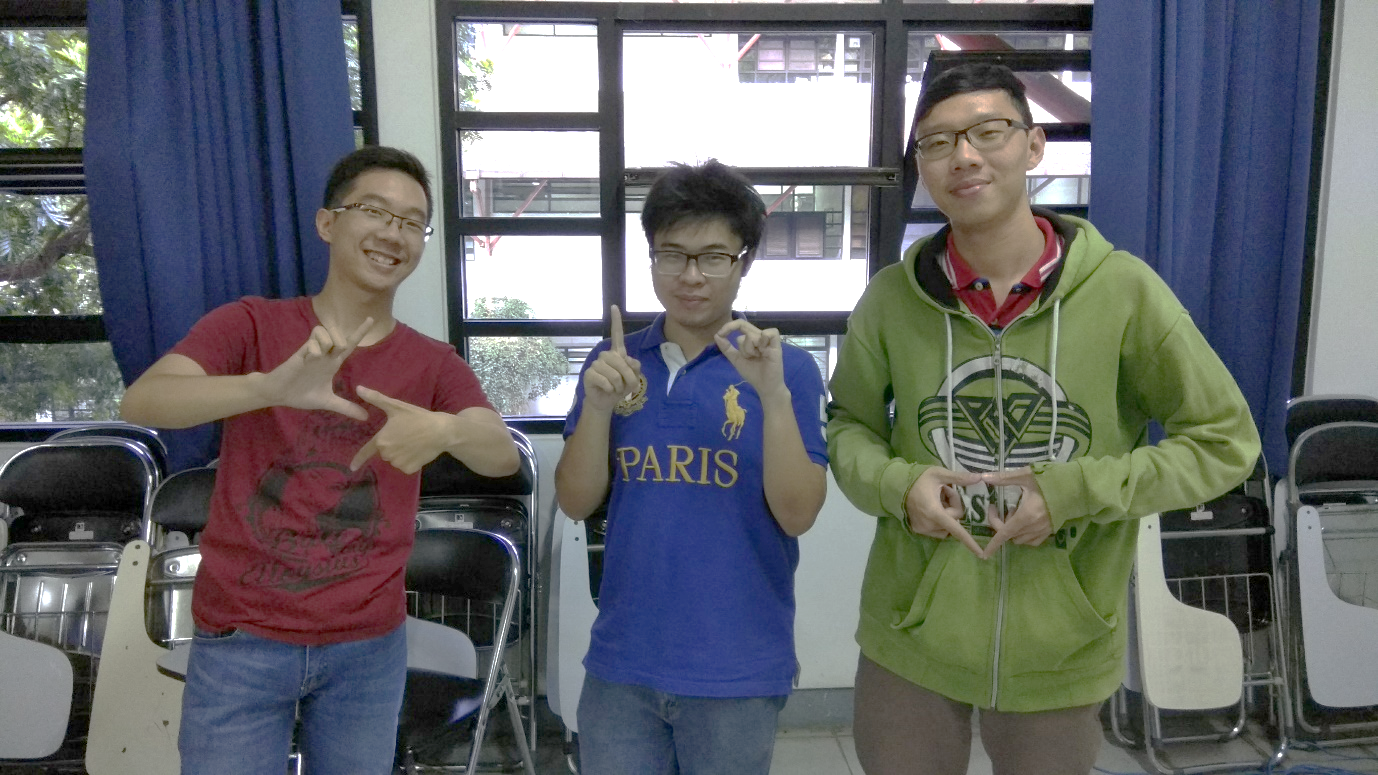
Jonathan Tjandra – 13516058

Shandy – 13516097

Jessin Donnyson – 13516112

Tugas Besar I

Aplikasi Aljabar Lanjar pada Metode Numerik



Abstrak

Laporan tugas pembuatan program untuk menghitung   
solusi SPL secara numerik.  
  
  
- IF2123 Aljabar Geometri -

# Bab I: Deskripsi Masalah

Tujuan utama dari tugas besar ini adalah membuat program untuk menghitung solusi Sistem Persamaan Linier (SPL) secara numerik dalam bahasa pemrograman *Java* dengan menggunakan metode eliminasi *Gauss* dan/atau *Gauss-Jordan*. SPL dapat memiliki solusi unik, banyak solusi, atau solusi tidak ada. SPL juga digunakan dalam menentukan persamaan polinom interpolasi.

Karena perhitungan menggunakan representasi bilangan titik-kambang (*floating point*) di dalam komputer, maka untuk meminumkan galat perhitungan, digunakan strategi *pivoting* dalam memilih baris yang dijadikan basis dalam operasi baris elementer. Bahasa *Java* digunakan sebagai bahan belajar penggunaan bahasa pemrograman selain *C* dan *Pascal* yang sudah digunakan selama ini.

Spesifikasi program :

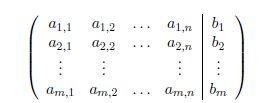
1. Program menggunakan bahasa *java*.
2. Program tidak perlu berbasis GUI, cukup berbasis teks saja.
3. Program harus mampu menerima input dari papan ketik dan file eksternal.
4. Keluaran program ditampilkan ke layar monitor dan disimpan dalam file eksternal (format bebas).
5. Program mampu memecahkan SPL yang diketahui *n* peubah dan *m* persamaan.
6. Program mampu memecahkan interpolasi fungsi berderajat *n* yang diketahui f(x) dan x nya sejumlah (*n+1)*.
7. Program mampu menangani SPL yang memiliki solusi unik dan menampilakan solusinya.
8. Program mampu menangani SPL yang memiliki solusi tak terbatas dan menampilakan solusinya dalam parameter.
9. Program mampu menangani SPL yang tidak memiliki dan menampilakan pesan ‘Tidak ada solusi’.

# Bab II: Landasan Teori

Beberapa teori dasar dalam aljabar geometri yang dipakai adalah eliminasi *Gauss*, eliminasi *Gauss-Jordan*, strategi *pivoting* ,serta interpolasi polinom.

## Eliminasi Gauss

Eliminasi Gauss adalah suatu metode untuk mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana lagi dengan melakukan operasi baris elementer. Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Jika matriks A adalah matriks persamaan linear, dan matriks B adalah matriks kolom hasil dari sistem persamaan linear, maka dapat dibuat matriks *augmented* sebagai berikut :

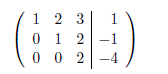


Gambar 2.1 *Augmented Matrix*

Operasi baris elementer (OBE) yang dilakukan pada matriks merupakan :

1. Tukar satu baris elemen dengan baris lainnya
2. Menambahkan kelipatan baris ke baris lainnya
3. Membagi suatu baris dengan suatu konstanta

## Tujuan akhir dari OBE dalam eliminasi *Gauss* adalah mendapatkan matriks *echelon*

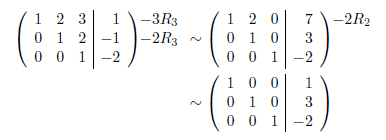


Gambar 2.2 *Echelon Matrix*

Setelah didapatkan matriks *echelon*, kemudian dilakukan subtitusi balik untuk mendapatkan hasil sistem persamaan linear tsb.

