

SISTEM PERSAMAAN LINEAR SIMULTAN

METODE INVERS

Oleh

Muh. Firdaus

F1D020054

1. Tujuan

Tujuan dari dilakukan pratikum ini adalah sebagai berikut:

- Untuk mengetahui prinsip kerja metode invers penyelesaian SPLS
- Untuk mengetahui prinsip penukaran baris dan kolom dalam penyelesaian SPLS serta pengaruhnya terhadap solusinya.

2. Algoritma Penyelesaian

```
import java.text.DecimalFormat;
public class Metode_InverseFinal{
    static DecimalFormat blkJoma = new DecimalFormat("0.0000");
    static int M_Row = 3;
    static int N_Collom = 2;

    public static double[][] invert(double a[][]){
        int n = a.length;
        double x[][] = new double[n][n];
        double b[][] = new double[n][n];
        int index[] = new int[n];
        for (int i=0; i<n; ++i)
            b[i][i] = 1;

        //mengubah ke MSA
        eliminasi(a, index);

        for (int i=0; i<n-1; ++i)
            for (int j=i+1; j<n; ++j)
                for (int k=0; k<n; ++k)
                    b[index[j]][k]
                        -= a[index[j]][i]*b[index[i]][k];

        public static void eliminasi(double a[], int index[]) {
            int n = index.length;
            double c[] = new double[n];

            for (int i=0; i<n; ++i)
                index[i] = i;

            for (int i=0; i<n; ++i){
                double c1 = 0;
                for (int j=0; j<n; ++j){
                    double c0 = Math.abs(a[i][j]);
                    if (c0 > c1) c1 = c0;
                }
                c[i] = c1;
            }

            int k = 0;
            for (int j=0; j<n-1; ++j){
                double p1l = 0;
                for (int i=j; i<n; ++i) {
                    double pi0 = Math.abs(a[index[i]][j]);
                    pi0 /= c[index[i]];
                    if (pi0 > p1l){
                        p1l = pi0;
                        k = i;
                    }
                }
            }
        }
    }
}
```

```

public static double[][] kali(double[][] matrix_1, double[][] matrix_2){
    int baris_1 = matrix_1.length;
    int kolom_1 = matrix_1[0].length;

    int baris_2 = matrix_2.length;
    int kolom_2 = matrix_2[0].length;

    double[][] hasilKali = new double[baris_1][kolom_2];

    double buffer;
    for (int i = 0; i < baris_1; i++){
        for (int j = 0; j < kolom_2 ; j++){
            buffer = 0;
            for (int k = 0; k < kolom_1; k++){
                buffer += matrix_1[i][k] * matrix_2[k][j];
            }
            hasilKali[i][j] = buffer;
        }
    }
    return hasilKali;
}

public static void printMatrix (double[][] matrikInput){
    for (int i=0; i<matrikInput.length; ++i) {
        for (int j=0; j<matrikInput[0].length; ++j){
            System.out.print(blkKoma.format(matrikInput[i][j])+" ");
        }
        System.out.println();
    }
}

public static void main(String argv[]) {
    double A [][] = {
        {0,3,2,1,3,6,8,9,2,1},
        {2,4,2,1,3,2,1,4,5,7},
        {9,3,1,6,7,5,3,4,5,3},
        {5,3,4,1,8,9,5,4,3,2},
        {4,4,3,2,1,5,6,7,8,9},
        {6,5,3,8,9,6,7,5,4,3},
        {4,7,5,1,9,7,5,6,7,8},
        {5,3,9,8,6,1,2,4,5,6},
        {5,3,1,2,8,9,6,4,5,6},
        {4,3,2,5,7,9,0,6,4,3}
    };
    double B [][] = {
        {1},{-1},{1},{-1},{1},{-1},{1},{-1},{1},{-54},
    };
}

```

Source code lengkap -> [src/Metode_InverseFinal.java · main · Muh. Firdaus / Praktikum Metode Numerik · GitLab](#)

3. Hasil Uji Coba dan Analisa Singkat

Pada matriks, operasi pembagian matriks tidak didefinisikan, akan tetapi operasi matriks yang serupa dengan pembagian adalah matriks inversi. Bila A adalah MBS, maka matriks inversinya adalah A^{-1} , sedemikian sehingga: $AA^{-1} = A^{-1}A = I$, dengan I adalah matriks identitas. Selain itu matriks inversi dapat digunakan untuk menyelesaikan sistem yang berbentuk: $AX = C$ atau $A^{-1}C$

