Laporan Tugas Besar 1
Pembuatan Library Pengolahan Matriks Menggunakan Java
IF2123 Aljabar Linier dan Geometri

Kelompok 20 - Kuli Java

Fawwaz Anugrah Wiradhika Dharmasatya	**13520086**
Andika Naufal Hilmy	**13520098**
Alifia Rahmah	**13520122**

DAFTAR ISI

BAB I: Deskripsi Masalah

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1} b$), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

Pada Tugas Besar 1 ini, kami membuat library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, library tersebut akan digunakan di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

BAB II: Teori Singkat

Matriks

Matriks adalah sebuah struktur data dua dimensi yang memiliki baris dan kolom. Terdapat beberapa jenis operasi pada matriks, di antaranya penjumlahan dan pengurangan dua matriks berdimensi sama, perkalian matriks, perkalian skalar dengan matriks, transpos matriks, mencari determinan, dan mencari balikan matriks. Terdapat beberapa jenis matriks, yaitu matriks identitas -- diagonal utama satu dan sisa elemen selain diagonal utama nol, matriks nol -- semua elemennya bernilai nol, matriks eselon baris, serta matriks eselon baris tereduksi.

Matriks eselon baris adalah matriks yang memiliki satu utama pada setiap baris, kecuali baris yang seluruhnya nol, di mana setiap satu utama harus merupakan angka bukan nol pertama, dan Di dalam dua baris berturutan yang tidak seluruhnya nol, maka satu utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada satu utama pada baris yang lebih tinggi. Matriks eselon baris tereduksi adalah matriks eselon baris di mana pada kolom yang terdapat satu utama, tidak boleh ada elemen bukan nol selain pada satu utama.

Eliminasi Gauss dan Eliminasi Gauss-Jordan

Metode eliminasi Gauss adalah salah satu metode yang dapat digunakan untuk menyelesaikan sistem persamaan linier (SPL) menggunakan matriks augmented. Matriks augmented adalah matriks yang menyatakan SPL secara ringkas. Proses eliminasi Gauss ini dilakukan dengan operasi baris elementer (OBE) terhadap matriks augmented hingga mendapatkan matriks eselon baris. Persamaan yang berkoresponden pada matriks eselon baris dilakukan dengan teknik penyulihan mundur hingga mendapatkan solusi unik, SPL disimpulkan tidak memiliki solusi, atau SPL disimpulkan memiliki banyak solusi. Untuk SPL dengan banyak solusi, diselesaikan dengan membuat solusi dalam bentuk parametrik. Metode eliminasi Gauss-Jordan adalah salah satu metode perluasan dari eliminasi Gauss. Pada eliminasi Gauss-Jordan, OBE dilakukan hingga mendapatkan matriks eselon baris tereduksi, sebelum melakukan penyulihan mundur.

Determinan

Determinan matriks A dilambangkan dengan det(A). Pada matriks 2×2, determinan matriks didapatkan dari selisih perkalian antara diagonal utama dan diagonal samping dari matriks. Pada matriks 3×3, determinan matriks dapat ditemukan dengan menggunakan operasi berikut

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

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

Terdapat beberapa teorema mengenai determinan, di antaranya:

- det(A) = 0, jika matriks A mengandung baris nol atau kolom nol
- $\det(A^T) = \det(A)$
- det(A) = det(B)det(C) jika A = BC
- $\det(A^{-1}) = 1/\det(A)$

Metode Reduksi Baris

Metode reduksi baris yaitu mencari determinan dengan melakukan OBE pada matriks hingga memperoleh matriks segitiga atas atau bawah. Determinan matriks segitiga ini dapat dihitung dengan

$$\det(A) = (-1)^p a'_{11}a'_{22}\ldots a'_{nn}$$

Di mana p menyatakan banyak operasi pertukaran baris dan $a'_{11}\ldots a'_{nn}$ menyatakan perkalian dari digonal utama.

Metode Ekspansi Kofaktor

Metode ekspansi kofaktor untuk mencari determinan dilakukan dengan memanfaatkan minor entri matriks tersebut. Minor entri adalah submatriks dengan semua elemennya tidak berada pada baris dan kolom tertentu di dalam matriks. Kofaktor entri pada baris i dan kolom j didapatkan dengan

$$C_{ij} = (-1)^{i+j}M_{ij},$$

Dengan M_{ij} adalah minor entri pada baris i dan kolom j. Dengan demikian, determinan matriks dapat dihitung dengan menjumlahkan semua kofaktor entri dari semua elemen pada salah satu baris atau kolom.

Matriks Kofaktor dan Adjoin

Matriks kofaktor adalah matriks dengen tiap elemen dengan baris i dan kolom j berisikan kofaktor entri pada baris i dan kolom j dari suatu matriks. Jika ditranspos, matriks kofaktor akan membentuk matriks adjoin, yang dapat digunakan untuk mencari balikan matriks.

Kaidah Cramer

Kaidah Cramer merupakan kaidah yang dapat digunakan untuk menyelesaikan SPL. Jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n variabel sedemikian sehingga $\det(A) \neq 0$, maka SPL tersebut memiliki solusi unik secara umum

$$x_i = \frac{\det(A_i)}{\det(A)}$$

Dengan A_i adalah matriks yang diperoleh dengan mengganti elemen-elemen pada kolom ke-i dengan matriks b.

Matriks Balikan

Sebuah matriks memiliki balikan jika determinan matriks tersebut tidak sama dengan nol. Untuk mencari matriks balikan, dapat dilakukan dengan membagi matriks adjoin dengan determinan matriks. Jika determinan matriks = 0, maka matriks balikan dari matriks tersebut tidak dapat ditemukan.

Interpolasi Polinom

Persoalan interpolasi polinom yaitu menentukan polinom $p_n(x)$ yang melewati seluruh titik-titik koordinat yang diberikan sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $i = 0, 1, 2, ..., n$. Setelah polinom interpolasi ditemukan, polinom tersebut dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang $[x_0, x_n]$. Persamaan pada interpolasi polinom dapat ditentukan dengan menggunakan metode eliminasi Gauss.

Regresi Linier Berganda

Regresi linier adalah metode untuk memprediksi nilai selain menggunakan interpolasi polinom. Rumus umum dari regresi linier yang bisa digunakan untuk menghitung regresi linier berganda yaitu $y_i = {}_0 + {}_i x_{1i} + {}_2 x_{2i} + \dots + {}_k x_{ki} + \epsilon_i$. Untuk mendapatkan nilai dari setiap , dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* yang dapat diselesaikan dengan menggunakan metode eliminasi Gauss, sebagai berikut

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

BAB III: Implementasi Pustaka

Struktur Class

- CramerMethod
    - public static Matriks solve(Matriks matA, Matriks matB)
      Menyelesaikan SPL dengan metode Cramer: det(matA)/det(matB)
- EchelonRedux
    - public static void selfReduce(Matriks m)
      Prosedur antara, mengubah matriks m menjadi matriks eselon baris
    - public static void selfReduce(Matriks m, boolean reduced_mode)
      Reduksi matriks m menjadi eselon baris atau eselon baris tereduksi jika reduced_mode = true (default false)
- GaussMethod
    - public static double[] gaussElim(Matriks m)
      Melakukan eliminasi Gauss dan menyimpan hasil dalam array of double
    - public static double[] gaussJordanElim(Matriks m)
      Melakukan eliminasi Gauss-Jordan dan menyimpan hasil dalam array of double
    - public static Matriks augment(Matriks a, Matriks b)
      Menggabungkan matriks a dan b dengan banyak baris yang sama menjadi satu matriks augmented
    - public static String toParamEq(Matriks m)
      Mengembalikan persamaan parametrik dari matriks m, dengan syarat matriks m sudah merupakan matriks eselon baris.
    - public static String printSol(double[] solution, boolean toTxt)
      Mengembalikan string berisi solusi eliminasi Gauss, mengeluarkan ke bentuk string yang dapat disimpan dalam file jika toTxt = true, dan mengeluarkan ke bentuk string yang dapat ditampilkan ke layar jika toTxt = false
    - public static boolean isUniqueSol (*Matriks m*)
      Mengembalikan true jika matriks eselon baris m mempunyai solusi unik.
    - public static boolean isNoSol(*Matriks m*)
      Mengembalikan true jika matriks eselon baris m tidak mempunyai solusi
    - public static boolean isManySol(*Matriks m*)
      Mengembalikan true jika matriks eselon baris m mempunyai solusi banyak.
- InversMethod
    - public static *Matriks* makeIdentity(int *dimension*)
      Membuat matriks identitas
    - public static *Matriks* solve(*Matriks a*,*Matriks b*)