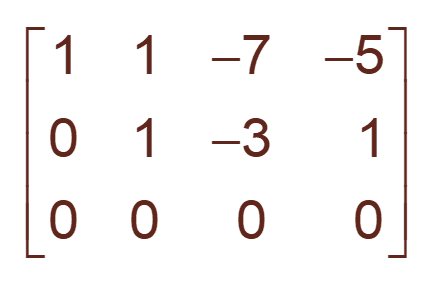
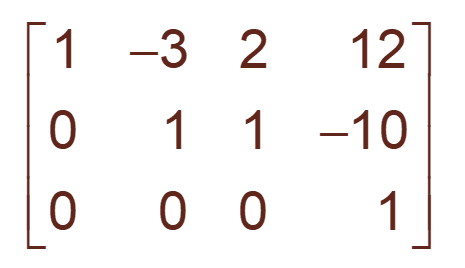
## Eliminasi Gauss-Jordan

Metode eliminasi *Gauss-Jordan* merupakan versi eliminasi *Gauss* yang dilanjutkan sampai ke tahap *row-reduced echelon*. Jika pada eliminasi *Gauss* dilakukan subtitusi balik, maka pada eliminasi *Gauss-Jordan* ini dilakukan OBE sehingga didapatkan matriks *row-reduced echelon*. Hasil SPL adalah elemen pada kolom paling kanan dari matriks *row-reduced echelon*.



Gambar 2.3 Konversi *Echelon* menjadi *Row-reduced Echelon*

Kasus khusus dalam eliminasi *Gauss* dan *Gauss-Jordan* terjadi jika matriks *Echelon* mempunyai satu baris atau lebih yang berisikan hanya 0, maka sistem persamaan linear tsb. memiliki tak-hingga solusi. Kasus lainnya terjadi saat matriks *Echelon* mempunyai satu baris atau lebih yang berisikan hanya 0 kecuali elemen paling kanan pada baris itu, maka sistem persamaan linear tsb. tidak memiliki solusi.

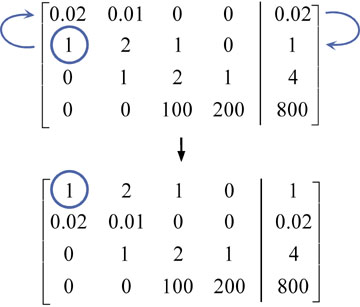


Gambar 2.4 Matriks SPL Solusi Banyak Gambar 2.5 Matriks SPL Tidak Punya Solusi

## Tatancang Pemorosan ( *Pivoting )*

Dalam melakukan OBE pada matriks SPL, apabila SPL diselesaikan dengan computer, maka akan timbul galat pembulatan pada solusinya karena operasi aritmerika bilangan titik-kambang (*floating point)*. Untuk memperoleh solusi SPl yang mengandung galat yang minimal akibat pembulatan, maka digunakan tatancang pemorosan (*pivoting strategy*).

*Pivoting* dilakukan dengan cara memilih *pivot* dari semua elemen pada kolom p yang mempunyai nilai mutlak terbesar, lalu pertukarkan baris yang mengandung pivot ke baris teratas.



Gambar 2.6 Ilustrasi Tatancang Pemorosan

## Interpolasi

Interpolasi adalah suatu metode untuk mengubah fungsi menjadi suatu polinom berderajat *n*. Interpolasi menggunakan sekumpulan (minimal *n + 1*) data domain dan range dari suatu fungsi, dan membuat suatu sistem persamaan linear yang menggambarkan fungsi tersebut dalam model polinomial.

*pn(x) = a0 + a1x + a2x2 + … + anxn*

Dengan menyulihkan (*xi, yi)* yang diketahuike dalam persamaan polinom di atas, untuk *i* = 0, 1, 2, *…, n,* akan diperoleh *n* buah sistem persamaan lanjar dalam a0, a1, a2, …, an,

a0 + a1x0 + a2x02 + ... + anx03 = y0

a0 + a1x1 + a2x12 + ... + anx13 = y1

... ...

a0 + a1xn + a2xn2 + ... + anxn3 = yn

Solusi dari sistem persamaan ini bisa diperoleh menggunakan metode eliminasi *Gauss / Gauss-Jordan*.

# Bab III: Implementasi Program

Program dibuat menggunakan 2 class, yaitu:

* Class Matrix yang berisi method untuk memproses sebuah matriks, mulai dari membentuk matriks (konstruktor), memproses matriks, dan menyimpan / menampilkan hasil akhir.
* Class MainMenu yang berisi menu utama yang digunakan untuk mengakses method-method pada class Matrix sehingga Matriks dapat diproses

## Class Matrix

Atribut:

* **int** row; /\* Baris \*/
* **int** col; /\* Kolom \*/
* **int** Solution\_type; /\* Tipe Solusi Gauss-Jordan \*/
* String format = "%7.3f "; /\* Format elemen yang diprint \*/
* **double** [][] matrix; /\* Array 2 dimensi (matriks) \*/

Method:

* **public** Matrix()

/\* Konstruktor matriks secara default \*/

* **public** Matrix(**int** row, **int** col)

/\* Konstruktor matriks jika diketahui baris dan kolom \*/

* **private** Matrix(Matrix A)

/\* Konstruktor matriks (untuk copy matriks) \*/

* **public** **double** Elmt(**int** row, **int** col)

/\* Mengakses elemen matriks \*/

* **public** **void** show()

/\* Menampilkan matriks ke layar \*/

* **public** **void** show(BufferedWriter writer)

/\* Menyimpan matriks ke file \*/

* **public** **void** showEx(Matrix B, BufferedWriter writer)

/\* Menampilkan matriks augmented \*/

* **public** **void** showFinal(BufferedWriter writer)

/\* Menampilkan matriks dengan format hasil akhir \*/

* **public** **void** showPolynom(BufferedWriter writer)

/\* Menampilkan polinom \*/

* **private** **boolean** eqDim(Matrix B)

/\* Check apakah dimensi sama \*/

* **public** **boolean** eq(Matrix B)

/\* Check apakah matriks sama \*/

* **public** Matrix plus(Matrix B)

/\* Mengembalikan A+B \*/

* **public** Matrix minus(Matrix B)

/\* Mengembalikan A-B \*/

* **public** Matrix times(Matrix B)

/\* Mengembalikan A\*B \*/

* **private** **void** swap(**int** i, **int** j)

/\* Menukar 2 baris pada matriks \*/

* **public** Matrix transpose()

/\* Transpose matriks \*/

* **public** **static** Matrix random(**int** M, **int** N)

