LAPORAN TUGAS BESAR I
IF2123 ALJABAR GEOMETRI

Disusun oleh :

Putu Gde Aditya Taguh Widiana (13517032)

Bram Musuko Panjaitan (13517089)

Louis Cahyadi (13517126)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2018

DAFTAR ISI

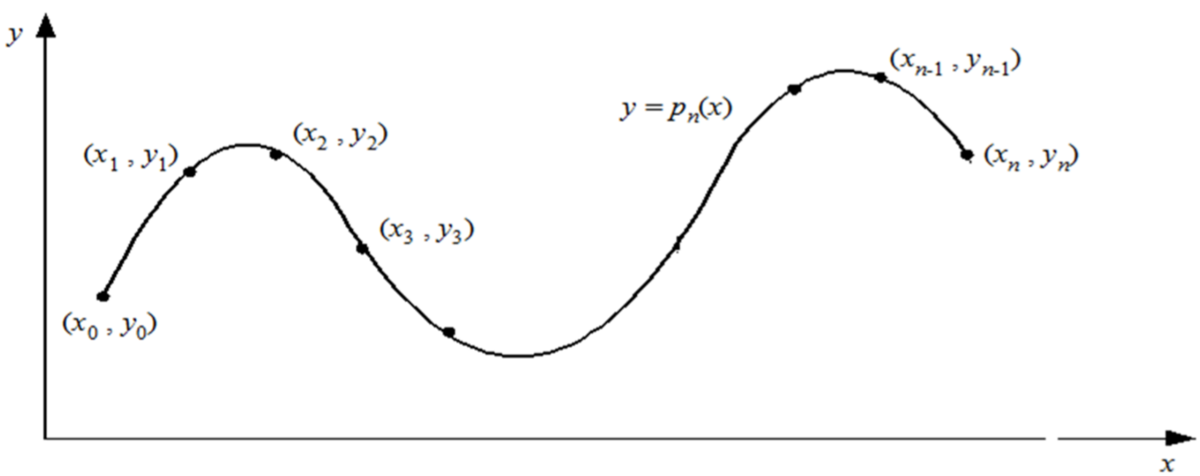
BAB 1. DESKRIPSI MASALAH

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

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\
\quad\vdots \qquad\qquad\qquad \vdots \\
\quad\vdots \qquad\qquad\qquad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m
\end{array}
$$

yang dalam hal ini x_i adalah peubah, a_{ij} dan b_i 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, (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). Tentukan polinom p_n(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk $i = 0, 1, 2, \dots, n$.



Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang $[x_0, x_n]$.

Polinom interpolasi derajat n yang menginterplolasi titik-titik (x_0, y_0), (x_1, y_1), ..., (x_n, y_n). adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0, y_0), (x_1, y_1), dan (x_2, y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x_0, y_0), (x_1, y_1), (x_2, y_2), dan (x_3, y_3), polinom yang menginterpolasi keempat titik tersebut adalah $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, 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 (x_i, y_i) ke dalam persamaan polinom $p_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ untuk $i = 0, 1, 2, \ldots, n$, akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \ldots, a_n$,

$$a_0 + a_1x_0 + a_2x_0^2 + \ldots + a_n x_0^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_n x_1^n = y_1$$

$$\ldots \qquad \ldots$$

$$a_0 + a_1x_n + a_2x_n^2 + \ldots + a_n x_n^n = y_n$$

Solusi sistem persamaan lanjar ini, yaitu nilai $a_0, a_1, \ldots, a_n$, 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 $p_2(x) = a_0 + a_1x + a_2x^2$. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah

$$a_0 + 8.0a_1 + 64.00a_2 = 2.0794$$

$$a_0 + 9.0a_1 + 81.00a_2 = 2.1972$$

$$a_0 + 9.5a_1 + 90.25a_2 = 2.2513$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan $a_0 = 0.6762$, $a_1 = 0.2266$, dan $a_2 = -0.0064$. Polinom interpolasi yang melaluiketiga buah titik tersebut adalah $p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$. Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: $p_2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)^2 = 2.2192$.

