**LAPORAN TUGAS BESAR**

**IF2123-01 Dasar Pemrograman**

**Semester I T. P. 2023/2024**

***Sistem Persamaan Linier, Determinan, dan Aplikasinya***

Disusun oleh:

**AmogusSussySus**

| Zachary Samuel Tobing | 13522016 |
| --- | --- |
| Akbar Al Fattah | 13522036 |
| Devinzen | 13522064 |



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2023**

**Bab 1 : Deskripsi Masalah**

**I. Abstraksi**

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 = 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).

Di dalam Tugas Besar 1 ini, Anda diminta membuat satu atau lebih *library* aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, gunakan *library* tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi. Penjelasan tentang interpolasi dan regresi adalah seperti di bawah ini.

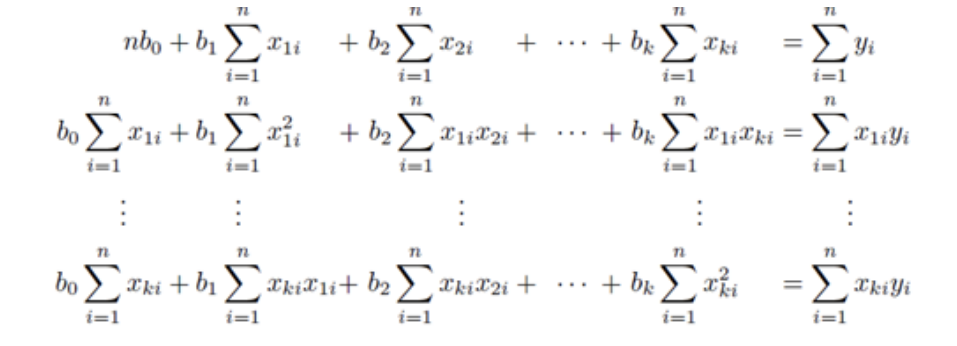
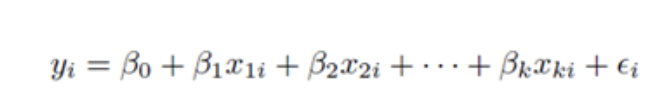
**II. Interpolasi Polinomial**

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x0, y0), (x1, y1), ..., (xn, yn). Tentukan polinom pn(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga yi = pn(xi) untuk i = 0, 1, 2, …, n. Setelah polinom interpolasi pn(x) ditemukan, pn(x) dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x0, xn].

Solusi sistem persamaan lanjar ini, yaitu nilai a0, a1, …, an, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk p2(x) = a0 + a1x + a2.

**III. Regresi Linier Berganda**

Regresi Linier (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Untuk mendapatkan nilai dari setiap βi dapat digunakan *Normal Estimation Equation for Multiple Linear Regression*. Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

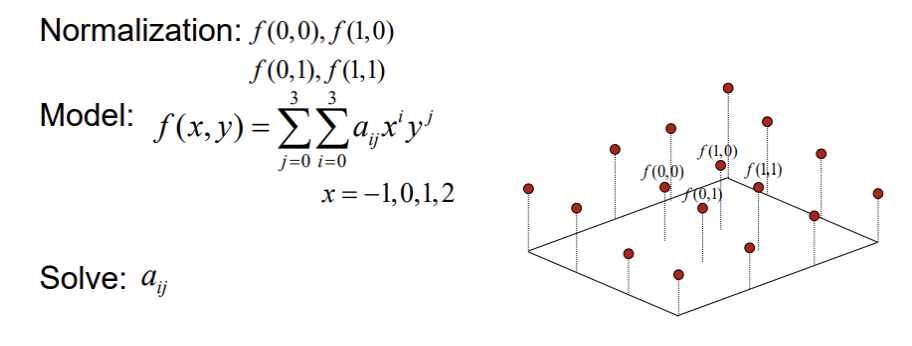
****

**IV. 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 linier. 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.

Perlu diketahui bahwa elemen pada matriks X adalah nilai dari setiap komponen koefisien aij yang diperoleh dari persamaan fungsi maupun persamaan turunan yang telah dijelaskan sebelumnya. Sebagai contoh, elemen matriks X pada baris 8 kolom ke 2 adalah koefisien dari pada ekspansi sigma untuk (1, 1) sehingga diperoleh nilai konstanta , sesuai dengan isi matriks X.

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. Tugas Anda pada studi kasus ini adalah membangun persamaan f(x, y) yang akan digunakan untuk melakukan interpolasi berdasarkan nilai f(a, b) dari masukan matriks 4 x 4. Nilai masukan a dan b berada dalam rentang [0, 1]. Nilai yang akan diinterpolasi dan turunan berarah disekitarnya dapat diilustrasikan pada titik berwarna merah pada gambar di bawah.



**Bab 2 : Teori Singkat**

**2.1 Metode Eliminasi Gauss**

Metode Eliminasi Gauss adalah metode untuk operasi nilai-nilai dalam matriks, untuk membuat matriks lebih sederhana lagi, dikembangkan dari metode eliminasi, dengan cara menghilangkan atau mengurangi jumlah variabel, untuk mendapatkan nilai variabel bebas. Metode ini digunakan untuk menyelesaikan Sistem Persamaan Linier.

**2.1.1 Matriks Eselon (*Row Echelon Form*)**

Matriks eselon adalah matriks yang memiliki 1 utama (*leading one*) pada setiap baris, kecuali baris yang seluruhnya nol. Matriks eselon memiliki sifat sebagai berikut:

* Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama).
* Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
* Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.
* Memiliki semua nilai nol di bawah 1 utama.

**2.1.2 Matriks Eselon Tereduksi (*Reduced Row Echelon*)**

Matriks eselon tereduksi adalah matriks eselon yang memiliki sifat tambahan yaitu memiliki semua nilai nol di bawah dan di atas 1 utama sehingga memiliki sifat sebagai berikut:

* Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama).
* Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
* Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.
* Memiliki semua nilai nol di bawah dan di atas 1 utama.

Metode Eliminasi Gauss terdiri dari 3 langkah yaitu:

1. Menyatakan Sistem Persamaan Linier dalam bentuk *Augmented*

Bentuk *augmented* adalah matriks dengan semua koefisien variabel bebas ditulis dalam bentuk matriks dan konstanta dari Sistem Persamaan Linier ditulis pada bagian kanan matriks koefisien variabel bebas.

1. Melakukan Operasi Baris Elementer

Operasi Baris Elementer (OBE) adalah suatu operasi pada sebuah baris pada matriks yang terdiri dari langkah-langkah berikut:

1. Kalikan sebuah baris dengan konstanta tidak nol.
2. Pertukarkan dua buah baris.
3. Tambahkan sebuah baris dengan kelipatan baris lainnya

Operasi Baris Elementer dilakukan dengan tujuan membentuk Matriks Eselon sehingga Operasi Baris Elementer dilakukan sehingga setiap baris memiliki 1 utama dan baris lainnya memiliki 0 pada sisi kiri di bawah 1 utama baris sebelumnya. Proses ini dilakukan sehingga menjadi Matriks Eselon.

**2.2 Metode Eliminasi Gauss Jordan**

Metode Eliminasi Gauss Jordan merupakan lanjutan Metode Eliminasi Gauss dengan tujuan membentuk Matriks Eselon Tereduksi (*Reduced Row Echelon*), sebuah matriks dengan 1 utama dan mengandung semua nilai nol di bawah dan di atas setiap 1 utama pada setiap baris. Metode Eliminasi Gauss Jordan dilakukan dengan langkah-langkah sebagai berikut:

1. Fase maju (*Forward Phase*)

Fase maju yang dimaksud adalah Metode Eliminasi Gauss yang sudah dijelaskan sebelumnya untuk membentuk Matriks Eselon dengan semua nilai nol di bawah 1 utama.

1. Fase Mundur (*Backward Phase*)

Fase Mundur adalah lanjutan fase maju yang bertujuan untuk membentuk Matriks Eselon Tereduksi dari Matriks Eselon yang sudah dibentuk Fase Maju. Fase Mundur dilakukan dengan melakukan Operasi Baris Elementer lagi, dengan tujuan membuat setiap nilai nol di atas 1 utama dari kolom teratas.

**2.3 Determinan**

Determinan matriks merupakan selisih antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder. Determinan dapat ditentukan dengan beberapa cara tergantung matriksnya sebagai berikut:

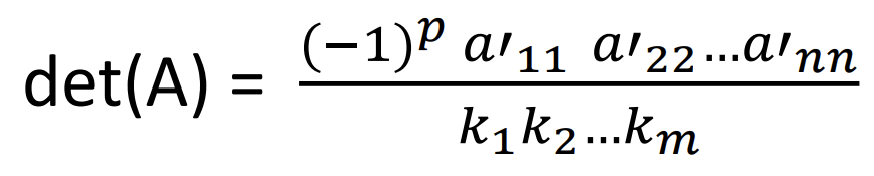
* Matriks Segitiga Bawah dan Matriks Segitiga Atas

Matriks Segitiga Bawah (*Lower Triangular*) adalah matriks yang semua elemen di bawah diagonal utama adalah nol. Matriks Segitiga Atas (*Upper Triangular*) adalah matriks yang semua elemen diatas diagonal utama adalah nol. Matriks Segitiga akan memiliki determinan berupa perkalian elemen pada diagonal utama.

* Reduksi Baris

Reduksi Baris adalah metode dengan menggunakan Operasi Baris Elementer pada matriks dengan tujuan membentuk Matriks Segitiga Bawah. Setiap operasi pada Operasi Baris Elementer berpengaruh pada Determinan akhir sebagai berikut:

* Setiap operasi pertukaran baris menambah *p* sebanyak satu.
* Perkalian baris matriks berupa *k.*
* Lanjutkan dengan metode mencari Determinan Matriks Segitiga.



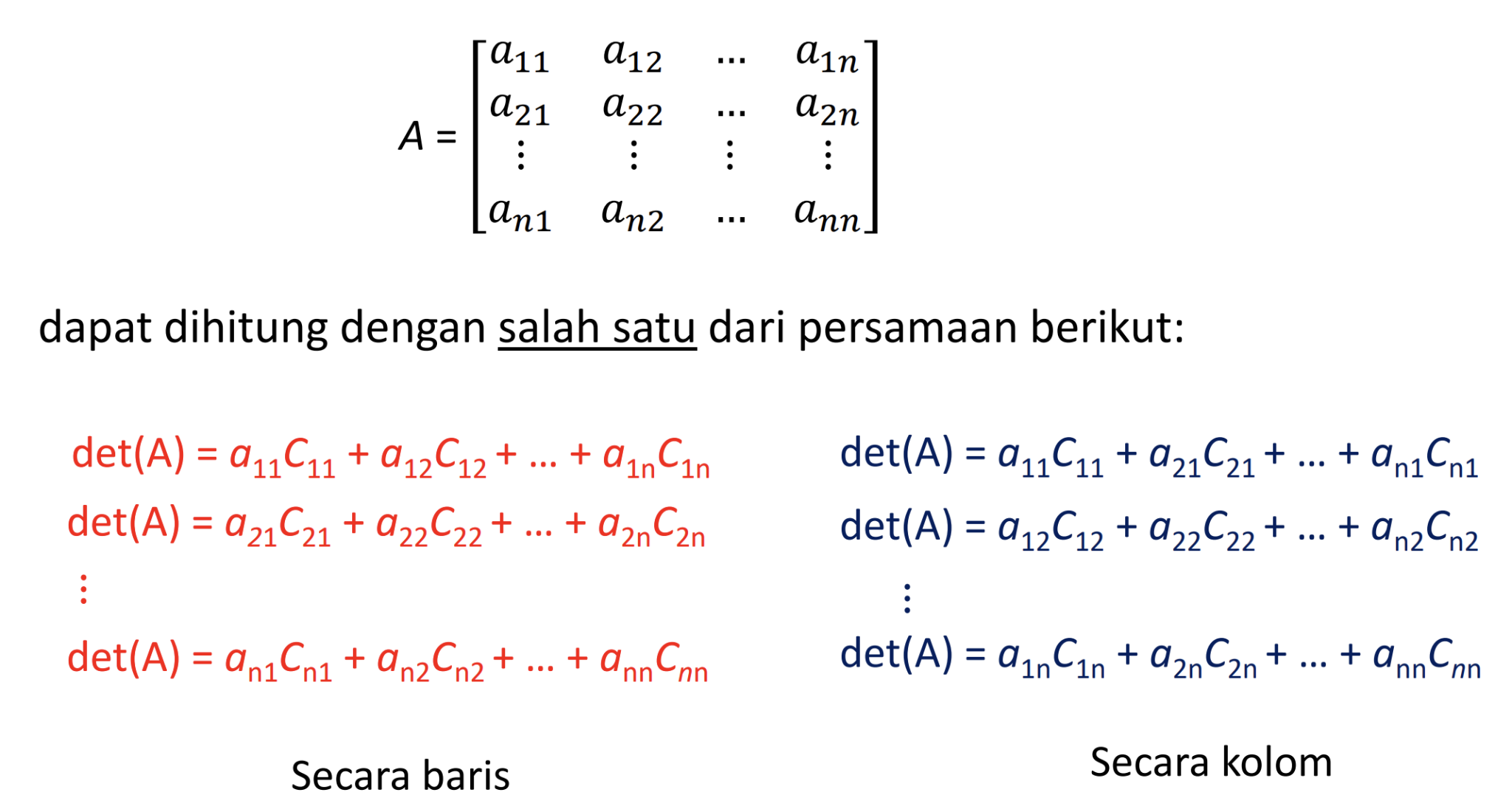
Beberapa teorema tentang Determinan yaitu:

* Jika A mengandung sebuah baris nol atau kolom nol, maka det(A) = 0.
* Jika adalah matriks transpose dari A, maka det() = det(A).
* Jika A = BC maka det(A) = det(B)det(C)
* Sebuah matriks hanya mempunyai balikan jika dan hanya jika det(A) tidak sama dengan 0.
* det() = 1/(det(A))

**2.4 Ekspansi Kofaktor**

Ekspansi Kofaktor adalah metode mencari determinan matriks dengan minor dan kofaktor. Minor () adalah determinan upa-matriks (*submatrix*) yang elemen-elemennya tidak berada pada baris i dan kolom j. Kofaktor () adalah perkalian minor dengan -1 pangkat i+j.

