**LAPORAN TUGAS BESAR**

**IF2123 – ALJABAR GEOMETRI**

**SISTEM PERSAMAAN LINIER DAN APLIKASINYA**

Disusun oleh:

1. Abel Stanley (13517068)
2. Samantha Olivia Tandri (13517123)
3. Saskia Imani (13517142)



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BAB I**

**DESKRIPSI MASALAH**

Sistem persamaan linier (SPL) dengan *n* peubah (variabel) dan *m* persamaan adalah berbentuk

*a11 x1 + a12 x2  + .... + a1n xn = b1*

*a21 x1 + a22 x2  + .... + a2n xn = b2*

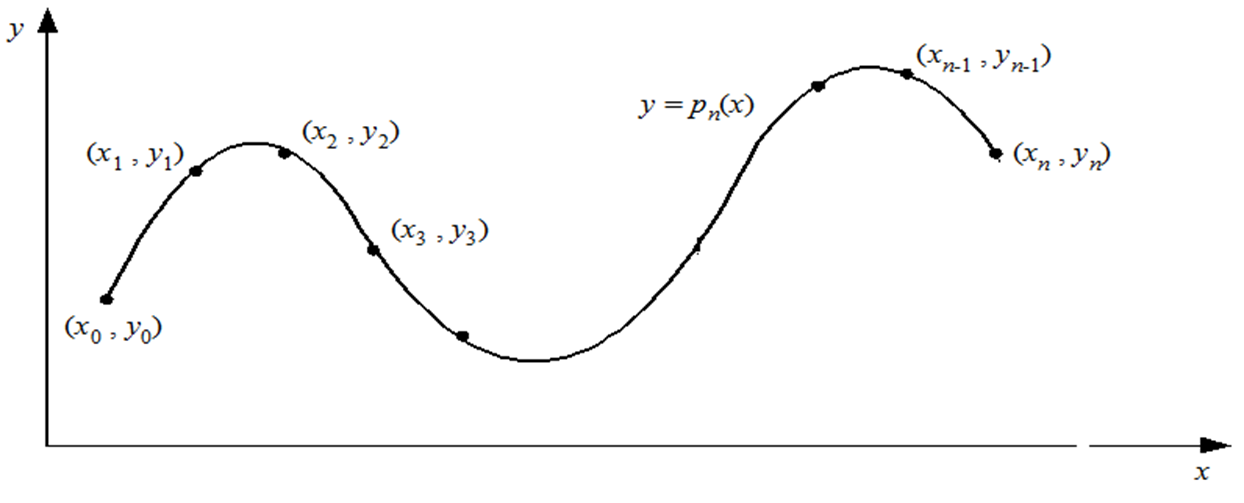
*: :*

*: :*

*am1 x1 + am2 x2 + .... + amn xn = bm*

yang dalam hal ini xi adalah peubah, aij dan bi adalah koefisien ∈ R. Sembarang SPL dapat diselesaikan dengan metode eliminasi Gauss dan metode eliminasi Gauss-Jordan. Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

Sistem persamaan linier memiliki banyak aplikasi dalam bidang sains dan rekayasa, salah satunya adalah mengestimasi nilai fungsi dengan interpolasi polinom. 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*].

**BAB II**

**TEORI SINGKAT**

**A. Eliminasi Gauss dan Gauss-Jordan**

Sebuah matriks *augmented* yang terbentuk dari sebuah sistem persamaan dapat diselesaikan dengan mengubah matriks tersebut ke dalam bentuk eselon baris atau eselon baris tereduksi.

