

PERBANDINGAN KOMPLEKSITAS ALGORITMA METODE-METODE PENYELESAIAN SISTEM PERSAMAAN LANJAR

Achmad Dimas Noorcahyo – NIM 13508076

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jalan Ganeca 10, Bandung
e-mail: if18076@students.if.itb.ac.id

ABSTRAK

Perkembangan ilmu Metode Numerik yang pesat saat ini membuat jumlah algoritma penyelesaian Sistem Persamaan Lanjar menjadi cukup banyak tersedia. Makalah ini bertujuan untuk membandingkan kinerja dan kemangkusan dari beberapa algoritma penyelesaian Sistem Persamaan Lanjar. Empat algoritma yang dibandingkan antara lain algoritma Metode Eliminasi Gauss, Metode Eliminasi Gauss-Jordan, Metode Dekomposisi LU Gauss, dan Metode Dekomposisi LU Crout. Cara yang digunakan dalam mengumpulkan data perbandingan adalah dengan mengkaji kompleksitas waktu dari masing-masing metode. Untuk melihat perilaku algoritma ketika masukan terus bertambah besar, pada tiap metode akan dihitung fungsi Kompleksitas waktu asimptotiknya dalam bentuk notasi O besar. Hasil pengkajian menunjukkan bahwa keempat metode penyelesaian Sistem Persamaan Lanjar ini memiliki kompleksitas asimptotik yang sama. Namun, melalui hasil perbandingan konstanta dapat disimpulkan bahwa Metode Eliminasi Gauss-Jordan memiliki waktu proses paling lama dibandingkan ketiga metode lainnya.

Kata kunci: Perbandingan kompleksitas, Eliminasi Gauss, Eliminasi Gauss-Jordan, Dekomposisi LU Gauss, Dekomposisi LU Crout.

1. PENDAHULUAN

Sistem Persamaan Lanjar adalah suatu masalah yang selalu muncul, tidak hanya dalam ranah matematika, tetapi juga dalam hampir semua bidang Rekayasa (Engineering). Ukuran Sistem Persamaan Lanjar yang muncul dalam praktik biasanya sangat besar baik dalam jumlah persamaan maupun jumlah variabel. Besarnya ukuran sistem menyebabkan penghitungan solusi dengan cara biasa (analitik) sudah tidak memungkinkan lagi untuk dilakukan. Dibutuhkan suatu metode penyelesaian matematika lain yang dapat diterapkan untuk

menyelesaikan masalah pencarian solusi dalam sistem seperti ini. Masalah ini membawa pada munculnya ilmu metode numerik. Metode Numerik adalah teknik penyelesaian persamaan matematika dengan menggunakan operasi-operasi dasar (tambah, kurang, kali, bagi) dan bekerja berdasarkan suatu urutan algoritma. Metode numerik dapat dipakai untuk menghitung solusi sistem persamaan yang besar dan rumit ketika metode analitik sudah tidak mampu dijalankan lagi [3].

Perkembangan kinerja komputer yang pesat saat ini mendorong kemajuan metode numerik, karena pada dasarnya penghitungan metode numerik membutuhkan kerja yang banyak dan berulang sehingga sulit jika dilakukan oleh manusia. Maka, penghitungan metode numerik biasanya diserahkan ke komputer [3].

Sebanding dengan itu, kemajuan juga terjadi pada bidang metode numerik Sistem Persamaan Lanjar. Berbagai macam metode pencarian solusi SPL secara numerik dikembangkan oleh para ahli sehingga menyebabkan pilihan algoritma penyelesaian Sistem Persamaan Lanjar menjadi semakin banyak.

Namun, ternyata banyaknya metode penyelesaian juga menyebabkan timbulnya suatu masalah baru, khususnya di kalangan pemrogram aplikasi metode numerik. Timbul suatu pertanyaan: Dari banyak algoritma untuk menyelesaikan Sistem Persamaan Lanjar, yang mana yang paling mangkus?

Pertanyaan ini menjadi penting mengingat semua algoritma penyelesaian Sistem Persamaan Lanjar akan diimplementasikan sebagai suatu program komputer. Kecepatan kinerja program akan ditentukan oleh seberapa mangkus Algoritma yang membangun program tersebut.

