

# Pencarian Langkah Terbaik Penyelesaian Sistem Persamaan Linear dengan Algoritma Breadth First Search

Bhimantyo Pamungkas<sup>1</sup>, Jimmy Karisma Ramadhan<sup>2</sup>, Rangga Wisnu Adi Permana<sup>3</sup>

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung  
E-mail : [if14016@students.if.itb.ac.id](mailto:if14016@students.if.itb.ac.id)<sup>1</sup>, [if14025@students.if.itb.ac.id](mailto:if14025@students.if.itb.ac.id)<sup>2</sup>,  
[if14036@students.if.itb.ac.id](mailto:if14036@students.if.itb.ac.id)<sup>3</sup>

## Abstrak

Persoalan sistem persamaan linear adalah salah satu persoalan bidang Matematika yang banyak digunakan. Salah satu penyelesaian sistem persamaan linear bisa didapat dengan metode Operasi Baris Elementer (OBE) yaitu membuat persamaan-persamaan awal pada sistem persamaan menjadi matriks lalu merubahnya menjadi matriks eselon ataupun matriks eselon tereduksi. Terkadang untuk menyelesaikan persoalan ini dengan metode OBE kita kesulitan dalam memilih baris mana yang terlebih dulu kita operasikan. Sehingga terkadang langkahnya sangat panjang dan tidak efisien. Dengan algoritma pencarian melebar (Breadth First Search atau BFS) kita dapat mencari langkah terbaik dalam mencari penyelesaian sistem persamaan linear.

**Kata kunci:** Sistem Persamaan Linear, Operasi Baris Elementer (OBE), dan Breadth First Search (BFS)

## 1. Pendahuluan

Persoalan sistem persamaan linear banyak dipergunakan dalam berbagai persoalan matematika. Oleh karena itu penyelesaian persoalan ini secara cepat efektif dan efisien sangat dibutuhkan. Penyelesaian sistem persamaan linear bisa dilakukan dengan berbagai macam cara, diantaranya:

- Metode Crammer
- Metode Subtitusi dan Eliminasi
- Metode Operasi Baris Elementer

Setiap metode tentunya memiliki kelebihan dan kekurangan masing-masing. Namun kami akan mencoba menyelesaikan persoalan ini dengan metode Operasi Baris Elementer serta Eliminasi Gauss-Jordan dimana Operasi Baris Elementer adalah operasi yang digunakan untuk merubah bentuk matriks yang merepresentasikan sistem persamaan linear tanpa merubah solusi dari sistem persamaan linear, sedangkan Eliminasi Gauss-Jordan adalah sistem eliminasi matriks yang menggunakan Operasi Baris Elementer sehingga membentuk matriks dengan bentuk eselon tereduksi.

## 2. Operasi Baris Elementer

Operasi Baris Elementer (OBE) adalah salah satu metode penyelesaian sistem persamaan linear.

Langkah utama dari OBE adalah :

- Kalikan sebuah baris dengan sebuah konstanta tak nol
- Pertukarkan dua baris
- Tambahkan perkalian dari suatu baris ke baris lainnya

Sebelum melakukan OBE, maka kita perlu merubah persamaan-persamaan linear menjadi sebuah matriks yang diperbanyak. Contoh matriks yang diperbanyak untuk sistem persamaan

$$\begin{aligned}x_1 + x_2 + 2x_3 &= 9 \\2x_1 + 4x_2 - 3x_3 &= 1 \\3x_1 + 6x_2 - 5x_3 &= 0\end{aligned}$$

adalah

$$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 2 & 4 & -3 & 1 \\ 3 & 6 & -5 & 0 \end{bmatrix}$$

Setelah mendapatkan matriks yang diperbanyak, barulah OBE dapat diterapkan hingga didapat matriks eselon tereduksi dengan eliminasi Gauss-Jordan. Esselon tereduksi adalah bentuk matriks dimana ditiap baris memiliki satu angka 1 utama dimana kolom yang terdapat angka 1 utama hanya terdapat angka 0 selain angka 1 utama dan semua

angka sebelum angka 1 utama dimana terdapat angka 1 utama adalah 0.

Contoh dari matriks eselon tereduksi :

$$\begin{bmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 9 \\ 0 & 0 & 1 & 6 \end{bmatrix}$$

Berikut adalah contoh pembentukan matriks eselon menggunakan Eliminasi Gauss-Jordan:

1.  $\begin{bmatrix} 1 & 1 & 2 & 9 \\ 2 & 4 & -3 & 1 \\ 3 & 6 & -5 & 0 \end{bmatrix}$  Persamaan awal

2.  $\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 2 & -7 & -17 \\ 3 & 6 & -5 & 0 \end{bmatrix}$

3.  $\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 2 & -7 & -17 \\ 0 & 3 & -11 & -27 \end{bmatrix}$

4.  $\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 3 & -11 & -27 \end{bmatrix}$

5.  $\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 0 & -\frac{1}{2} & -\frac{3}{2} \end{bmatrix}$

