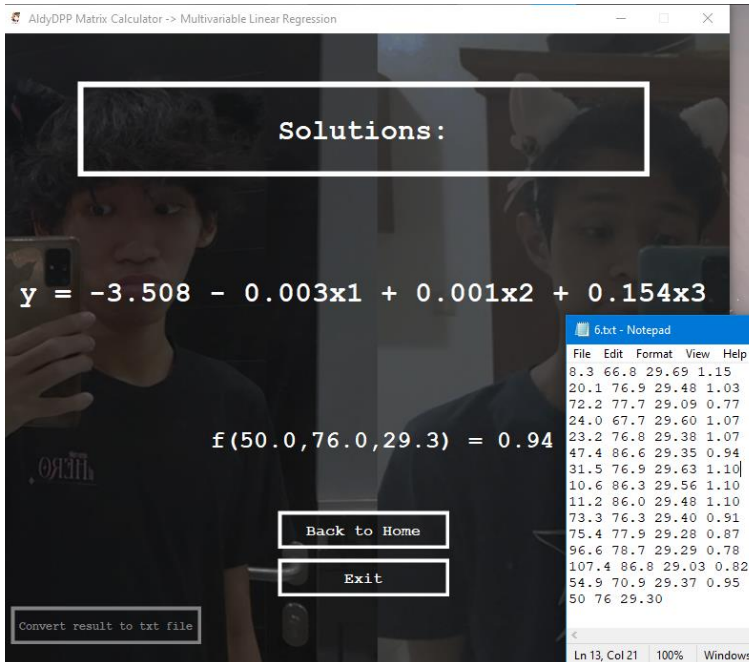
Diambil contoh absis $x = 7$ dan didapatkan hasil -2.2×10^7. Nilai tersebut sangat jauh dari nilai $f(x)$ yang sebenarnya, yaitu jika dimasukkan kalkulator, $f(7)$ seharusnya 0.046. Hal ini disebabkan interpolasi hanya dilakukan pada *range* nilai [0,2] sementara nilai yang ditafsir tidak dalam range tersebut. Apabila dimasukkan nilai yang *range* nya sesuai, akan didapatkan nilai yang lebih akurat. Berikut gambar output program dengan contoh absis $x = 1.554$. Nilai $f(1.554)$ dapat dilihat berada dalam kisaran yang lebih wajar.



Gambar 5.20. Hasil uji test case studi kasus nomor 5c dengan contoh absis x = 1.554.

6. Regresi Linier Berganda untuk Estimasi Faktor Koreksi *Nitrous Oxide*

Pada soal, disajikan tabel 20 persamaan faktor koreksi untuk zat Nitrous Oxide yang terdiri dari tiga buah: *humidity* (kelembapan), temperatur, dan *pressure* (tekanan udara). Dilakukan regresi linier pada 20 persamaan tersebut untuk menentukan persamaan nilai faktor koreksi Nitrous Oxide, kemudian diaproksimasikan nilai faktor koreksi untuk *humidity* bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30. Berikut adalah tampilan output program.