Untuk menjawabnya, dalam makalah ini penulis akan membandingkan kemangkusan dari beberapa algoritma saja yang dinilai paling terkenal dan paling umum digunakan. Penyelesaian Sistem Persamaan Lanjar tersebut yaitu Metode Eliminasi Gauss, Metode Eliminasi Gauss-Jordan, Metode LU Gauss, dan Metode Dekomposisi LU Crout [3, 4].

2. OPERASI DASAR

Pada bagian ini akan dibahas operasi-operasi matriks yang mendasar dalam penyelesaian Sistem Persamaan Linier.

Dalam Aljabar Linier, matriks dipakai secara luas untuk merepresentasikan Sistem Persamaan Linier. Cara perrepresentasiannya adalah sebagai berikut [1] :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n &= b_2 \\ \cdot & \\ \cdot & \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Menjadi :

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \quad (1)$$

Dan, biasanya agar singkat direpresentasikan dalam bentuk Matriks teraugmentasi [1]:

$$\begin{pmatrix} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} & b_n \end{pmatrix} \quad (2)$$

2.1 Operasi Baris Elementer

Operasi Baris Elementer merupakan operasi matriks yang menjadi dasar utama dari Metode eliminasi Gauss, eliminasi Gauss-Jordan dan Dekomposisi LU Gauss.

Operasi Baris Elementer adalah operasi yang dilakukan pada baris matriks yang ,dalam hubungannya dengan matriks teraugmentasi, mengubah bentuk Sistem Persamaan Linier menjadi sederhana namun tidak mengubah solusi akhirnya [1]. Perubahan baris matriks yang dapat dilakukan dengan Operasi Baris Elementer adalah sebagai berikut :

- Mengalikan baris dengan bilangan konstan bukan 0
- Menukar dua baris
- Menambahkan kelipatan suatu baris dengan baris lainnya

2.2 Teknik Penyulihan

Teknik penyulihan adalah teknik yang digunakan untuk mendapatkan penyelesaian akhir Sistem Persamaan Linier. Teknik penyulihan terbagi menjadi dua, yaitu

teknik penyulihan mundur (*backward substitution*) dan teknik penyulihan maju (*forward substitution*) [3]. Teknik penyulihan mundur hanya dapat diterapkan ketika sistem persamaan linier berbentuk matriks segitiga atas.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (3)$$

Penghitungan diselesaikan dari baris paling bawah lalu mundur ke baris atas seperti berikut :

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}} \\ x_{n-1} &= \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} \\ &\vdots \end{aligned}$$

Sehingga didapat rumus umum teknik penyulihan maju

$$x_k = \frac{b_k - \sum_{j=k+1}^n a_{kj}x_j}{a_{kk}} \quad (4)$$

Pada k=n-1,n-2,...,1, dan dengan syarat $a_{kk} \neq 0$

Jika dituliskan dalam bentuk prosedur algoritma dengan keluaran vektor x sebagai berikut [3] :

procedure sulihmundur (A : matriks, b : vektor, n : **integer**, output x : array)

KAMUS LOKAL

i, j : **integer**
sigma : **real**

ALGORITMA

```
xn ← bn / an,n
i traversal [n-1..1]
  sigma ← 0
  j traversal [i+1..n]
    sigma ← sigma + ai,j * xj
xk ← (bk - sigma) / ak,k
```

Untuk Teknik Penyulihan Maju hanya dapat dilakukan pada Sistem Persamaan Linier berbentuk Matriks segitiga bawah. Prinsip penghitungan Teknik Penyulihan Mundur sama dengan Teknik penyulihan maju, namun karena teknik diterapkan pada matriks segitiga bawah, maka penghitungan dimulai dari baris paling atas, lalu maju ke bawah.

$$a_{iq} = \frac{a_{iq} - \sum_{k=1}^{i-1} a_{ik} a_{kq}}{a_{ii}} \quad (5)$$

pada $i=2,3,\dots,n$ dan dengan syarat $a_{ii} \neq 0$

Dengan dasar rumus ini, algoritma Penyulihan Maju juga dapat dituliskan dengan mudah. Algoritma penyulihan maju tidak akan dituliskan lagi karena fokus utama makalah ini bukan pada algoritmanya tetapi pada kompleksitas waktu dari masing-masing algoritma tersebut. Karena dasar prosesnya sama, maka kompleksitas waktu **procedure** sulihmundur dan **procedure** sulihmaju adalah sama.