/\* Membuat matriks acak \*/

* **public** **static** Matrix unsolvable(**int** M, **int** N,**int** X)

/\* Membuat unsolvable matrix \*/

* **public** **static** Matrix interpolate(Matrix func, **int** deg)

/\* Membuat matrix untuk interpolasi polinom \*/

* **public** Matrix solve(Matrix result)

/\* Menyelesaikan sistem persamaan linear / interpolasi \*/

* **void** fillMatrix()

/\* Isi matriks dengan input user (dimensi diinput oleh user) \*/

* **void** fillMatrix(**int** row, **int** col)

/\* Isi matriks dengan input user (dimensi sudah diketahui) \*/

* **void** fillMatrixExt(Matrix M2)

/\* Membaca baris, kolom, dan isi matrix dari file external, M2 adalah matrix untuk menampung sisi ‘hasil’ dari SPL. \*/

* **void** save(String input, BufferedWriter writer)

/\* Menyimpan string input kedalam file external dan mencetak string ke layar. \*/

## Class MainMenu

Atribut :

* **private** Scanner in = **new** Scanner (System.in);

/\* Scanner \*/

* Matrix M1 = **new** Matrix();

/\* Matriks persamaan linear \*/

* Matrix M2 = **new** Matrix();

/\* Matriks ‘hasil’ setiap persamaan linear \*/

* **boolean** X;

/\* Flag untuk kedua menu didalam fungsi run \*/

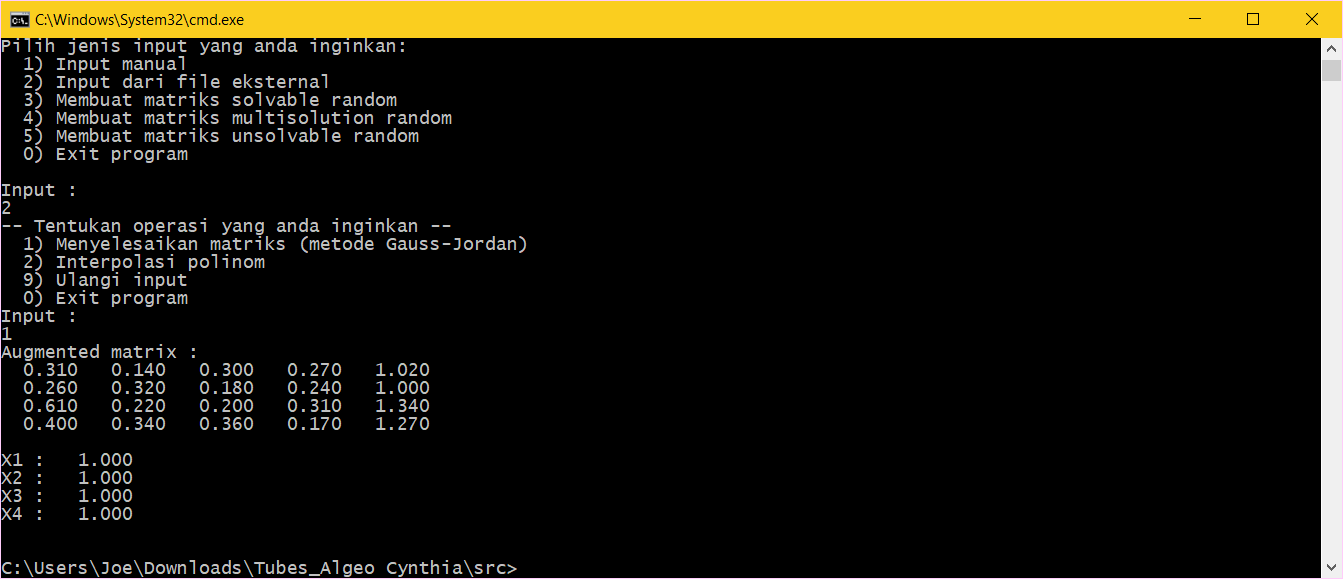
Method :

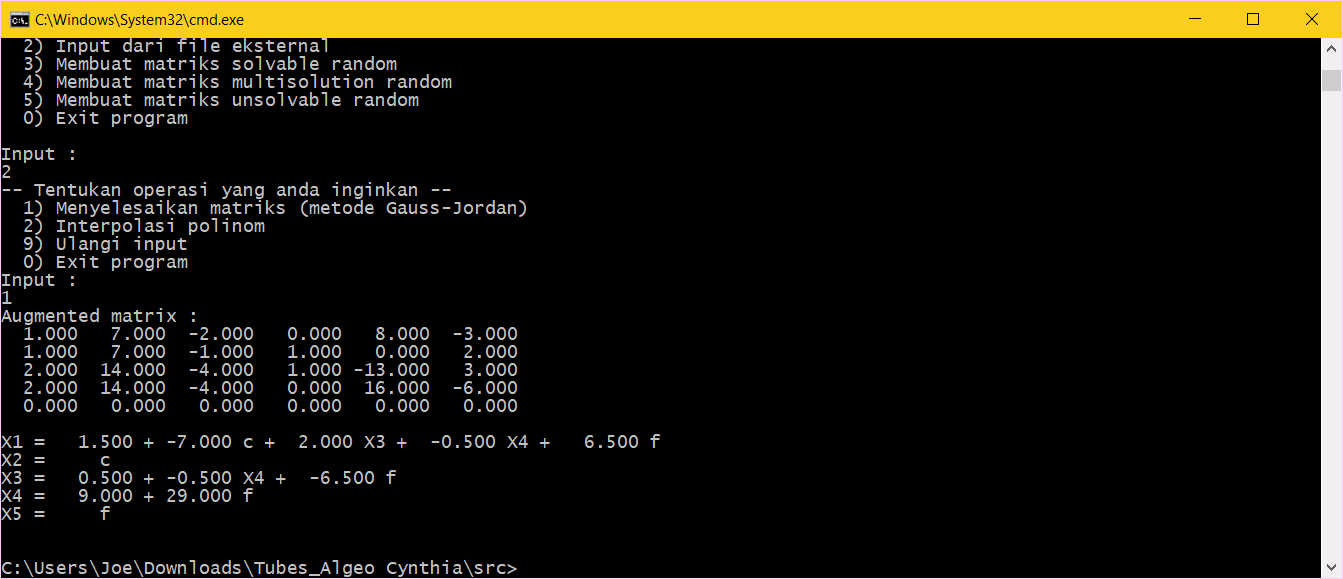
* **public** **void** run()

/\* Isi dari menu (memakai label untuk menjalankan nested menu) \*/

* **public** **static** **void** main(String[] args)