Gambar 5.21 Hasil uji test case studi kasus nomor 6

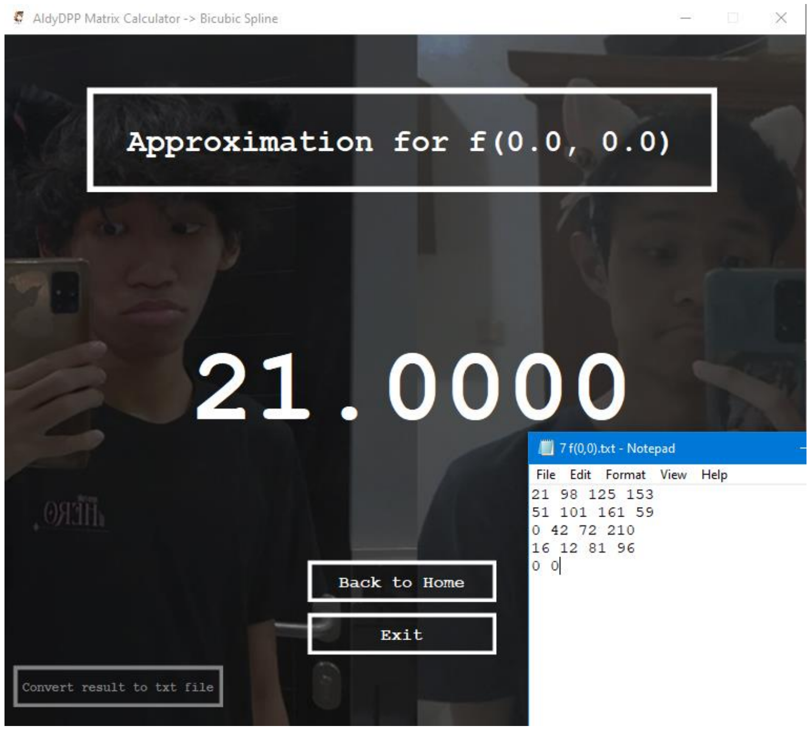
Nilai faktor koreksi Nitrous Oxide yang ditafsirkan untuk *humidity* bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30 adalah **0.94**.

7. Interpolasi Bicubic Spline

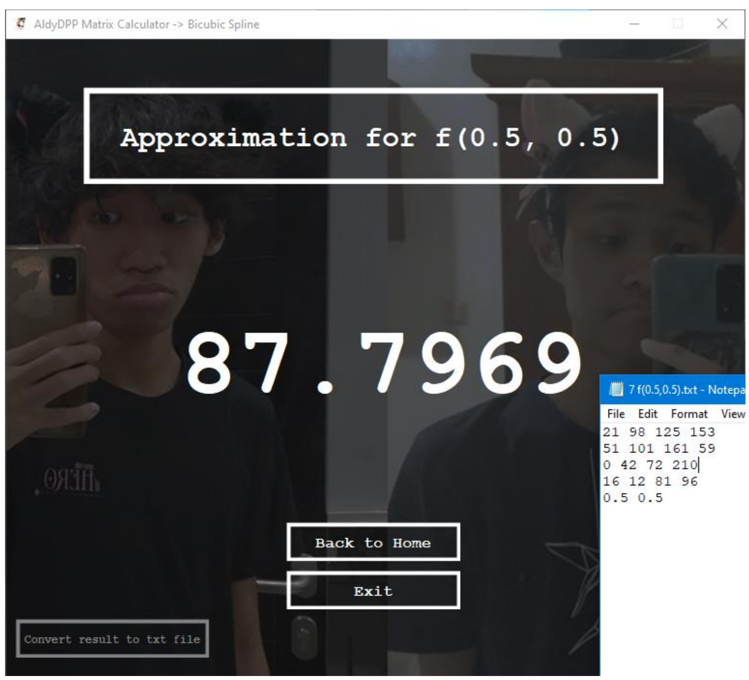
Studi kasus ini cukup *straightforward*. Berikut disajikan hasil estimasi beserta gambar tampilan output program untuk kalkulasi interpolasi *Bicubic Spline* matriks pada soal.

Hasil estimasi empat nilai (a, b):