    - public static int ZeroUntilMainOne(double[] *row*)
      Mengembalikan indeks baris dengan elemen nol terakhir sebelum ‘1’ utama
    - public static *Matriks* invers(*Matriks m*)
      Menentukan invers dari sebuah matriks
    - private static boolean isAllZero(double[] *row*)
      Menentukan sebuah list/array/baris apakah semua nilainya nol
- Main
    - public static void main(*String*[] *args*)
      Program utama - menjalankan keseluruhan program
- Matriks
    - public Matriks(int *nbaris*, int *nkolom*)
      Konstruktor matriks
    - public static *Matriks* makeIdentity(int *dimension*)
      Membuat matriks identitas
    - public void normalize()
      Mengonversi elemen negatif nol menjadi positif nol pada matriks
    - public void swapRows(int *r1*, int *r2*)
      Mempertukarkan dua baris pada matriks
    - public void rowReduce(int *r1*, int *r2*, int *colidx*)
      Mengurangkan semua elemen pada r1 dengan semua elemen r2 * multiplier satu per satu dimana multiplier = matriks[r1][colidx]/matriks[r2][colidx]
    - public *String* repr()
      Mengembalikan bentuk string dari sebuah matriks
    - public *String* repr_forIO()
      Mengembalikan bentuk string dari sebuah matriks dalam bentuk GUI
    - public static *Matriks* parseMatrix(*String text*, int *NBaris*, int *NKolom*)
      Membaca matriks dari masukan pengguna pada GUI
    - public void writeMatriks()
      Mencetak matriks pada layar (terminal mode)
    - public *String* SimpanHasil()
      Mengembalikan string hasil dari penyelesaian SPL
    - public *Matriks* cloneMatriks()
      Melakukan *hard copy*/kloning matriks (beda pointer, beda referensi) agar operasi pada matriks tidak mempengaruhi matriks awal
    - public boolean isSquare()
      Mengembalikan true jika jumlah baris sama dengan jumlah kolom
    - public boolean isIdentity()
      Mengembalikan true jika matriks adalah matriks identitas

- public static *Matriks* kali(*Matriks m1*, *Matriks m2*)
  Mengembalikan hasil kali dua matriks
- public *Matriks* scalarMult(double *x*)
  Mengembalikan hasil kali skalar matriks dengan konstanta
- public double determinantReduction()
  Mengembalikan determinan dari sebuah matriks dengan metode reduksi
- public *Matriks* submatrix(int *row*, int *col*)
  Mengembalikan submatriks dari matriks tanpa semua elemen pada baris row dan kolom col
- public double minor(int *row*, int *col*)
  Mengembalikan minor dari elemen matriks / determinan submatrix
- public *Matriks* cofactor()
  Mengembalikan matriks kofaktor dari sebuah matriks
- public *Matriks* transpose()
  Mengembalikan sebuah matriks hasil transpose dari matriks awal
- public double determinantCofactor()
  Menghasilkan nilai determinan dari sebuah matriks menggunakan metode ekspansi kofaktor

- RegresiLib
  - public static double sigma2Var(double[]*var1*,double[]*var2*)
    Menghasilkan jumlah dari perkalian tiap elemen berindeks sama di var1 dan var2[$\Sigma(var1_i * var2_i)$]
- PolynomialInterpretationLib
  - public static *Matriks* createAugmented(*Matriks coordMatrix*)
    Mengembalikan matriks augmented untuk digunakan dalam perhitungan interpolasi polinomial

Garis Besar Program

Program ini dibuat untuk:

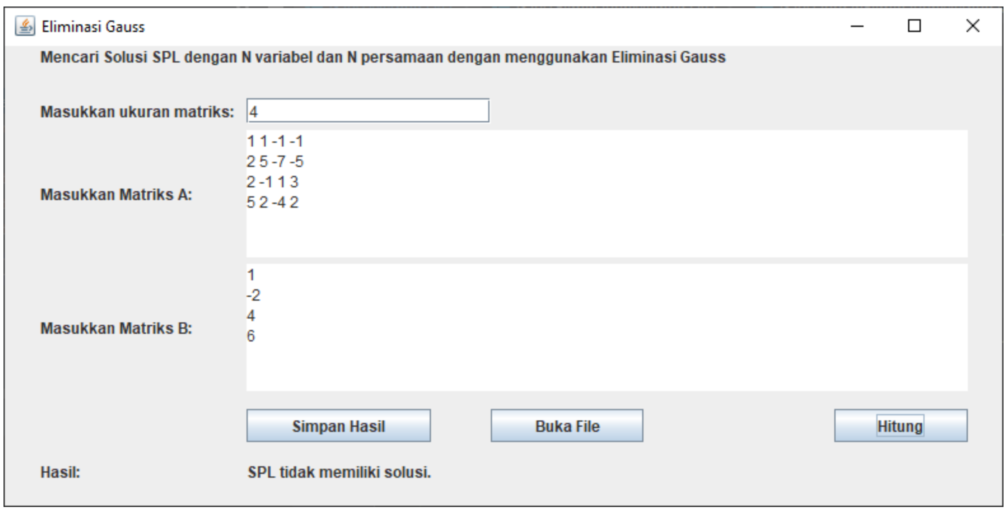
1. Menyelesaikan sistem persamaan linear (SPL) dengan metode Cramer, Eliminasi Gauss, Eliminasi Gauss-Jordan, dan metode matriks balikan
2. Menentukan determinan dari sebuah matriks dengan metode reduksi dan ekspansi kofaktor
3. Menentukan invers sebuah matriks dengan metode OBE dan matriks adjoin
4. Melakukan interpolasi polinomial dari sejumlah titik
5. Melakukan regresi linear berganda dari sejumlah data eksperimen

BAB IV: Eksperimen

1. Penyelesaian SPL berbentuk matriks Ax = b

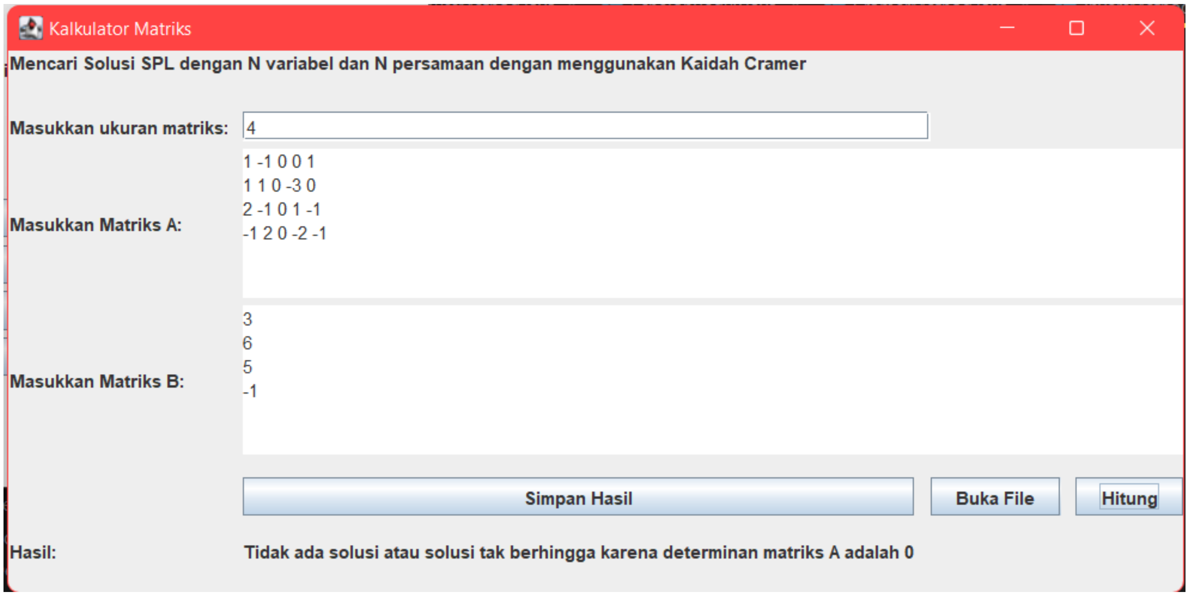
a.