Setelah mencari Minor dan Kofaktor, Determinan dapat dihitung dengan jumlah perkalian kofaktor dengan nilai pada indeks bersangkutan. Penjumlahan dapat dilakukan pada salah satu baris atau kolom.

`

**2.5 Matriks Invers**

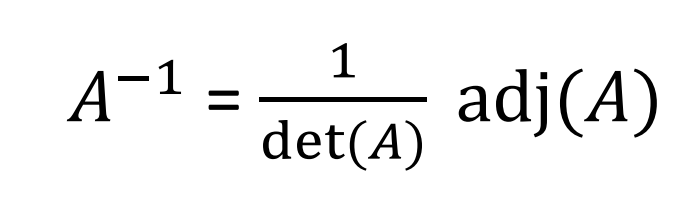
Invers matriks adalah kebalikan (invers) dari sebuah matriks yang apabila matriks tersebut dikalikan dengan inversnya, akan menjadi matriks identitas. Invers matriks dilambangkan dengan . Suatu matriks dikatakan memiliki invers jika determinan dari matriks tersebut tidak sama dengan nol.

Invers Matriks dapat ditentukan dengan cara sebagai berikut:

* Eliminasi Gauss Jordan

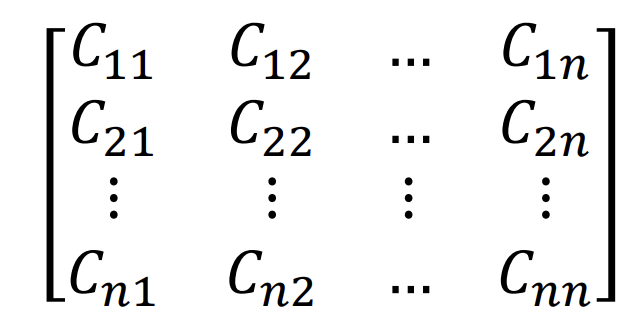
Dengan menggunakan sifat perkalian Matriks dan Matriks Invers menjadi Matriks Identitas, maka Matriks ditulis dalam bentuk *augmented* dengan Matriks pada sisi kiri dan Matriks Identitas pada sisi kanan Matriks. Kemudian, lakukan OBE (Eliminasi Gauss Jordan) untuk memperoleh Matriks Identitas pada sisi kiri dan akan diperoleh Matriks Invers pada sisi kanan.

* Matriks Adjoin dan Determinan



**2.6 Matriks Kofaktor**

Matriks Kofaktor adalah matriks yang mengandung Kofaktor pada setiap indeks bersangkutan.



**2.7 Adjoin**

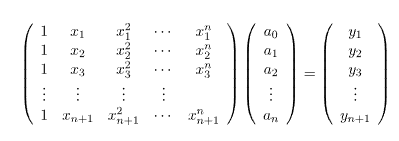
Adjoin adalah *transpose* dari Matriks Kofaktor.

**2.8 Kaidah Cramer**

Jika Ax = b adalah Sistem Persamaan Linier yang terdiri dari n persamaan linier dengan n peubah (*variable*) sedemikian sehingga det(A) tidak sama dengan 0, maka SPL tersebut memiliki solusi yang unik yaitu:

Dengan adalah matriks dengan kolom i diubah dengan b.

**2.9 Interpolasi Polinomial**

Interpolasi Polinomial adalah metode untuk mencari fungsi polinomial berderajat n yang melewati n+1 titik. Untuk mendapatkan fungsi polinomial, kita harus menyelesaikan SPL berikut

Kita perlu mencari solusi, yaitu matriks a. Setelah kita menemukan solusinya, nilai a0, a1, … ,an adalah nilai koefisien dari fungsi polinomial yang dicari. Maka akan didapat fungsi polinomial

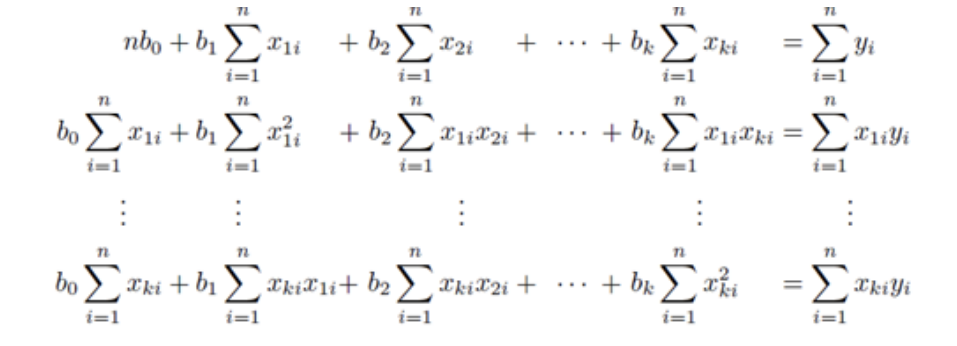


**2.10 Regresi Linier Berganda**

Analisis regresi digunakan untuk mengukur seberapa besar pengaruh antara variabel bebas dan variabel terikat. Apabila hanya terdapat satu variabel bebas dan satu variabel terikat, maka regresi tersebut dinamakan Regresi Linier Sederhana (Juliandi, Irfan, & Manurung, 2014). Sebaliknya, apabila terdapat lebih dari satu variabel bebas atau variabel terikat, maka disebut Regresi Linier Berganda. Regresi Linier Berganda merupakan model regresi yang melibatkan lebih dari satu variabel independen. Analisis Regresi Linier Berganda dilakukan untuk mengetahui arah dan seberapa besar pengaruh variabel independen terhadap variabel dependen (Ghozali, 2018).

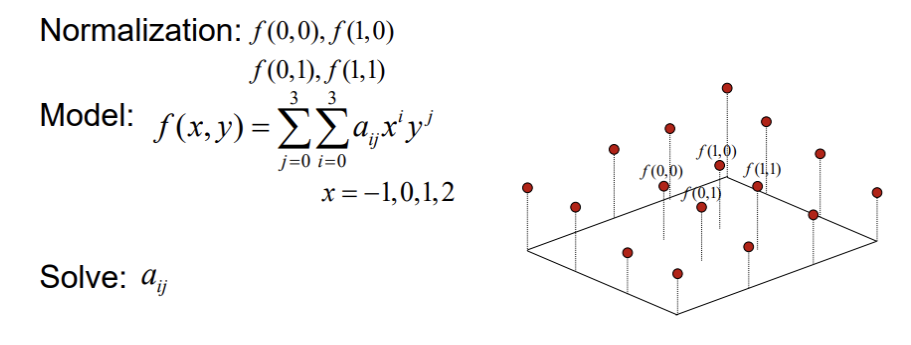
Persamaan Regresi Linier adalah:

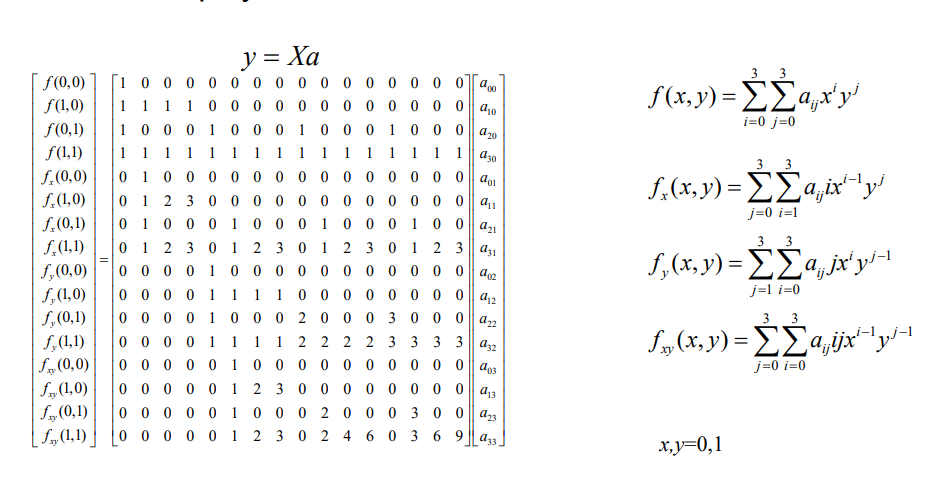
Untuk mendapatkan nilai dari setiap βi dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:

****

dengan n adalah banyak sampel, k adalah banyak variabel peubah, dan bi adalah koefisien persamaan regresi.

**2.11 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 linier. 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.

Selain melibatkan model dasar, juga digunakan model turunan berarah dari kedua sumbu, baik terhadap sumbu x, sumbu y, mapun keduanya. 

Matriks X dikonstruksi dari persamaan yang ada di sebelah kanan. Matriks y berasal dari input user. Kita perlu untuk mencari solusi, yang disimpan di matriks a. Untuk itu, kita perlu menyelesaikan SPL di atas dengan metode invers.

Setelah kita berhasil menentukan solusi untuk matriks a, matriks a diubah ke matriks 4x4 dengan i menjadi kolom dan j menjadi baris. Kemudian, titik yang ingin ditaksir nilainya diinput dengan persamaan f(x,y) (persamaan yang paling atas di gambar ini).

**Bab 3 : Implementasi Pustaka dan Program**

**3.1 Default Package**

**3.1.1 Class Main**

Main digunakan sebagai pusat pengendali berjalannya program. Main terdiri dari pilihan *matrix* m, menu dan submenu. Menu diterima sebagai pilihan antara 7 opsi yang dapat dilakukan pada program.

Submenu digunakan sebagai pilihan antara 2 opsi untuk memasukkan SPL melalui *keyboard* atau *file* yang berisi matriks *augmented*, kemudian 4 opsi dalam pilihan menu 1 (Sistem Persamaan Linier) yaitu semua jenis metode yang dapat digunakan, atau pilihan antara 2 opsi dalam pilihan menu 2 (Determinan), yaitu antara metode Reduksi Baris atau Ekspansi Kofaktor. Program akan langsung berakhir jika memilih angka diluar jangkauan opsi.

Program akan berlangsung sesuai opsi menu dan submenu yang dimasukkan dalam program, dengan setiap ketentuan dan prosesnya masing-masing. Program juga akan menampilkan pilihan menu lagi sesudah menyelesaikan salah satu proses program yang dipilih. Program akan berhenti apabila memilih menu 7 (Keluar) atau memasukkan opsi diluar pilihan menu dan submenu.

**3.2 Package LinearAlgebra**

**3.2.1 Class Matrix**

Matrix digunakan sebagai pernyataan bentuk data matriks dan operasi dasar yang dibutuhkan dalam proses matriks. Matrix memiliki tipe data dengan atribut sebagai berikut:

1. Mat (array of double)

Menyimpan isi matriks.

1. row (integer)

Menyatakan berapa banyak baris pada matriks tersebut.

1. col (integer)

Menyatakan berapa banyak kolom pada matriks tersebut.

Method yang terdapat pada class ini adalah:

1. inputMatrix

Mengisi Matrix secara *traversal* berdasarkan masukan baris dan kolom.

1. printMatrix

Mengeluarkan/mencetak isi Matrix secara *traversal* setiap baris dan kolom.

1. swapRow

Menukar baris Matrix yang diinginkan.

1. Transpose

Membuat matriks transpos (baris dan kolom ditukar posisinya).

1. PlusMinus

Melakukan penjumlahan atau pengurangan matriks sesuai kebutuhan.

1. Multiply

Melakukan perkalian matriks.

1. AllRowZero

Menentukan apakah baris yang sedang diperiksa di matriks semuanya berisi angka 0

1. Augment

Menyatukan matriks ax dan matriks b menjadi matriks *augmented*

1. toColMatrix

Mengubah matriks berukuran nxm menjadi matriks kolom.

**3.2.2 Class Determinant**

Class Determinant digunakan untuk menghitung determinan dari sebuah matriks dengan masukan sebuah matriks. Determinant memiliki 2 method, yaitu ReduksiBaris dan EkspansiKofaktor.

1. ReduksiBaris

ReduksiBaris menghitung determinan dengan langkah berikut:

1. Ubah matriks masukan menjadi Matriks Segitiga Atas, dengan tujuan menghitung determinan dengan mengalikan semua bilangan pada diagonal utama.
2. Hitung berapa kali pertukaran baris terjadi
3. Hitung determinan dengan mengalikan semua elemen di diagonal utama dan tentukan tanda (positif atau negatif) berdasarkan ganjil genapnya banyaknya pertukaran baris. (lihat teori dasar determinan bagian reduksi baris untuk keterangan lebih lanjut)

2. EkspansiKofaktor

EkspansiKofaktor menghitung determinan dengan langkah berikut:

1. Buat kasus bagi matriks persegi 1x1.
2. Kemudian, ekspansi kofaktor dilakukan dengan metode rekursi dari matriks masukan hingga basis matriks persegi 1x1 tersebut. Determinan kemudian dihitung dengan metode ekspansi kofaktor biasa.

**3.2.3 Class EquationSystem**

EquationSystem digunakan sebagai wadah untuk program dari submenu Sistem Persamaan Linier, yang tujuannya mencari solusi dari Sistem Persamaan Linier.