BAB 2. TEORI SINGKAT

I. Metode Eleminasi Gauss

Sebuah matriks disebut ***row-echelon form*** jika memenuhi properti – properti berikut ini :

1. Jika sebuah baris tidak semuanya bernilai nol maka bilangan bukan nol pertama yang muncul harus bernilai 1, selanjutnya akan disebut ***leading 1***.
2. Jika terdapat baris yang seluruhnya terdiri dari nol maka baris tersebut diletakkan dibagian bawah matriks
3. Untuk setiap dua baris yang berurutan dan tidak semuanya bernilai nol maka ***leading 1*** dari baris yang bawah terletak lebih di sebelah kanan daripada ***leading 1*** dari baris yang atas.

Metode eleminasi Gauss ialah salah satu metode penyelesaian sistem persamaan linear yang ditulis dalam bentuk matriks augmented. Pertama-tama matriks augmented yang merepresentasikan sistem persamaan linear tersebut dirubah menjadi matriks ***row-echelon form*** dengan menggunakan operasi – operasi berikut :

1. Mengalikan sebuah baris dengan sebuah bilangan konstan tak nol.
2. Menukar dua buah baris.
3. Menjumlahkan kelipatan suatu baris ke baris yang lain.

Metode eliminasi sebuah matriks dapat diubah menjadi ***row-echelon form*** dapat digambarkan dengan langkah – langkah berikut :

1. Tentukan kolom paling kiri yang tidak seluruhnya bernilai nol
2. Lakukan penukaran pada baris paling atas dengan baris lainnya, jika diperlukan, untuk membawa bilangan bukan nol ke baris paling atas dari kolom yang ditemukan pada langkah pertama
3. Jika element matriks pada baris dan kolom pada dua langkah sebelumnya bernilai a maka kalikan baris pertama tersebut dengan 1/a untuk membuat ***leading 1***.
4. Jumlahkan kelipatan baris pertama yang sesuai dengan baris – baris dibawahnya supaya pada kolom tersebut, di baris – baris dibawahnya, setiap element bernilai nol.
5. Lakukan kembali langkah 1 – 4 untuk baris berikutnya.

Setelah matriks augmented menjadi matriks ***row-echelon form*** maka dapat diperoleh solusi dari sistem persamaan linear yang diminta dengan ***back substitution*** dengan langkah – langkah sebagai berikut :

1. Selesaikan persamaan – persamaan untuk setiap ***leading variable***
2. Dimulai dari persamaan terbawah dan bekerja ke atas, lakukan substitusi setiap persamaan ke persamaan – persamaan yang ada di atasnya.

II. Metode Eleminasi Gauss-Jordan

Sebuah matriks disebut ***reduced row-echelon form*** jika matriks tersebut ***row-echelon form*** dan memenuhi sebuah syarat tambahan yaitu setiap kolom yang memuat ***leading 1***, element – element yang lain bernilai nol. Untuk mendapatkannya lakukan langkah langkah seperti yang digunakan pada metode eleminasi Gauss ditambah pada bagian akhirnya langkah berikut :

Dimulai dari baris paling bawah dan bekerja ke atas, lakukan penjumlahan kelipatan dari setiap baris terhadap baris – baris yang ada diatasnya untuk membuat nilai diatas setiap ***leading 1*** bernilai nol.

Untuk mendapatkan penyelesaiannya metode eleminasi Gauss Jordan juga menggunakan ***back substitution*** yang digunakan pada metode eleminasi Gauss.

III. Pseudo-code

Merubah matriks M menjadi ***row-echelon form*** dengan progaram berikut ini

```
RowEchelonForm
i = 1
j = 1
while (i ≤ BanyakBaris and j < BanyakKolom)
        while (isKolNol(j,i)) //true jika pada kolom j dibawah M[i][j] semuanya bernilai nol
                j += 1
        if (j < Kolom)
                tukar baris i dengan indeks tak nol di kolom j dibawah i
                buat element M[i][j] menjadi leading 1
                buat element pada kolom j dibawah M[i][j] bernilai nol
                i += 1
                j += 1
```