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



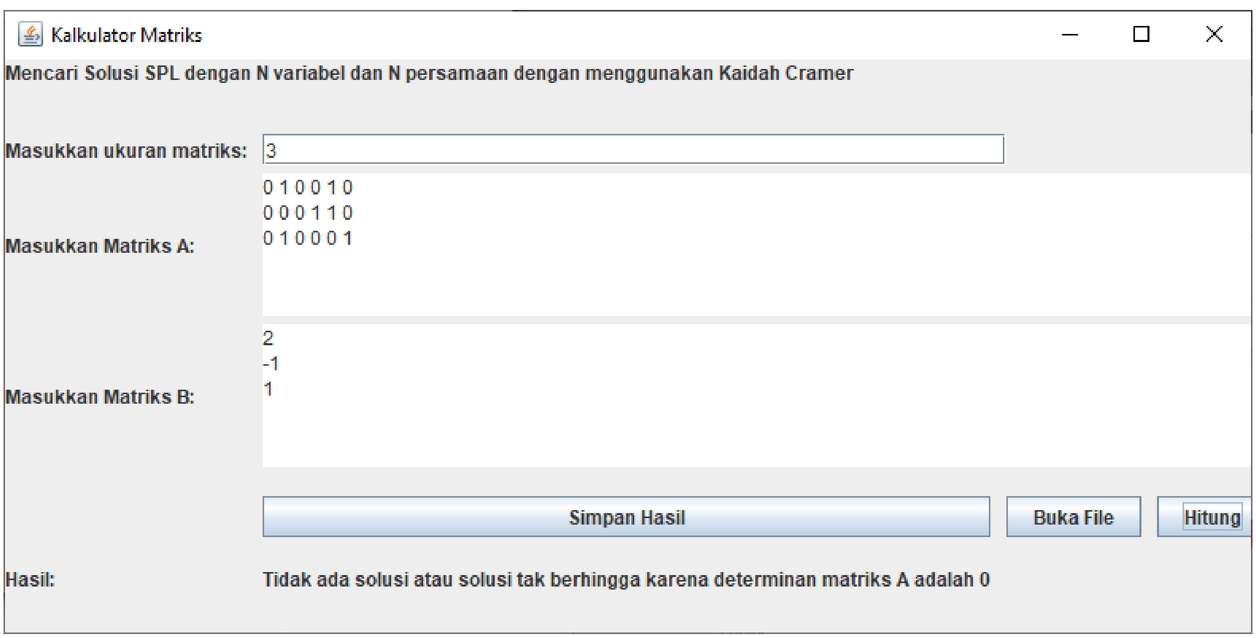
b.

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



c.

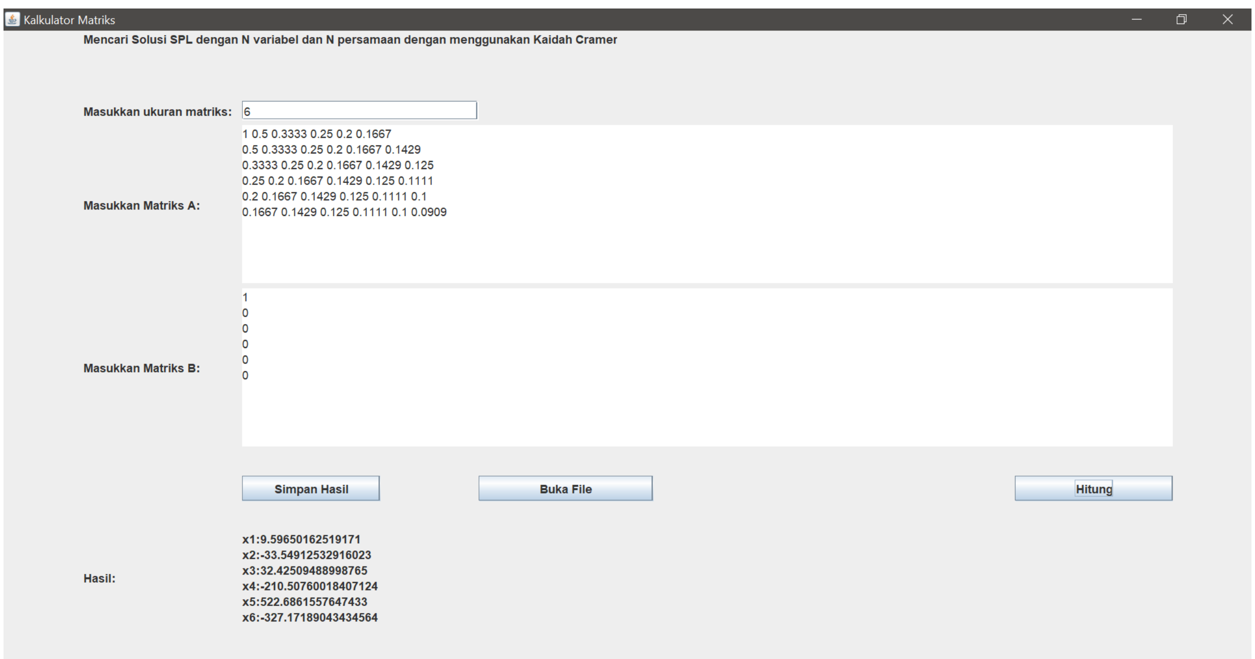
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$



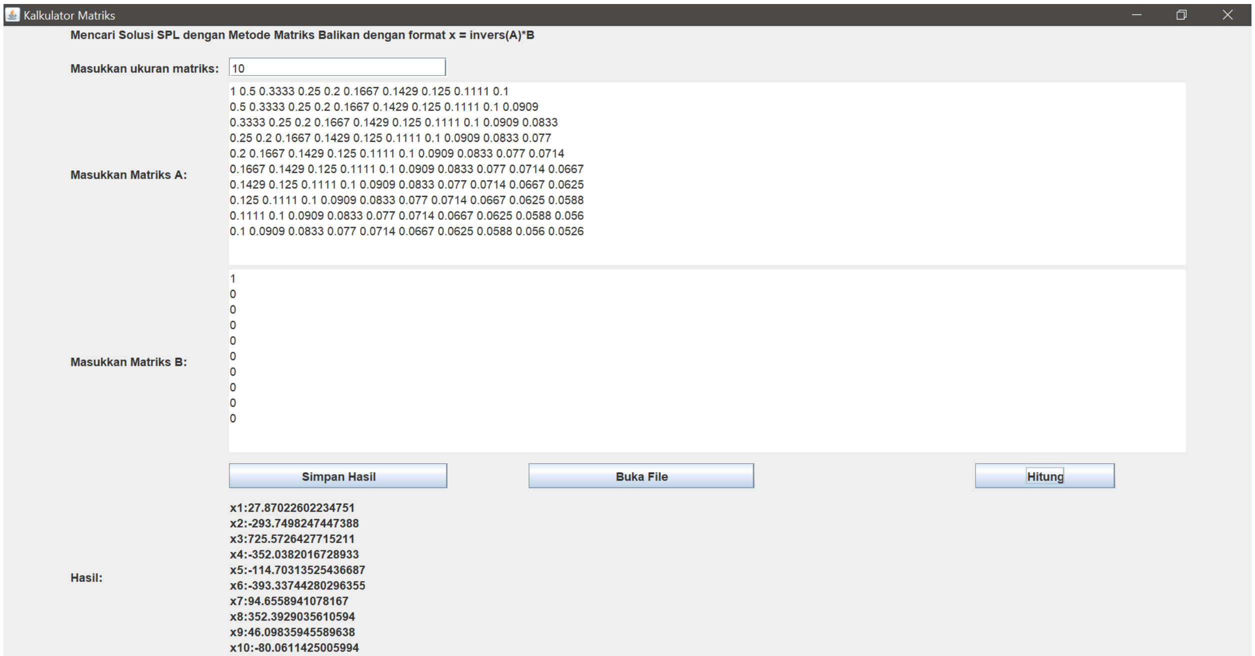
d.