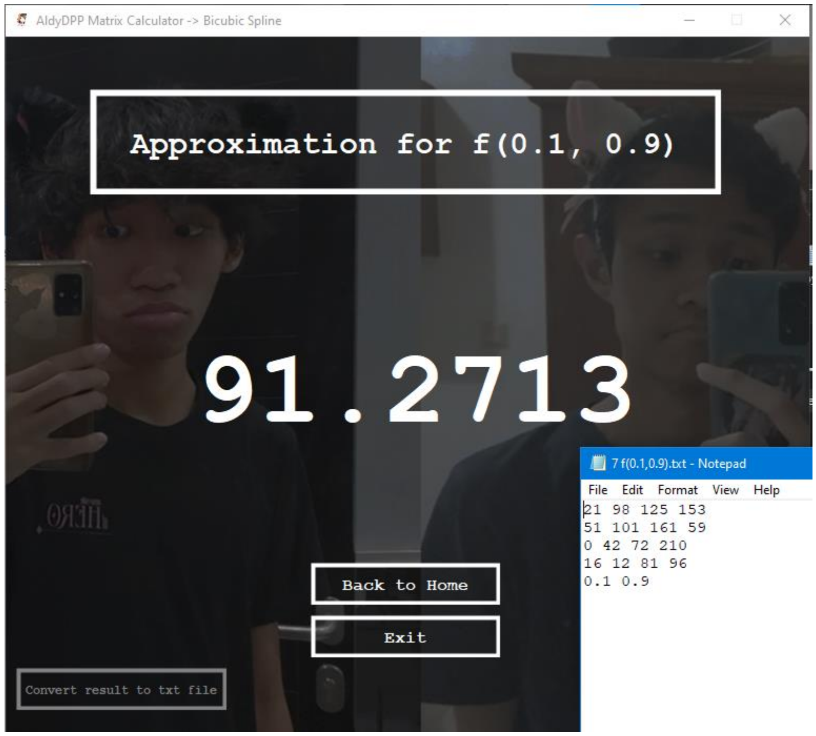
- $f(0, 0) = 21$
- $f(0.5, 0.5) = 87.79$
- $f(0.1, 0.9) = 91.27$
- $f(0.25, 0.25) = 82.14$



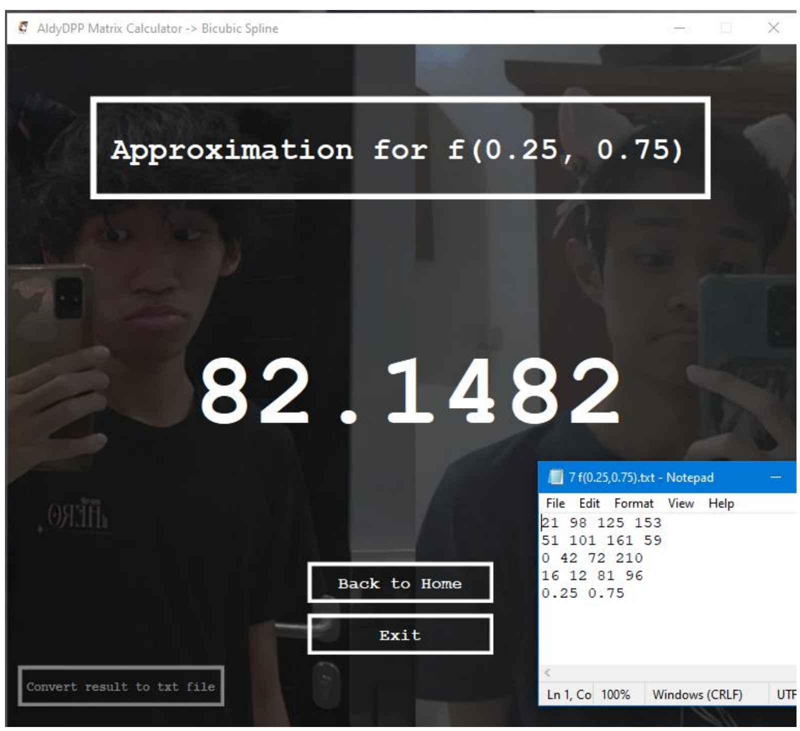
Gambar 5.21. Hasil uji test case studi kasus nomor 7 untuk nilai (a, b) = (0, 0)



Gambar 5.22. Hasil uji test case studi kasus nomor 7 untuk nilai (a, b) = (0.5, 0.5)



Gambar 5.23. Hasil uji test case studi kasus nomor 7 untuk nilai (a, b) = (0.1, 0.9)



Gambar 7.24. Hasil uji test case studi kasus nomor 7 untuk nilai (a, b) = (0.25, 0.75)

8. Bonus (Perbesaran *Image*)

Berikut disajikan gambar sebelum dan sesudah diperbesar menggunakan program (ukuran file gambar berbeda).

a. mario

Gambar 7.25. mario.png, sebelum diperbesar

Gambar 7.26 mario2.png, setelah diperbesar

b. floppyThing.jpg

Gambar 7.27. floppyThing.jpg sebelum diperbesar

Gambar 7.28. bigFloppyThing.png setelah diperbesar

BAB 5

KESIMPULAN DAN SARAN

Kesimpulan

Kami berhasil membuat program kalkulator matriks berbasis GUI sesuai dengan spesifikasi yang tertera pada tugas besar ini.

