ANALISIS KINERJA DEKOMPOSISI CROUT SEBAGAI PENYELESAIAN SISTEM PERSAMAAN LINIER BERUKURAN BESAR

Supriyono[1], Daniel Syamsudin[2]
[1]*Sekolah Tinggi Teknologi Nuklir – BATAN*
Jl. Babarsari Kotak Pos 6101/YKBB Yogyakarta.
E-mail: masprie_sttn@yahoo.com
[2]*Jurusan Teknik Informatika, Universitas Islam Indonesia*
Jl. Kaliurang Km. 14 Yogyakarta

Abstrak

Persoalan bidang ilmu rekayasa penyelesaiannya banyak yang berbentuk model sistem persamaan linier berukuran besar, contohnya adalah perhitungan deformasi bangunan akibat gempa atau gaya luar lainnya. Saat ini, untuk menyelesaikan persoalan system persamaan linier berukuran besar tersebut, metode yang sering digunakan oleh para praktisi maupun para peneliti adalah metode Gauss. Ada kelemahan dalam eliminasi Gauss, yaitu melibatkan dua langkah utama : eliminasi maju (forward elimination) dan pensubstitusian mundur (backward substitution). Dari dua Kegiatan tersebut, eliminasi maju merupakan bagian terbesar dari perhitungan. Akibatnya Waktu eksekusi menjadi lama. Untuk mengurangi jumlah perhitungan pada proses eliminasi maju tersebut, Crout menawarkan suatu proses dekomposisi, yaitu memecah suatu matriks [A] atas [L] dan [U]. Secara teoritis dapat diuraikan bahwa dengan dekomposisi Crout ternyata ada suatu operasi perhitungan yang dapat dipersingkat. Oleh karena itu dalam Penelitian ini diuji penyelesaian menggunakan dekomposisi Crout untuk menyelesaikan system persamaan linier yang berukuran besar. Ternyata dengan komputasi numeris terbukti benar bahwa dekomposisi Crout waktu eksekusinya jauh lebih pendek dibanding dengan menggunakan metode Gauss-Jordan, Gauss-Seidel maupun dengan dekomposisi LU sekalipun. Dalam Penelitian ini, dianalisis pula kompleksitas algoritma dekomposisi Crout, untuk membuktikan kebenaran hasil komputasi berupa waktu eksekusi dekomposisi Crout lebih pendek dibandingkan dengan metode yang lain. Sehingga dengan Penelitian ini, dari hasil analisis kinerjanya, dekomposisi Crout sudah waktunya untuk digunakan oleh para peneliti dan praktisi untuk menyelesaikan sistem persamaan linier yang berukuran besar.

Kata Kunci: *Sistem Persamaan Linier, Berukuran Besar, Dekomposisi Crout, Waktu Eksekusi, Kompleksitas Algoritma*

1. PENDAHULUAN

Banyak persoalan dalam bidang rekayasa dan teknik sipil khususnya tentang perubahan deformasi pada struktur bangunan penyelesaiannya berbentuk sistem persamaan linier dengan ukuran matriks yang besar [1]. Persoalan sistem persamaan linier berukuran besar, penyelesaiannya tidak mungkin dapat diselesaikan secara analitik. Oleh karena itu penyelesaian secara numeris adalah satu-satunya cara untuk menyelesaikannya.

Seiring dengan perkembangan dunia komputer, penelitian dalam bidang metode numeris juga ikut berkembang, sehingga banyak metode penyelesaian secara numeris yang terpublikasikan baik melalui jurnal ilmiah maupun pada buku-buku textbook. Dengan banyaknya metode penyelesaian secara numeris, perlu dilakukan penelitian tentang metode mana yang terbaik, agar pengguna metode numerik dapat memilih metode yang baik dan efisien.

Dalam penelitian ini, telah dikaji dan dianalisis suatu Dekomposisi Crout [2] sebagai salah satu metode penyelesaian sistem persamaan linier. Adapun bentuk kajian dan anlisisnya adalah membuktikan secara numeris keunggulan Dekomposisi Crout dibandingkankan dengan metode standart, yaitu metode Gauss-Jordan dan metode LU. Untuk lebih meyakinkan hasil secara numeris, dalam Penelitian ini dianalisis pula kompleksitas algoritmanya, yaitu dengan menggunakan fungsi langkah pada algoritma [3] masing-masing metode.

Untuk menguji kebenaran analisis, dibangun suatu perangkat lunak yang berfungsi sebagai alat uji maupun sebagai alat hitung. Hasil pengujian dibuat tabulasi waktu eksekusi untuk masing-masing ukuran matriks berukuran besarnya, misalnya matriks ukuran 1000 x 1000, 2000 x 2000, dst. Untuk pengujian (sebagai alat uji) dengan ukuran matriks yang besar inputnya digunakan teknik randomisasi, sedangkan untuk perhitungan (sebagai alat hitung) inputnya bisa dilakukan dengan pengisian (*entry data*) atau dengan membangkitkannya melalui otomatisasi pembangkitan data dengan formula.

