

Penyelesaian Persamaan Linear Simultan Metode Eliminasi Gauss

Ahmad Zainudin, S.ST, M.T
Workshop Metode Numerik

2014

Persamaan Linier Simultan

- **Persamaan linier simultan** adalah suatu bentuk persamaan-persamaan yang secara bersama-sama menyajikan banyak variabel bebas
- Bentuk persamaan linier simultan dengan m persamaan dan n variabel bebas

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3$$

.....

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m$$

- a_{ij} untuk $i=1$ s/d m dan $j=1$ s/d n adalah koefisien atau persamaan simultan
- x_j untuk $i=1$ s/d n adalah variabel bebas pada persamaan simultan

Persamaan Linier Simultan

- Penyelesaian persamaan linier simultan adalah penentuan nilai x_i untuk semua $i=1$ s/d n yang memenuhi semua persamaan yang diberikan.

$$AX = B$$

Matrik **A** = Matrik Koefisien/ Jacobian.

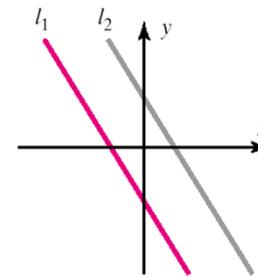
Vektor **x** = vektor variabel

vektor **B** = vektor konstanta.

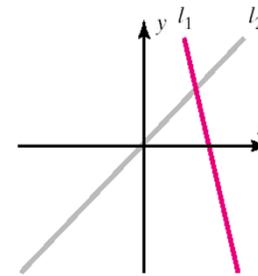
$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Persamaan Linier Simultan

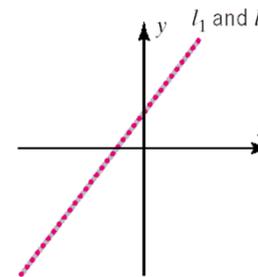
- Persamaan Linier Simultan atau Sistem Persamaan Linier mempunyai kemungkinan solusi :
 - Tidak mempunyai solusi
 - Tepat satu solusi
 - Banyak solusi



(a) No solution



(b) One solution



(c) Infinitely many solutions

Figure 1.1.1

Augmented Matrix

- matrik yang merupakan perluasan matrik A dengan menambahkan vector B pada kolom terakhirnya, dan dituliskan:
- **Augmented (A) = [A B]**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} & b_m \end{bmatrix}$$

Konsep Metode Eliminasi Gauss

- Metode Eliminasi Gauss merupakan metode yang dikembangkan dari metode eliminasi, yaitu menghilangkan atau mengurangi jumlah variable sehingga dapat diperoleh nilai dari suatu variable bebas.

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right]$$

Konsep Metode Eliminasi Gauss

- Suatu metode dimana bentuk matrik di atas, pada bagian kiri diubah menjadi matrik segitiga atas atau segitiga bawah dengan menggunakan **OBE (Operasi Baris Elementer)**.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix} \longrightarrow \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1n} & d_1 \\ 0 & c_{22} & c_{23} & \dots & c_{2n} & d_2 \\ 0 & 0 & c_{33} & \dots & c_{3n} & d_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{nn} & d_n \end{bmatrix}$$

Konsep Metode Eliminasi Gauss

- Sehingga penyelesaian dapat diperoleh dengan :

$$x_n = \frac{d_n}{c_{nn}}$$

$$x_{n-1} = \frac{1}{c_{n-1,n-1}} (-c_{n-1,n}x_n + d_{n-1})$$

.....

$$x_2 = \frac{1}{c_{22}} (d_2 - c_{23}x_3 - c_{24}x_4 - \dots - c_{2n}x_n)$$

$$x_1 = \frac{1}{c_{11}} (d_1 - c_{12}x_2 - c_{13}x_3 - \dots - c_{1n}x_n)$$

Algoritma Metode Eliminasi Gauss

- (1) Masukkan matrik A, dan vektor B beserta ukurannya n
- (2) Buat augmented matrik $[A|B]$ namakan dengan A
- (3) Untuk baris ke i dimana $i=1$ s/d n, perhatikan apakah nilai $a_{i,i}$ sama dengan nol :

Bila ya :

pertukarkan baris ke i dan baris ke $i+k \leq n$, dimana $a_{i+k,i}$ tidak sama dengan nol, bila tidak ada berarti perhitungan tidak bisa dilanjutkan dan proses dihentikan dengan tanpa penyelesaian.

Bila tidak : lanjutkan

- (1) Untuk baris ke j, dimana $j = i+1$ s/d n

Lakukan operasi baris elementer:

◇ Hitung $c = \frac{a_{j,i}}{a_{i,i}}$

- ◇ Untuk kolom k dimana $k=1$ s/d $n+1$

hitung $a_{j,k} = a_{j,k} - c.a_{i,k}$

Algoritma Metode Eliminasi Gauss

(2) Hitung akar, untuk $i = n$ s/d 1 (bergerak dari baris ke n sampai baris pertama)

$$x_i = \frac{1}{a_{i,i}} (b_i - a_{i,i+1}x_{i+1} - a_{i,i+2}x_{i+2} - \dots - a_{i,n}x_n)$$

dimana nilai $i+k \leq n$

Program Metode Eliminasi Gauss

- Tentukan Ordo dan Masukkan matrik A

```
printf("Tentukan ordo matrik : ");
scanf("%d",&n);
printf("Masukkan matrik A : \n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("Matrik A[%d][%d] : ",i+1,j+1);
        scanf("%lg",&A[i][j]);
    }
}
```