3. METODE NUMERIK PENYELESAIAN SISTEM PERSAMAAN LANJAR

Dalam dunia penyelesaian Sistem Persamaan Lanjar sebenarnya ada dua metode numerik umum yaitu Metode Langsung (*Direct Method*) dan Metode Lelaran (*Iterative Method*) [3].

Namun, dalam metode Lelaran banyak hal yang perlu dipertimbangkan untuk menentukan kualitas dari suatu metode. Selain kecepatan program, metode ini juga mempertimbangkan ketelitian yang dihasilkan. Dalam makalah ini hanya dibahas Metode Langsung, karena metode langsung amat bergantung pada kecepatan program, sehingga dengan hanya menghitung kompleksitas waktunya sudah cukup merepresentasikan kualitas metode.

3.1 Eliminasi Gauss

Eliminasi Gauss bekerja dengan proses sebagai berikut :

- Melakukan Operasi Baris Elementer pada matriks Sistem Persamaan Lanjar sedemikian rupa sehingga membentuk matriks baru yang berbentuk segitiga atas.
- Melakukan teknik Penyulihan Mundur pada matriks segitiga atas yang didapat di langkah pertama.

Dua langkah ini sudah dapat menghasilkan solusi.

Metode Gauss dalam bentuk notasi algoritmik [3] :

```
procedure EliminasiGauss (A:matriks,
b:vektor, n:integer, output x : vektor)
```

KAMUS LOKAL
i, k, j : integer
m : real

ALGORITMA

```
k traversal [1..n-1]
  i traversal [k+1..n]
    m ← ai,k/ak,k
    j traversal [k..n]
      ai,j ← ai,j - m*ak,j
    bi ← bi - m* bk
```

sulihmundur (A, b, n, x)

3.2 Eliminasi Gauss Jordan

Eliminasi Gauss Jordan memiliki kesamaan dengan eliminasi Gauss. Proses utama dalam elimiisasi Gauss-Jordan adalah Operasi Baris Elementer. Pada Eliminasi Gauss-Jordan, Operasi Baris Elementer diterapkan pada matriks Sistem Persamaan Lanjar sampai terbentuk matriks satuan.

Karena sudah berbentuk matriks satuan, maka solusi langsung didapat, tanpa perlu Penyulihan Mundur.

Algoritma Eliminasi Gauss-Jordan seperti berikut [3] :

```
procedure EliminasiGaussJordan
(A:matriks, b:vektor, n:integer, output
x : vektor)
```

KAMUS LOKAL
i, k, j : integer
m, tampung : real

```
ALGORITMA
k traversal [1..n]
  tampung ← ak,k
  j traversal [1..n]
    ak,j ← ak,j / tampung

  bk ← bk / tampung
  i traversal [1..n]
    if (i ≠ k) then
      m ← ai,k
      j traversal [1..n]
        ai,j ← ai,j - m*ak,j
      bi ← bi - m* bk

  i traversal [1..n]
    xi ← bi
```

3.3 Dekomposisi LU Crout

Dekomposisi LU adalah cara penyelesaian Sistem Persamaan Lanjar dengan terlebih dahulu mamfaktorkan matriks Sistem Persamaan Lanjar menjadi dua matriks, Matriks pertama adalah matriks segitiga bawah dengan diagonal semua bernilai satu, sedangkan matriks kedua adalah matriks segitiga atas

Ilustrasi metode dekomposisi LU sebagai berikut [3] :

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad (6)$$

Difaktorkan menjadi :

$$\begin{matrix} \text{Matriks L} & \text{Matriks U} \end{matrix}$$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n1} & \cdots & l_{nn-1} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{pmatrix} \quad (7)$$

Jika A adalah matriks Sistem Persamaan Lanjar :

$$Ax = b \leftrightarrow LUx = b$$

Dengan mengasumsikan $Ux = z$, $Lz = b$ dapat diselesaikan dengan teknik penyulihan maju. Setelah z didapat maka $Ux = z$ dapat diselesaikan dengan Teknik penyulihan mundur [3].

Teknik pemfaktoran matriks A menjadi LU dapat dilakukan dengan dua cara. Cara pertama dengan menggunakan Reduksi Crout (**Dekomposisi LU Crout**).