Matriks eselon baris tereduksi

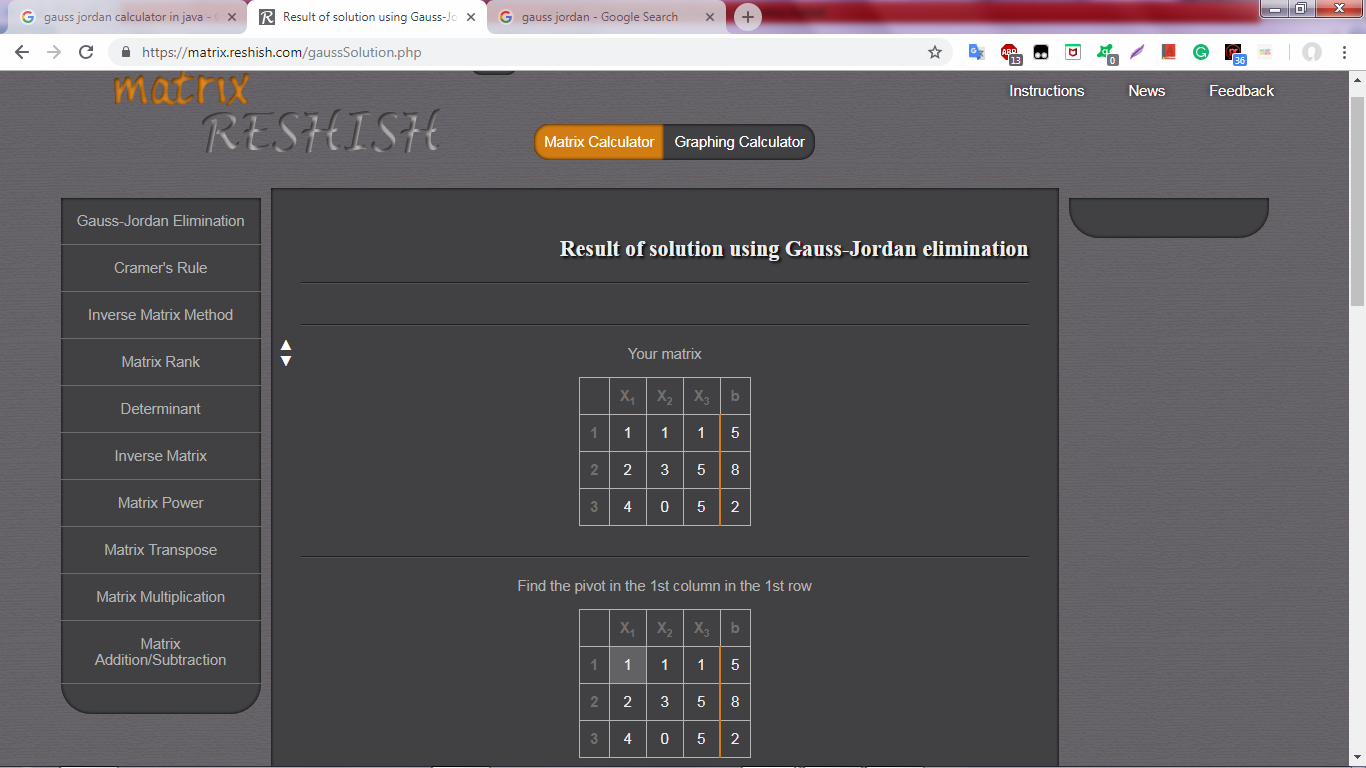
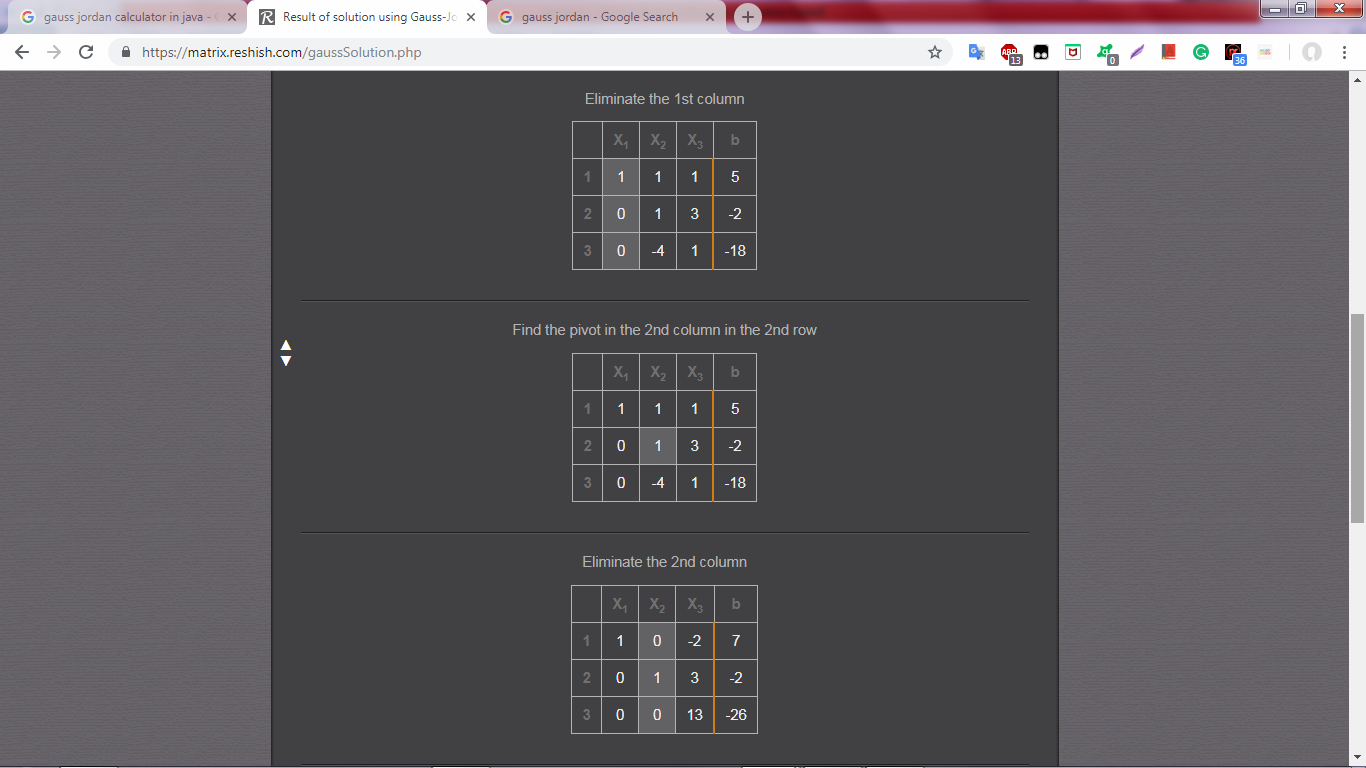
Matriks eselon baris

Kedua bentuk ini dapat dicapai dengan menggunakan operasi baris elementer (OBE), yaitu operasi-operasi yang tidak akan mengubah solusi dari sebuah matriks *augmented* atau sistem persamaan linier. Operasi-operasi yang termasuk operasi baris elementer antara lain:

1. Menukar dua buah baris mana pun.
2. Mengkalikan setiap elemen pada satu baris dengan sebuah nilai yang bukan nol.
3. Menambahkan elemen suatu baris ke elemen baris yang lain.