6.  $\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 0 & 1 & 3 \end{bmatrix}$

7.  $\begin{bmatrix} 1 & 0 & \frac{11}{2} & \frac{35}{2} \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 0 & 1 & 3 \end{bmatrix}$

8.  $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$  Solusi

Didapat solusi yaitu  $x_1 = 1, x_2 = 2, x_3 = 3$

Terkadang kita mengalami kesulitan untuk menentukan baris mana yang terlebih dulu dioperasikan dengan eliminasi Gauss-Jordan. Sehingga walaupun solusi didapatkan, namun langkahnya bisa bervariasi. Untuk persoalan yang

sama bisa jadi langkah yang diambil menjadi sangat singkat atau bahkan sangat rumit dan berulang-ulang sehingga seakan-akan tidak ada solusi.

### 3. Algoritma Breadth First Search (BFS)

Algoritma BFS adalah salah satu jenis algoritma traversal di dalam graf. Secara umum langkah-langkah algoritma BFS adalah sebagai berikut :

- Mulai traversal dari simpul v
- Kunjungi simpul v
- Kunjungi simpul yang bertetangga dengan simpul v
- Kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya.
- Jika graf berbentuk pohon berakar, maka semua simpul pada aras  $d$  dikunjungi lebih dahulu sebelum simpul-simpul pada aras  $d + 1$ .

### 4. Pencarian Langkah Terbaik OBE dengan Algoritma BFS

Untuk mencari langkah terbaik OBE dengan BFS dibutuhkan struktur pohon yang memiliki atribut matriks dan penyimpan alamat *parent*.

Matriks digunakan untuk merepresentasikan SPL yang sedang diproses, elemen dari matriks adalah integer. Alamat *parent* digunakan untuk menyimpan urutan langkah ketika sudah mendapatkan solusi yang merupakan langkah terbaik dari OBE.

Pohon dibangun bersamaan dengan berjalannya algoritma BFS. Dan akan berhenti ketika mendapatkan solusi yang optimum. Jika pada kedalaman yang sama terdapat lebih dari satu solusi maka diambil solusi yang paling kiri pembaca.

Pohon mempunyai anak berdasarkan baris yang dapat dioperasikan.

Pseudocode :

```

procedure OBE (input M : Matriks input n,m : integer output
L : list of Matriks)
{
  Procedure untuk mencari langkah OBE terbaik dalam mencari
solusi SPL dengan menggunakan algoritma BFS.
  Masukan : M merupakan matriks yang menjadi representasi SPL
yang akan dicari solusinya matriks berukuran
[0..m,0..n].Masukan : n+1 adalah jumlah baris dari matriks
M.
  Masukan : m+1 adalah jumlah kolom dari matriks M.
  Keluaran : L merupakan List of Matriks yang merupakan
langkah terbaik dalam menyelesaikan SPL dengan metode OBE.
}
Deklarasi
i, j, k, l, r      : integer
q                 : antrian
P, Pt            : Pohon
Ma               : Matriks

procedure BuatAntrian(input/output q : antrian)

```

```

{
    Membuat antrian kosong.
}
Procedure MasukanKeAntrian(input/output q : antrian, input
P : Pohon)
{
    Memasukan Pohon P ke dalam antrian q.
}
procedure HapusAntrian(input/output q : antrian, output P :
Pohon)
{
    Menghapus P yang merupakan kepala dari antrian q.
}
function AntrianKosong(input q : antrian) → boolean
{
    Menghasilkan true jika antrian q kosong.
}
procedure BuatAkar(input/output P : Pohon, input Ma :
Matriks)
{
    Membuat Pohon P dengan Matriks Ma dan Parent
menunjuk ke null (akar tidak punya parent).
}
procedure BangunPohon(input/output P : Pohon, input Ma :
Matriks, input Pt : Pohon)
{
    Membuat Pohon P dengan Matriks Ma dan
Parent menunjuk ke alamat Pohon Pt.
}
function IsSolusi(input Ma : Matriks) → boolean
{
    true jika Matriks Ma tidak dapat di OBE lagi
ini termasuk solusi tunggal, solusi banyak
ataupun tidak mempunyai solusi
}

function getMatriks(input P : Pohon) → Matriks
{
    Mengembalikan Matriks dari Pohon P.
}
function IsBarisKosong(input Ma : Matriks, input i :
integer) → boolean
{
    true jika Matriks pada baris i semua elemen
selain kolom terakhir adalah 0.
}
function Parent(input P : Pohon) → Pohon
{
    Mengembalikan Pohon yang menjadi parent dari
pohon P.
}
procedure MasukDepan(input/output L : List of Pohon, input
M : Matriks)
{
    Memasukkan Matriks M kedalam list L sebagai
elemen paling depan dari list L.
}

Algoritma
BuatAntrian(q)
BuatAkar(P,M)
MasukKeAntrian(q,P)
while (not AntrianKosong(q)) and (IsSolusi(Ma)) do
    HapusAntrian(q, P)
    i ← 0
    while i ≤ n do
        Ma ← getMatriks(P)
        if not IsBarisKosong(Ma,i) then
            j ← 0
            while Ma[i,j] ≠ 0 do
                j ← j+1
            endwhile
            r ← i + 1
            while Ma[r,j] ≠ 0 and r ≠ i do
                if r = n then
                    r ← 0
                else
                    r ← r+1
                endif
            endwhile
            if Ma[i,j] ≠ 1 then
                for k ← j to m do
                    Ma[i,k] ← Ma[i,k] / Ma[i,j]
                endfor
            endif
            if r ≠ i then
                for l ← 0 to m do
                    Ma[r,l] ← Ma[r,l] -
                    (Ma[i,j]/Ma[r,j])
                endfor
            endif
            BangunPohon(Pt, Ma, P)
            MasukanKeAntrian(q, Pt)
        endif
        if IsSolusi(Ma) then
            i ← n+1
        else
            i ← i+1
        endif
    endwhile
}