Reduksi Crout didasarkan pada kesamaan :

$$A=LU$$

Dengan menyelesaikan persamaan ini bergantian dimulai dari baris pertama U, baris kedua L, baris kedua U, baris kedua L, dan seterusnya, maka diperoleh rumus reduksi Crout sebagai berikut [3] :

$$u_{pj} = a_{pj} - \sum_{k=1}^{p-1} l_{pk} u_{kj} \quad (8)$$

Dengan $p = 1, 2, \dots, n$ serta $j = p, p+1, \dots, n$

dan

$$l_{iq} = \frac{a_{iq} - \sum_{k=1}^{q-1} l_{ik} u_{kq}}{u_{qq}} \quad (9)$$

Dengan $q = 1, 2, \dots, n-1$ serta $i=q+1, q+2, \dots, n$

Namun, biasanya dalam representasi fisik di algoritma, hasil matriks L dan matriks U dalam reduksi Crout tidak akan diletakkan dalam matriks yang terpisah. Untuk menghemat ruang struktur data, maka Matriks L dan Matriks U akan diletakkan dalam satu matriks. Ini bisa dilakukan karena matriks L berbentuk matriks segitiga bawah, sedangkan U berbentuk matriks segitiga bawah.

Algoritma untuk Reduksi Crout [5] :

procedure crout(n:integer, input/output A:matriks)

KAMUS LOKAL

i, j, k, p : integer

t : real

ALGORITMA

```

p traversal [1..n]
  j traversal [k..n]
    t ← Ap,j
    k traversal [1..p-1]
      t ← t - (Ap,k*Ak,j);
    Ap,j ← t;

i traversal [p+1..n]
  t ← Ai,p;
  h traversal [1..p-1]
    t ← t - (Ai,k*Ak,p)
  Ai,p ← t/Ap,p

```

Metode pemfaktoran yang kedua adalah dengan menggunakan Eliminasi Gauss. Metode ini dinamakan **Dekomposisi LU Gauss**.

Metode Dekomposisi LU Gauss memproses matriks A menggunakan Operasi Baris Elementer hingga terbentuk Matriks segitiga atas yang tidak lain adalah matriks U. Adapun, matriks L diisi oleh faktor pengali dari proses Operasi Baris Elementer pada indeks yang bersesuaian .

4. ANALISIS KOMPLEKSITAS

Dalam Analisis Kompleksitas Waktu pada Algoritma Sistem Persamaan Lanjar, Operasi yang dominan dan menjadi Operasi Abstrak adalah Perkalian, Pembagian, Penjumlahan, dan Pengurangan. Dalam analisis bab ini, Pengurangan dimasukkan dalam kategori Penjumlahan, serta Pembagian dimasukkan dalam kategori Perkalian.

4.1 Kompleksitas Waktu Teknik Penyulihan

Karena pada dasarnya langkah pada operasi Teknik Penyulihan Mundur dan Teknik Penyulihan Maju adalah

sama, hanya berbeda arah pengerjaan, maka fungsi Kompleksitas keduanya juga sama.

Kompleksitas Waktu dari Teknik Penyulihan Mundur dapat dianalisis dari **procedure** sulihmundur di Upabab 2.2.

Pertama dihitung jumlah kalang yang dilalui :

Untuk $i = n-1$,
 Jumlah eksekusi kalang : 1
 Untuk $i = n-2$
 Jumlah eksekusi kalang : 2
 ⋮
 Untuk $i = 2$,
 Jumlah eksekusi kalang : $n-2$
 Untuk $i = 1$,
 Jumlah eksekusi kalang : $n-1$

Maka, Jumlah Operasi perkalian dan pembagian di dalam kalang adalah :

$$T(n) = 1 + 2 + \dots + (n-1)$$

$$= \sum_{i=1}^{n-1} i = n(n-1)/2$$

Selain itu, operasi $x_n \leftarrow b_n / a_{n,n}$ sebelum kalang, memuat 1 operasi perkalian .

Sedangkan operasi $x_k \leftarrow (b_k - \text{sigma}) / a_{k,k}$ memuat 1 operasi penjumlahan dan 1 operasi perkalian.

Sehingga secara total, **Teknik Penyulihan Mundur** memiliki :

Kompleksitas Waktu Penjumlahan :

$$T(n) = \frac{1}{2}n^2 - \frac{1}{2}n + 1 \tag{10}$$

Kompleksitas Waktu Perkalian :

$$T(n) = \frac{1}{2}n^2 - \frac{1}{2}n + 2 \tag{11}$$

Teknik Penyulihan Maju Memiliki fungsi kompleksitas yang sama.