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

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



Solusi Matriks Hilbert Untuk n=6

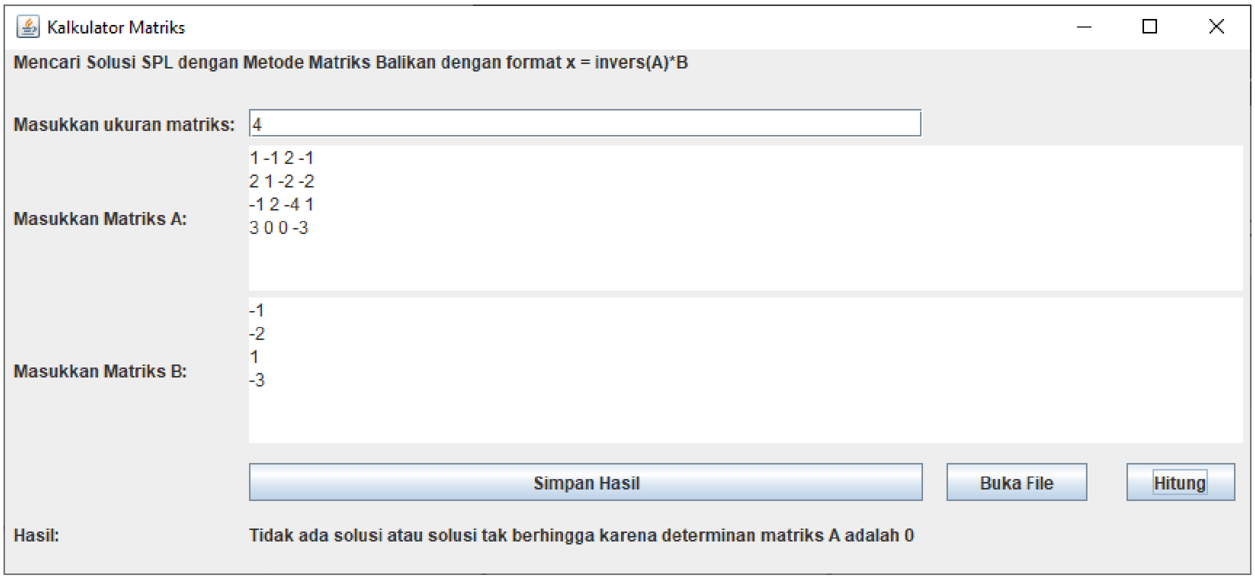


Solusi Matriks Hilbert Untuk n=10

2. Penyelesaian SPL berbentuk matriks augmented

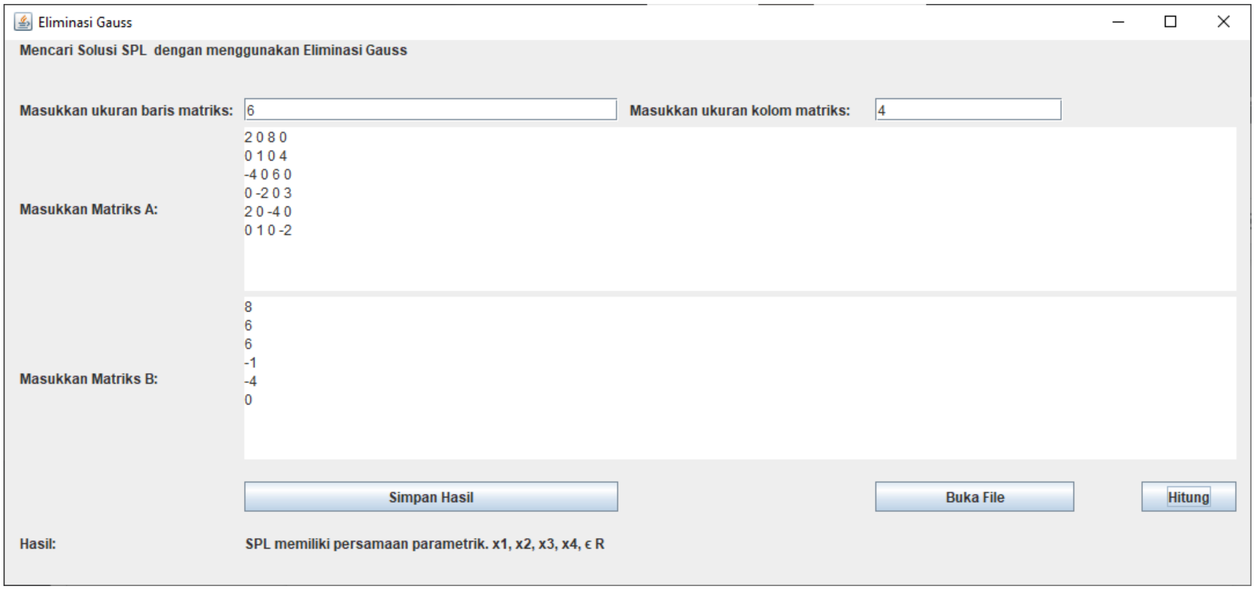
a.

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



b.

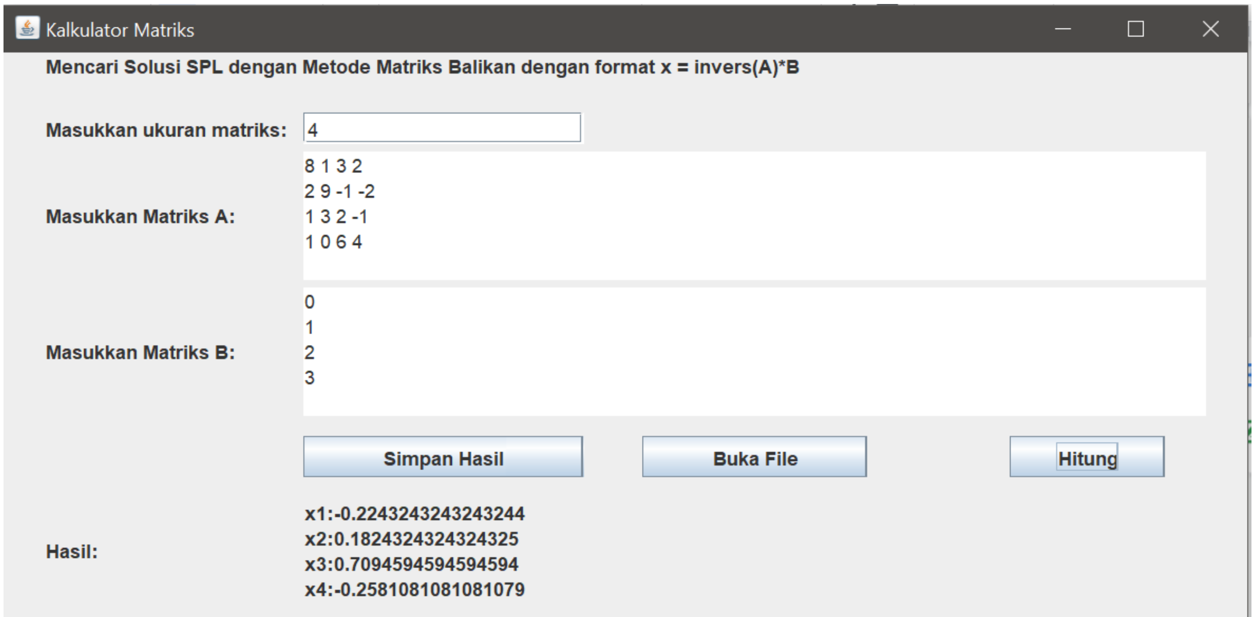
$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$$