```

```

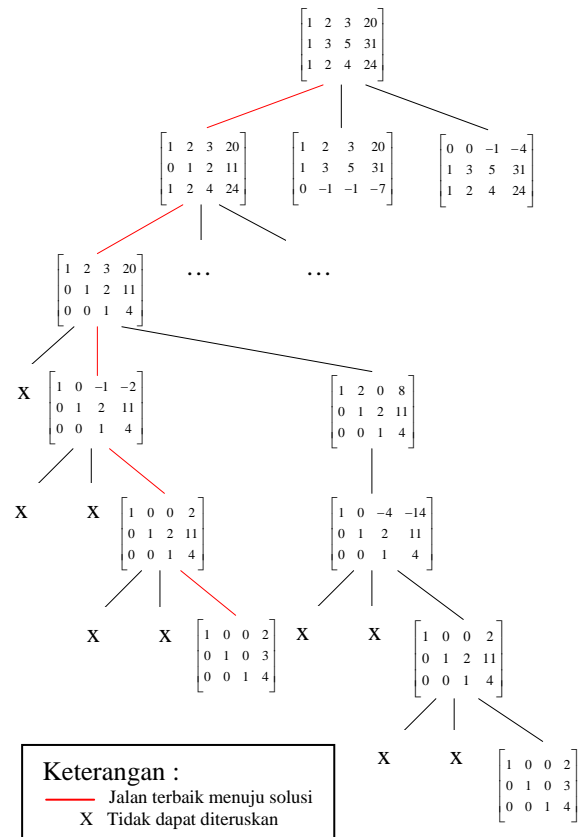
endif
endif
endfor
endwhile
while Parent(P) ≠ null do
    MasukDepan(L, Ma)
    P ← Parent(P)
    Ma ← getMatriks(P)
endwhile

```

Misal diberikan persamaan :

$$\begin{aligned}
 x_1 + 2x_2 + 3x_3 &= 20 \\
 x_1 + 3x_2 + 5x_3 &= 31 \\
 x_1 + 2x_2 + 4x_3 &= 24
 \end{aligned}$$

Pencarian langkah terbaik untuk solusi sistem persamaan linear di atas dapat dilihat pada gambar berikut :



Gambar 4.1 Sebagian dari pohon solusi

Dari gambar di atas dapat dilihat bahwa jalan terbaik adalah :

$$\begin{matrix}
 \begin{bmatrix} 1 & 2 & 3 & 20 \\ 1 & 3 & 5 & 31 \\ 1 & 2 & 4 & 24 \end{bmatrix} & \begin{bmatrix} 1 & 2 & 3 & 20 \\ 0 & 1 & 2 & 11 \\ 0 & 0 & 1 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 2 & 3 & 20 \\ 0 & 1 & 2 & 11 \\ 0 & 0 & 1 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 2 & 11 \\ 0 & 0 & 1 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 2 & 11 \\ 0 & 0 & 1 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 4 \end{bmatrix} \\
 0 & 1 & 2 & 3 & 4 & 5 \\
 & & & & & \text{(solusi)}
 \end{matrix}$$

## 5. Kesimpulan

Sistem persamaan linear dapat diselesaikan dengan berbagai cara diantaranya dengan merepresentasikan sistem persamaan linear kedalam bentuk matriks lalu melakukan OBE terhadap matriks lalu mencari solusi dengan Eliminasi Gauss-Jordan. Langkah-langkah pengoperasian matriks dengan OBE ini bisa bervariasi dan kerap membingungkan. Untuk mencari langkah yang terbaik menuju matriks solusi kita dapat memakai algoritma pencarian melebar (Breadth First Search) dengan menggunakan representasi pohon dengan matriks sebagai elemen dari pohon.

## Daftar Pustaka

1. Munir, Rinaldi, *Diktat Kuliah IF2251 Strategi Algoritmik*, Penerbit ITB, Bandung, 2006.
2. Anton, Howard, *Dasar-dasar Aljabar Linear Jilid I*, Erlangga, Bandung, 2006.