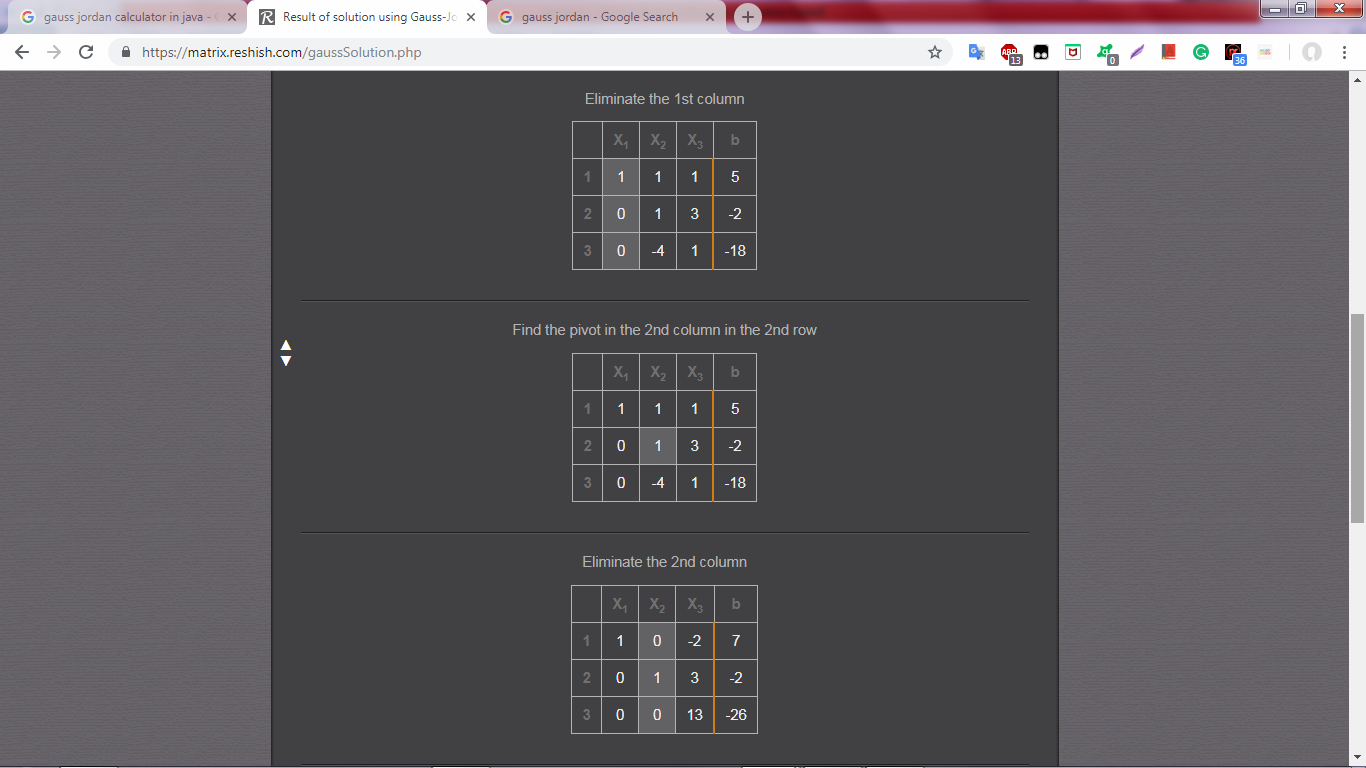
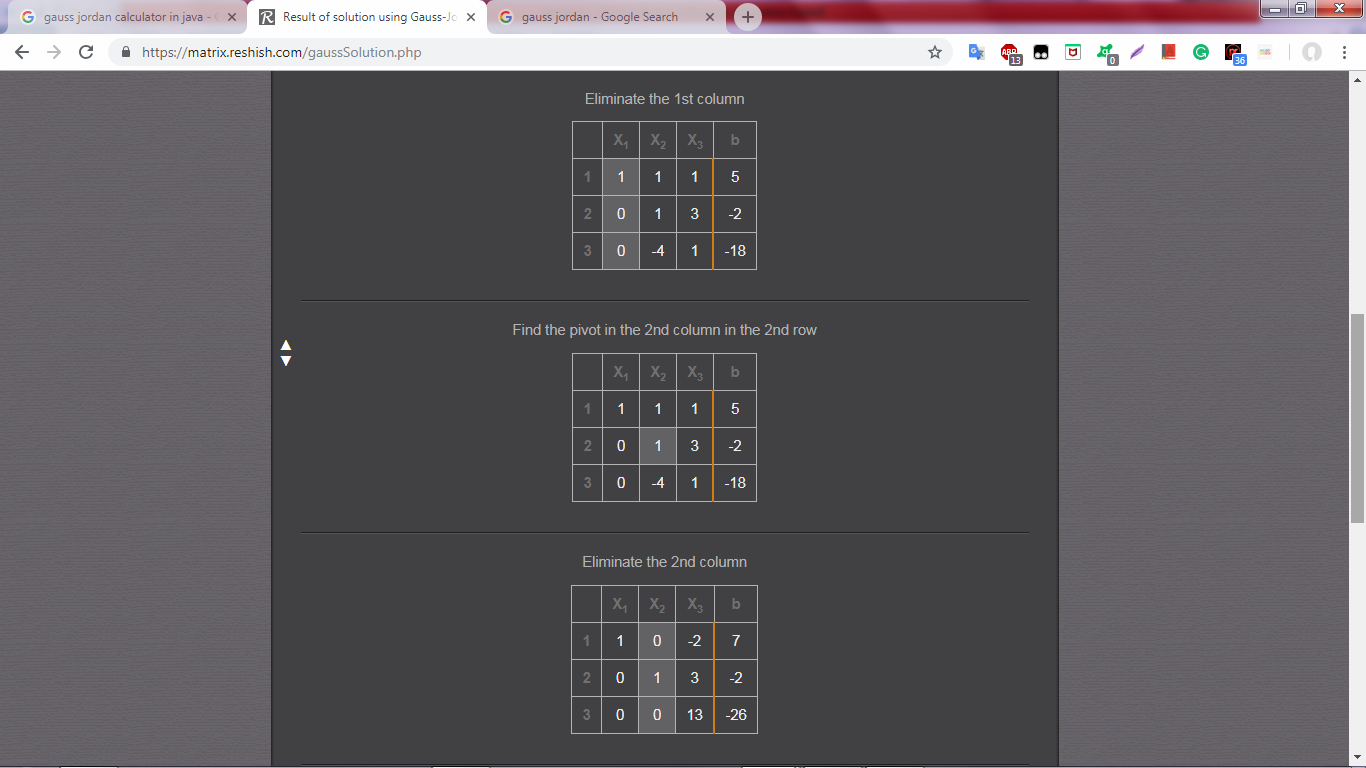
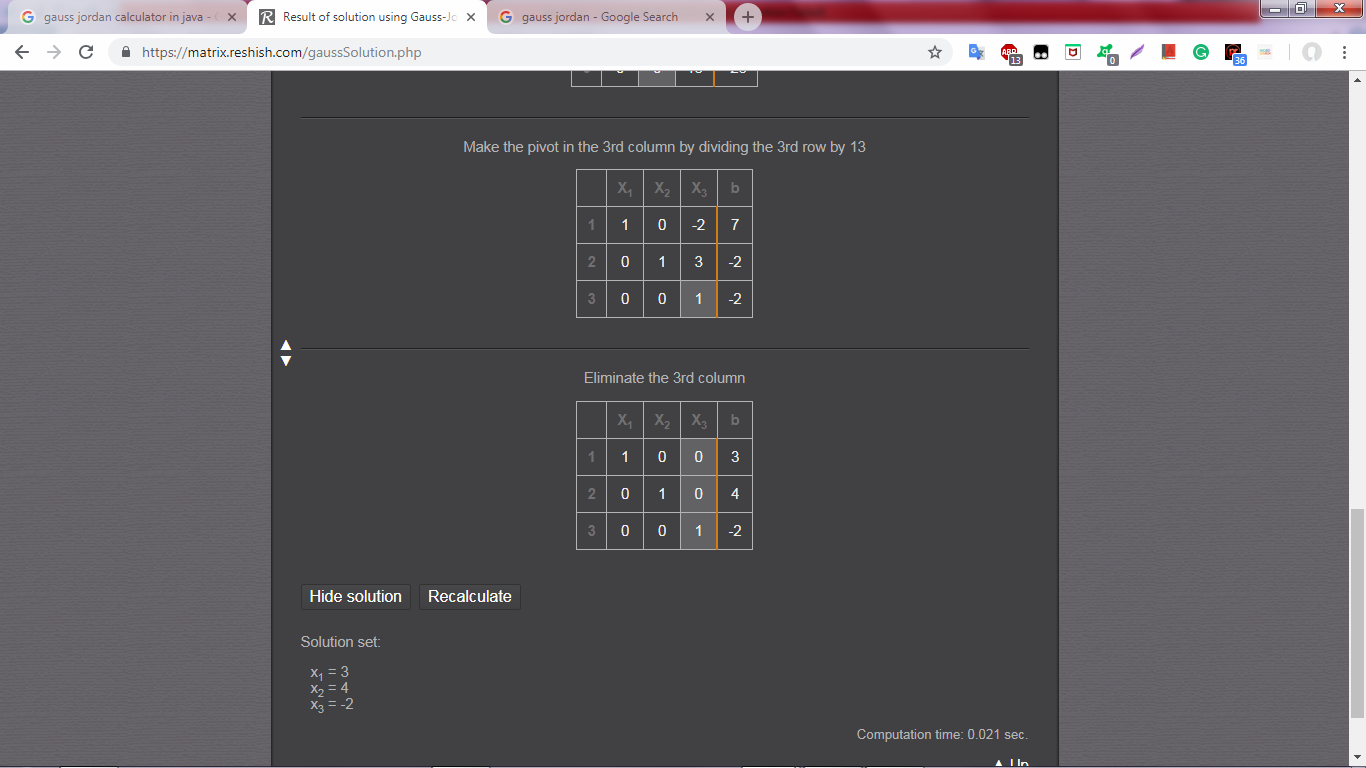
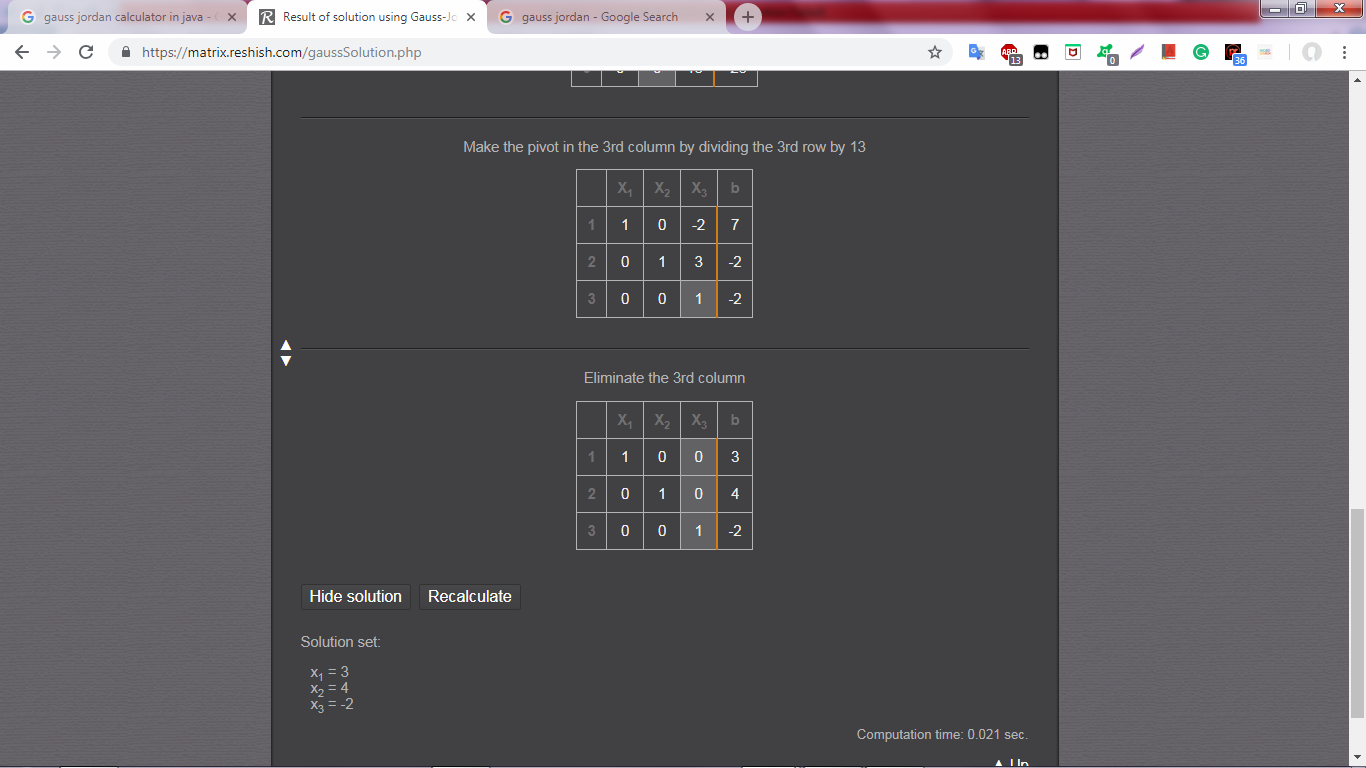
Untuk mencapai bentuk matriks eselon baris, elemen pada baris pertama kolom pertama dijadikan sebagai acuan. Jika nilai elemen ini adalah nol, maka baris pertama ditukar dengan baris lain yang elemen pada kolom pertamanya tidak nol. Kemudian, untuk setiap baris di bawah baris pertama, kolom pertama dijadikan nol. Setelah itu, berlanjut ke kolom selanjutnya dicari acuan atau pivot poin yang harus bernilai satu lalu seperti proses sebelumnya yaitu baris dibawah pivot poin pada kolom yang sama harus dijadikan nol hingga mencapai baris terakhir. Setelah tercapai matriks yang berbentuk segitiga ataupun segitiga bawah maka dilakukan *back subtitution.*