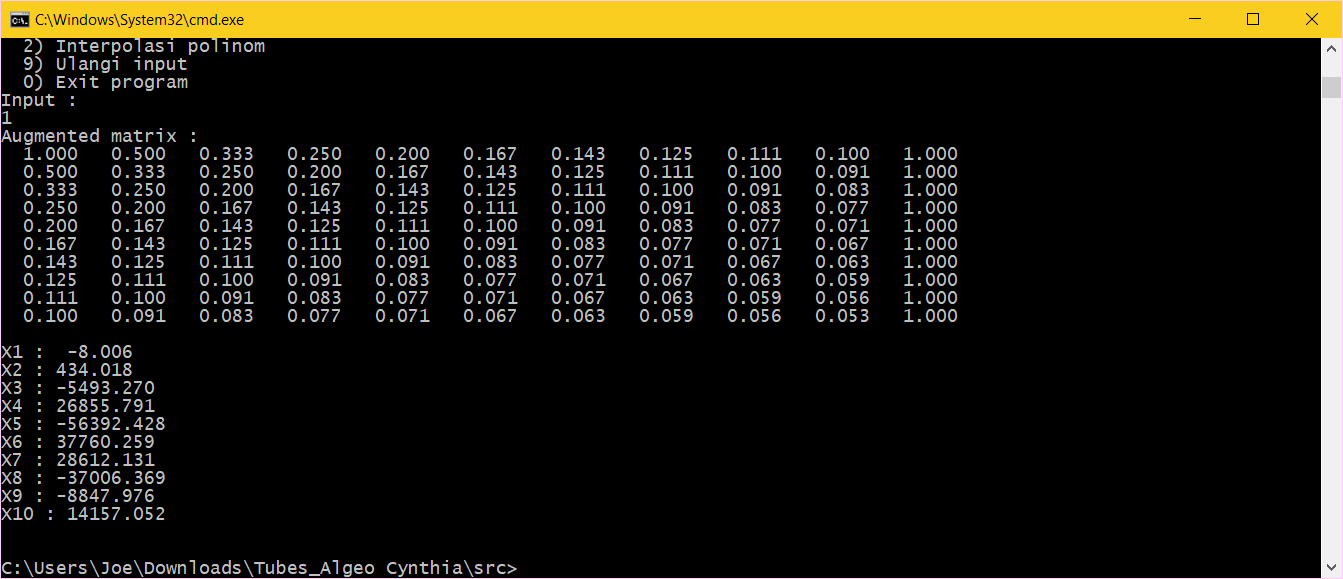
# Bab IV: Eksperimen

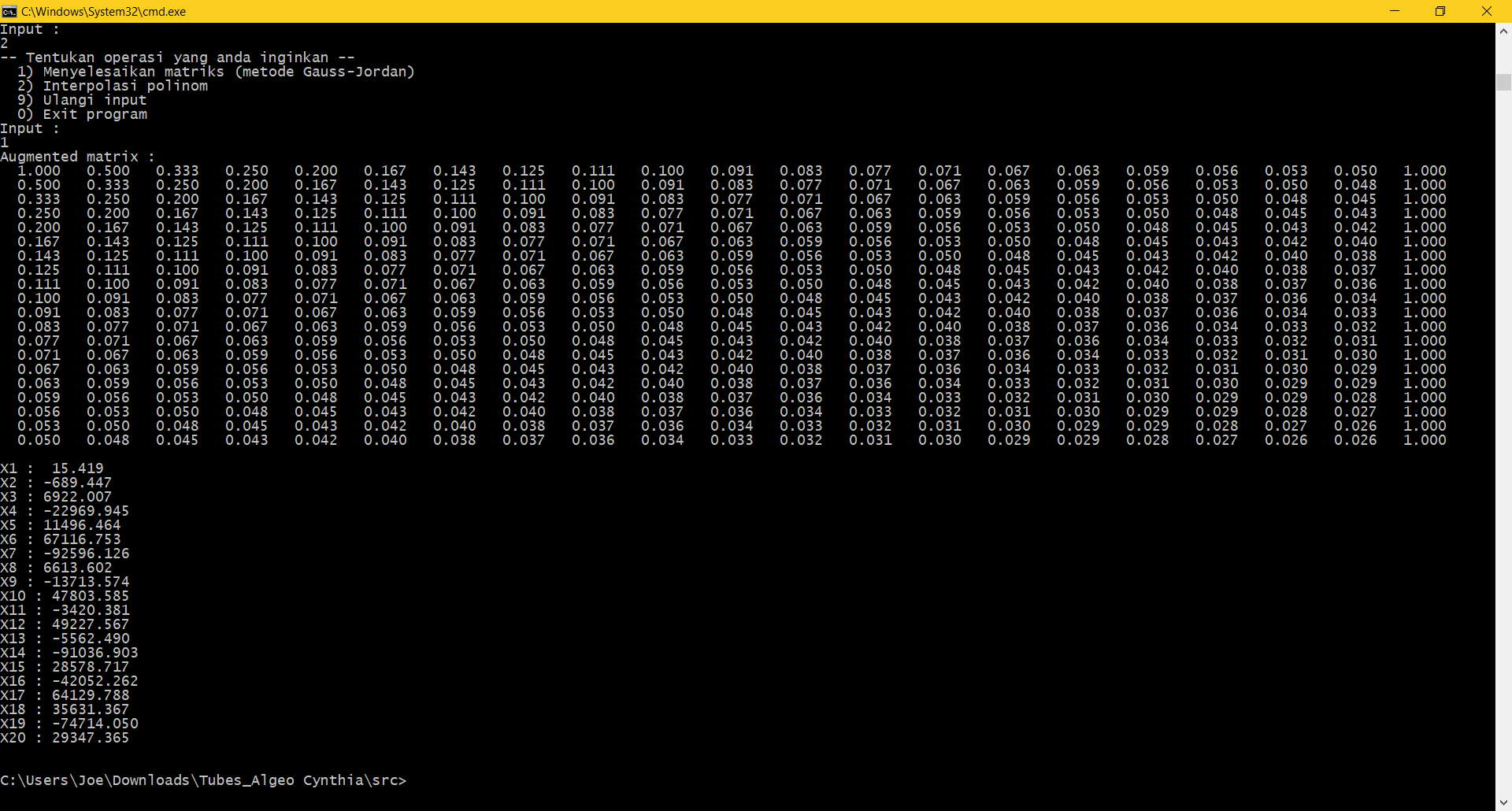
1. Soal a) 
2. Soal b)