Nilai X dapat dihitung dengan mengalikan matriks inversi dari koefisien matriks A dengan ruas kanan dari sistem persamaan yaitu C. Metode Gauss-Jordan dapat digunakan untuk mencari matriks inversi, untuk itu koefisien matriks ditingkatkan dengan matriks identitas. Metode Gauss-Jordan dipakai untuk mereduksi koefisien matriks menjadi matriks identitas, setelah selesai, sisi kanan dari matriks yang ditingkatkan merupakan matriks inversi

```

Inverse Matrix A :
-0.0321  0.5381  0.0687  0.4254  -0.0271  0.0670  -0.3940  -0.0571  -0.1394  -0.1391
-0.1604  0.4636  -0.1741  0.3804  0.0771  0.3741  -0.1887  -0.2155  -0.3779  -0.0235
-0.0303  -0.0182  -0.0704  0.1577  0.0456  -0.0094  -0.0175  0.0705  -0.0964  -0.0113
-0.0524  -0.0339  -0.0901  -0.0637  0.0527  0.1476  -0.0656  0.0098  0.0003  0.0783
0.1155  -0.2699  0.1404  -0.3166  -0.1830  -0.1925  0.2847  0.1023  0.2209  -0.0021
-0.1122  0.0846  -0.1652  0.2106  0.1115  0.1266  -0.1540  -0.0766  -0.0627  0.0930
0.0274  -0.1214  -0.0157  -0.0430  0.0388  0.0456  0.0322  0.0077  0.0727  -0.0901
0.1867  0.2405  0.1152  0.0120  -0.1253  -0.1299  -0.1229  0.0428  0.0084  -0.0153
0.0252  -1.4913  0.2926  -0.9101  0.2073  -0.3824  1.0598  0.1025  0.1485  0.2177
-0.0291  0.7708  -0.1726  0.3500  -0.0923  0.1359  -0.5347  -0.0028  0.0852  -0.1144

Matrix B :      Matriks B Transpose * Inverse Matrix A =
1.0000      6.0165
-1.0000     -0.5580
1.0000      0.2400
-1.0000     -4.4411
1.0000      1.3670
-1.0000     -5.7476
1.0000      5.1300
-1.0000      0.7204
1.0000     -7.3409
-54.0000     4.1782

```

Gambar 2.1 Percobaan Pertama

Percobaan pada **Gambar 2.1** dilakukan dengan memberikan masukan matriks persegi A dengan ordo 10 dimana A(0,0) diisikan nilai 0. Oleh karena nilai A(0,0) memiliki nilai 0 dan berada pada diagonal matriks, maka akan dilakukan penukaran baris pertama dengan baris yang lain. Selanjutnya program akan melakukan perhitungan untuk mendapatkan *invers* dari matriks A dengan algoritma eliminasi Gauss-Jordan. Matriks kedua adalah matriks B diisi dengan nilai sesuai pada gambar dan khusus baris dan kolom terakhir matriks B diisi dengan dua digit NIM terakhir praktikan. Hasil dari inversi matriks A akan dikalikan dengan matriks B yang kemudian hasilnya ditunjukkan seperti pada gambar diatas.

```

Inverse Matrix A :
0.35 -0.62  0.47 -0.44 -0.20 -0.86  0.07  0.51  0.75 -0.09
0.18 -0.54  0.19 -0.41 -0.08 -0.43  0.21  0.26  0.42  0.02
-0.04  0.02 -0.08  0.18  0.05  0.02 -0.03  0.05 -0.12 -0.01
-0.07  0.04 -0.11 -0.02  0.06  0.20 -0.09 -0.03 -0.05  0.08
-0.08  0.31 -0.07  0.15 -0.09  0.28  0.05 -0.17 -0.25 -0.03
-0.04 -0.10 -0.09  0.04  0.08 -0.02 -0.08  0.00  0.10  0.10
-0.06  0.14 -0.11  0.15  0.08  0.26 -0.07 -0.12 -0.13 -0.10
0.35 -0.28  0.29 -0.36 -0.20 -0.54  0.08  0.30  0.40  0.01
-1.06  1.73 -0.85  1.58  0.70  2.21 -0.23 -1.45 -2.38  0.07
0.53 -0.89  0.42 -0.94 -0.35 -1.20  0.13  0.80  1.39 -0.04

Matrix B :      Matriks B Transpose * Inverse Matrix A =
1.00      7.51
-1.00     0.69
1.00      0.18
-1.00     -4.55
1.00      0.65
-1.00     -5.54
1.00      4.79
-1.00      1.40
1.00     -11.40
-54.00     6.27