Penyelesaian metode eleminasi Gauss memanfaatkan matriks Tampung untuk menampung penghitungan dari back substitution yang digunakan. Untuk menandakan variable X_k adalah variable bebas pada matriks Tampung[k][BanyakKol + 1] = -1.

```
Gauss

RowEchelonForm
for (i = BanyaknyaBaris; i ≥ 1; i--)
j = indeksPivot(i) //indeks dari leading 1 di baris i
for (k = indeksPivot(i + 1) – 1; k > indeksPivot(i); k--)
        Tampung[k][BanyakKolom] = -1
Tampung[j][BanyakKolom] = M[i][Kol]

i = BanyakBaris
while (i ≥ 1 dan lanjut = true)
        if (Semua element pada baris i nol kecuali pada kolom terakhir)
              lanjut = false
        else if (semua element pada baris i nol)
              i = i – 1
        else
              j = indeksPivot(i)
              for (k = BanyakKolom -1; k > j; k--)
                    if (X_k variabel bebas)
                          Tampung[j][k] = M[i][k]
                    Else
                          Tampung[j][Kol] = Tampung[j][Kol] -
    (Tampung[k][Kol])*(M[i][k]);
                          for (l = k + 1;l < Kol;l++)
                                Tampung[j][l] = Tampung[j][l] -
    (Tampung[k][l])*(M[i][k]);
If (lanjut) //ada solusi
        For (i = BanyakKol – 1; i ≥ 1; i--)
              If  (Tampung[i][BanyakKol + 1] = -1)
                    X_i variabel bebas
              Else
                    X_i  = Tampung[i][BanyakKolom]
                    For (j = i+1; j < BanyakKolom; j++)
                          If (Tampung[i][j] != 0)
                                Tulis - X_j di belakang hasil X_i
Else
        Tulis "Tidak ada solusi"
```

Merubah matriks M menjadi ***reduced row-echelon form*** dengan program berikut ini

```
ReducedRowEchelonForm

RowEchelonForm() //Membuat matriks M menjadi row-echelon form
While (i ≥ 1)
    While (is baris i nol semua)
        i += 1
    j = indeksPivot(i)
    buat semua element pada kolom j diatas M[i][j] bernilai nol
    i = i - 1
```

Mendapatkan solusi dengan metode eleminasi Gauss-Jordan dengan program berikut ini

GaussJordan

```
ReducedRowEchelonForm()
While (i ≥ 1 dan lanjut = true)
        If (Baris i nol semua kecuali kolom terakhir)
                lanjut = false
                Tulis “Tidak ada solusi”
        Else if (Baris i nol semua)
                i = i  - 1
        else
                j = indeksPivot(i)
                for (k = indeksPivot(i+1) – 1; k > indeksPivot(i); k++)
                        Xk variabel bebas
                Xj  = M[i][BanyakKolom] …..
                For (k = j + 1; k < BanyakKolom; k++)
                        If (M[i][k] != 0)
                                Tulis – M[i][k] Xk
        i = i - 1
```

BAB 3. IMPLEMENTASI PROGAM DALAM JAVA

Dalam tugas besar Aljabar Geometri ini, kelompok kami membagi file menjadi 2 bagian, file yg pertama bernama *matriks.java* dan file yang kedua bernama *DriverMatriks.java*. Tujuan kita membagi file ini menjadi 2 adalah untuk mempermudah proses pengerjaan saat kita membuat *code*. File *matriks.java* dibuat khusus untuk menyimpan segala fungsi yang dibutuhkan untuk melaksanakan proses pengerjaan, sedangkan file *DriverMatriks.java* dibuat khusus untuk menjadi program utama dan akan memanggil fungsi-fungsi dari file *matriks.java*.

File *matriks.java* mempunyai *class* yang bernama matriks. Di dalam *class* tersebut kita menyimpan beberapa fungsi dan prosedur yang memiliki kegunaan yang berbeda-beda, kita juga membuat 1 tipe bentukan yang bernama matriks, tipe data bentukan tersebut menyimpan array 2 dimensi, baris efektif, dan kolom efektif.