Berikut adalah method yang terdapat pada class EquationSystem

1. CramerRule

CramerRule berfungsi untuk menyelesaikan SPL dengan metode Cramer's Rule, yang dilakukan dengan langkah sebagai berikut:

1. Menentukan determinan awal. Jika determinan sama dengan nol, maka tidak ada solusinya atau berupa parametrik.
2. Metode Cramer kemudian membuat 3 matriks berbeda yaitu *clone* yang mula-mula sama dengan matriks masukanuntuk matriks dengan salah satu kolomnya diubah menjadi kolom *b*, matriks *reset* yang sama dengan matriks masukan untuk mengubah kembali matriks *clone* yang diproses menjadi matriks semula, dan matriks *konstan* untuk menyimpan nilai kolom *b* untuk memisahkan dan memudahkan proses.
3. Setiap solusi persamaan linier kemudian dilakukan dengan rumus Cramer dengan determinan matriks yang telah diubah kolomnya (*Clone*) dibagi dengan determinan matriks awal (m) dengan matriks *clone* diubah kembali menjadi *reset* dan mengulanginya untuk setiap kolom lainnya.
4. Dibuat sebuah SPLTuplesuntuk menyimpan semua informasi yang dibutuhkan, yaitu matriksnya, solusinya, apakah ada nilai solusinya, dan apakah solusinya parametrik.

2. Gauss

Method Gauss berfungsi untuk menyelesaikan SPL dengan Eliminasi Gauss (Gauss), dan dilakukan dengan langkah sebagai berikut:

1. Memanggil metode membuat matriks menjadi matriks eselon.
2. Mengecek baris. Jika semua elemen baris terakhir nol, maka solusinya berupa parametrik (sifat). Jika ada baris lain yang semua elemennya nol tapi nilai b nya tidak nol, maka tidak ada solusinya (sifat).
3. Informasi mengenai matriksnya, solusinya, apakah parametrik dan apakah ada solusinya dimasukkan dalam tipe data SPLTuples.

3. ParametricGauss

Method ParametricGauss berfungsi untuk mencetak solusi parametrik yang didapat dari eliminasi Gauss dengan prekondisi ada solusinya dan solusinya parametrik dengan langkah berikut:

1. Memanggil metode untuk merubah menjadi matriks.
2. Mencari nilai tidak nol dari baris bawah kanan untuk dijadikan parameter.
3. Karena sudah menjadi matriks eselon tereduksi, maka ada *leading one* setiap baris sehingga setiap baris yang memiliki parameter yang sama dengan baris di bawahnya yang sudah diselesaikan, cukup dikurangi dengan nilai parameternya di barisnya.
4. Mengubah matriks untuk menghilangkan kolom yang semua elemennya nol dan menggeser kolom lain.

4. GaussJordan

Eliminasi Gauss Jordan (GaussJordan) bertujuan untuk membuat sebuah *tuple* SPL yang berisi matriks awal, solusi, apakah parametrik, dan apakah ada solusinya.

1. Memanggil metode membuat matriks menjadi matriks eselon tereduksi.
2. Mengecek baris. Jika semua elemen baris terakhir nol, maka solusinya berupa parametrik (sifat). Jika ada baris lain yang semua elemennya nol tapi nilai b nya tidak nol, maka tidak ada solusinya (sifat).
3. Informasi mengenai matriksnya, solusinya, apakah parametrik dan apakah ada solusinya dimasukkan dalam SPLTuples.

5. ParametricGaussJordan

ParametricGaussJordan berfungsi untuk mengoutputkan solusi parametrik dari eliminasi Gauss-Jordan. Langkah-langkahnya adalah

1. Memanggil metode untuk merubah menjadi matriks eselon tereduksi.
2. Mencari nilai tidak nol dari baris bawah kanan untuk dijadikan parameter.
3. Karena sudah menjadi matriks eselon, maka ada *leading one* setiap baris sehingga setiap baris yang memiliki parameter yang sama dengan baris di bawahnya yang sudah diselesaikan, cukup dikurangi dengan nilai parameternya di barisnya.
4. Mengubah matriks untuk menghilangkan kolom yang semua elemennya nol dan menggeser kolom lain.

6. Inverse

Method Inverse bertujuan untuk menyelesaikan SPL dengan metode matriks balikan/invers.

1. Memanggil metode membuat matriks menjadi matriks invers.
2. Mengecek baris. Akan tetapi, berbeda dengan eliminasi Gauss dan eliminasi Gauss Jordan, untuk matriks invers tidak dapat ditentukan apakah tidak memiliki solusi atau bersifat parametrik.
3. Informasi mengenai matriksnya, solusinya, apakah parametrik dan apakah ada solusinya dimasukkan dalam SPLTuples.

**3.2.4 Class Inverse**

Class Inverse digunakan untuk mencari matriks invers dari sebuah matriks yang dilakukan dengan 2 metode, Eliminasi Gauss Jordan dan Matriks Adjoin, keduanya dengan asumsi bahwa matriks yang dimasukkan adalah matriks persegi dan ada inversnya.

1. Eliminasi Gauss Jordan (*GaussJordan*) = membuat matriks eselon tereduksi (*doubleInv*) dari matriks *augmented* yang terdiri dari nilai ax dan matriks identitas dengan langkah berikut:

* Membuat matriks dengan matriks ax awal dan disambungkan dengan matriks identitas dengan panjang kolom yang sama pada bagian akhir.
* Melakukan eliminasi Gauss Jordan dengan fungsi matriks eselon tereduksi (*RREF*) pada kelas *Echelon*.

1. Invers (*GetInverseGJ*) = mendapatkan invers matriks dari jumlah kolom terakhir matriks Gauss Jordan (*RREF*) yang sama dengan jumlah kolom matriks awal, sesuai hukumnya.
2. Matriks Adjoin (*Adjoin*) = mendapatkan invers matriks dari mencari matriks kofaktor dan mencari *transpose* sehingga mendapatkan matriks adjoin dari matriks awal dengan langkah berikut:

* Mencari determinan dari matriks (*EkspansiKofaktor*) pada kelas *Determinant* dan menangani matriks 1x1. Jika determinan sama dengan nol, maka tidak ada inversnya.
* Mencari determinan dari setiap kofaktor dengan menangani tanda plus atau minus untuk setiap kolom dan baris.
* Membagi matriks kofaktor dengan determinan yang diperoleh.
* Membuat matriks *transpose* dari matriks kofaktor.

**3.2.5 Class Echelon**

Echelon digunakan untuk mengubah sebuah matriks menjadi matriks eselon atau matriks eselon baris tereduksi.

Echelon memilki 2 method, yaitu:

1. REF

Mengubah sebuah matriks menjadi matriks eselon secara *traversal* dengan langkah berikut:

* Mengecek bilangan pertama tidak nol yang muncul di sebuah baris sesuai kolom iterasi *traversal* dan menukarnya dengan baris yang memiliki bilangan pertama nol dengan kolom yang sama (jika ada).
* Kemudian, setiap baris akan dikurangi dengan mengurangi dirinya dengan baris yang sudah memiliki *leading one* sesuai perbandingan rasio (*Ratio*) antara bilangan pertama tidak nol pada baris tersebut dengan nilai pada baris di atasnya.
* Terakhir, *divider* digunakan untuk membagi setiap nilai pada baris dengan bilangan pertama tidak nol pada setiap baris, dengan tujuan membuat *leading one* pada setiap baris.

1. RREF

Mengubah sebuah matriks menjadi matriks eselon tereduksi dengan langkah berikut:

* Memanggil method REF untuk membuat matriks tersebut menjadi matriks eselon.
* Karena sudah matriks eselon, maka semua baris ada *leading one*, sehingga rasio (*Mul*) hanya nilai setelah *leading one* setiap baris.
* Setiap kolom akan dikurangi dengan perkalian baris setelahnya dengan rasio(*mul*), sehingga semua nilai di atas dan di bawah *leading one* menjadi nol.

**3.3 Package SPLTuples**

**3.3.1 Class SPLTuples**

SPLTuples adalah sebuah tipe data *tuple* buatan untuk membantu menyelesaikan Sistem Persamaan Linier (SPL). Atribut SPLTuples terdiri dari atas:

1. Mat (tipe data Matrix)

Berisi matriks yang sudah diolah dari method yang ada di class EquationSystem.

1. Solution (array of double)

Berisi data solusi SPL jika dan hanya jika SPL memiliki solusi unik. Jika tidak (solusi tidak ada atau solusinya parametrik), isi semua elemen Solution adalah NaN.

1. isSolvable

isSolvable menyatakan apakah SPL dapat diselesaikan atau tidak.

1. isParametric

isParametric menyatakan apakah SPL memiliki solusi parametrik atau solusi unik.

**3.4 Package PITuple**

**3.4.1 Class PITuple**

PITuple adalah sebuah tipe data *tuple* buatan untuk membantu penyelesaian masalah Interpolasi Polinomial. Atribut dari PITuple adalah

1. taksir (double)

taksir adalah nilai f(x) yang ingin ditaksir dari titik x dengan fungsi polinomial yang dibentuk dari interpolasi polinomial.

1. xValues (array double)

xValues adalah array yang berisi nilai x dari titik-titik yang diinput sebelum nilai x yang ditaksir.

1. yValues (array of double)

yValues adalah array yang berisi nilai y dari titik-titik yang diinput sebelum nilai x yang ditaksir.

**3.5 Package MulRegTuple**

**3.5.1 Class MulRegTuple**

MulRegTuple adalah sebuah tipe data *tuple* buatan untuk membantu menyelesaikan masalah Regresi Linier Berganda. Atribut yang dimiliki MulRegTuple adalah:

1. variables (integer)

variables adalah banyak variabel peubah yang akan diregresi (banyak x yang dianggap berbeda).

1. samples (integer)

samples adalah banyak sampel data yang diinput dan diregresikan.

1. inputMat (Matrix)

inputMat adalah matriks yang didapatkan dari input pengguna.

1. NEE (Matrix)

NEE adalah matriks yang mengandung *Normal Estimation Equation for Multiple Linear Regression.*

1. xtest (array of double)

xtest adalah nilai sampel baru yang akan ditaksir.

**3.6 Package BicubicSITuple**

**3.6.1 Class BicubicSITuple**

BicubicSITuple adalah sebuah tipe data *tuple* buatan untuk membantu menyelesaikan masalah *Bicubic Spline Interpolation*, yaitu menaksir sebuah nilai berdasarkan titik dan sampel yang sebelumnya telah dimasukkan.

Atribut yang terdapat pada BicubicSITuple adalah

1. Mat (Matrix)

Mat berfungsi untuk menyimpan matriks yang diinput.

1. xTaksir (double)

xTaksir berfungsi untuk menyimpan nilai x dari titik yang ingin ditaksir.

1. yTaksir (double)

yTaksir berfungsi untuk menyimpan nilai y dari titik yang ingin ditaksir.

**3.7 Package Menu**

**3.7.1 Class EnhancedIO**

EnhancedIO digunakan untuk memudahkan format permintaan dan penerimaan kode agar lebih mudah ditulis. Beberapa fungsi di antaranya adalah:

1. Meminta *input* sebuah matriks persegi dari *keyboard* (*InputSquareMatrixKeyboard*) = sebuah prosedur untuk memudahkan masukan matriks persegi dengan ukuran yang dimasukkan, membuat matriks baru, dan memasukkan nilai dengan fungsi *inputMatrix*.
2. Meminta *input* sebuah SPL dari *keyboard* (*InputSPLKeyboard*) = sebuah prosedur untuk masukan SPL dengan banyaknya persamaan, banyaknya peubah (variabel), dan membuat matriks SPL dalam bentuk matriks *augmented.*
3. Meminta *input* sebuah SPL dari *file* (*InputSPLFile*) = sebuah prosedur untuk masukan SPL dari sebuah *file* dengan pertama mencari *file*, kemudian menghitung jumlah kolom dan baris yang ada, membuat matriks baru sesuai jumlahnya, dan *parsing filenya* untuk memasukkan nilai-nilai SPL nya.
4. Meminta *input* sebuah matriks dari *file* (*InputSquareMatrixFile*) = sebuah prosedur untuk masukan matriks dari sebuah *file* dengan pertama mencari *file*, kemudian menghitung jumlah kolom dan baris yang ada, membuat matriks baru sesuai jumlahnya, dan *parsing filenya* untuk memasukkan nilai-nilai matriks nya.
5. Mengeluarkan nilai *double* (*OutputDoublePrecision4*) = mengeluarkan nilai double dengan presisi 4 angka di belakang floating point.
6. Mengeluarkan determinan (*OutputDet*) = mengeluarkan determinan dengan *double precision*.
7. Mengeluarkan solusi SPL (*OutputSPL*) = mengeluarkan solusi SPL sesuai solusinya. Jika ada solusinya, dikeluarkan dengan *double precision* dan *round to even*.
8. Mengeluarkan solusi SPL parametrik (*OutputParametric*) = mengeluarkan solusi SPL jika SPL memiliki solusi parametrik dan dilakukan dengan format parametrik.
9. Mengeluarkan solusi SPL invers matriks (*OutputSPLInverse*) = mengeluarkan solusi SPL jika ada solusinya, dan mengirimkan pesan jika tidak.
10. Mengeluarkan fungsi (*OutputFunction*) = mengeluarkan fungsi dengan semua koefisien variabel sudah dicantumkan. Jika fungsi berupa fungsi regresi, akan menambahkan “ei” (nilai galat) di akhir.
11. Mengeluarkan *path file* dalam *string* (*findFileDir*) = mengeluarkan nama *path file* dengan pertama menerima masukan nama *file* untuk diproses.
12. Menerima *input* dan membuat *tuple* untuk *Bicubic Spline Interpolation* dari *keyboard* (*InputBicubicKeyboard*) = menerima nilai untuk matriks 4x4 (16 nilai) sesuai ketentuan dan nilai yang ingin ditaksir (x dan y) kemudian membuat *tuple* untuk *Bicubic Spline*.
13. Menerima *input* dan membuat *tuple* untuk *Bicubic Spline Interpolation* dari *file* (*InputBicubicFile*) = menerima nilai untuk matriks 4x4 (16 nilai) sesuai ketentuan dari file dengan mencari *file* dan kemudian memasukkan nilai dengan *parsing* serta nilai yang ingin ditaksir (x dan y) kemudian membuat *tuple* untuk *Bicubic Spline*.
14. Menerima *input* dan membuat *tuple* untuk interpolasi polinomial dari *keyboard* (*InputPIKeyboard*) = menerima jumlah titik dan nilai titik sesuai jumlah dan nilai yang ingin ditaksir (x dan y) dan membuat sebuah *tuple* berdasarkan semua informasinya.
15. Menerima *input* dan membuat *tuple* untuk interpolasi polinomial dari *file* (*InputPIFile*) = dilakukan dengan pertama mencari *file,* kemudian menghitung jumlah titik berdasarkan isi *file* dan nilai titik sesuai jumlah titik dari *parsing* isi *file*  yang dimasukkan dalam *array* nilai x dan y, serta nilai yang ingin ditaksir (x dan y) dan membuat sebuah *tuple* berdasarkan semua informasinya.
16. Menerima *input* dan membuat *tuple* untuk regresi linier berganda dari *keyboard* (*InputRegKeyboard*) = dilakukan dengan menerima jumlah variabel peubah, banyaknya sampel, melakukan proses sigma dengan pemanggilan metode sigma, menerima masukan nilai yang ingin ditaksir, dan menaruhnya semua dalam *tuple.*
17. Menerima *input* dan membuat *tuple* untuk regresi linier berganda dari *file* (*InputRegFile*) = dilakukan dengan pertama mencari *file*, membaca jumlah peubah, sampel, dan nilai yang ingin ditaksir dari jumlah yang ada di *file*, melakukan *parsing* untuk menerima elemennya, melakukan proses sigma dengan pemanggilan metode sigma, dan menaruhnya semua dalam *tuple.*
18. Mengeluarkan solusi unik SPL ke dalam file .txt (*OutputSPLUniqueFile*)
19. Mengeluarkan solusi parametrik SPL ke dalam file .txt (*OutputSPLParaFile*)
20. Mengeluarkan keterangan tidak ada solusi SPL ke dalam file.txt (*OutputSPLNoSolutionFile*)
21. Menu untuk memilih ingin mengoutputkan solusi SPL ke file .txt (*OutputSPLFile*)
22. Mengeluarkan determinan suatu matriks ke dalam file .txt (*OutputDetFile*)
23. Mengeluarkan matriks balikan suatu matriks ke dalam file.txt (*OutputInverseMatrixFile*)
24. Mengeluarkan keterangan bahwa matriks tidak punya invers ke dalam file .txt (*OutputNoInverseFile*)
25. Mengeluarkan luaran Bicubic Spline Interpolation ke dalam file.txt (*OutputBSIFile*)
26. Mengeluarkan luaran Interpolasi Polinomial ke dalam file.txt (*OutputPIFile*)
27. Mengeluarkan luaran Regresi Linear Berganda ke dalam file.txt (*OutputMulRegFile*)

**3.8 Package Misc**

**3.8.1 Class Misc**

Misc berisi method-method tambahan yang tidak termasuk kategori package dan class di atas. Method yang ada di Misc adalah

1. Sigma

Melakukan operasi sigma (penjumlahan) sesuai aturan pada Regresi Linier Berganda secara *traversal*.

1. GenerateBicubicSplineMat Membuat matriks untuk *Bicubic Spline* sesuai dengan nilai-nilai f, fx, fy untuk setiap nilai -1,0,1,2 dengan memanggil metode f, fx, fy, dan fxy di bawah ini.
2. f

Menghitung nilai f(x, y) untuk nilai sesuai aturan pada *Bicubic Spline Interpolation*.

1. fx

Menghitung nilai turunan berarah f terhadap sumbu x sesuai aturan pada *Bicubic Spline Interpolation*.

1. fy

Menghitung nilai turunan berarah f terhadap sumbu y sesuai aturan pada *Bicubic Spline Interpolation*.

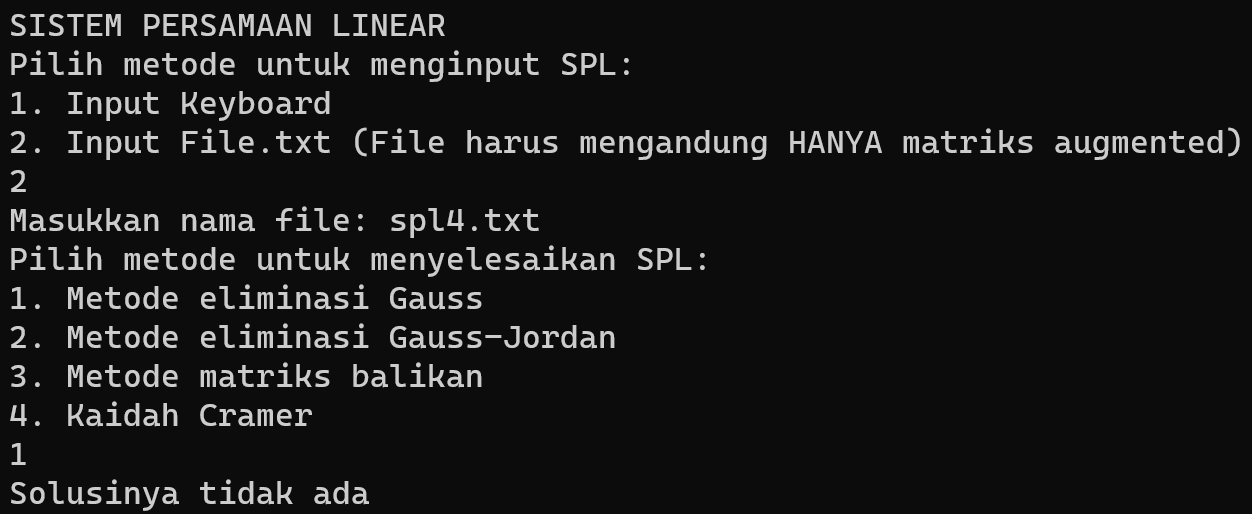
1. fxy

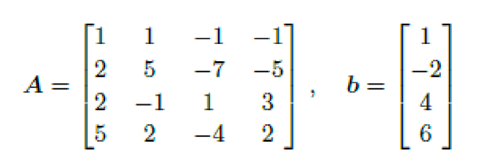
Menghitung nilai turunan berarah f terhadap sumbu x kemudian terhadap sumbu sesuai aturan pada *Bicubic Spline Interpolation*.

**Bab 4 : Eksperimen dan Studi Kasus**

**4.1 Solusi SPL Ax = b**

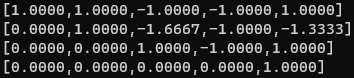
**4.1.a**

****

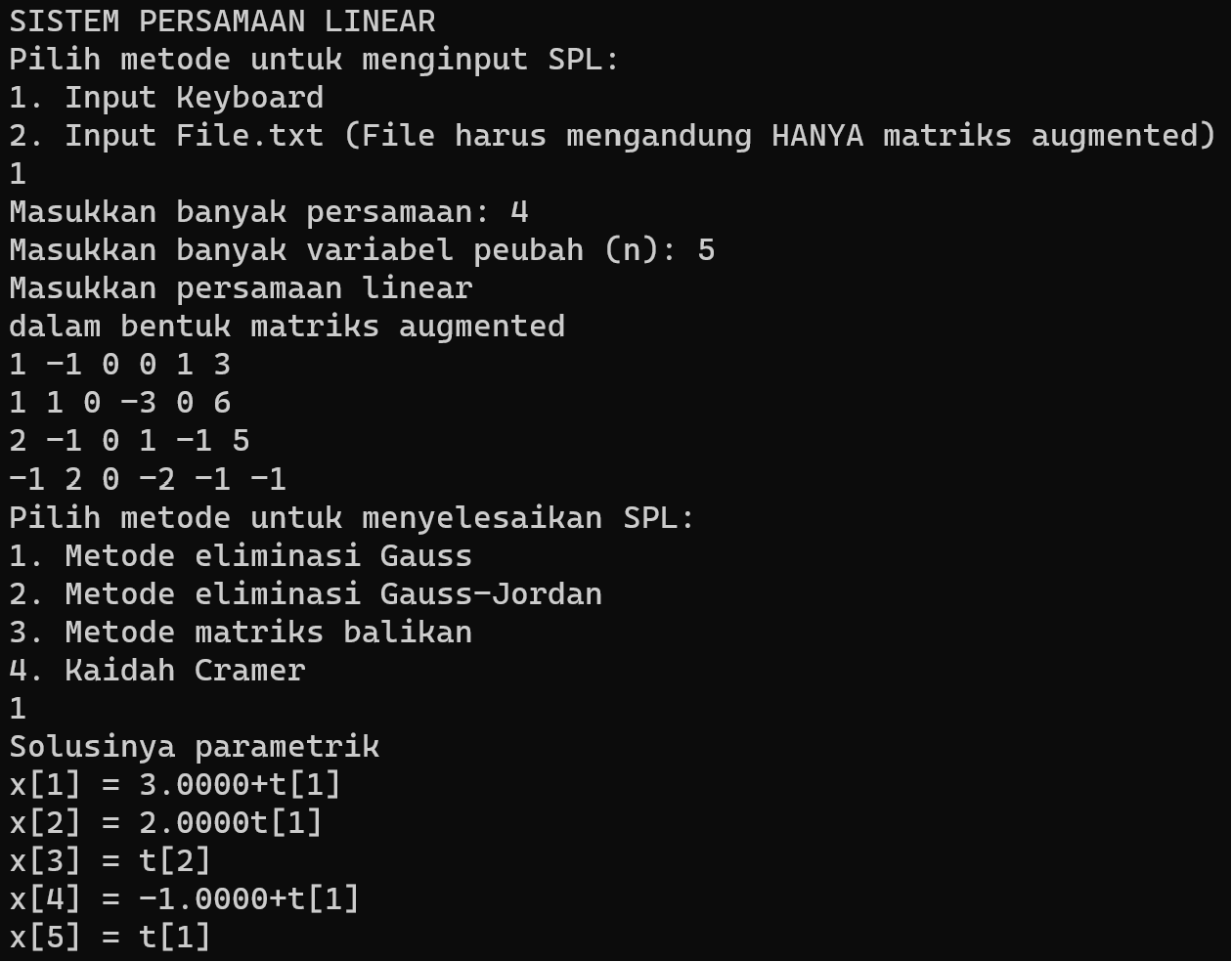
Keterangan: isi dari file spl4.txt adalah SPL di bawah ini dalam bentuk matriks augmented.

Analisis :

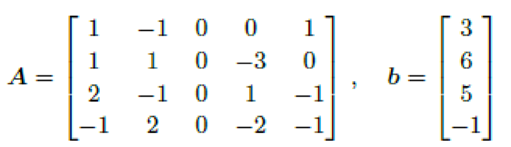
SPL di atas dimasukkan dari *file* spl4.txt(sudah ada di *folder* “test”) dan menggunakan eliminasi Gauss dan ditemukan bahwa solusinya tidak ada, karena hasil SPL yang didapat setelah melakukan eliminasi gauss adalah sebagai berikut.



Baris terbawah matriks hasil SPL tersebut bernilai nol semua dan kolom paling kanan terbawah nilainya ≠ 0, sehingga SPL di atas tidak punya solusi.