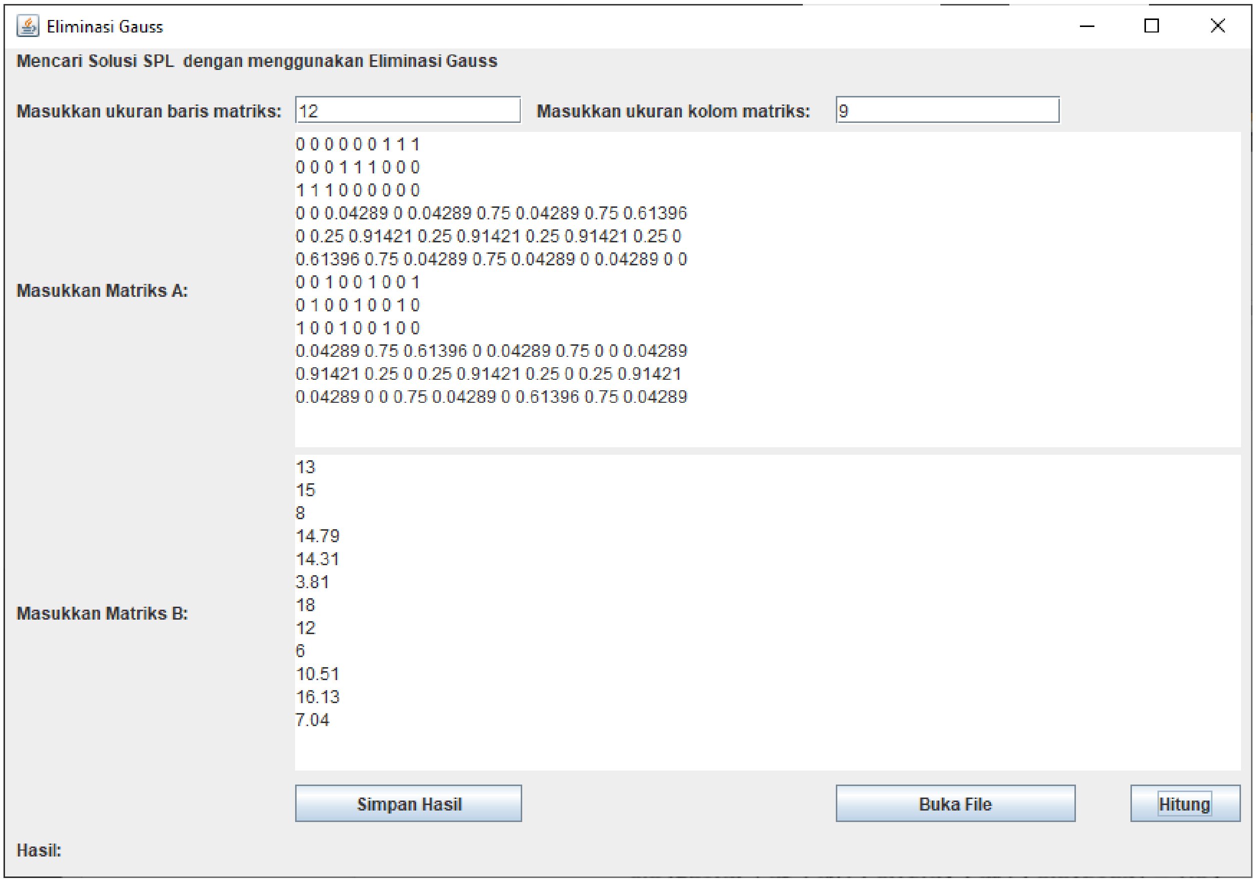
Eliminasi Gauss
Mencari Solusi SPL dengan menggunakan Eliminasi Gauss
Masukkan ukuran baris matriks:
6
Masukkan ukuran kolom matriks:
4
Masukkan Matriks A:
2 0 8 0
0 1 0 4
-4 0 6 0
0 -2 0 3
2 0 -4 0
0 1 0 -2
Masukkan Matriks B:
8
6
6
-1
-4
0
Simpan Hasil
Buka File
Hitung
Hasil:
SPL memiliki persamaan parametrik. x1, x2, x3, x4, ϵ R

3. Penyelesaian SPL berbentuk kumpulan persamaan

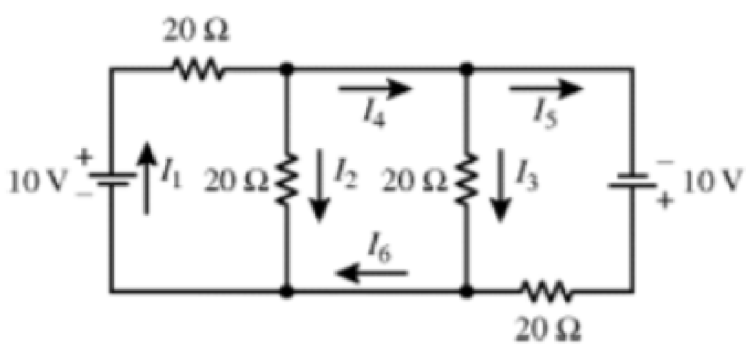
a.
$$
\begin{aligned}
8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\
2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\
x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\
x_1 + 6x_3 + 4x_4 &= 3
\end{aligned}
$$

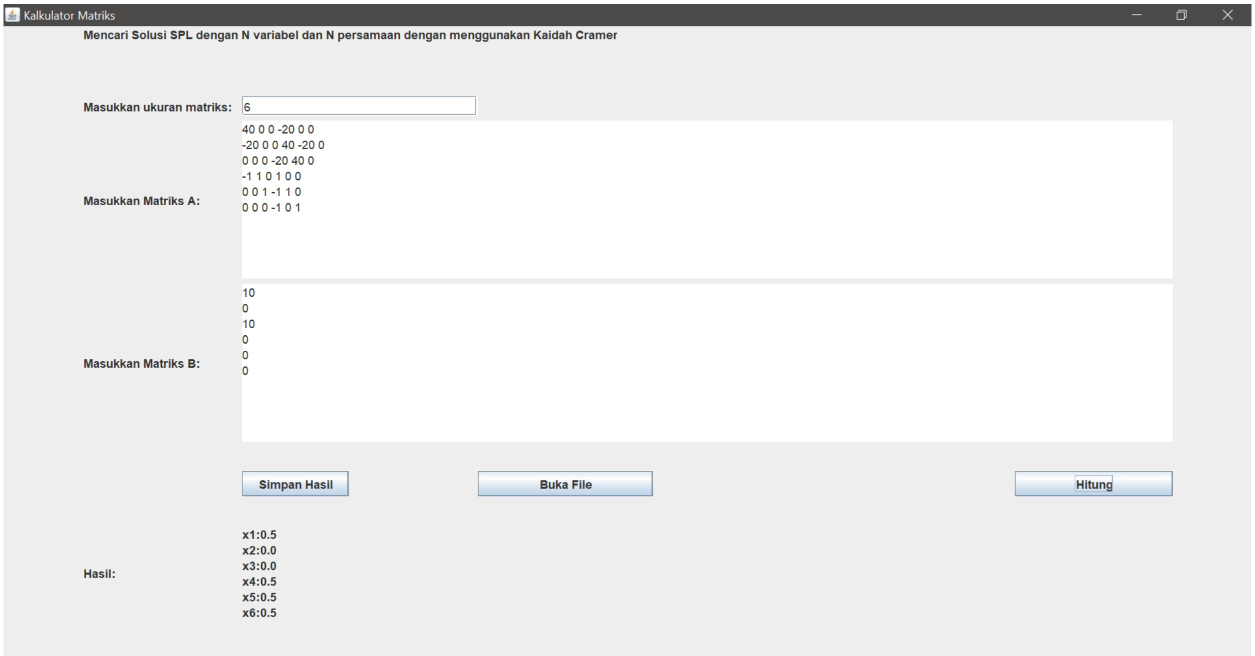


b.
$$
\begin{aligned}
x_7 + x_8 + x_9 &= 13.00 \\
x_4 + x_5 + x_6 &= 15.00 \\
x_1 + x_2 + x_3 &= 8.00 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\
0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\
0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81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