Fungsi dan prosedur yang kami gunakan di dalam *matriks.java* antara lain :

Prosedur :

1. bacaUkuranMatriks() = membaca ukuran matriks yang diinput dari keyboard
2. bacaUkuranMatriksInterpolasi() = membaca baris dari matriks interpolasi dari keyboard
3. bacaMatriks() = membaca isi matriks dari keyboard
4. bacaMatriksInterpolasi() = membaca isi matriks interpolasi dari keyboard
5. matriksInterpolasi() = mengubah matriks interpolasi yang diinput user menjadi matriks interpolasi yang siap untuk dicari solusinya
6. matriksInterpolasiExt() = mengubah matriks interpolasi yang dibaca dari file external menjadi matriks interpolasi yang siap untuk dicari solusinya
7. solusiInterpolasi() = menghasilkan nilai dari dengan metode Gauss f(x) ke layar dan ke file external
8. tulisMatriks() = menampilkan matriks ke layar
9. tukarBaris(int i, int j) = menukar baris i dan j pada matriks
10. buatLeadingPoint(int i, int j) = menghasilkan matriks yang memiliki leading point 1 pada baris i
11. buatKolomNolBawah(int j, int i) = Membuat kolom j nol dimulai dari baris ke i + 1
12. Gauss() = Merubah matriks menjadi Row-Echelon-Form
13. SolusiGauss() = menghasilkan penyelesaian dari matriks dengan metode Gauss ke layar dan ke file external
14. SolusiInterpolasiGauss() = menghasilkan nilai dari dengan metode Gauss (x) ke layar dan ke file external
15. buatKolomNolAtas(int i, int j) = Membuat kolom j berisi nol semua diatas indeks i
16. GaussJordan() = Merubah matriks menjadi Row-Echelon-Form tereduksi
17. SolusiGaussJordan() = menghasilkan penyelesaian dari matriks dengan metode Gauss-Jordan ke layar dan ke file external
18. bacaFileExtSPL() = membaca SPL dari file external
19. bacaFileExtInterpolasi() = membaca matriks interpolasi dari file external

fungsi :

1. float pangkat(float x,int i) = mengembalikan nilai x^i
2. boolean isKolNol(int i,int j) = true bila pada kolom j dimulai dari baris i sampai Brs = 0
3. int indeksTakNol(int j, int i) = Mengeluarkan indeks pertama yang tidak bernilai 0 dari matriks pada kolom j, dimulai dari baris ke i
4. boolean isNolSemua(int i) = Mengembalikan true jika dalam baris tersebut 0 semua kecuali kolom augmented
5. indeksPivot(int i) = Mengembalikan indeks pivot point pada baris i.Dengan asumsi bukan baris yang berisi 0 semua (isNolSemua = false)
6. isBarisNol(int i) = Menghasilkan true jika baris i bukan augmented isinya 0 semua

Fungsi dan prosedur yang kami gunakan di dalam *DriverMatriks.java* antara lain :

Prosedur :

1. tampilMenu() = untuk menampilkan menu utama pada layar dan membaca pilihan menu yang akan digunakan dari user
2. menuSPL() = untuk membaca metode yang akan digunakan dari user
3. pilihanInput() = untuk membaca cara menginput nilai matriks dari user

Fungsi : -

BAB 4. EKSPERIMEN

Berikut adalah hasil program kami dalam menangani kasus-kasus yang diberikan.

1. SPL berbentuk
   x1 + 3x2 - 2x3 + 2x5 = 0
   2x1 + 6x2– 5x3 - 2x4 + 4x5 - 3x6 = -1
   5x3 + 10x4 + 15x6 = 5
   2x1 + 6x2 + 8x4 + 4x4 + 18x6 = 6

   Berikut adalah tampilan keluaran dari program kami :

```
Masukan banyaknya baris : 4
Masukan banyaknya kolom : 7
1 3 -2 0 2 0 0
2 6 -5 -2 4 -3 -1
0 0 5 10 0 15 5
2 6 0 8 4 18 6

1.0 3.0 0.0 4.0 2.0 0.0 0.0
0.0 0.0 1.0 2.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 1.0 0.33333334
0.0 0.0 0.0 0.0 0.0 0.0 0.0

x6 = 0.33333334
x5 variabel bebas
x4 variabel bebas
x3 = 0.0 -(2.0)x4
x2 variabel bebas
x1 = 0.0 -(3.0)x2 -(4.0)x4 -(2.0)x5
```