```

Gambar 2.2 Percobaan Kedua

Percobaan pada **Gambar 2.2** dilakukan dengan memberikan masukan matriks persegi A dengan ordo 10 dimana nilai tiga angka digonal matriks yaitu A(0,0), A(1,1) dan A(2,2) diisikan nilai 0. Selain itu dilakukan perubahan angka dibelakang dengan maksimal 2 angka dibelakang koma. Hasil yang didapat akan seperti pada gambar di atas.

```

Inverse Matrix A :
0.35136569 -0.61973996 0.46742278 -0.44400164 -0.19693820 -0.86468262 0.06552129 0.51001810 0.75221986 -0.08618584
0.17946714 -0.53855513 0.18774756 -0.40950313 -0.08044515 -0.43353394 0.21447594 0.26196955 0.41803011 0.02499500
-0.04048034 0.01957783 -0.07851527 0.17509876 0.04806760 0.02065484 -0.03086570 0.04805135 -0.11840780 -0.01224160
-0.07311378 0.03732627 -0.10845899 -0.02403604 0.05922435 0.20443393 -0.09189627 -0.03001227 -0.04562276 0.07597946
-0.08391377 0.31425497 -0.07333419 0.15006976 -0.08945012 0.27842001 0.04883529 -0.17371692 -0.24704551 -0.03077492
-0.04478670 -0.10088929 -0.08872277 0.04303390 0.07629849 -0.02352107 -0.07627272 0.00411135 0.09836319 0.10344117
-0.05910107 0.13979614 -0.10560639 0.15304079 0.07709981 0.25579271 -0.07144802 -0.12027996 -0.12845335 -0.10200889
0.35466901 -0.27528275 0.28670361 -0.36169553 -0.19712312 -0.54452426 0.07979252 0.30026225 0.39686983 0.00737956
-1.05718091 1.72701809 -0.85035013 1.58386390 0.70125016 2.20977789 -0.22860334 -1.44616574 -2.37973966 0.06501697
0.53128391 -0.89310628 0.41980496 -0.94285906 -0.34865542 -1.20445448 0.13198847 0.79669444 1.39458301 -0.03518233

Matrix B :      Matriks B Transpose * Inverse Matrix A =
1.00000000      7.51203296
-1.00000000     0.68916807
1.00000000     0.17746236
-1.00000000    -4.55047023
1.00000000     0.64790958
-1.00000000    -5.54367831
1.00000000     4.79262119
-1.00000000     1.40365616
1.00000000    -11.40003437
-54.00000000     6.27257621

```

Gambar 2.3 Percobaan Ketiga

Percobaan pada **Gambar 2.3** dilakukan dengan memberikan masukan matriks A sama dengan nilai pada matriks A pada **Gambar 2.2** dengan ordo 10 dimana nilai tiga angka digonal matriks yaitu $A(0,0)$, $A(1,1)$ dan $A(2,2)$ diisikan nilai 0. Selain itu dilakukan perubahan angka dibelakang dengan maksimal angka dibelakang koma. Hasil yang didapat akan seperti pada gambar di atas.

4. Kesimpulan

- Dalam menyelesaikan persamaan linear simultan dapat menggunakan metode invers matriks. Proses invers matriks dilakukan dengan menggunakan algoritma eliminasi Gauss-Jordan. Dengan menggunakan algoritma ini, invers dari sebuah matriks persegi akan mudah diperoleh. Eliminasi Gauss-Jordan dilakukan untuk mengubah koefisien matriks sebelah kiri ruas persamaan menjadi matriks identitas, sementara ruas kanan persamaan yang ditingkatkan merupakan invers matriks.
- Penukaran baris dan kolom dari sebuah matriks untuk menyelesaikan persamaan linear simultan akan mempengaruhi terhadap solusi yang didapatkan. Solusi yang didapatkan bisa memiliki tepat satu solusi, banyak solusi maupun tanpa solusi.

Referensi

- [1] Vlandari, R. T., "Metode Numerik: Teori, Kasus, dan Aplikasi", Surabaya: Mavendra Pers, 2017.
- [2] Luknanto, J., "Metoda Numerik", Bahan kuliah Metoda Numerik Teknik Sipil UGM, Jogjakarta, 2001.
- [3] Atmika, A. I K., "Diktat Metode Numerik", Denpasar: Universitas Udayana, 2016.