- Masukkan matrik B

```
printf("Masukkan matrik vektor B : \n");
for(i=0;i<n;i++){
    printf("Matrik B[%d] : ",i+1);
    scanf("%lg",&B[i]);
    A[i][j]=B[i];
}
```

Program Metode Eliminasi Gauss

- Buat augmented matrik [A|B]

```
printf("Augmented matrik [A|B] : \n");
for(i=0;i<n;i++){
    for(j=0;j<n+1;j++){
        printf("%4lg",A[i][j]);
    }
    printf("\n");
}
```

Program Metode Eliminasi Gauss

(1) Untuk baris ke j , dimana $j = i+1$ s/d n

Lakukan operasi baris elementer:

◇ Hitung $c = \frac{a_{j,i}}{a_{i,i}}$

◇ Untuk kolom k dimana $k=1$ s/d $n+1$

hitung $a_{j,k} = a_{j,k} - c.a_{i,k}$

```
for(i=0; i<n; i++)
{
    for(j=i+1; j<=n; j++)
    {
        c = A[j][i] / A[i][i];
        for(k=0; k<=n+1; k++)
        {
            A[j][k] = A[j][k] - c * A[i][k];
        }
    }
}
```

Program Metode Eliminasi Gauss

(2) Hitung akar, untuk $i = n$ s/d 1 (bergerak dari baris ke n sampai baris pertama)

$$x_i = \frac{1}{a_{i,i}} (b_i - a_{i,i+1}x_{i+1} - a_{i,i+2}x_{i+2} - \dots - a_{i,n}x_n)$$

dimana nilai $i+k \leq n$

```
for (i=0; i<n; i++)
{
    b[i] = A[i][n];
}
x[n-1] = b[n-1]/A[n-1][n-1];
for (k=n-2; k>=0; k--)
{
    sigma = 0;
    for (j=k+1; j < n ; j++)
    {
        sigma = sigma + A[k][j] * x[j];
    }
    x[k] = (b[k] - sigma) / A[k][k];
}
```

Program Metode Eliminasi Gauss

- Cetak Matrik setelah OBE

```
printf("Matrik setelah OBE : \n");
for(i=0; i<n; i++){
    for(j=0; j<n+1; j++){
        printf("%4lg", A[i][j]);
    }
    printf("\n");
}
```

- Cetak Penyelesaian Persamaan Simultan

```
printf("Penyelesaian persamaan simultan : \n");
for(i=0; i<n; i++){
    printf("x-%d = %lg\n", i+1, x[i]);
}
```

Pengujian Program (1)

Selesaikan sistem persamaan linier berikut :

$$x_1 + x_2 + x_3 = 6$$

$$x_1 + 2x_2 - x_3 = 2$$

$$2x_1 + x_2 + 2x_3 = 10$$



Pengujian Program

```
c:\Data\Workshop Metode Numerik\Program>gauss
```

```
Tentukan ordo matrik : 3
```

```
Masukkan matrik A :
```

```
Matrik A[1][1] : 1
```

```
Matrik A[1][2] : 1
```

```
Matrik A[1][3] : 1
```

```
Matrik A[2][1] : 1
```

```
Matrik A[2][2] : 2
```

```
Matrik A[2][3] : -1
```

```
Matrik A[3][1] : 2
```

```
Matrik A[3][2] : 1
```

```
Matrik A[3][3] : 2
```

```
Masukkan matrik vektor B :
```

```
Matrik B[1] : 6
```

```
Matrik B[2] : 2
```

```
Matrik B[3] : 10
```

```
Augmented matrik [A|B] :
```

```
1 1 1 6
```

```
1 2 -1 2
```

```
2 1 2 10
```

```
Matrik setelah OBE :
```

```
1 1 1 6
```

```
0 1 -2 -4
```

```
0 0 -2 -6
```

```
Penyelesaian persamaan simultan :
```

```
x-1 = 1
```

```
x-2 = 2
```

```
x-3 = 3
```

Pengujian Program (2)

- Angga anak Pak Purwoko memiliki setumpuk kartu. Keseluruhan kartu dapat dipilah menjadi dua bagian menurut bentuknya. Satu jenis berbentuk persegi yang di dalamnya terdapat gambar seekor kerbau dan empat ekor burung. Satu jenis lagi berbentuk segitiga yang di dalamnya terdapat gambar seekor kerbau dan dua ekor burung. Berapa banyak kartu persegi dan segitiga yang harus diambil dari tumpukan kartu agar jumlah gambar kerbau 33 dan jumlah gambar burung 100.



Pengujian Program (3)

Tentukan nilai a , b , c , dan d dari persamaan di bawah ini

$$-a + 2b - 3c + 4d = 20$$

$$4a - 3b + 2c - d = 0$$

$$2a - 2b - 2c + 2d = 0$$

$$5a + 4b - c - d = 12$$



Laporan Sementara

- Selesaikan pengujian program (1),(2) dan (3)
- Tuliskan persamaan dan hasil program

- **Tugas Laporan Resmi :**
Bandingkan dengan hasil perhitungan secara manual untuk masing-masing pengujian program