2. SPL berbentuk matriks augmented

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

   Berikut adalah tampilan keluaran dari program kami :

```
Masukan banyaknya baris : 4
Masukan banyaknya kolom : 5
1 -1 2 -1 -1
2 1 -2 -2 -2
-1 2 -4 1 1
3 0 0 -3 -3

1.0 -1.0 2.0 -1.0 -1.0
0.0 1.0 -2.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0

x4 variabel bebas
x3 variabel bebas
x2 = 0.0 -(-2.0)x3
x1 = -1.0 -(-1.0)x4
```

3. SPL berbentuk matriks augmented

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

Berikut adalah tampilan keluaran dari program kami :

```
Masukan banyaknya baris : 6
Masukan banyaknya kolom : 5
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

1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 2.0
0.0 0.0 1.0 0.0 1.0
0.0 0.0 0.0 1.0 1.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0

x4 = 1.0
x3 = 1.0
x2 = 2.0
x1 = 0.0
```

4. SPL berbentuk

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

Berikut adalah tampilan keluaran dari program kami :

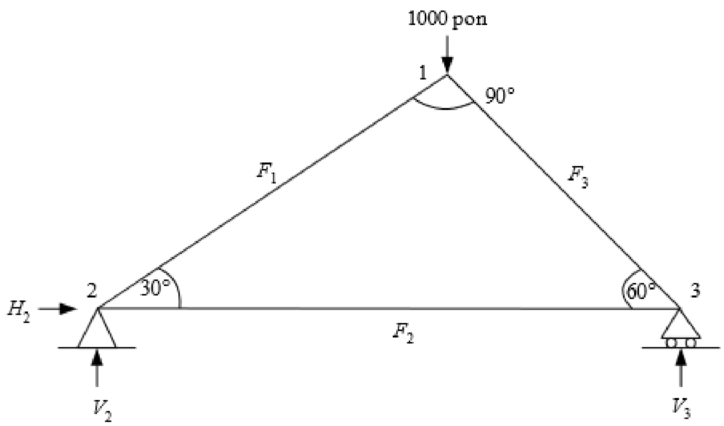
```
1
Masukan banyaknya baris : 12
Masukan banyaknya kolom : 10
0 0 0 0 0 0 1 1 1 13
0 0 0 1 1 1 0 0 0 15
1 1 1 0 0 0 0 0 0 8
0 0 0.04289 0 0.04289 0.75 0.04289 0.75 0.61396 14.79
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0 14.31
0.61396 0.75 0.042890 0.75 0.04289 0 0.04289 0 0 3.81
0 0 1 0 0 1 0 0 1 18
0 1 0 0 1 0 0 1 0 12
1 0 0 1 0 0 1 0 0 6
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.04289 10.51
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.91421 16.13
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.04289 7.04

1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0
0.0 1.0 3.65684 1.0 3.65684 1.0 3.65684 1.0 0.0 57.24
0.0 0.0 1.0 0.0 1.0 17.486593 1.0 17.486593 14.314759 344.8356
0.0 0.0 0.0 1.0 1.0 1.0 0.0 0.0 0.0 15.0
0.0 0.0 0.0 0.0 1.0 16.486593 1.0 17.486593 13.314759 326.8356
0.0 0.0 0.0 0.0 0.0 1.0 0.034232058 1.034232 0.85284543 19.535557
0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 13.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 900926.8 4505418.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 5.000844
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 15.5
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.01064682
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -15.5

Tidak ada solusi
```

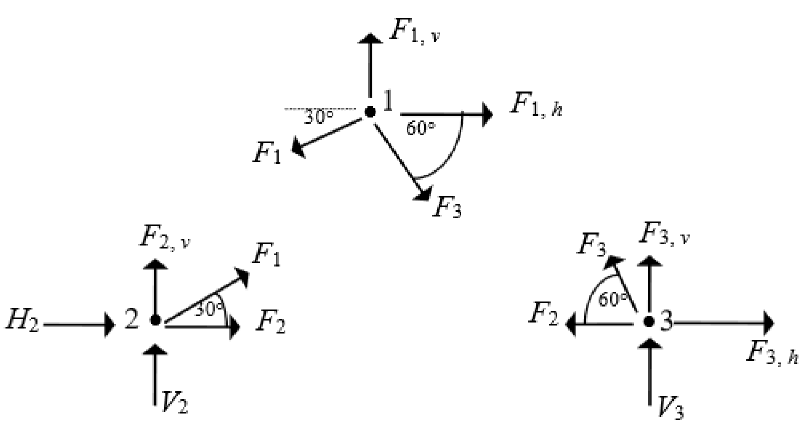
5. Misalkan seorang insinyur Teknik Sipil merancang sebuah rangka statis yang berbentuk segitiga (Gambar 1). Ujung segitiga yang bersudut 30□ bertumpu pada sebuah penyangga statis, sedangkan ujung segitiga yang lain bertumpu pada penyangga beroda.