Contoh Eliminasi Gauss-Jordan.



Menjadikan semua angka dibawah pivot point pertama menjadi 0.

Pivot poin pertama berada di kolum 1 dan baris 1.



Menjadikan angka di kolom 2 menjadi 0 selain pivot poin.

Pivot poin kedua berada di kolum 2 dan baris 2.

Pivot poin ketiga berada di kolom ketiga baris ketiga.

Menjadikan angka di kolom 3 menjadi 0 selain pivot poin.

Setelah terbentuk matriks yang berbentuk matriks eselon tereduksi maka hasil penyelesaian SPL dapat diperoleh , seperti pada soal diatas maka hasilnya adalah x1=3, x2=4, x3=-2.

Dalam penyelesaian Gauss-Jordan digunakan beberapa prosedur seperti pivot dan swap. Prosedur-prosedur tersebut digambarkan sebagai berikut

Type Matrix : array [1..100] of array [1..100] of double

Function gjordan(Matrix matriks, array aug, integer m, integer n) → Matrix

{KAMUS}

a : Matrix

N ← b.length

i traversal[0..N]

j traversal[0..N]

a[i][j] ← A[i][j]

i traversal[0..N]

a[i][N+i] ← 1.0

i traversal[0..N]

a[i][N+N] ← b[i]

solve()

→ a → ←

Procedure solve()

{KAMUS}

max : integer

p traversal [0..N]

max ← p

p traversal [p+1..N]

if (Math.abs(a[i][p]) > Math.abs(a[max][p]))

max ← i

swap(p, max)

if (Math.abs(a[p][p]) <= eps)

continue

pivot(p, p)

Procedure swap(integer row1, integer row2)

{KAMUS}

temp : array

a : Matrix

temp ← a[row1]

a[row1] ← a[row2]

a[row2] ← temp

Procedure pivot(integer p, integer q)

{KAMUS}

alpha : real

i traversal[0..N]

alpha = a[i][q]/a[p][q]

j traversal[0..N+N]

if(i != p && j != q) then

a[i][j] -= alpha \* a[p][j]

i traversal[0..N]

if (i != p) a[i][q] = 0.0 then

j traversal[0..N+N]

if (j != q) a[p][j] /= a[p][q] then

a[p][q] = 1.0

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Procedure lsolve(input Matriks A, input array B)

n,m,mn : int

a : Matriks

n : b.length

m: A.length

mn : A[0]

i traversal [i.m]

mn : int

alpha : real

mn = mn

//build augmented matrix

a : Matriks

I traversal[0..m]

for (int j = 0; j < mn; j++)

a[i][j] ← A[i][j]