Saran

Untuk kedepannya, program dapat dibuat dengan lebih rapi dan teliti, penggunaan metode dan class harus lebih sesuai, dan harus meminimalisasi *bug* dengan lebih teliti membaca spesifikasi sebelum mengimplementasi. Untuk pelaksanaan tugas besar, menurut kami ada baiknya tugas tidak diberi kebebasan sebesar ini (perlu ada restriksi atau *constraint* lebih). Hal ini karena terlalu bebasnya format input/output mungkin akan membuat proses pengerjaan kurang terarah bagi mahasiswa yang belum terlalu berpengalaman dalam memprogram. Tidak perlu sespesifik itu, mungkin mahasiswa dapat diarahkan untuk membuat *class-class* tertentu sehingga lebih mudah untuk pembagian tugas.

Refleksi

Dalam pengerjaan tugas besar ini, kami menyadari dan mempelajari banyak hal. Kami menyadari betapa mudahnya pengerjaan tugas apabila dikerjakan jauh-jauh hari dan dilakukan secara bertahap, ketimbang apabila mengerjakan di seminggu terakhir sebelum hari pengmumpulan. Kami juga mempelajari, khususnya Aldy yang menangani error dan GUI, bahwa *debugging* adalah skill tersendiri yang terlepas dari skill memprogram. Akibat harus mengintegrasikan fungsi yang dibuat oleh Dito ke dalam tampilan UI, ada banyak sekali error dan/atau output salah yang terjadi akibat penamaan variabel yang membingungkan, kurangnya komentar, dsb.

Kemudian dari pengerjaan tugas ini, kami pun menyadari kembali betapa berpengaruhnya *passion* terhadap motivasi belajar. Ketika mengerjakan, kami tidak pernah berpikir bahwa tugas ini adalah beban. Memang, tugas ini cukup sulit dan banyak, serta kadang-kadang membuat frustasi. ***However***, perjalanannya dapat dinikmati karena kami suka berpetualang di dunia informatika. Kami rasa manfaat yang telah kami dapatkan dari tugas besar ini sangat banyak, dan kami nantikan tugas-tugas besar maupun kecil berikutnya.

“本当にありがとうございました！！！”

- ALDY dan DITO -

BAB 6

PEMBAGIAN KERJA

No	Deskripsi	Penanggung Jawab
1	Tipe-Tipe Bentukan Matriks	Aldy (13522022)
2	Tipe Bentukan SolutionType	Aldy
3	Determinan, Gauss-Jordan, Inverse	Aldy
4	Input/Output + parsing .*txt* File	Aldy
5	Ekspansi Kofaktor, Metode Cramer	Dito (13522089)
6	Sistem Persamaan Linier	Dito
7	Interpolasi Polinom	Dito
8	Interpolasi *Bicubic Spline*	Dito
9	Regresi Linier Multivariabel	Dito
10	Bonus (Image enlarger)	Dito
11	GUI dan Error Handling	Aldy
12	Laporan	Dito, Aldy

Mengetahui,

Renaldy Arief Susanto

(Aldy - 13522022)

Abdul Rafi Radityo Hutomo

(Dito - 13522089)

REFERENSI

Elementary Linear Algebra: Applications Version. H. Anton, and C. Rorres. Wiley, Eleventh edition, (2014)

BiLinear, Bicubic, and In Between Spline Interpolation. Rowe, D.B. Department of Biophysics, Medical College of Wisconsin. (2018)

OLS Estimation of the Multiple (Three-Variable) Linear Regression Model. Abbott. Queen's Economics Department. Note 12 of ECONOMICS 351

Giassa. (2010). "*IV. Generalized Bicubic Interpolation*": Image Processing – Generalized Bicubic Interpolation. Giassa. https://www.giassa.net/?page_id=371

Keys, Robert G. (1982). Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions On Acoustics, Speech, and Signal Processing*, 1153 – 1160.