4.2 Kompleksitas Waktu Metode Eliminasi Gauss

Penghitungan Kompleksitas dilakukan berdasar pada **procedure** EliminasiGauss yang ada di Upabab 3.1

Pertama, dihitung Jumlah Operasi

$$a_{i,j} \leftarrow a_{i,j} - m^*a_{k,j}$$

yang ada di dalam kalang j

Untuk $k=1$,

$$\text{Jumlah Operasi} : (n-1) \cdot n$$

Untuk $k=2$,

$$\text{Jumlah Operasi} : (n-2) \cdot (n-1)$$

⋮

Untuk $k = n-1$,

$$\text{Jumlah Operasi} : 1 \cdot 2$$

Sehingga untuk operasi $a_{i,j} \leftarrow a_{i,j} - m^*a_{k,j}$
 $T(n) = (n-1) \cdot n + (n-2)(n-1) + \dots + 1 \cdot 2$

$$= \sum_{i=1}^{n-1} (n-i)(n-i+1)$$

$$= (n^3 - n)/3$$

Selanjutnya dihitung operasi $m \leftarrow a_{i,k}/a_{k,k}$ yang memuat 1 perkalian dan $m \leftarrow a_{i,k}/a_{k,k}$ yang memuat 1 penjumlahan dan 1 perkalian, sebelum dan sesudah kalang j

Jumlah masing-masingnya adalah

$$T(n) = (n-1) + (n-2) + \dots + 1$$

$$= \sum_{i=1}^{n-1} i = n(n-1)/2$$

Karena Metode Eliminasi Gauss diakhiri dengan Teknik Penyulihan Mundur, maka, Kompleksitas waktu total metode ini harus dijumlahkan dengan kompleksitas waktu teknik penyulihan mundur.

Secara Total, **Metode Eliminasi Gauss** memiliki:

Kompleksitas Waktu Penjumlahan :

$$T(n) = \frac{1}{3}n^3 + \frac{3}{2}n^2 - \frac{11}{6}n + 2 \tag{12}$$

Kompleksitas Waktu perkalian

$$T(n) = \frac{1}{3}n^3 + n^2 - \frac{4}{3}n + 1 \tag{13}$$

4.3 Kompleksitas Waktu Metode Eliminasi Gauss-Jordan

Penghitungan Kompleksitas dilakukan berdasar pada **procedure** EliminasiGaussJordan yang ada di Upabab 3.2

Dimulai dengan menghitung operasi

$$a_{k,j} \leftarrow a_{k,j} / \text{tampung}$$

yang ada di dalam kalang j dan hanya memuat operasi perkalian

Operasi ini berada pada dua kalang yang memiliki pencacah dari 1 sampai n. Maka, Kompleksitas waktu operasi ini adalah :

$$T(n) = n^2$$

Kemudian, jumlah operasi $b_k \leftarrow b_k / t$ tampung yang hanya memuat 1 perkalian adalah :

$$T(n) = n$$

Perhitungan dilanjutkan untuk operasi yang ada di dalam kalang i. Pada kalang ini, ada pernyataan kondisional i tidak samadengan k. Namun, kondisi ini tidak dipenuhi hanya satu kali saja (ketika indeks sama) dari total n kali kalang. Jadi kalang i ini total berjumlah n-1 kali.

Sehingga,

Operasi $a_{i,j} \leftarrow a_{i,j} - m * a_{k,j}$ memiliki jumlah operasi :

$$T(n) = n^2(n-1)$$

Operasi $b_i \leftarrow b_i - m * b_k$ yang memiliki 1 operasi penjumlahan dan 1 operasi perkalian memiliki jumlah :

$$T(n) = n(n-1)$$

Eliminasi Gauss Jordan sudah otomatis menghasilkan matriks identitas, maka tidak usah diterapkan Teknik Penyulihan lagi.

Maka, Metode **Eliminasi Gauss-Jordan** memiliki total :

Kompleksitas Waktu Penjumlahan :

$$T(n) = n^3 - n \tag{14}$$

Kompleksitas Waktu Perkalian

$$T(n) = n^3 + n^2 \tag{15}$$

4.4 Kompleksitas Waktu Metode Dekomposisi LU

Untuk Metode Dekomposisi LU dengan Reduksi Crout, maka langkah awal adalah menjalankan reduksi Crout untuk faktorisasi LU. Algoritma reduksi Crout dijalankan dengan procedure crout yang ada pada Upabab 3.3.