   Rangka mendapat gaya eksternal sebesar 1000 pon. Gaya ini disebar ke seluruh bagian rangka. Gaya F menyatakan tegangan atau kompresi pada anggota rangka. Reaksi eksternal (H2, V2, dan V3) adalah gaya yang mencirikan bagaimana rangka berinteraksi dengan permukaan pendukung. Engsel pada simpul 2 dapat menjangkitkan gaya mendatar dan tegak pada permukaan, sedangkan gelinding pada simpul 3 hanya menjangkitkan gaya tegak.



Gambar 1 Gaya-gaya pada rangka statis tertentu

   Struktur jenis ini dapat diuraikan sebagai sistem persamaan aljabar lanjar simultan. Diagram gaya-benda-bebas diperlihatkan untuk tiap simpul dalam Gambar 2.



Gambar 2 Diagram gaya-benda-bebas untuk simpul-simpul rangka statis

Menurut hukum Newton, resultan gaya dalam arah mendatar maupun tegak harus nol pada tiap simpul, karena sistem dalam keadaan diam (statis). Oleh karena itu, untuk simpul 1,

Σ FH = 0 = -F1 cos 30° + F3 cos 60° + F1, h

Σ FV = 0 = -F1 sin 30° - F3 sin 60° + F1, v

untuk simpul 2,

Σ FH = 0 = F2 + F1 cos 30° + F2, h + H2

Σ FV = 0 = F1 sin 30° - F2, v + V2

dan untuk simpul 3,

Σ FH = 0 = -F2 - F3 cos 60° + F3, h

Σ FV = 0 = F3 sin 60° + F3, v + V3

Gaya 1000 pon ke bawah pada simpul 1 berpadanan dengan F1, v = -1000, sedangkan semua Fi, v dan Fi, h lainnya adalah nol. Persoalan rangka statis ini dapat dituliskan sebagai sistem yang disusun oleh enam persamaan lanjar dengan 6 peubah yang tidak diketahui:

Σ FH = 0 = -F1 cos 30° + F3 cos 60° + F1, h = -0.866F1 + 0.5 F3

Σ FV = 0 = -F1 sin 30° - F3 sin 60° + F1, v = -0.5F1 – 0.866 F3 + 1000

Σ FH = 0 = F2 + F1 cos 30° + F2, h + H2 = F2 + 0.866F1 + 0 + H2

Σ FV = 0 = F1 sin 30° - F2, v + V2 = 0.5 F1 + V2

Σ FH = 0 = -F2 - F3 cos 60° + F3, h = -F2 – 0.5 F3

Σ FV = 0 = F3 sin 60° + F3, v + V3 = 0.866 F3 + V3

Keenam persamaan di atas ditulis ulang kembali dalam susunan yang teratur berdasarkan urutan peubah F1, F2, F3, H2, V2, V3:

$$
\begin{array}{rrrrrrl}
-0.866F_1 & & + \ 0.5\,F_3 & & & & = 0 \\
-0.5F_1 & & & -0.866\,F_3 & & & = -1000 \\
-0.866F_1 & -F_2 & & & -H_2 & & = 0 \\
-0.5\,F_1 & & & & & -V_2 & = 0 \\
 & -F_2 & -0.5\,F_3 & & & & = 0 \\
 & & -0.866\,F_3 & & & -V_3 & = 0
\end{array}
$$

Tentukan solusi sistem di atas!