4. Studi kasus SPL: rangkaian listrik

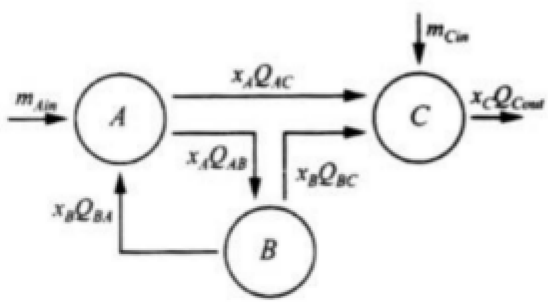




Solusi Permasalahan Rangkaian Elektrik

5. Studi kasus SPL: sistem reaktor

Lihatlah sistem reaktor pada gambar berikut



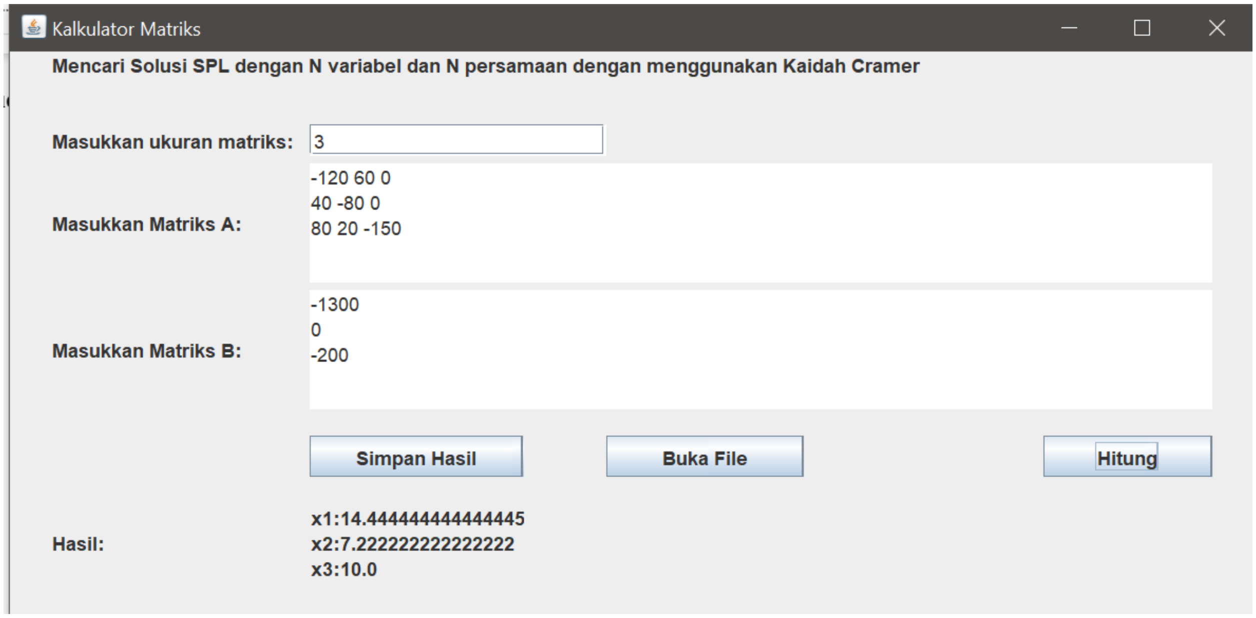
Dengan laju volume Q dalam m^3/s dan input massa m_{in} dalam mg/s. Konserva massa pada tiap inti reaktor adalah sebagai berikut:

$$\text{A:}\quad m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A = 0$$

$$\text{B:}\quad Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B = 0$$

$$\text{C:}\quad m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C = 0$$

Tentukan solusi x_A, x_B, x_C dengan menggunakan parameter berikut : $Q_{AB} = 40$, $Q_{A(}$ = 80, $Q_{BA} = 60$, $Q_{BC} = 20$ dan $Q_{Cout} = 150$ m^3/s dan $m_{Ain} = 1300$ dan $m_{Cin} = 200$ mg/s.



Solusi dari permasalahan sistem reaktor

6. Studi Kasus Interpolasi

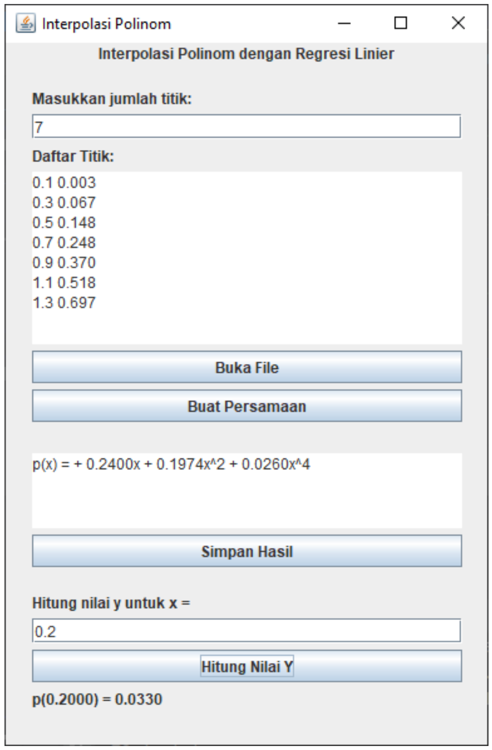
Subsoal a

a. Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai x yang akan dicari nilai fungsi $f(x)$.

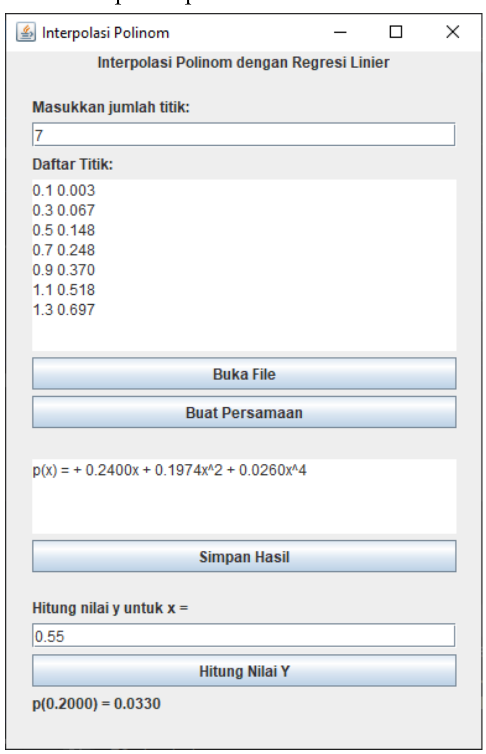
x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
$f(x)$	0.003	0.067	0. 148	0.248	0.370	0.518	0.697

Lakukan pengujian pada nilai-nilai default berikut:

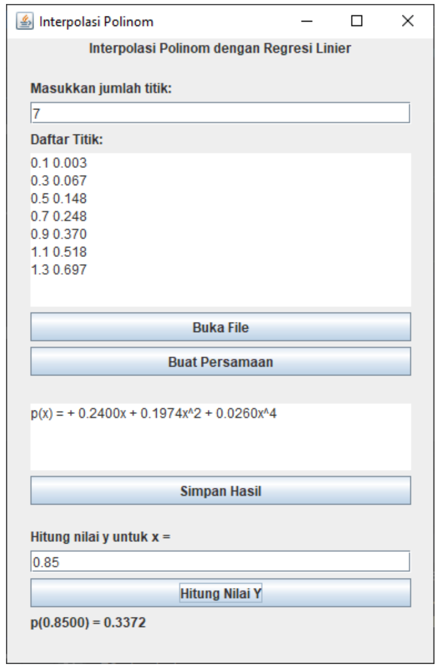
$x = 0.2$ f(x) = ?
$x = 0.55$ f(x) = ?
$x = 0.85$ f(x) = ?
$x = 1.28$ f(x) = ?