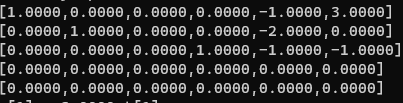
**4.1.b.**

SPL yang diinput dengan keyboard pada contoh eksekusi di atas adalah:

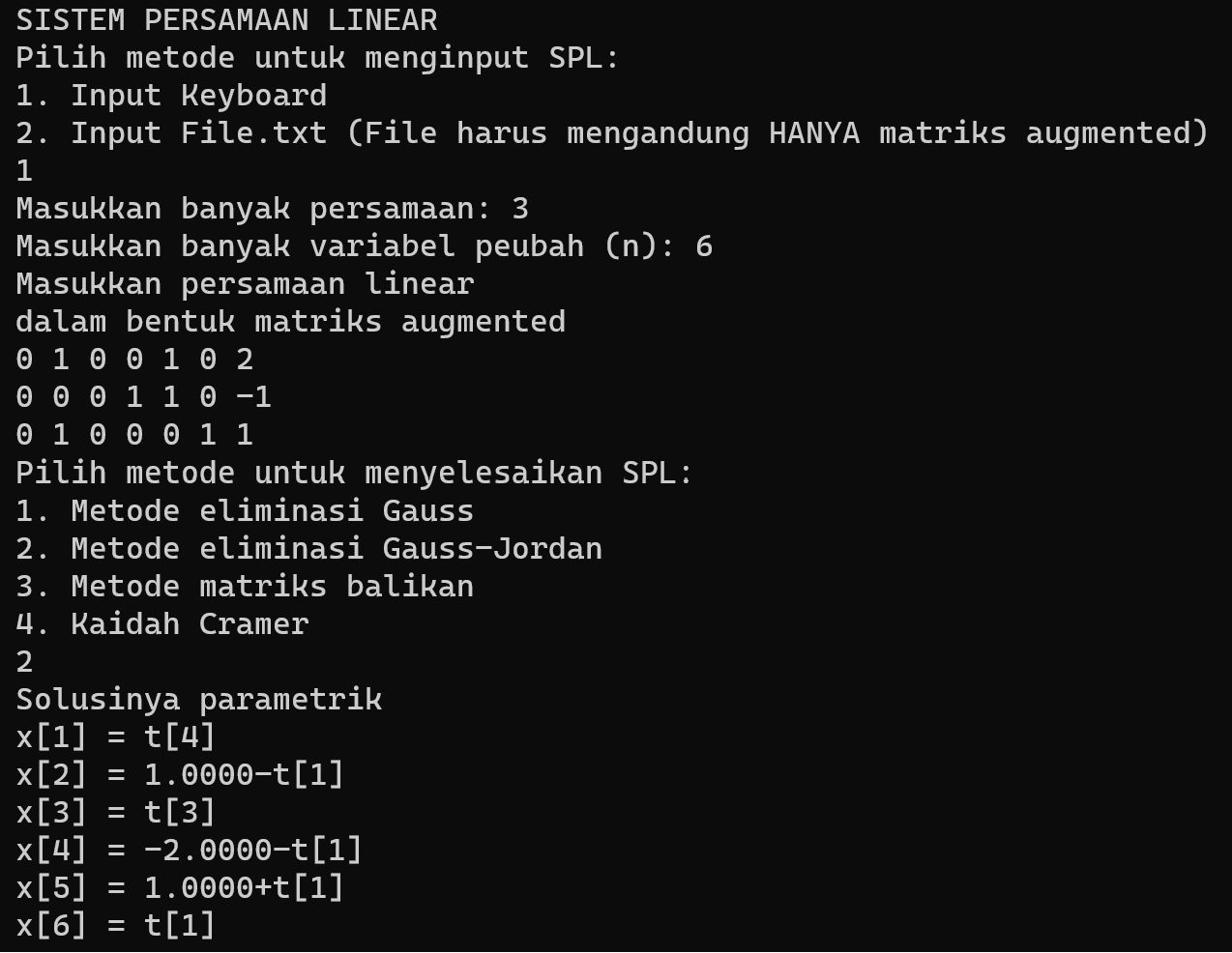


Analisis:

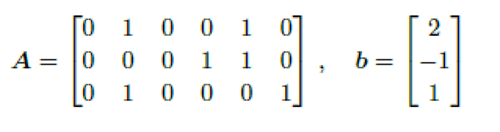
Solusi SPL di atas parametrik karena matriks yang dihasilkan setelah melakukan eliminasi Gauss adalah:



Karena ada baris yang semuanya bernilai 0 dan kolom paling kanan bernilai 0 juga (yang disebabkan oleh banyak persamaan lebih banyak daripada banyak variabel, mengakibatkan barisan paling bawah otomatis bernilai 0 semua termasuk kolom paling kanan), maka SPL ini memiliki solusi parametrik.

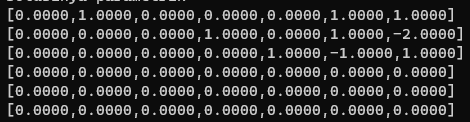
**4.1.c**

Keterangan: SPL yang diinput pada eksekusi di atas adalah:



Analisis:

Solusi SPL di atas parametrik karena matriks yang dihasilkan setelah melakukan eliminasi Gauss-Jordan adalah:



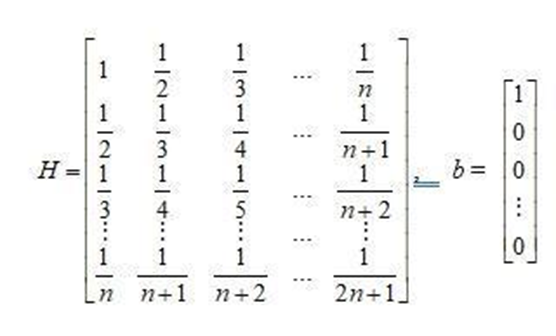
Karena ada baris yang semuanya bernilai 0 dan kolom paling kanan bernilai 0 juga (yang disebabkan oleh banyak persamaan lebih banyak daripada banyak variabel, mengakibatkan barisan paling bawah otomatis bernilai 0 semua termasuk kolom paling kanan), maka SPL ini memiliki solusi parametrik.

**4.1.d**

**Matriks Hilbert dengan N = 6**

****

Keterangan: Isi dari file hilbert6.txt adalah

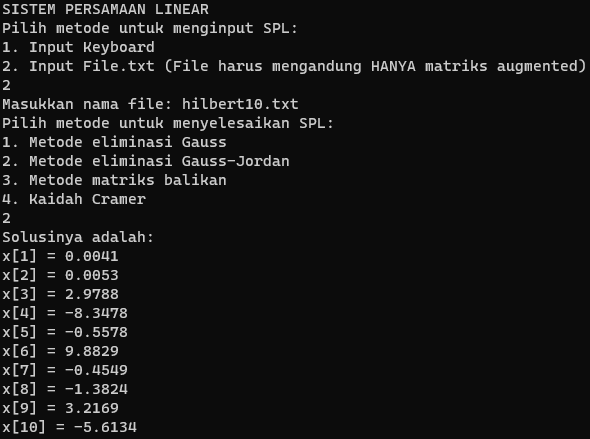


dengan n = 6 dan ditulis dalam bentuk matriks augmented.

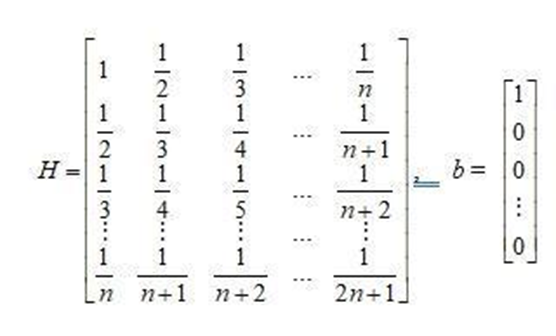
Analisis:

Persamaan di atas memiliki solusi unik jika dihitung menggunakan Kaidah Cramer karena nilai D ≠ 0. (Matriks Hilbert memiliki determinan yang sangat kecil hingga mendekati 0, namun tidak pernah sama dengan 0)

**Matriks Hilbert dengan N=10**

****

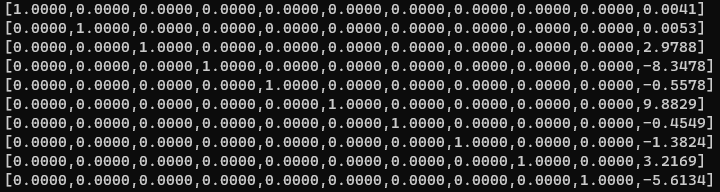
Keterangan: Isi dari file hilbert10.txt adalah



dengan n = 10 dan ditulis dalam bentuk matriks augmented.

Analisis:

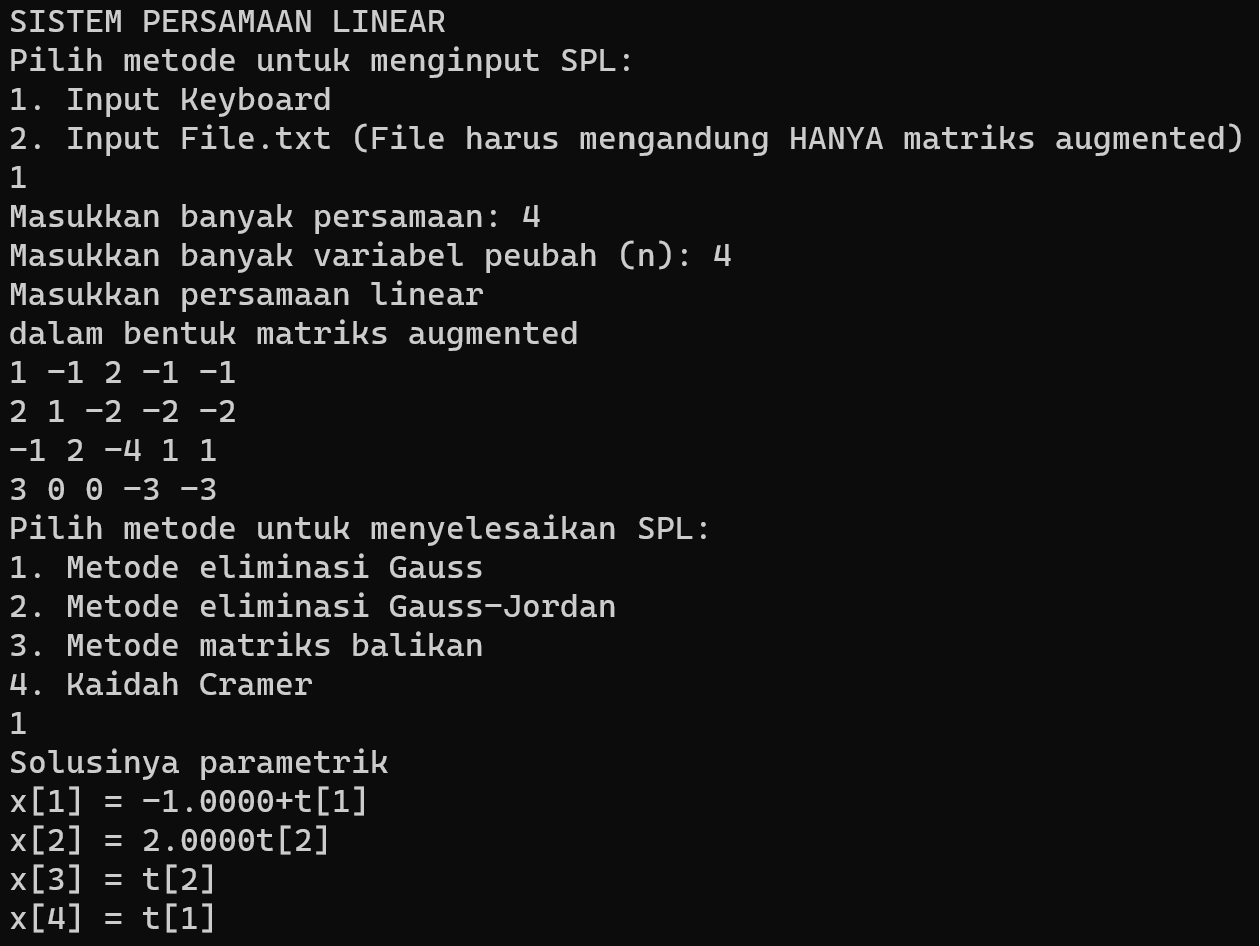
Solusi SPL di atas berupa solusi unik karena matriks yang dihasilkan setelah melakukan eliminasi Gauss-Jordan adalah:



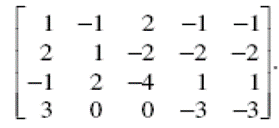
Karena tidak ada baris yang memiliki nilai 0 semua, maka solusinya unik.

**4.2 SPL Berbentuk Matriks Augmented**

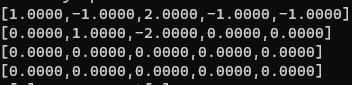
**4.2.a**

****

Keterangan: SPL yang diinput dengan keyboard pada eksekusi di atas adalah:

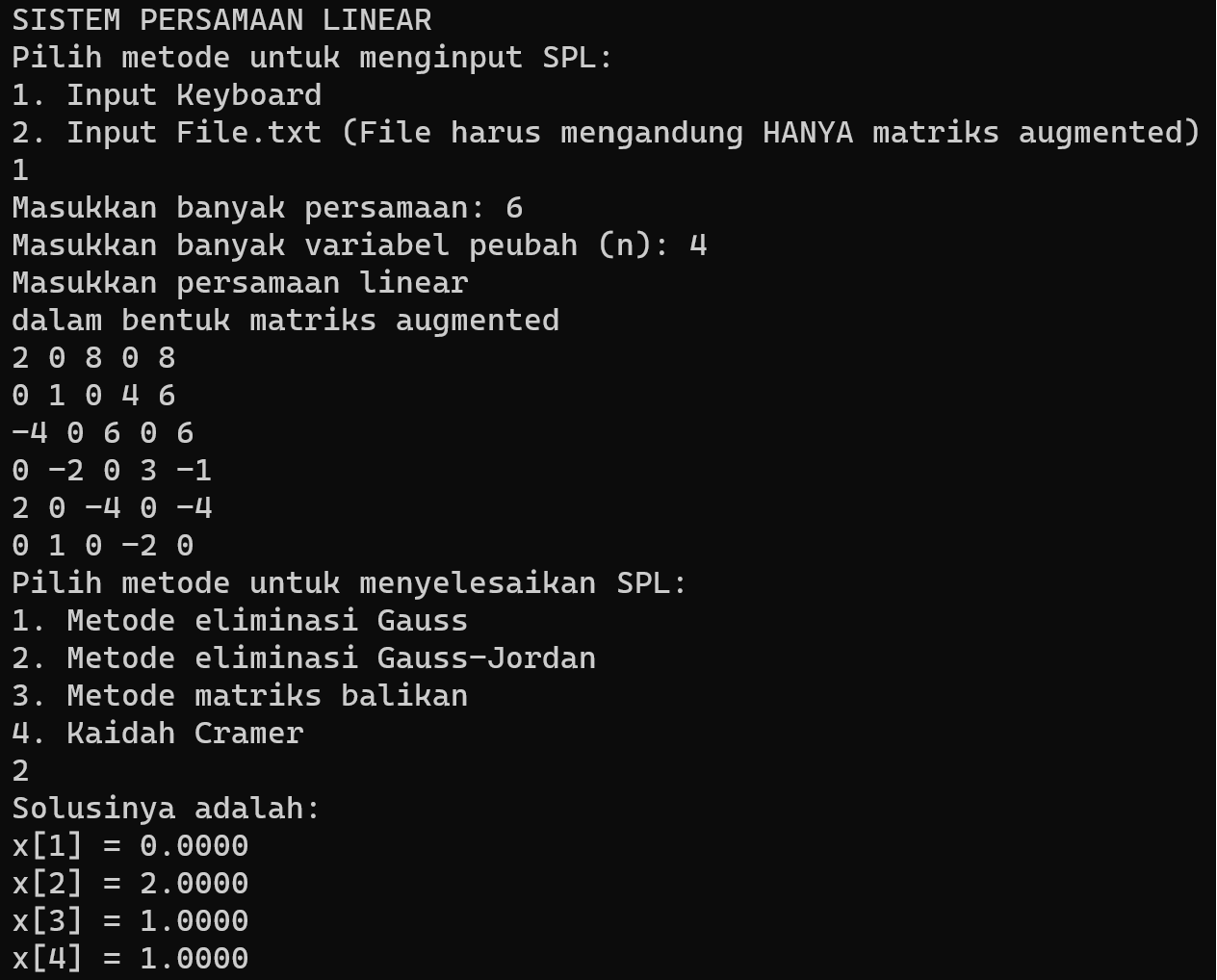


Analisis:

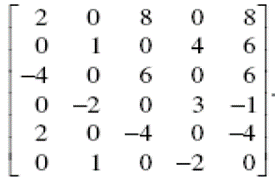
Solusi SPL di atas parametrik karena matriks yang dihasilkan setelah melakukan eliminasi Gauss adalah:

Karena ada baris yang semuanya bernilai 0 dan kolom paling kanan bernilai 0 juga, SPL ini memiliki solusi parametrik.