Keluaran dari program kami adalah :

F1 = 500.02197

F2 = -433.01904

F3 = 866.0381

H2 = 2.5809946E-5

V2 = -250.01099

V3 = -749.98895

Gambar hasil programnya bisa dilihat di bawah ini :

```
-0.866 0.0 0.5 0.0 0.0 0.0 0.0
-0.5 0.0 -0.866 0.0 0.0 0.0 -1000.0
-0.866 -1.0 0.0 -1.0 0.0 0.0 0.0
-0.5 0.0 0.0 0.0 -1.0 0.0 0.0
0.0 -1.0 -0.5 0.0 0.0 0.0 0.0
0.0 0.0 -0.866 0.0 0.0 -1.0 0.0

1.0 0.0 0.0 0.0 0.0 0.0 500.02197
0.0 1.0 0.0 0.0 0.0 0.0 -433.01904
0.0 0.0 1.0 0.0 0.0 0.0 866.0381
0.0 0.0 0.0 1.0 0.0 0.0 2.5809946E-5
0.0 0.0 0.0 0.0 1.0 0.0 -250.01099
0.0 0.0 0.0 0.0 0.0 1.0 -749.98895

x6 = -749.98895
x5 = -250.01099
x4 = 2.5809946E-5
x3 = 866.0381
x2 = -433.01904
x1 = 500.02197
```

6. (Interpolasi) 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) = ?

Matriks keluaran dari program kami adalah :

```
1.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.022976523
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.23999915
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.19740225
0.0 0.0 0.0 1.0 0.0 0.0 0.0 -2.1718442E-5
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.026077244
0.0 0.0 0.0 0.0 0.0 1.0 0.0 -2.7132126E-5
0.0 0.0 0.0 0.0 0.0 0.0 1.0 7.701633E-6
```

Berikut adalah hasil untuk tiap-tiap nilai x yang diuji :

a. x = 2

```
Masukkan nilai x = 0.2
f(x) = -0.022976523 + 0.23999915x + 0.19740225x^2 + (-2.1718442E-5)x^3 + 0.026077244x^4 + (-2.7132126E-5)x^5 + 7.701633E-6x^6
f(0.2) = 0.032960936
```

b. x = 0.55

```
Masukkan nilai x = 0.55
f(x) = -0.022976523 + 0.23999915x + 0.19740225x^2 + (-2.1718442E-5)x^3 + 0.026077244x^4 + (-2.7132126E-5)x^5 + 7.701633E-6x^6
f(0.55) = 0.17111866
```

c. x = 0.85

```
Masukkan nilai x = 0.85
f(x) = -0.022976523 + 0.23999915x + 0.19740225x^2 + (-2.1718442E-5)x^3 + 0.026077244x^4 + (-2.7132126E-5)x^5 + 7.701633E-6x^6
f(0.85) = 0.33723587
```

d. x = 1.28

```
Masukkan nilai x = 1.28
f(x) = -0.022976523 + 0.23999915x + 0.19740225x^2 + (-2.1718442E-5)x^3 + 0.026077244x^4 + (-2.7132126E-5)x^5 + 7.701633E-6x^6
f(1.28) = 0.6775419
```

7. (Interpolasi) Harga rumah baru dari tahun 1950 hingga 1969 mengalami perubahan yang tercatat sebagai berikut :

Tahun	Harga ($ juta)
1950	33,525
1955	46,519
1960	53,941
1965	72,319
1966	75,160
1967	76,160
1968	84,690
1969	90,866

Berdasarkan data tersebut prediksilah harga rumah baru pada tahun 1957, 1964, 1970, 1975 (atau nilai lain sesuai masukan user) dengan menggunakan polinom interpolasi.

Keluaran matriks dari program kami :

```
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.27816704E8
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 -565377.2
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 362.05377
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 -0.13254866
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 4.330479E-5
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 -3.4125378E-9
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 -5.7843296E-12
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.5752768E-15
```

a. Tahun : 1957

```
Masukkan nilai x = 1957
f(x) = 3.27816704E8 + -565377.2x + 362.05377x^2 + -0.13254866x^3 + 4.330479E-5x^4 + -3.4125378E-9x^5 + -5.7843296E-12x^6 + 1.5752768E-15x^7
f(1957.0) = 48.0
```