Operasi pertama yang memuat perkalian dan penjumlahan adalah $t \leftarrow t - (A_{p,k} * A_{k,j})$ yang ada di kalang k.

Untuk p = 1,

Tidak masuk kalang k

Untuk p = 2,

Jumlah Operasi = (n-1). 1

Untuk p = 3,

Jumlah Operasi = (n-2). 2

⋮

Untuk p = n,

Jumlah Operasi = 1. (n-1)

Maka, Jumlah operasi $t \leftarrow t - (A_{p,k} * A_{k,j})$ adalah :

$$T(n) = (n-1).1 + (n-2).2 + \dots + 1. (n-1)$$

$$= \sum_{i=1}^{n-1} (n-i)i$$

$$= (n^3 - n)/6$$

Kemudian operasi berikutnya adalah

$t \leftarrow t - (A_{i,k} * A_{k,p})$ yang berada dalam kalang h.

Untuk p = 1,

Tidak masuk kalang h

Untuk p = 2,

Jumlah Operasi = (n-2). 1

Untuk p = 3,

Jumlah Operasi = (n-3). 2

⋮

Untuk p = n-1,

Jumlah Operasi = 1. (n-2)

Maka, Jumlah operasi $t \leftarrow t - (A_{i,k} * A_{k,p})$ adalah :

$$T(n) = (n-2).1 + (n-3).2 + \dots + 1.(n-2)$$

$$= \sum_{i=1}^{n-2} (n-i-1)i$$

$$= (n^3 - 3n^2 + 2n)/6$$

Untuk operasi $A_{i,p} \leftarrow t / A_{p,p}$ yang ada di baris terakhir jumlahnya adalah :

$$T(n) = (n-1) + (n-2) + \dots + 1$$

$$= \sum_{i=1}^{n-1} i = n(n-1)/2$$

Maka, total, **Reduksi Crout** memiliki kompleksitas :

Kompleksitas Waktu Penjumlahan :

$$T(n) = \frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n \tag{16}$$

Kompleksitas Waktu Perkalian :

$$T(n) = \frac{1}{3}n^3 - \frac{1}{3}n \tag{17}$$

Algoritma Metode Dekomposisi LU Crout merupakan proses yang terdiri atas Reduksi Crout untuk pemfaktoran, kemudian dilanjutkan dengan Teknik penyulihan Maju dan Teknik Penyulihan Mundu untuk penyelesaian. Maka, formula Kompleksitas metode ini :

Kompleksitas waktu Dekomposisi LU Crout = Kompleksitas waktu reduksi crout + 2 x Kompleksitas waktu teknik penyulihan

Sehingga, **Metode Dekomposisi LU Crout** memiliki :

Kompleksitas Waktu Penjumlahan :

$$T(n) = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n + 2 \quad (18)$$

Kompleksitas Waktu Perkalian :

$$T(n) = \frac{1}{3}n^3 + n^2 - \frac{4}{3}n + 4 \quad (19)$$

Untuk Metode Dekomposisi LU Gauss, metode ini merupakan proses yang terdiri atas Eliminasi Gauss untuk pemfaktoran, kemudian dilanjutkan dengan Teknik penyulihan Maju dan Teknik Penyulihan Mundur untuk penyelesaian. Maka, formula Kompleksitas metode ini :

Kompleksitas waktu Dekomposisi LU Gauss = Kompleksitas waktu reduksi Eliminasi Gauss + Kompleksitas waktu teknik penyulihan

Karena dalam metode Eliminasi Gauss sudah ada proses penyulihan, maka hanya butuh satu teknik penyulihan lagi, sehingga dalam formula kompleksitas waktunya tidak dikali 2.