**4.2.b**

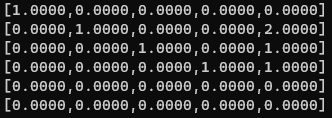
****

Keterangan: SPL yang diinput dengan keyboard pada eksekusi di atas adalah:



Analisis:

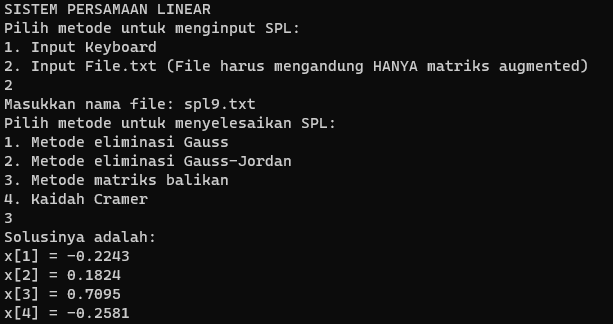
Perhatikan bahwa banyak persamaan di SPL di atas lebih banyak daripada banyak variabelnya. (4 variabel 6 persamaan). Empat persamaan di atas memiliki solusi unik karena matriks yang dihasilkan setelah melakukan eliminasi Gauss adalah (perhatikan 4 baris teratas saja).



Karena 4 baris pertama pada matriks di atas menunjukkan solusi unik, kita perlu mengecek kekonsistenan solusi unik tersebut dengan menguji solusi tersebut ke persamaan 5 dan 6. Ternyata setelah dilakukan, hasilnya konsisten, sehingga SPL di atas memiliki solusi yang unik.

**4.3 SPL berbentuk persamaan**

**4.3.a**

****

Keterangan: SPL yang diinput dalam bentuk matriks augmented di file spl9.txt pada eksekusi di atas adalah:

8*x*1 + *x*2 + 3*x*3 + 2*x*4 = 0

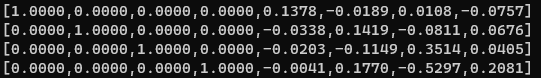
2*x*1 + 9*x*2 -*x*3 - 2*x*4 = 1

*x*1 + 3*x*2 + 2*x*3 - *x*4 = 2

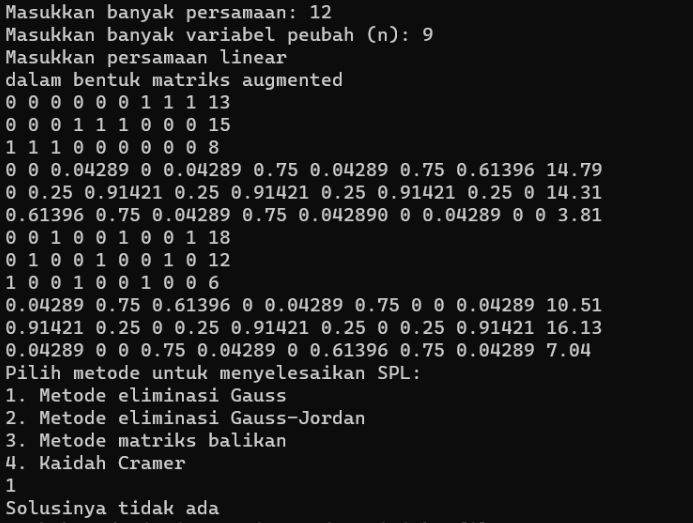
*x*1 + 6*x*3 + 4*x*4 = 3

Analisis:

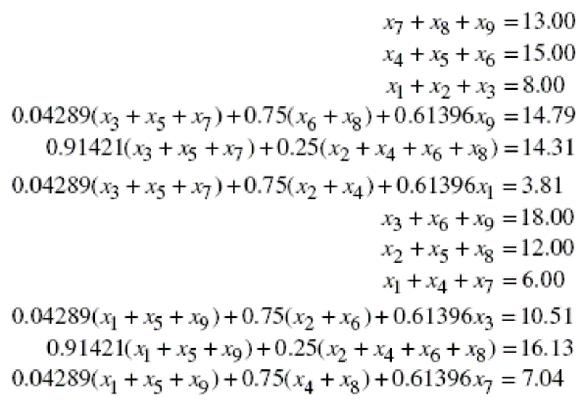
Solusi SPL di atas berupa solusi unik karena berhasil didapatkan bentuk [I|A-1] setelah melakukan metode SPL invers.



**4.3.b**

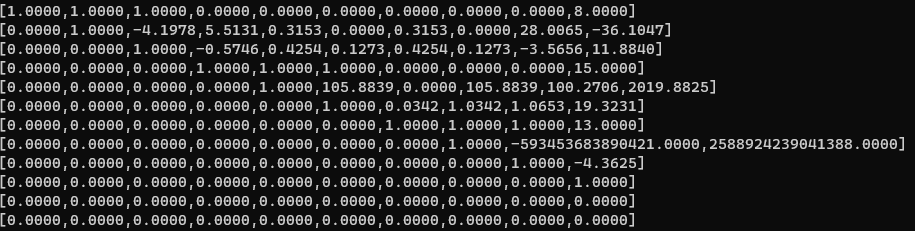


Keterangan: SPL yang diinput pada eksekusi di atas adalah



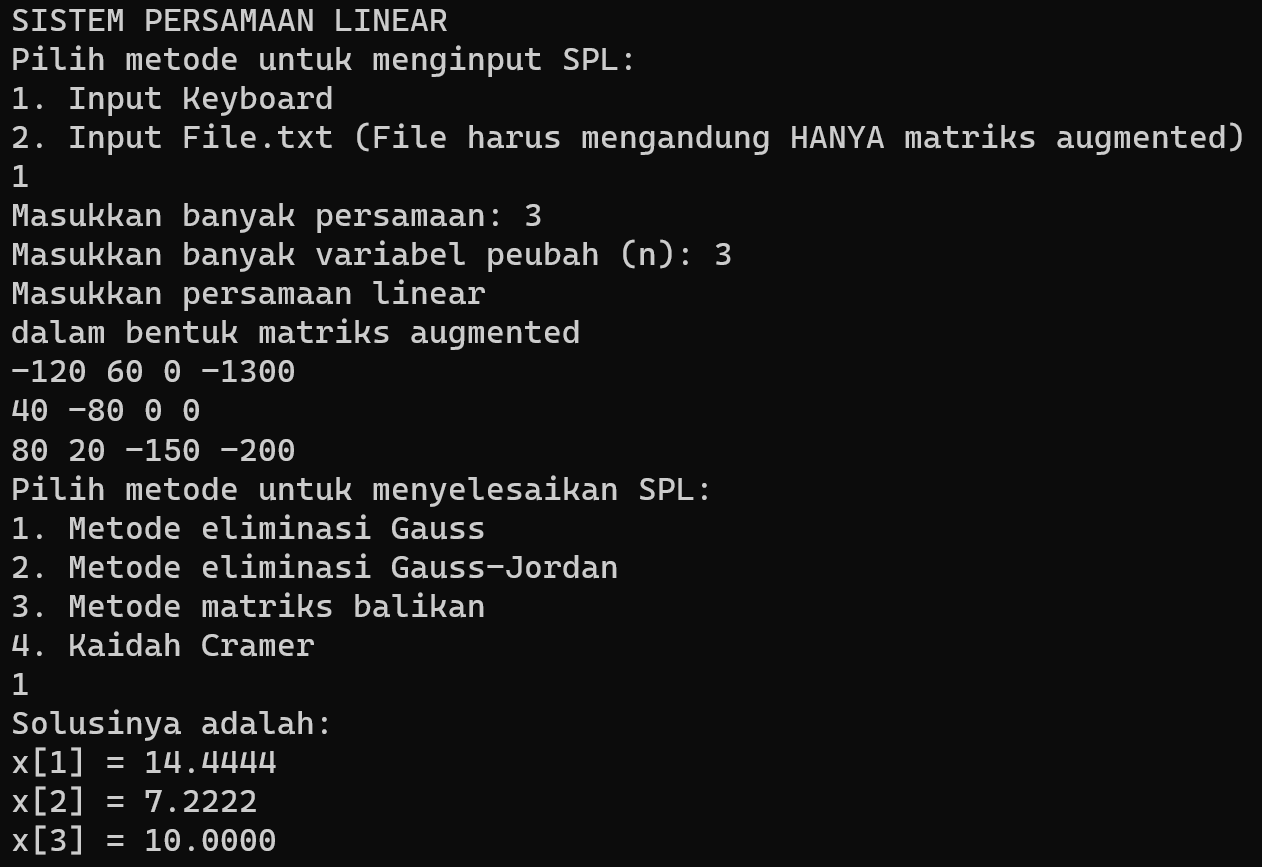
Analisis:

Persamaan di atas tidak memiliki solusi karena matriks hasil eliminasi Gauss di atas adalah



Terdapat 1 baris di mana ada baris yang elemennya 0 semua, kecuali untuk kolom paling kanan. Maka, SPL di atas tidak ada solusi.

**4.4 Penerapan SPL**

****

Keterangan: SPL yang diinput dengan keyboard pada eksekusi di atas adalah SPL untuk soal 4 dari Spesifikasi Tugas Besar 1 IF2123 Aljabar Linear dan Geometri bagian studi kasus.

Analisis:

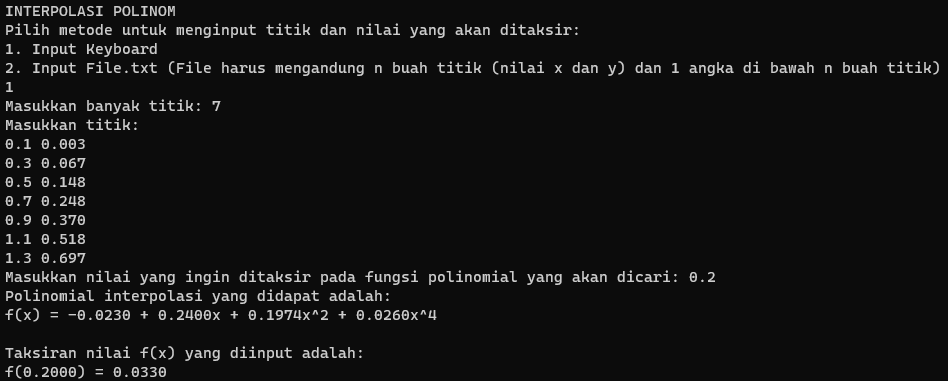
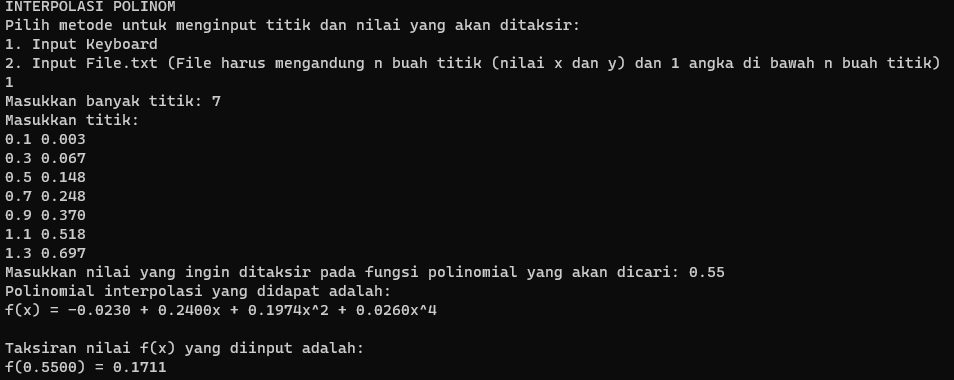
Solusi SPL di atas berupa solusi unik karena matriks yang dihasilkan setelah melakukan eliminasi Gauss adalah:

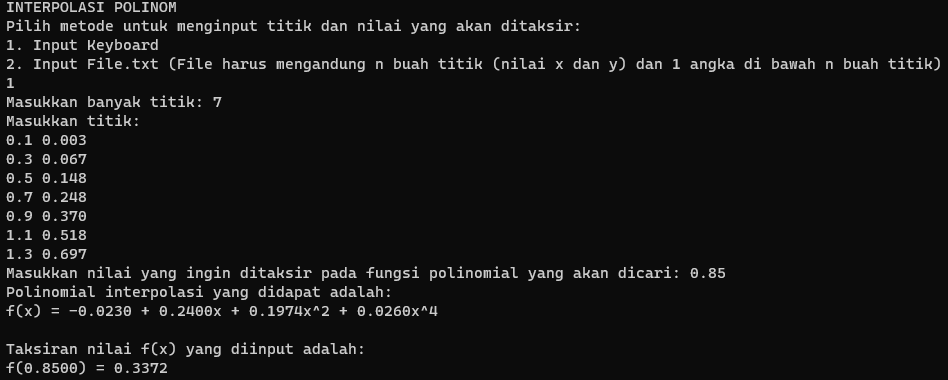


Karena tidak ada baris yang memiliki nilai 0 semua, maka solusinya unik.