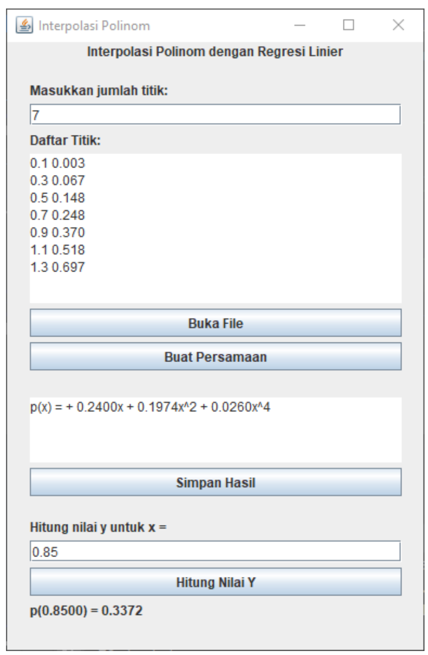
Interpolasi polinom untuk x = 0.2



Interpolasi polinom untuk x = 0.55



Interpolasi polinom untuk x = 0.85



Interpolasi polinom untuk x = 1.28

Subsoal b

b. Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2021 hingga 31 Agustus 2021:

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2021	6,567	12.624
30/06/2021	7	21.807
08/07/2021	7,258	38.391
14/07/2021	7,451	54.517
17/07/2021	7,548	51.952
26/07/2021	7,839	28.228
05/08/2021	8,161	35.764
15/08/2021	8,484	20.813
22/08/2021	8,709	12.408
31/08/2021	9	10.534

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

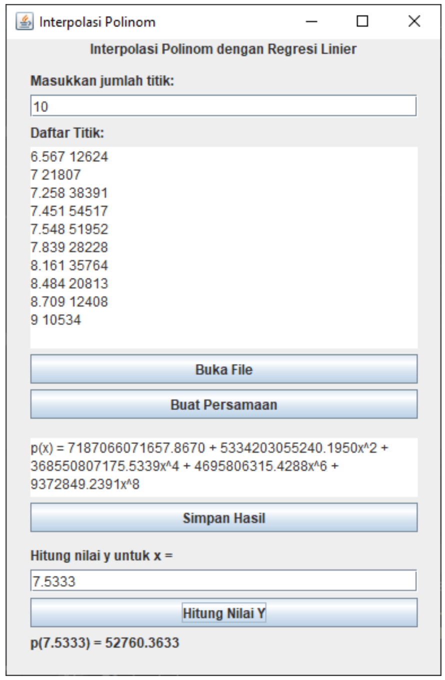
tanggal(desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut)

Sebagai **contoh**, untuk tanggal 17/06/2021 (dibaca: 17 Juni 2021) diperoleh tanggal(desimal) sebagai berikut:

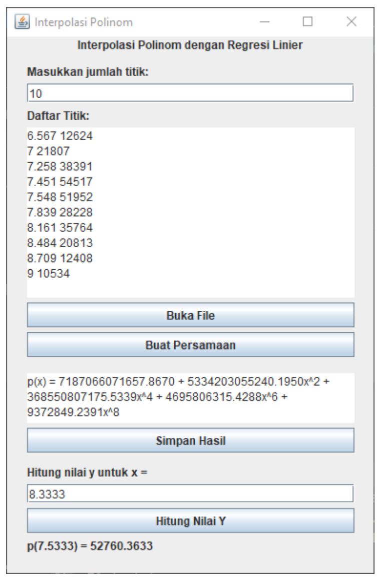
$$\text{Tanggal(desimal)} = 6 + (17/30) = 6{,}567$$

Gunakanlah data di atas dengan memanfaatkan **polinom interpolasi** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

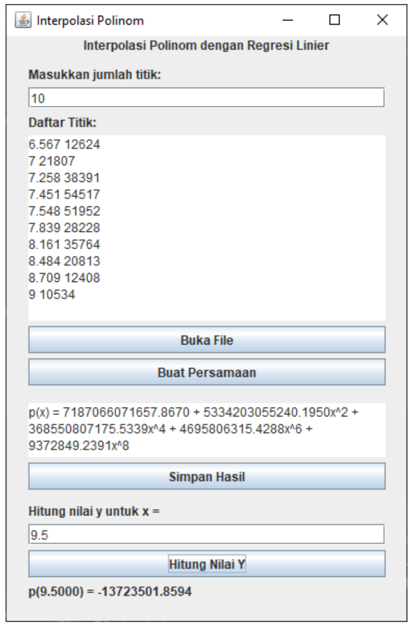
a. 16/07/2021
b. 10/08/2021
c. 05/09/2021
d. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2021.



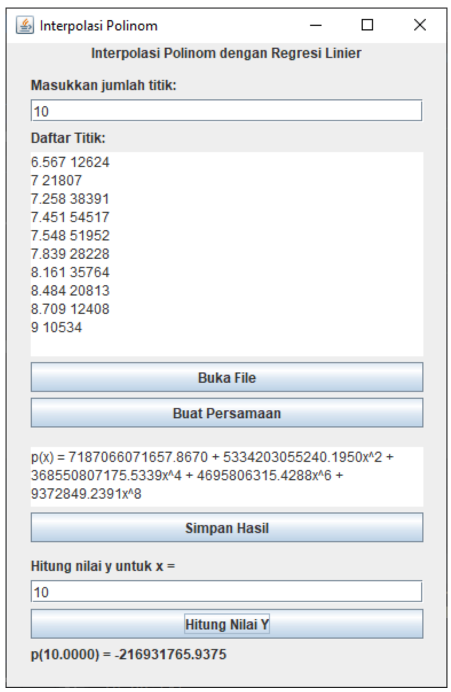
Prediksi tanggal 16/07/2021 (7 + (16/30) = 7.5333)



Prediksi tanggal 10/08/2021 (8 + (10/30) = 8.3333)



Prediksi tanggal 15/09/2021 (9 + (15/30) = 9.5)



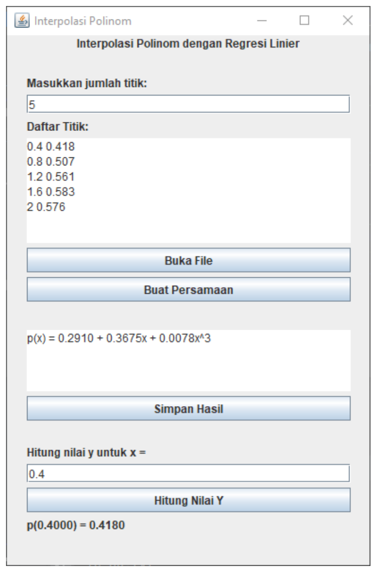
Prediksi tanggal 30/09/2021 (9 + (30/30) = 10)

Subsoal c

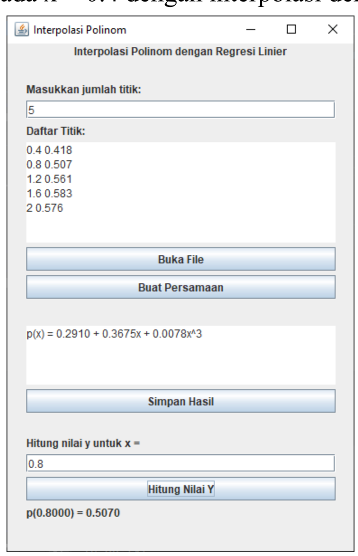
c. Sederhanakan fungsi

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

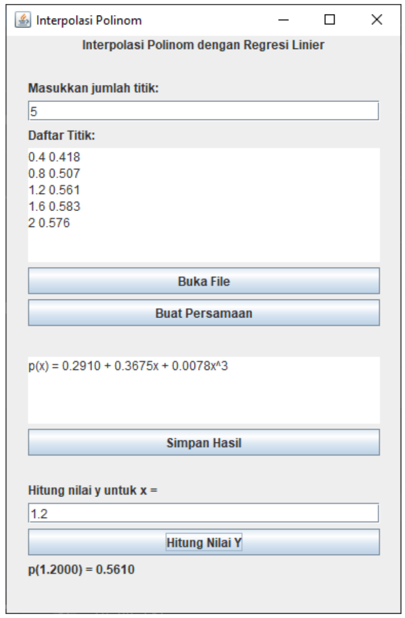
dengan polinom interpolasi derajat n di dalam selang [0, 2]. Sebagai contoh, jika n = 5, maka titik-titik x yang diambil di dalam selang [0, 2] berjarak $h = (2 – 0)/5 = 0.4$.