Metode Dekomposisi LU Gauss memiliki :

Kompleksitas Waktu Penjumlahan :

$$T(n) = \frac{1}{3}n^3 + 2n^2 - \frac{7}{3}n + 3 \quad (20)$$

Kompleksitas Waktu Perkalian :

$$T(n) = \frac{1}{3}n^3 + \frac{3}{2}n^2 - \frac{11}{6}n + 3 \quad (21)$$

5. PERBANDINGAN KOMPLEKSITAS

Tabel 1 Tabel perbandingan kompleksitas waktu metode-metode penyelesaian Sisten Persamaan Lanjar

Metode Eliminasi Gauss
Penjumlahan : $T(n) = \frac{1}{3}n^3 + \frac{3}{2}n^2 - \frac{11}{6}n + 2$
Perkalian : $T(n) = \frac{1}{3}n^3 + n^2 - \frac{4}{3}n + 1$
Metode Eliminasi Gauss-Jordan
Penjumlahan : $T(n) = n^3 - n$
Perkalian : $T(n) = n^3 + n^2$
Metode Dekomposisi LU Crout
Penjumlahan : $T(n) = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n + 2$
Perkalian : $T(n) = \frac{1}{3}n^3 + n^2 - \frac{4}{3}n + 4$

Metode Dekomposisi LU Gauss
Penjumlahan : $T(n) = \frac{1}{3}n^3 + 2n^2 - \frac{7}{3}n + 3$
Perkalian : $T(n) = \frac{1}{3}n^3 + \frac{3}{2}n^2 - \frac{11}{6}n + 3$

Dari data analisis yang dirangkum di atas, hampir tidak ada perbedaan yang jauh dari keempat Metode Penyelesaian Sistem Persamaan Lanjar tersebut.

Semua fungsi kompleksitas waktu berbentuk polinomial yang derajat paling besarnya adalah 3 sehingga keempatnya memiliki Kompleksitas Asimptotik $O(n^3)$.

Artinya keempat algoritma ini tumbuh sebanding dengan fungsi yang sama, yaitu n^3 , apabila diberi masukan semakin besar.

Namun, walau tidak signifikan, keempat metode ini dapat dibandingkan dengan melihat konstanta yang mengikuti. Yang paling mencolok adalah pada Metode Eliminasi Gauss-Jordan. Metode ini memiliki konstanta 1 di depan n^3 . Berbeda dengan ketiga metode lain yang memiliki konstanta 1/3 di depan n^3 .

Ini menandakan Proses komputasi yang dibutuhkan pada Metode Gauss-Jordan memakan waktu lebih lama jika dibandingkan metode yang lain. Walaupun ordenya sama, namun untuk n cukup besar, bedanya cukup terlihat. Misalnya untuk $n=100$, Jika metode lain menghasilkan $T(n)$ sekitar 333.000, maka Metode Gauss-Jordan menghasilkan $T(n)$ sekitar 1.000.000.

Untuk ketiga metode lain sebenarnya juga masih dapat dibandingkan dengan cara melihat konstanta di depan n^2 , namun dengan signifikansi makin kecil.

6. KESIMPULAN

Dalam makalah ini, dibandingkan empat metode untuk menyelesaikan Sistem Persamaan Lanjar antara lain Metode Eliminasi Gauss, Metode Eliminasi Gauss-Jordan, Metode Dekomposisi LU Crout, dan Metode Dekomposisi LU Gauss. Hasil perbandingan memperlihatkan bahwa keempat metode ini memiliki Kompleksitas Asimptotik yang sama yaitu $O(n^3)$. Kesamaan kompleksitas ini memberikan gambaran perilaku dari keempat metode bahwa apabila diberi masukan yang semakin besar, keempatnya akan tumbuh dengan sebanding. Namun, dengan pendekatan perbandingan konstanta, berhasil ditunjukkan bahwa ternyata Metode Eliminasi Gauss-Jordan memiliki Proses perhitungan waktu paling lama dibandingkan ketiga metode lainnya. Untuk Ketiga

metode lain, karena tingkat signifikansinya semakin kecil, maka dianggap memiliki kinerja yang sama baik.

DAFTAR REFERENSI

- [1] Anton, Howard & Chris Rorres, "*Elementary Linear Algebra*", John Wiley & Sons, 2000
- [2] Munir, Rinaldi, "*Matematika Diskrit*", Penerbit Informatika, 2005
- [3] Munir, Rinaldi, "*Metode Numerik*", Penerbit Informatika, 2003
- [4] <http://www.htdp.org/2001-09-22/Book/node147.htm>
Tanggal Akses : 18 Desember 2009, 14.12 WIB
- [5] http://www.damtp.cam.ac.uk/lab/people/sd/lectures/nummet/h98/linear.htm#L_1_Linear_equations
Tanggal Akses : 19 Desember 2009, 07.36 WIB