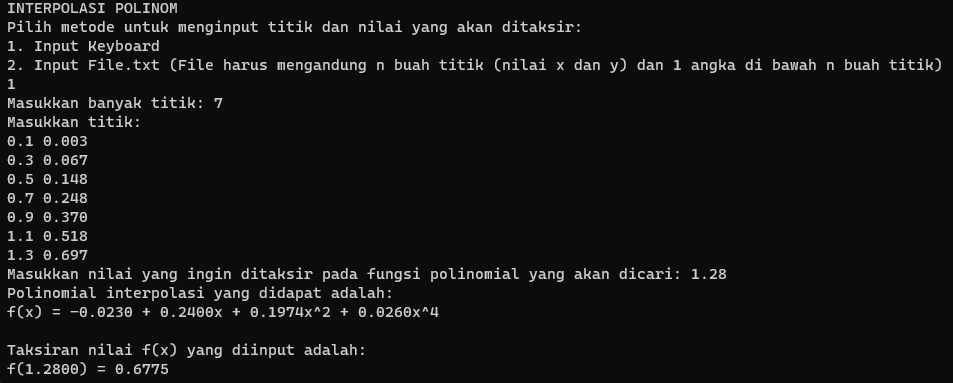
**4.5 Interpolasi Polinomial**

**4.5.1 a**

****



Analisis:



Analisis:

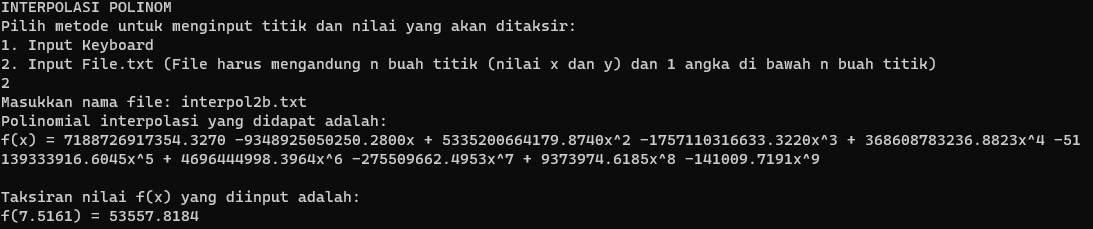
Fungsi polinomial yang didapatkan dari interpolasi polinomial dari soal 5a dari Spesifikasi Tugas Besar 1 IF2123 Aljabar Linear dan Geometri bagian studi kasus adalah

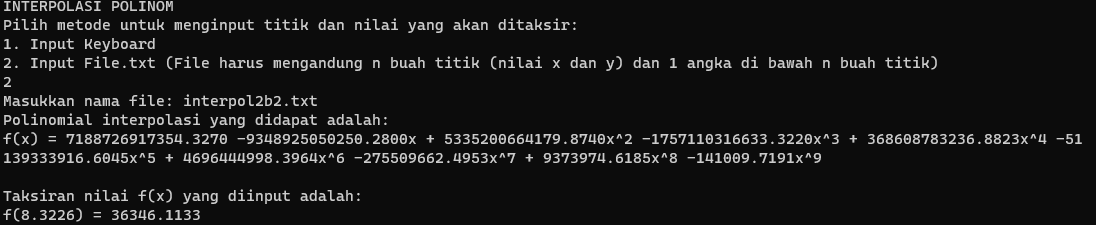
f(*x*) = -0,0230 + 0,2400*x* + 0,1974*x*2 + 0,0260*x*4

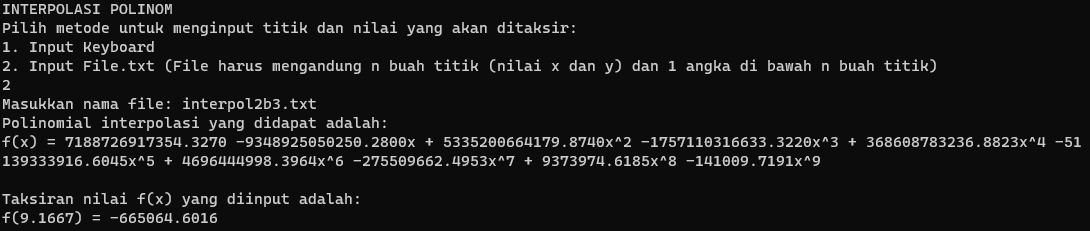
Dan nilai taksiran yang diperoleh adalah:

f(0,2) = 0,0330; f(0,55) = 0,1711; f(0,85) = 0,3372; f(1,28) = 0,6775

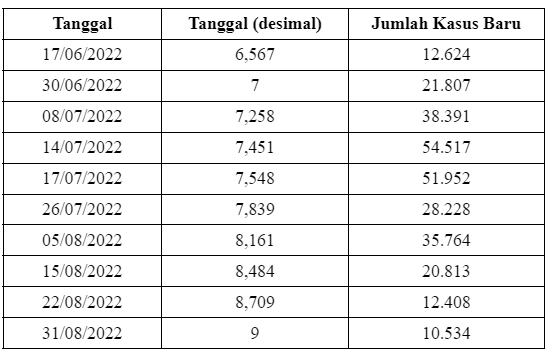
**4.5.b**







Keterangan: Data titik yang diinput dari ketiga file untuk interpolasi polinomial di atas adalah:



Tanggal yang diinput pada file di atas adalah

1. 16/07/2022 (7.5161, di file interpol2b.txt)
2. 10/08/2022 (8.3226, di file interpol2b2.txt)
3. 05/09/2022 (9.1667, di file interpol2b3.txt)

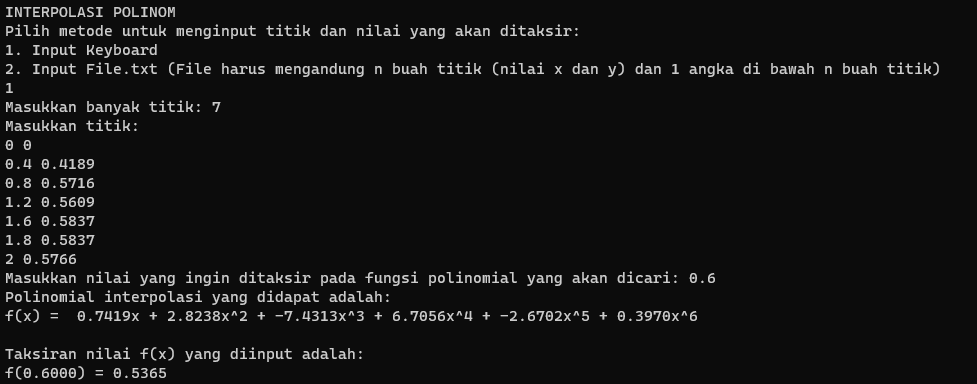
Analisis:

Fungsi polinomial yang didapatkan adalah



Perhatikan bahwa untuk tanggal 05/09/2022 didapat nilai negatif. Hal ini disebabkan karena fungsi polinomial di atas memiliki derajat yang sangat tinggi (derajat 9) sehingga f(x) semakin sering untuk berosilasi. Akibatnya, ada beberapa nilai yang akan bernilai negatif karena frekuensi osilasi yang sangat tinggi berdasarkan fungsi yang didapatkan dari interpolasi polinomial. Fenomena ini disebut fenomena Runge.

**4.5.3 c**



Keterangan: Input diatas berdasarkan soal studi kasus Interpolasi bagian c di Spesifikasi Tugas.

Analisis:

Fungsi f(x) yang didapatkan dari interpolasi polinomial adalah:

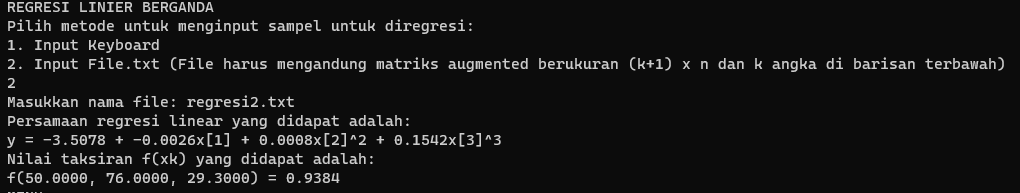


Dan taksiran dari fungsi di atas untuk x = 0.6 adalah

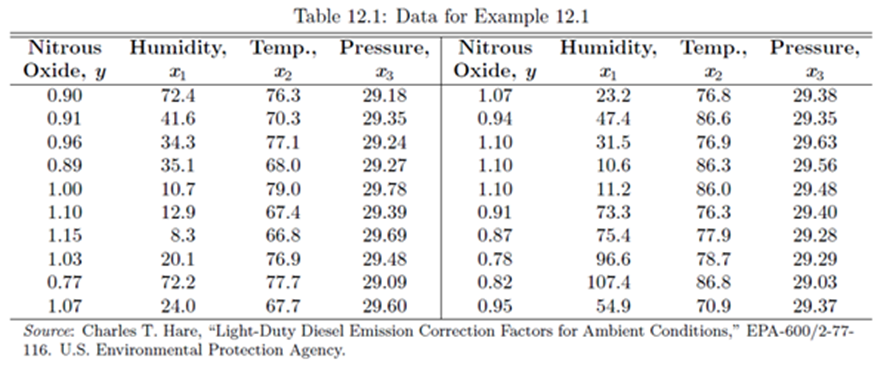


Namun, nilai f(0.6) di grafik seharusnya adalah f(0.6) = 0.468. Hasil taksiran polinomial di atas kurang begitu akurat jika menggunakan interpolasi polinomial, namun sudah mendekati hasil riil.

**4.6 Regresi Linear Berganda**

****

Keterangan: Data regresi yang diinput dari file regresi2.txt pada eksekusi di atas adalah



Dengan input yang ditaksir adalah humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Analisis:

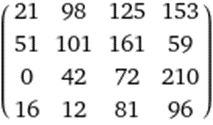
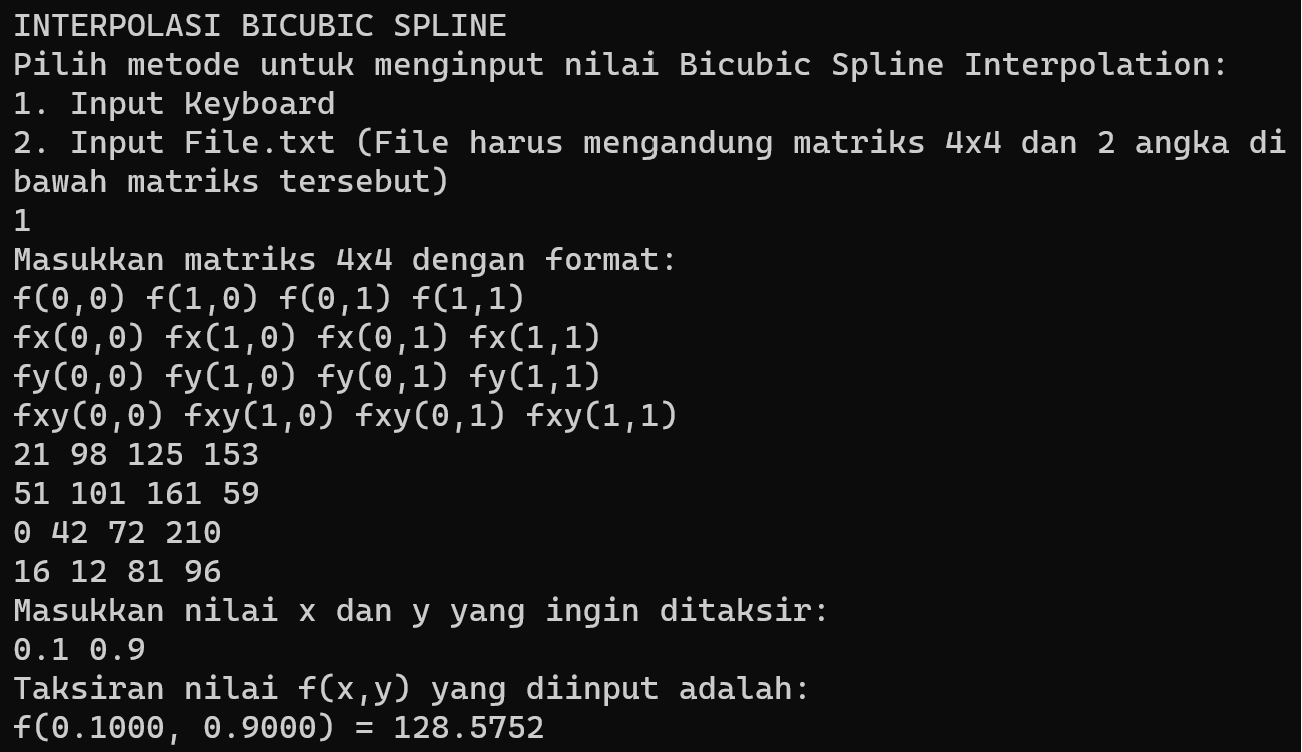
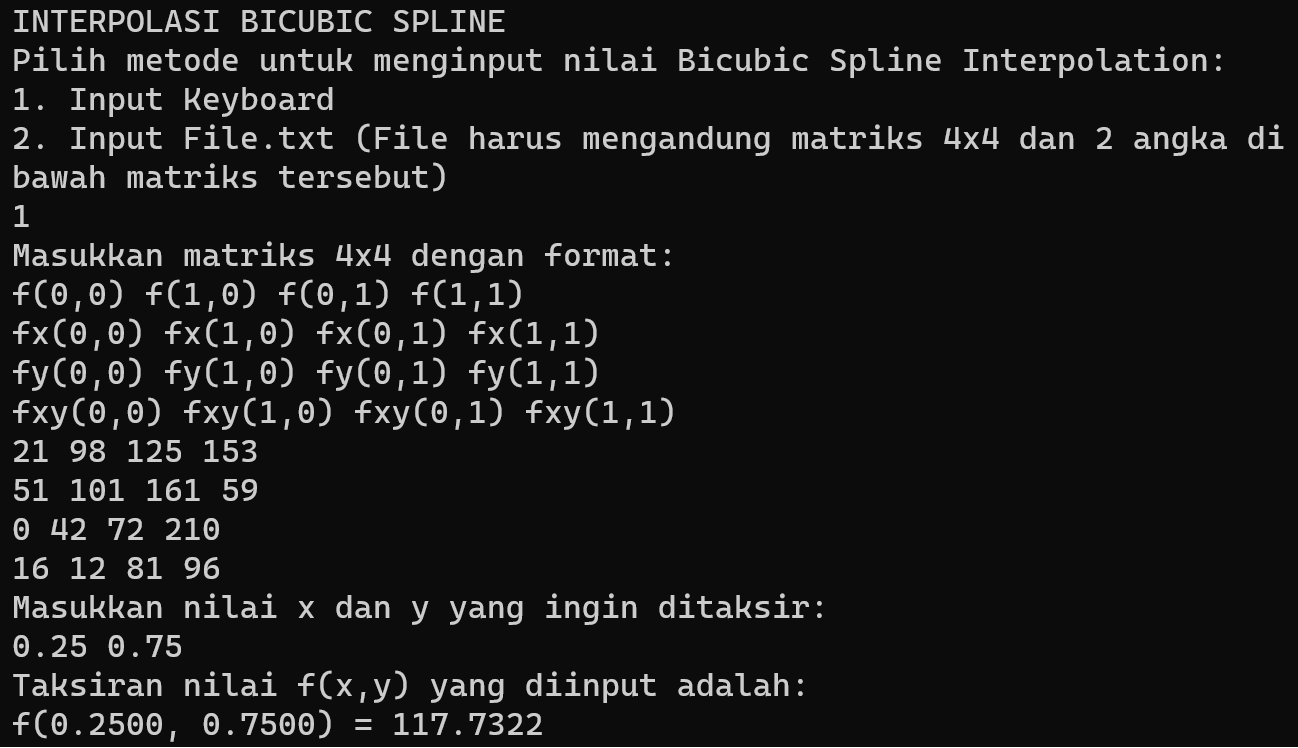
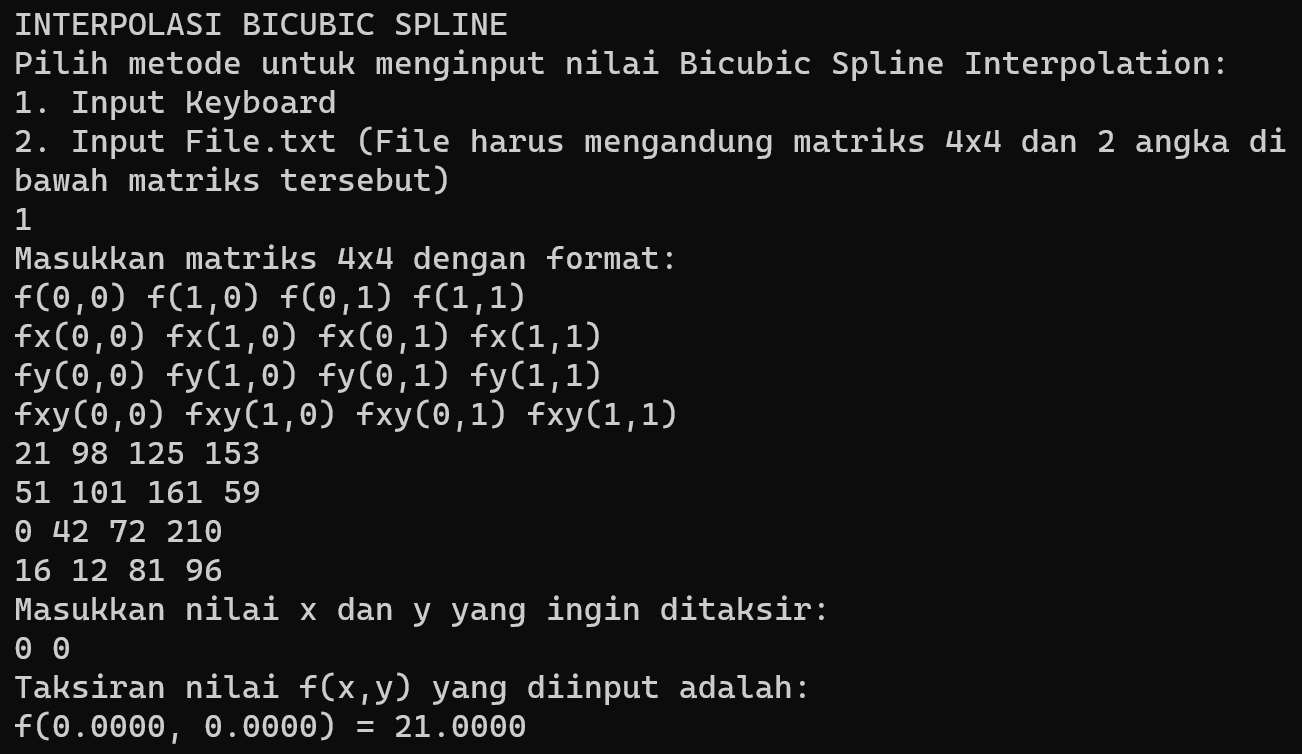
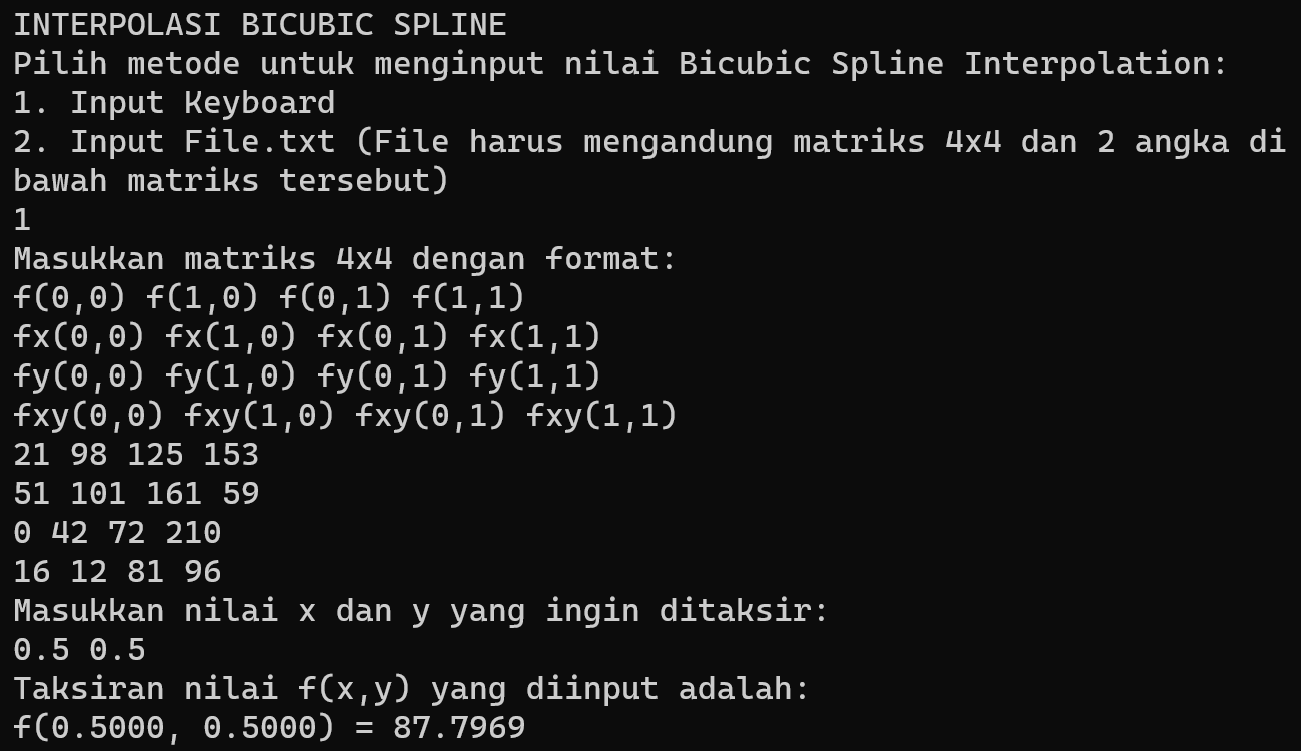


Nilai taksiran f(50,76,29.3) yang didapatkan adalah



Nilai tersebut masih mendekati dengan data input.

**4.7 Nomor 7 (Interpolasi Bicubic Spline)**

Keterangan: Input yang diberikan sesuai spesifikasi adalah

Analisis:

Taksiran nilai yang didapatkan adalah

f(0,0) = 21; f(0.5,0.5) = 87,7969; f(0.25,0.75) = 117.7322; f(0.1,0.9) = 128,5752

**Bab 5 : Kesimpulan, Saran, Komentar, dan Refleksi**

**5.1 Kesimpulan**

Hasil yang didapatkan dari pengerjaan tugas besar ini adalah sebuah program yang berfungsi untuk melakukan operasi aljabar linear dan aplikasi aljabar linear, seperti interpolasi polinomial, regresi linear berganda, dan bicubic spline interpolation. Selain itu, didapatkan juga kekurangan dari interpolasi polinomial, yaitu adanya fenomena Runge yang terjadi jika derajat polinomial yang didapat sangat tinggi sehingga keakuratan dari metode ini masih kurang akurat, terlebih lagi jika titik yang ingin ditaksir terletak di ujung kanan dan kiri fungsi polinomial yang didapat

**5.2 Saran**

Saran yang kami berikan untuk pengembangan berikutnya adalah

1. Untuk regresi linear berganda, sebaiknya diperjelas pada spesifikasi bahwa variabel peubah ada lebih dari atau sama dengan 1, atau perjelas lebih detail lagi arti variabel-variabel yang ada di normal estimation equation.
2. Untuk Bicubic Spline Interpolation, sebaiknya diperjelas apa yang harus dimasukkan ke dalam matriks X jika ada kasus 0n di mana n≤0, karena nilai 0n di mana n≤0 tidak terdefinisi.

**5.3 Refleksi**

Refleksi kami terhadap tugas ini adalah:

1. Kami harus memperbaiki cara untuk handling floating point.
2. Kami harus berkomunikasi lebih baik lagi untuk meminimalisasi terjadinya *clashing* baik saat push kode atau saat melakukan bagian lain dari tugas besar.
3. Kami harus memperbaiki diri kami sendiri untuk lebih terstruktur dan terarah saat pengerjaan tugas berkelompok.

**Daftar Referensi**

[Paul Bialek - Polynomial Interpolation Using Matrices](https://www.youtube.com/watch?v=z1YUTRG3ngM)

[Yonsei University - Runge Phenomenon](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiduqDd3d6BAxVYVWwGHbehBHU4ChAWegQICBAB&url=https%3A%2F%2Fweb.yonsei.ac.kr%2F_ezaid%2Fboard%2Fdownload.aspx%3Fmethod%3Ddownload%26pfkHomepageNo%3D1710%26fkBoardEntryPkNo%3D20%26attacheFileChoice%3D1%26pkNo%3D8599&usg=AOvVaw2QofVfsE55-C8s8MH__z6M&opi=89978449)

[TileStats - Multiple linear regression](https://www.youtube.com/watch?v=AP_K7SaKkIE)

[Marquette University - BiLinear, Bicubic, and In Between Spline Interpolation](https://www.mssc.mu.edu/~daniel/pubs/RoweTalkMSCS_BiCubic.pdf)

[Engineering at Alberta - Polynomial Interpolation](https://engcourses-uofa.ca/introduction-to-numerical-analysis/polynomial-interpolation/)

[Rinaldi Munir - Matriks Eselon](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-02-Matriks-Eselon-2023.pdf)

[Rinaldi Munir - Sistem Persamaan Linier Bagian 1](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-03-Sistem-Persamaan-Linier-2023.pdf)

[Rinaldi Munir - Sistem Persamaan Linier Bagian 2](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2023.pdf)

[Rinaldi Munir - Sistem Persamaan Linier Bagian 3](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-05-Sistem-Persamaan-Linier-2023.pdf)

[Rinaldi Munir - Determinan bagian 1](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-08-Determinan-bagian1-2023.pdf)

[Rinaldi Munir - Determinan Bagian 2](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-09-Determinan-bagian2-2023.pdf)