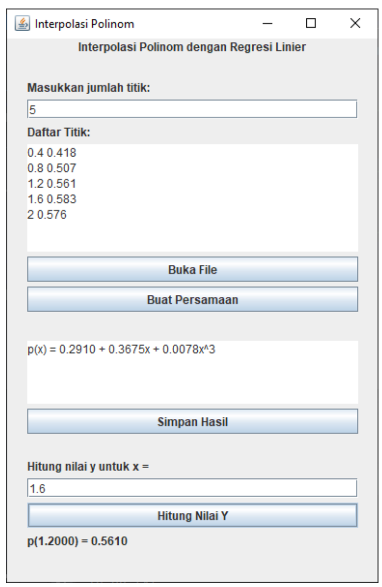
f(x) pada x = 0.4 dengan interpolasi derajat 5



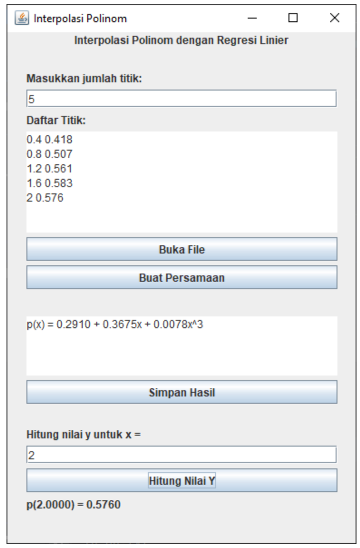
f(x) pada x = 0.8 dengan interpolasi derajat 5



f(x) pada x = 1.2 dengan interpolasi derajat 5



f(x) pada x = 1.6 dengan interpolasi derajat 5



f(x) pada x = 2 dengan interpolasi derajat 5

7. Studi Kasus Regresi Linier Berganda

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3	Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	1.10	11.2	86.0	29.48
1.10	12.9	67.4	29.39	0.91	73.3	76.3	29.40
1.15	8.3	66.8	29.69	0.87	75.4	77.9	29.28
1.03	20.1	76.9	29.48	0.78	96.6	78.7	29.29
0.77	72.2	77.7	29.09	0.82	107.4	86.8	29.03
1.07	24.0	67.7	29.60	0.95	54.9	70.9	29.37

Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

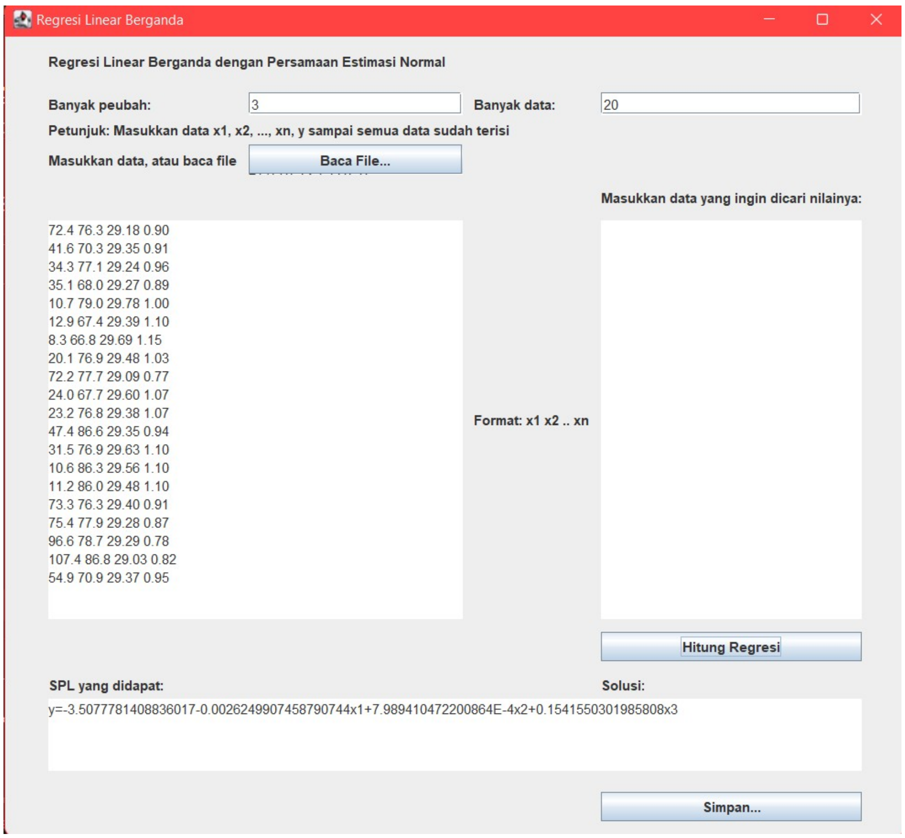
Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

$$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 \quad = 19.42$$

$$863.1b_0 \ + 54876.89b_1 + 67000.09b_2 \ + 25283.395b_3 = 779.477$$

$$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$$

$$587.84b_0 \ + 25283.395b_1 + 44976.867b_2 \ \ + 17278.5086b_3 \ = 571.1219$$



BAB V: Kesimpulan, Saran, dan Refleksi

Kesimpulan

Dengan menggunakan pemrograman, beberapa persoalan aljabar linier dan geometri, terutama yang bersifat repetitif, dapat diselesaikan. Bagi program, menghitung angka yang banyak, besar, dan repetitif bukan masalah, sehingga pembuatan program dapat memudahkan berbagai persoalan matematika. Modularitas program juga sangat penting agar program berjalan dengan efektif dan memudahkan pengembangan bersama-sama.

Secara keseluruhan, program kami dapat berjalan dengan baik. Meskipun demikian, terdapat kekurangan dalam kasus eliminasi Gauss dan Gauss-Jordan, di mana kami belum dapat membuat persamaan parametrik dalam eliminasi Gauss dengan banyak solusi berjalan dengan baik.

Saran

Untuk kelangsungan pengembangan program ini, program ini bisa diperluas fungsionalitasnya seperti membuat tab khusus untuk menyelesaikan permasalahan kelistrikan, reaktor pabrik, prediksi menggunakan regresi beserta grafiknya, dsb.

Refleksi

Dari tugas ini kami belajar tentang cara membuat algoritma yang bisa menyelesaikan berbagai permasalahan yang apabila diselesaikan secara manual akan menjadi sangat repetitif.

Kami juga belajar cara beradaptasi dengan cepat karena tugas besar ini menggunakan bahasa Java yang tergolong baru bagi kami dalam menggunakannya sehingga bisa dibilang pembuatan tubes ini dimulai dari nol.

Kedepannya juga mungkin kami akan lebih strict dalam membagi tugas serta membuat semacam “milestone” agar pemantauan progressnya bisa lebih detail.

DAFTAR REFERENSI

- Materi Kuliah IF2123 Aljabar Linier dan Geometri oleh Rinaldi Munir dan tim https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/algeo21-22.htm
- Gireesh Sundaram - Building Linear Regression (Least Squares) with Linear Algebra https://towardsdatascience.com/building-linear-regression-least-squares-with-linear-algebra-2adf071dd5dd
- Statmat - Contoh Soal Pembahasan Regresi Linier Berganda https://www.statmat.net/regresi-linier-berganda/
- JetBrains IntelliJ IDEA Documentation - Compile and build applications with IntelliJ IDEA https://www.jetbrains.com/help/idea/compiling-applications.html
- Dokumentasi Javax Swing API https://docs.oracle.com/javase/7/docs/api/javax/swing/