1. Soal c)

Dicoba Matrix Hilbert untuk n=10 dan 20





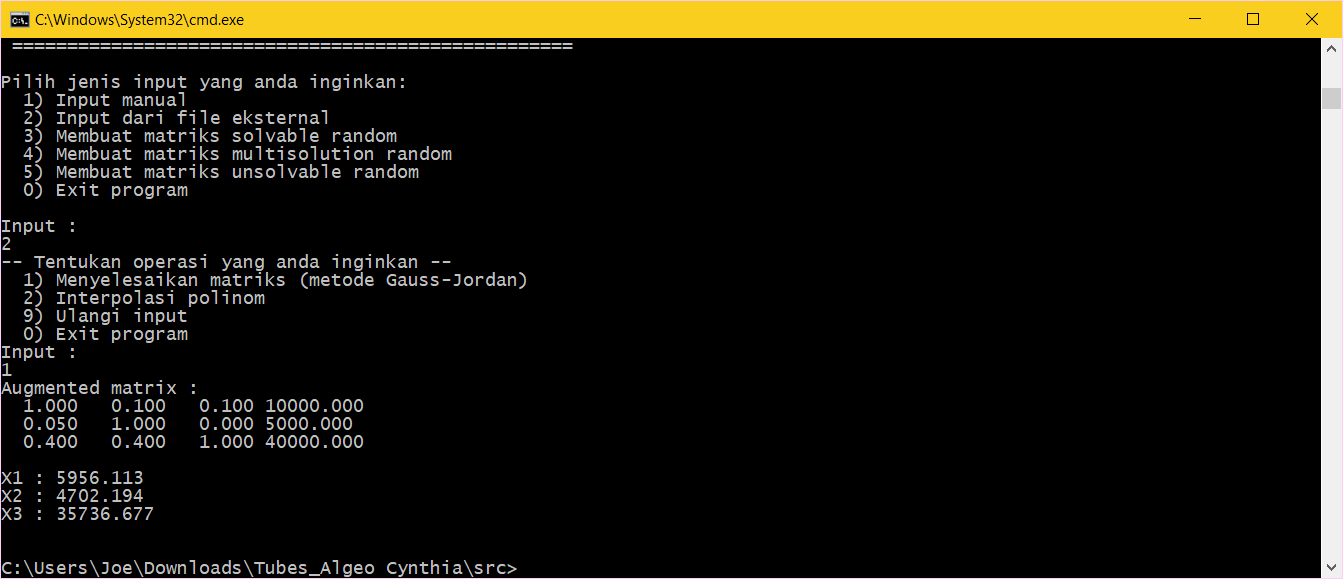
1. Soal d)

Dengan keterangan:

X1 = CSR

X2 = Pajak daerah

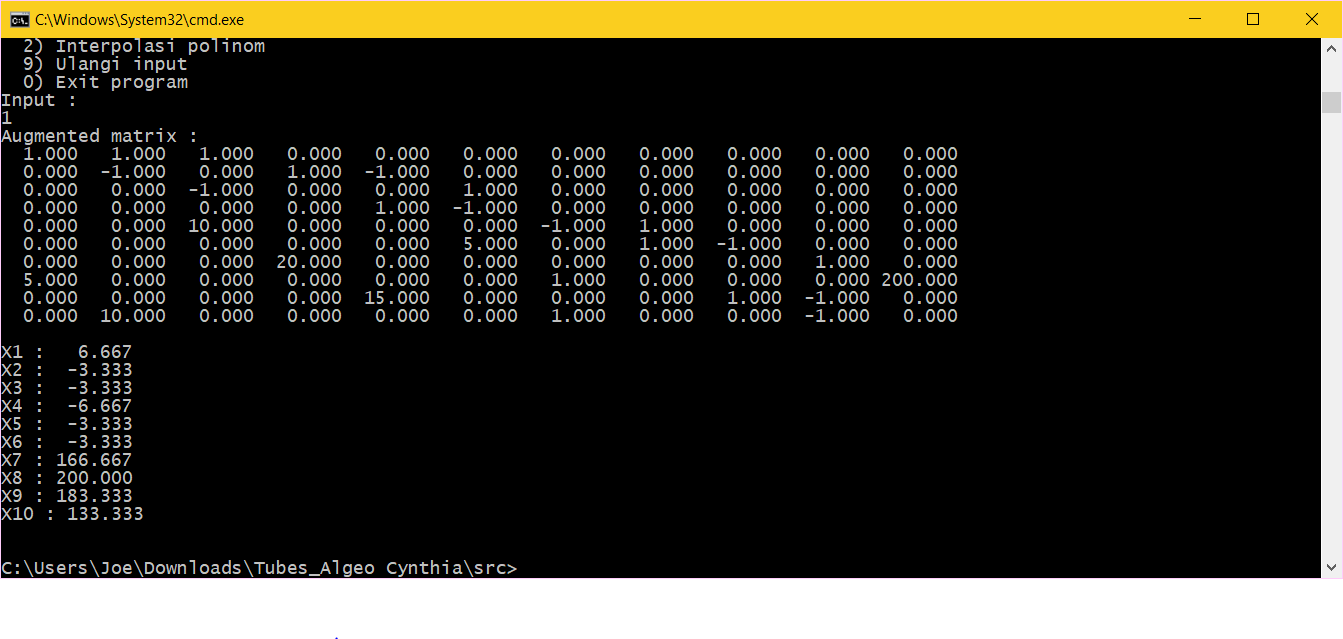
X3 = Pajak federal



1. Soal e)

Dengan keterangan:

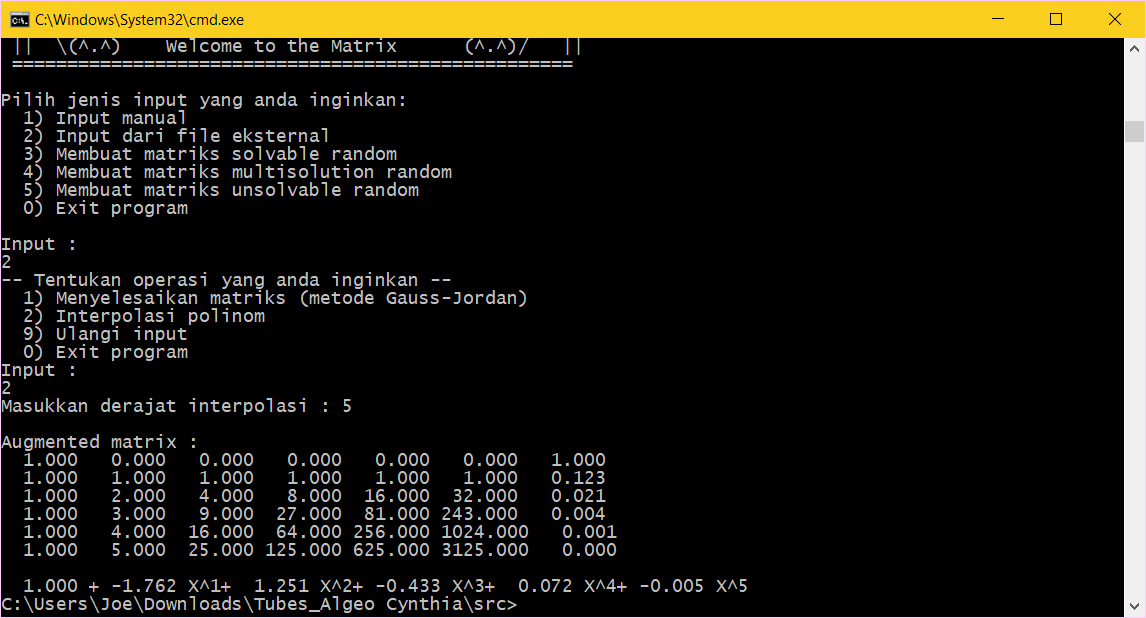
|  |  |
| --- | --- |
| X1 = i12 | X6 = i43 |
| X2 = i52 | X7 = V2 |
| X3 = i32 | X8 = V3 |
| X4 = i65 | X9 = V4 |
| X5 = i54 | X10 = V5 |