b. Tahun : 1964

```
Masukkan nilai x = 1964
f(x) = 3.27816704E8 + -565377.2x + 362.05377x^2 + -0.13254866x^3 + 4.330479E-5x^4 + -3.4125378E-9x^5 + -5.7843296E-12x^6 + 1.5752768E-15x^7
f(1964.0) = 80.0
```

c. Tahun : 1970

```
Masukkan nilai x = 1970
f(x) = 3.27816704E8 + -565377.2x + 362.05377x^2 + -0.13254866x^3 + 4.330479E-5x^4 + -3.4125378E-9x^5 + -5.7843296E-12x^6 + 1.5752768E-15x^7
f(1970.0) = 64.0
```

d. Tahun : 1975

```
Masukkan nilai x = 1975
f(x) = 3.27816704E8 + -565377.2x + 362.05377x^2 + -0.13254866x^3 + 4.330479E-5x^4 + -3.4125378E-9x^5 + -5.7843296E-12x^6 + 1.5752768E-15x^7
f(1975.0) = 176.0
```

8. Sederhanakan fungsi $y = 1/(\sqrt{}(x^2 + 2)$ dengan polinom interpolasi derajat n di dalam selang [1, 5]. Sebagai contoh, jika n = 5, maka titik-titik x yang diambil di dalam selang [1, 5] berjarak h = (5 – 1)/5.

   Kami memilih n = 5 dengan titik-titik sebagai berikut :

   a. (1.0 , 0.577)
   b. (1.8 , 0.436)
   c. (2.6 , 0.337)
   d. (3.4 , 0.271)
   e. (4.2, 0.225)
   f. (5.0, 0.192)

Hasilnya adalah :

```
Masukkan nilai x = 1
f(x) = 0.8025519 + -0.23823737x + -8.855611E-4x^2 + 0.017029895x^3 + -0.0037131822x^4 + 2.5432833E-4x^5
f(1.0) = 0.577
```

Fungsi yang merupakan keluaran dari program kami adalah :

$F(x) = 0.8025519 - 0.23823737x - 8.855611(10^{-4})x^2 + 0.017029895x^3 + 0.0037131822x^4 + 2.5432833(10^{-4})x^5$

BAB 5 : KESIMPULAN, SARAN, DAN REFLEKSI

Kesimpulan

Kami sudah membuat program untuk menyelesaikan SPL dan membuat interpolasi grafik dari titik-titik yang ada. Kami melihat bahwa penyelesaian SPL dan interpolasi grafik sangat berguna dan dapat diaplikasikan dalam berbagai bidang di kehidupan sehari hari, mulai dari hal yang sederhana sampai hal yang paling kompleks.

Saran

Kami berharap untuk tugas-tugas kedepannya dapat seperti tugas ini yang memiliki aplikasi yang sangat luas pada kehidupan sehari-hari, sehingga apa yang nantinya kami buat dapat berguna dan dapat menjadi solusi bagi permasalahan banyak orang.

Refleksi

Dalam membuat tugas ini, kami mengalami beberapa hambatan. Yang pertama adalah kurangnya pemahaman kami terhadap paradigma pemrograman berorientasi objek, seperti yang digunakan pada java, sehingga kami cukup sulit dalam mengerti dan memahami cara kerja dari program kami.

Hambatan yang kedua adalah kurangnya pengetahuan kami mengenai bahasa pemrograman yang digunakan, yaitu bahasa Java, sehingga pada beberapa tahapan seperti berurusan dengan file eksternal menjadi sedikit sulit.

Kami tau bahwa masih banyak kekurangan dalam kemampuan kami, oleh karena itu kami akan melakukan usaha-usaha yang terbaik sehingga segala sesuatu yang menjadi hambatan bagi kami saat ini tidak lagi menjadi hambatan bagi kami pada kemudian hari.

DAFTAR REFERENSI

https://stackoverflow.com/

https://en.wikipedia.org/wiki/Gaussian_elimination

https://en.wikibooks.org/wiki/Linear_Algebra/Gauss-Jordan_Reduction

https://www.mkyong.com/java/how-to-read-file-from-java-bufferedreader-example/

https://www.javatpoint.com/java-bufferedreader-class

https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html

https://www.programcreek.com/2011/03/java-write-to-a-file-code-example/