I traversal[0..m]

a[i][mn] = b[i];

p transversal[0..n]

// find pivot row and swap

max : int max ← p

p transversal[p+1..n]

if (Math.abs(A[i][p]) > Math.abs(A[max][p])) then

max = i

temp ← A[p]

A[p] ←A[max]

A[max] ←temp

t ← b[p]

b[p] ←b[max]

b[max] ← t

// pivot within A and b

alpha: float

I traversal [p+1..n]

if (A[i][p] == 0.0 && A[p][p] == 0.0)

alpha ← 0

else

alpha ← A[i][p] / A[p][p

b[i] -= alpha \* b[p]

j traversal [p..n]

A[i][j] -= alpha \* A[p][j]

M: Matriks

M ←rref(a,m,n)

printmatrix(m,mn,M)

for (int i = 0; i < m; i++)

b[i] ← M[i][mn-1]

// back substitution

x : Array

I traversal [n-1..0]

sum ←float = 0.0

j traversal [1..1+n]

sum += A[i][j] \* x[j]

x[i] ← (b[i] - sum) / A[i][i]

solutionType : char = 'U'

temp : Array

I traversal [0..N]

temp[i] ← M[i][mn]

}

solutionType = checksolution(m,mn,M,temp)

if (solutionType = 'U') then

printSolution(x,n)

→ x

/\*

**B. Fungsi Interpolasi**

Polinom interpolasi derajat n yang menginterplolasi titik-titik (*x0, y0*)*,* (*x1, y1*)*, ...,* (*xn, yn*) adalah berbentuk *pn*(*x*) = *a0* + *a1x* + *a2x2* + … + *anxn*. Jika hanya ada dua titik, (*x0, y0*) dan (*x1, y1*), maka polinom yang menginterpolasi kedua titik tersebut adalah *p1*(*x*) = *a0* + *a1x* yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (*x0*, *y0*), (*x1*, *y1*), dan (*x2*, *y2*), maka polinom yang menginterpolasi ketiga titik tersebut adalah *p2*(*x*) = *a0* + *a1x* + *a2x2* atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (*x0*, *y0*), (*x1*, *y1*), (*x2*, *y2*), dan (*x3*, *y3*), polinom yang menginterpolasi keempat titik tersebut adalah *p3*(*x*) = *a0* + *a1x* + *a2x2* + *a3x3*, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat *n* untuk *n* yang lebih tinggi asalkan tersedia (*n*+1) buah titik data. Dengan menyulihkan (*xi*, *yi*) ke dalam persamaan polinom *pn*(*x*) = *a0* + *a1x* + *a2x2* + … + *anxn* untuk *i* = 0, 1, 2, …, *n*, akan diperoleh *n* buah sistem persamaan lanjar dalam *a0*, *a1*, *a2*, …, *an*,

*a0 + a1x0 + a2x02 + ... + an x0n = y0*

*a0 + a1x1 + a2x12 + ... + an x1n = y1*

*... ...*

*a0 + a1xn + a2xn2 + ... + an xnn = yn*

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* + *a2x2*. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah

*a0* + 8.0*a1* + 64.00*a2* = 2.0794

*a0* + 9.0*a1* + 81.00*a2* = 2.1972

*a0* + 9.5*a1* + 90.25*a2* = 2.2513

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan *a0* = 0.6762, *a1* = 0.2266, dan *a2* = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah *p2*(*x*) = 0.6762 + 0.2266*x* - 0.0064*x2*. Dengan menggunakan polinom ini, maka nilai fungsi pada *x* = 9.2 dapat ditaksir sebagai berikut: *p2*(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

Dalam program, polinom interpolasi ditentukan dengan menggunakan prosedur Gauss atau GaussJordan. Fungsi untuk mencari nilai ordinat dari absis dan polinom diberi nama SolusiInterpolasi dan dapat digambarkan sebagai berikut:

polinom: array [1...100] of float

function Solusi (array polinom, integer l, float x) → float

KAMUS LOKAL

float y

ALGORITMA

y ← 0

i traversal [1...l]

y ← y + polinom[i+1] \* (x^i)

→ y

**BAB III**

**IMPLEMENTASI PROGRAM**

Bahasa Java merupakan bahasa yang diperuntukkan *object-oriented programming* (OOP) atau pemrograman berorientasi objek. Ini berarti Bahasa Java menggunakan kelas-kelas untuk membentuk objek, dan menentukan atribut serta perilaku setiap objek di dalam masing-masing kelas objek.

Dalam pengerjaan tugas ini, kelompok kami tidak sepenuhnya memanfaatkan fitur OOP yang terdapat pada Java dan memilih untuk melakukan pemrograman yang lebih bersifat prosedural, dengan pertimbangan bahwa pemahaman kelompok terhadap bahasa pemrograman prosedural sudah lebih baik dan bahasa Java dapat diakali untuk berlaku secara prosedural. Karena itu, dalam program buatan kelompok kami, semua kelas di dalam kelas utama yang ada pada program merupakan sebuah fungsi atau prosedur (biasa disebut *method* atau metode dalam Java), beserta satu *method* program utama yang mengandung variabel “global”.