1. Soal f)

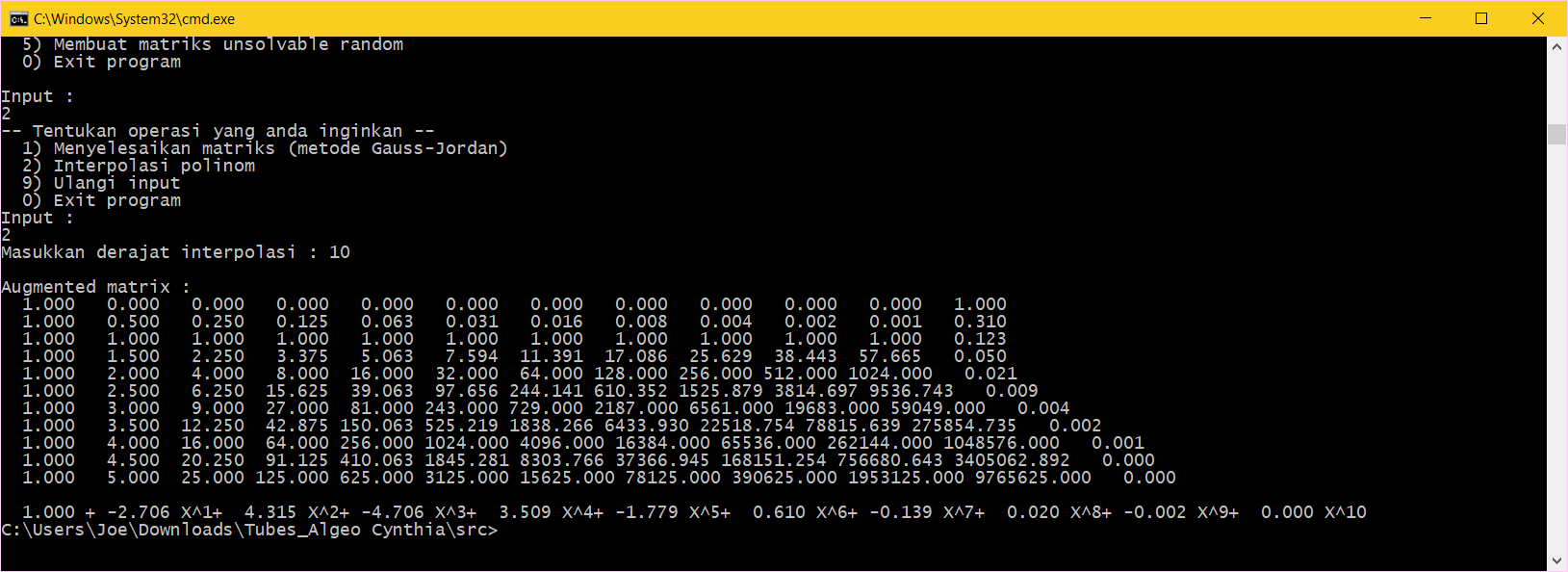
**Percobaan dengan selang [0..5],**

Untuk n = 5, didapatkan :



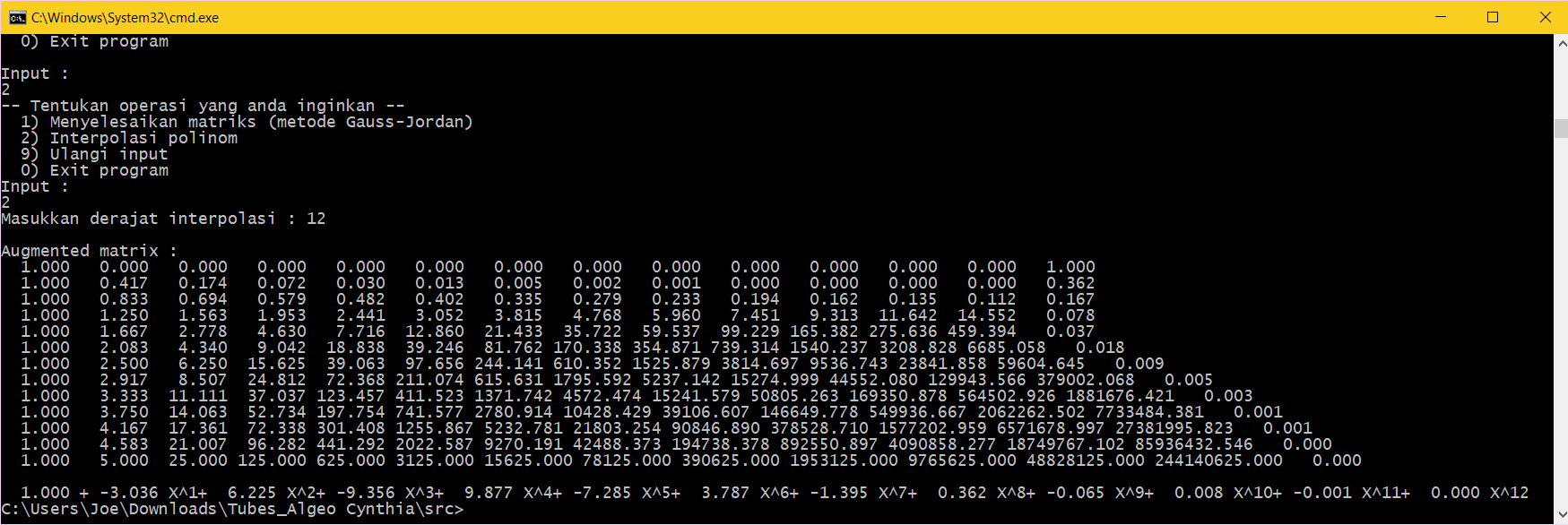
f(x) = 1 - 1.762x + 1.251 x2 - 0.433 x3 + 0.072 x4 - 0.005 x5

Untuk n=10, didapatkan :



f(x) = 1 -2.706 x + 4.315 X2 + -4.706 X3 + 3.509 X4 + -1.779 X5 + 0.610 X6 + -0.139 X7 + 0.020 X8+ -0.002 X9 + 0.000 X10

Untuk n=12, didapatkan :