**A. Variabel dan Konstanta Program**

Berikut merupakan variabel-variabel yang terdapat di dalam program:

1. ***Array* dan Matriks**
   1. matriks : *array* dua dimensi yang menyimpan nilai matriks *augmented*
   2. aug : *array* (kolom) *augmented* dari matriks *augmented*
   3. interpolasi : *array* dua dimensi berisi titik-titik yang akan diinterpolasi
   4. solusi : *array string* yang mengandung solusi matriks *augmented*
   5. persamaan : *array double* yang mengandung konstanta persamaan interpolasi
2. **Scanner**: sebuah kelas *built-in* Java yang berfungsi membaca masukan
3. **Integer dan Double**
   1. m: *integer* yang menyimpan jumlah baris *matriks* dan *interpolasi*
   2. n: *integer* yang menyimpan jumlah kolom *matriks*
   3. l: *integer* yang menyimpan derajat fungsi interpolasi
   4. x: *double* yang menyimpan nilai absis untuk persamaan interpolasi
   5. y: *double* yang menyimpan nilai ordinat untuk persamaan interpolasi
   6. opsi1: *integer* yang menyimpan input pengguna untuk pilihan menu pertama
   7. opsi2: *integer* yang menyimpan input pengguna untuk pilihan menu kedua
   8. opsi3: *integer* yang menyimpan input pengguna untuk pilihan menu ketiga

**B. Komponen Program**

Berikut merupakan fungsi dan prosedur yang terdapat di dalam program berserta parameternya (di dalam tanda kurung):

1. **public class Main**

Main adalah kelas utama yang mencakup keseluruhan program, yaitu kelas-kelas fungsi dan prosedur serta kelas program utama.

1. **public static void MatriksDariLayar**

Prosedur MatriksDariLayar adalah prosedur yang berfungsi menerima masukan pengguna dan memasukkannya ke dalam variabel matriks *augmented*. Spesifikasi prosedur berupa jumlah baris (m), jumlah kolom (n), matriks itu sendiri (matriks), kolom *augmented* matriks (aug), dan Scanner untuk input dari pengguna yang digunakan (scanner).

1. **public static int CountRow**

Fungsi CountRow adalah fungsi yang membaca file eksternal, menghitung jumlah baris pada file eksternal, dan mengeluarkan jumlah baris yang terdapat dalam file eksternal tersebut. Parameter input fungsi ini adalah sebuah *string* yang berisi alamat file eksternal (source). Parameter output fungsi adalah jumlah baris (n).

1. **public static int CountColumn**

Fungsi CountColumn adalah fungsi yang membaca file eksternal, menghitung jumlah kolom pada file eksternal, dan mengeluarkan jumlah kolom yang terdapat dalam file eksternal tersebut. Parameter input fungsi ini adalah sebuah *string* yang berisi alamat file eksternal (source). Parameter output fungsi adalah jumlah kolom (n).

1. **public static void MatriksDariFile**

Prosedur MatriksDariFile adalah prosedur yang berfungsi membaca sejumlah nilai dari file eksternal dan memasukkannya ke dalam variabel matriks *augmented*. Spesifikasi prosedur berupa jumlah baris (m) dan jumlah kolom (n) yang dihitung menggunakan fungsi CountRow dan CountColumn, matriks itu sendiri (matriks), kolom *augmented* matriks (aug), dan alamat file eksternal (source).

1. **public static void MatriksKeLayar**

Prosedur MatriksKeLayar adalah prosedur yang berfungsi mencetak isi variabel matriks *augmented* ke layar. Spesifikasi prosedur antara lain jumlah baris (m), jumlah kolom (n), matriks itu sendiri (matriks), dan kolom *augmented* matriks (aug).

1. **public static void MatriksKeFile**

Prosedur MatriksKeFile adalah prosedur yang berfungsi mencetak isi variabel matriks *augmented* ke suatu file eksternal. Spesifikasi prosedur antara lain jumlah baris (m), jumlah kolom (n), matriks itu sendiri (matriks), kolom *augmented* matriks (aug), dan alamat file eksternal (source).

1. **public static void InterpolasiDariLayar**

Prosedur InterpolasiDariLayar adalah prosedur yang berfungsi menerima masukan pengguna dan memasukkannya ke dalam variabel *array* dua dimensi interpolasi. Spesifikasi prosedur berupa jumlah baris (m), *array* interpolasi (interpolasi), dan Scanner yang digunakan untuk input pengguna (scanner).

1. **public static void InterpolasiDariFile**

Prosedur InterpolasiDariFile adalah prosedur yang berfungsi membaca sejumlah absis dan ordinat titik dari file eksternal dan memasukkannya ke dalam variabel *array* dua dimensi interpolasi. Spesifikasi prosedur berupa jumlah baris (m) yang dihitung menggunakan fungsi CountRow, *array* interpolasi (interpolasi), dan alamat file eksternal (source).

1. **public static void Matrikskan**

Prosedur Matrikskan adalah prosedur yang berfungsi membaca *array* absis dan ordinat titik interpolasi dan menggunakan nilai-nilai tersebut untuk membentuk sebuah matriks *augmented* yang merepresentasikan persamaan interpolasi titik-titik tersebut. Parameter prosedur adalah jumlah baris (m), jumlah kolom (n), matriks *augmented* yang akan dibuat (matriks), kolom *augmented* matriks (aug), dan *array* interpolasi (interpolasi).

1. **public static void Gauss**

Prosedur Gauss adalah prosedur yang berfungsi mengubah matriks *augmented* menjadi sebuah matriks segitiga atas dengan menggunakan metode eliminasi Gauss. Parameter prosedur adalah matriks augmented (matriks), kolom augmented (aug), jumlah baris (m), jumlah kolom(n).