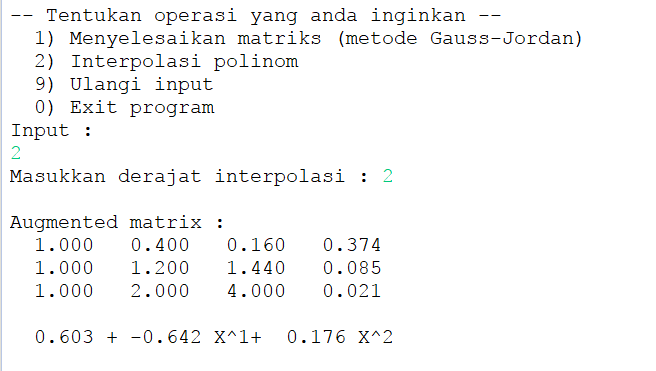
f(x) = 1 - 3.036 X + 6.225 X2 - 9.356 X3 + 9.877 X4 -7.285 X5 + 3.787 X6 - 1.395 X7 + 0.362 X8 - 0.065 X9 + 0.008 X10 -0.001 X11 + 0.000 X12

**Percobaan dengan selang [-2..2]**

NB : Selang minus tidak bisa dipakai karena terdapat fungsi akar.

Untuk n=5 , didapatkan:

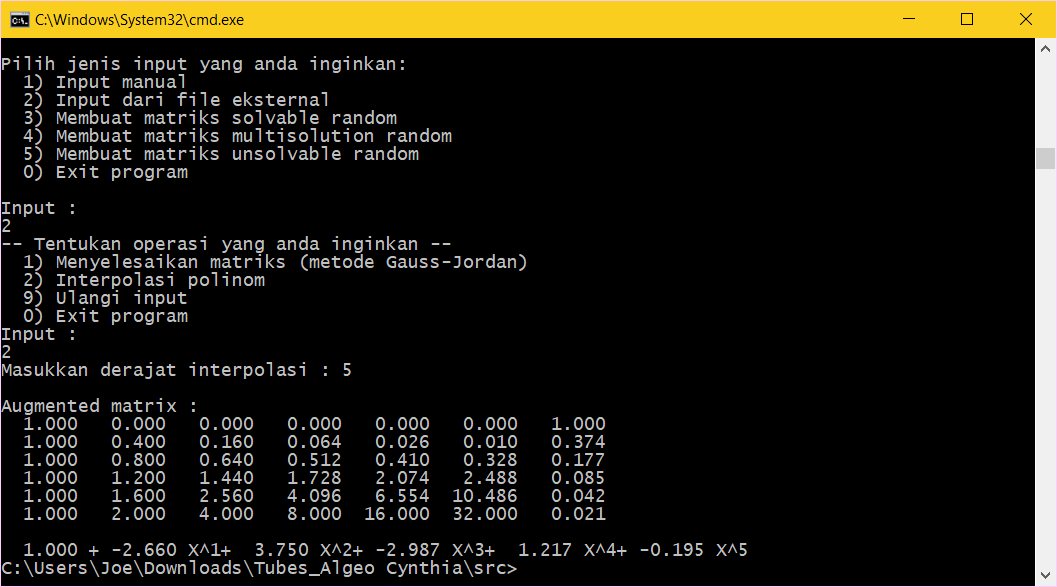
**Nb: n= 3 -> (-2, -1.2 , -0.4 tidak bisa)**



f(x) = 0.603 – 0.642 X + 0.176 X2

Untuk n = 10, didapatkan :

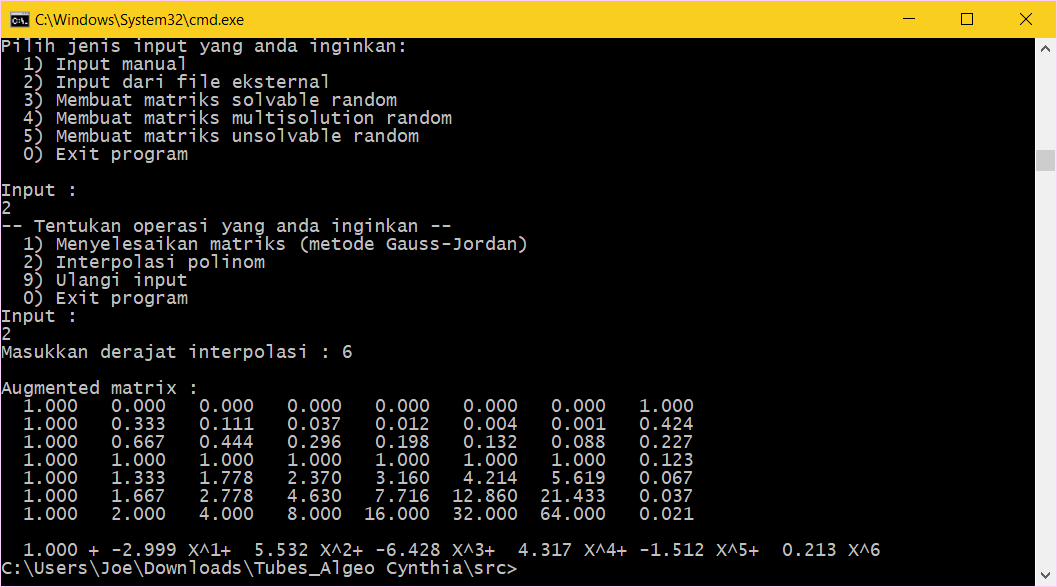
**Nb: n=10 -> 6 (-2,-1.6,-1.2,-0,8,-0.4 tdk bisa),**

****

f(x) = 1 - 2.66 x + 3.75 x2 – 2.987 x3 + 1.217 x4 - 0.195 x5

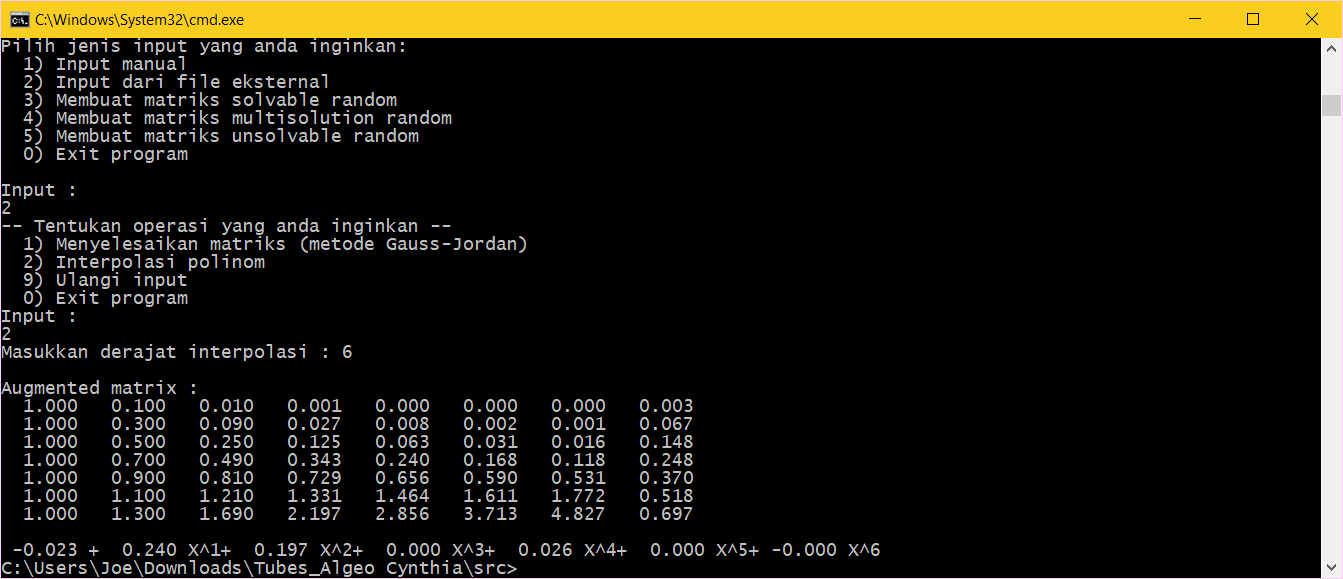
Untuk n = 12, didapatkan :