1. **public static void GaussJordan**

Prosedur GaussJordan adalah prosedur yang berfungsi mengubah matriks *augmented* menjadi sebuah matriks yang hanya mengandung satu nilai bukan nol pada setiap baris efektifnya dengan menggunakan metode eliminasi Gauss-Jordan. Parameter prosedur adalah matriks augmented (matriks), kolom augmented (aug), jumlah baris (m), jumlah kolom(n).

1. **public static void CekSolusi**

Prosedur CekSolusi adalah prosedur yang berfungsi menentukan jenis solusi matriks *augmented*, yaitu antara solusi tunggal, tidak ada solusi, atau solusi banyak. Prosedur kemudian mengisikan solusi ke dalam sebuah *array* solusi (solusi). Parameter prosedur adalah matriks augmented (matriks), kolom augmented (aug), jumlah baris (m), jumlah kolom(n).

1. **public static double SolusiInterpolasi**

Fungsi SolusiInterpolasi adalah fungsi yang memasukkan absis dari pengguna ke dalam persamaan interpolasi yang didapatkan dari solusi matriks *augmented*, dan kemudian mengeluarkan hasil berupa ordinat dari titik yang dicari. Parameter input fungsi adalah array koefisien interpolasi (persamaan) dan absis dari titik yang dicari (x). Parameter output fungsi adalah ordinat titik yang dicari (y).

1. **public static main**

Kelas main adalah fungsi yang berlaku sebagai program utama. Pada kelas ini terdapat inisiasi dan deklarasi variabel “global” program.

**C. Alur Kerja Program**

Pada menu pertama, pengguna diminta memilih antara sistem persamaan linier, interpolasi, atau keluar dari program. Jika pengguna memilih keluar dari program, program berakhir. Jika tidak, maka muncul menu kedua.

Pada menu kedua, pengguna diminta memilih sumber input, antara input pengguna atau file eksternal.

Terdapat dua klausa *if-then* yang dieksekusi berdasarkan pilihan pengguna:

* Jika pengguna memilih sistem persamaan linier, pengguna diminta memasukkan spesifikasi matriks *augmented*, dan matriks akan dibuat oleh program.
* Jika pengguna memilih interpolasi, pengguna diminta memasukkan titik-titik persamaan interpolasi, serta absis dari titik yang akand dicari, dan *array* interpolasi akan dibuat oleh program. Program kemudian mengubah *array* tersebut menjadi sebuah matriks *augmented*.

Pada menu ketiga, pengguna diminta memilih metode penyelesaian matriks *augmented*, yaitu Gauss atau Gauss-Jordan. Program kemudian menyelesaikan matriks sesuai metode yang dipilih dan menampilkan hasil penyelesaian.

Jika pengguna memilih interpolasi, program akan memasukkan absis dari pengguna ke dalam persamaan interpolasi yang didapat dari hasil penyelesaian matriks *augmented* dan menampilkan hasil berupa ordinat titik yang dicari.

**BAB IV**

**EKSPERIMEN**

1. x1=-4\*x4 - 2\*x5 - 3\*x2

x2= a

x3=-2\*x4

x4=b

x5=c

x6=1/3

1. x1 = -1+b

x2 = 2\*a

x3 = a

x4 = b

1. x1=-1+b

x2=2\*a

x3=a

x4=b

1. Tidak ada solusi
2. x1 = 500.02200096804256

x2 = -433.0190528383249

x3 = 866.0381056766498

x4 = 4.8074772256248556e-14

x5 = -250.01100048402128

x6 = -249.99633317199292

**BAB V**

**KESIMPULAN, REFLEKSI, DAN SARAN**

1. **Kesimpulan**

Berdasarkan pemrograman pada tugas besar yang kami buat, Sistem Persamaan Linear (SPL) dapat diselesaikan dengan beberapa cara. Beberapa diantaranya digunakan dalam pembuatan program ini yaitu Eliminasi Gauss dan Eliminasi Gauss Jordan. Kedua metode tersebut dapat membentuk bentuk matriks eselon dan bentuks matriks eselon tereduksi. Dengan kemudian dilanjutkan dengan sistem back subtitution dapat menghasilkan penyelesaian dari SPL.

Kasus Interpolasi juga dapat diselesaikan dengan menggunakan beberapa diantaranya yaitu metode eliminasi Gauss dan metode eliminasi Gauss-Jordan dimana setelah didapatkan persamaan yang akan memenuhi kondisi titik-titik tersebut maka jika diberikan suatu absis maka kita dapat mencari hasil atau ordinatnya.

1. **Refleksi**

Berdasarkan proses pengerjaan tugas besar aljabar geometri ini, kami sebagai satu kelompok mengakui bahwa kami sangat kurang dalam memanajemen waktu dan menentukan prioritas dalam menyelesaikan permasalahan. Sehingga sampai batas akhir kami masih berkutat menyelesaikan tugas ini. Kami juga berharap setelah pengerjaan tugas ini kami bisa memperbaiki kesalahan kami dan bekerja lebih baik pada tugas-tugas selanjutnya.

1. **Saran**

Saran pengembangan yang dapat kami berikan yaitu adalah Semoga tugas ini bukan hanya tugas yang dikerjakan tanpa menambah pengetahuan namun dapat juga meningkatkan rasa keingintahuan kami untuk menggali lebih dalam lagi mengenai hal yang terkait tugas ini.

**DAFTAR REFERENSI**