**Nb: n=12 -> 7 (-2,-1.66,-1.33,-1,-0.66,-0.33 tidak bisa)**



f(x) = 1 - 2.999 x + 5.532 x2 – 6.428 x3 + 4.317 x4 - 1.512 x5 + 0.213 x6

1. Soal g)



Dari persamaan polynomial diatas, nilai f(x) bisa didekati sehingga:

f(x) = -0.022976562500000

+ 0.240000000000002 X

+ 0.197395833333324 X2

+ 0.000000000000019 X3

+ 0.026041666666648 X4

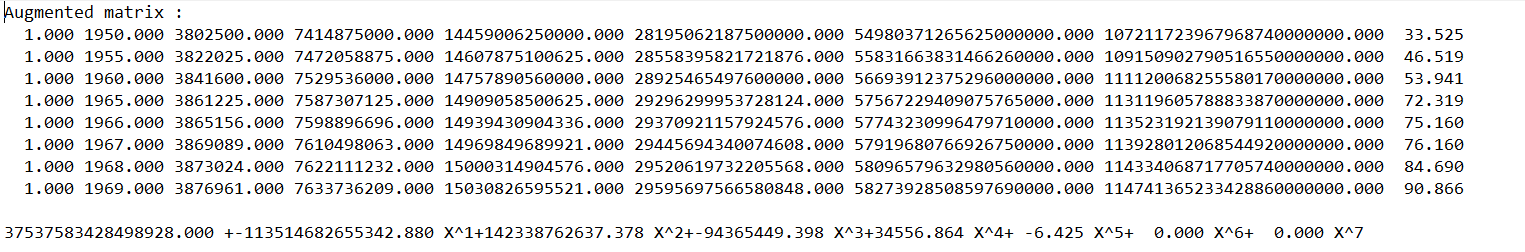
+ 0.000000000000007 X5

- 0.000000000000001 X6

|  |  |
| --- | --- |
| x | f(x) |
| 0.2 | 0.032960938 |
| 0.55 | 0.171118652 |
| 0.85 | 0.33723584 |
| 1.28 | 0.677541837 |

1. Soal h)

Dengan menggunakan X = tahun, dan Y = Harga( $ juta)



f(x) = 37537583428498928.000000000000000

- 113514682655342.880000000000000 X

+ 142338762637.378330000000000 X2

- 94365449.398418200000000 X3

+ 34556.864376035150000 X4

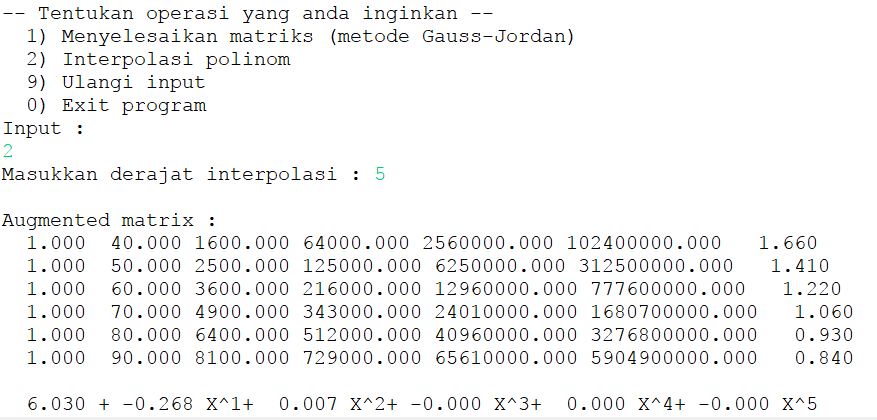
- 6.424597200754641 X5

+ 0.000387140093764 X6

+ 0.000000023130007 X7

1. Soal i)

Dengan x = suhu (T) dan y = v (10-5 ft2/detik), didapatkan:



f(X) = 6.030000000000042

|  |  |
| --- | --- |
| Suhu | Viskositas |
| 62 | 1.185904975 |
| 75 | 1.031296635 |
| 88 | 0.854492961 |

- 0.267816666666671 X

+ 0.006737500000000 X2

- 0.000091666666667 X3

+ 0.000000625000000 X4

- 0.000000001666667 X5

# Bab V: Kesimpulan

1. Kesimpulan

Penyelesaian SPL dan Interpolasi fungsi dapat dilakukan dengan menggunakan matriks dan metode *Gauss* dan *Gauss-Jordan*. Proses penyelesaian dapat diotomasi dengan menggunakan program komputer.

1. Hasil yang dicapai

Penulis & tim telah berhasil membuat program untuk memecahkan SPL dan interpolasi fungsi. Program telah melampaui standar capaian dan spek yang ditentukan.

1. Saran

Untuk orang yang menjadikan laporan ini referensi, pengaturan desimal *floating-point* dan strategi *pivoting* sangat mempengaruhi tingkat akurasi hasil akhir. Dalam menampilkan solusi SPL yang memiliki parametris, bisa dicoba menggunakan algoritma lain yang lebih general. Perhatikan pula standar code yang baik :> .

# Daftar Pustaka

http://chronicles.blog.ryanrampersad.com/2011/03/text-based-menu-in-java/

<http://arifhidayat659.blogspot.sg/2014/04/metode-eliminasi-gauss-dan-gauss-jordan.html>

<https://penma2b.id/2017/04/04/eliminasi-gauss-jordan-spl-3-variabel/>

Munir,Rinaldi*.*2011*.Aplikasi Aljabar Lanjar pada Metode Numerik*

Stewart, et.ol.2007.*College Algebra: Matrices and Determinants*

Treil, Sergei.2009.*Linear Algebra Done Wrong*