Hasil penelitian menunjukkan bahwa Dekomposisi Crout ternyata lebih baik dibandingkan dengan metode-metode yang lain, karena Dekomposisi Crout di dalam algoritmanya maksimal hanya ada kalang di dalam kalang saja ($O(n^2)$). Sedangkan metode Gauss-Jordan terdapat kalang di dalam kalang dan di dalamnya ada kalang lagi ($O(n^3)$), sedangkan metode Gauss-Seidel ada persoalan dengan penghentian proses iterasi, sehingga ada satu kalang yang besarnya tak dapat diprediksi, yaitu pada proses kapan iterasi akan berhenti. Dalam kenyataannya untuk dapat dihasilkannya solusi dengan galat yang kecil, maka penghentian iterasi harus distop dengan mengkondisian toleransi error yang kecil sekali. Akibatnya ukuran kalang menjadi besar. Untuk dekomposisi LU walaupun besar langkahnya adalah ($O(n^2)$), tetapi suku-suku polinomialnya lebih banyak dibandingkan dengan Dekomposisi Court.

2. DASAR TEORI

Suatu sistem persamaan linier berikut [2]:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots\cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots\cdots + a_{2n}x_n &= b_2 \\ &\cdots \\ &\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots\cdots + a_{nn}x_n &= b_n \end{aligned} \qquad (1)$$

dapat dicari nilai $x_1, x_2, \ldots\ldots, x_n$ dengan *a* adalah koefisien-koefisien konstan dan *b* adalah konstanta-konstanta serta n adalah banyaknya persamaan.

Untuk n yang berukuran kecil, misalkan n $\leq$ 3 sangat mudah diselesaikan secara analitik, misalkan diselesaikan dengan aturan Cramer. Tetapi untuk n yang besar, maka teknik-teknik penyelesaian berbasis komputer harus digunakan. Metode-metode tersebut antara lain:

2.1 Metode Gauss–Jordan

Menurut [2] metode Gauss-Jordan adalah pengembangan dari metode Eliminasi Gauss, yaitu dengan merubah bentuk persamaan (1) menjadi bentuk:

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

Bentuk persamaan (2) dimanipulasi sedemikian rupa dengan eliminasi maju, sehingga dihasilkan bentuk matriks sebagai berikut:

$$\left[\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & b_1^* \\ 0 & 1 & \cdots & 0 & b_2^* \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & b_n^* \end{array}\right] \qquad (3)$$

Hasil persamaan (3) di atas mempunyai makna bahwa hasil penyelesaian system persamaan linier adalah : $x_1 = b_1^*$, $x_2 = b_2^*$, ….. , $x_n = b_n^*$.

2.2 Metode Gauss–Seidel

Menurut [2] Metode Gauss-Seidel adalah metode dengan langkah iterasi, yaitu dengan menghitung nilai salah satu x yang tergantung dengan nilai x yang lain. Iterasi Metode gauss tersebut dari persamaan (1) bentuk persamaannya ditulis sebagai persamaan (3) berikut:

$$\begin{cases} x_1 = \dfrac{b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n}{a_{11}} \\ x_2 = \dfrac{b_2 - a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n}{a_{11}} \\ \vdots \\ x_n = \dfrac{b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{nn-1}x_{n-1}}{a_{11}} \end{cases} \qquad (4)$$

Adapaun iterasi akan berhenti jika telah tercapai kondisi konvergen, yaitu $\left|\dfrac{x_i^{\,j} - x_i^{\,j-1}}{x_i^{\,j}}\right| \times 100\%$ lebih kecil toleransi error.

Untuk penyelesaian dengan metode Gauss, baik untuk metode Gauss-Jordan dan metode Gauss Seidel kemungkinan ada persoalan dengan proses pembagian dengan nol. Untuk menghindari kejadian proses yang terbagi dengan nol, maka dalam penyelesaian dengan metode Gauss digunakan teknik pivoting.

2.3 Dekomposisi LU

Jika persamaan (1) disusun ulang menjadi persamaan $[A]\{X\} - \{b\} = 0$, maka persamaan tersebut dapat disusun lagi menjadi suatu system segitiga atas:

$$\begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{Bmatrix} \qquad (5)$$

atau dapat ditulis sebagai $[U]\{X\} - \{c\} = 0$ dan segitiga atas:

$$[L] = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \qquad (6)$$

Akibatnya, dekomposisi LU adalah $[A] = [L][U]$.

Dengan dapat dihasilkannya matriks $[L]$ seperti persamaan (6), maka dengan substitusi maju dapat dihasilkan suatu vektor $\{c\}$ sebagai penyelesaian system persamaan linier yang memenuhi syarat $[U]\{X\} - \{c\} = 0$.

2.4 Dekomposisi Crout

Menurut [2], Metode Crout dapat diturunkan dengan menggunakan perkalian matriks untuk menghitung persamaan $[A] = [L][U]$ dan memberikan hasil $l_{i1} = a_{i1}$ untuk $i = 1,2,\cdots,n$ dan $u_{1j} = \frac{a_{ij}}{l_{11}}$ untuk $j = 2,3,\cdots,n$. Sehingga untuk $j = 2,3,\cdots,n-1$ dan $i = j, j+1,\cdots,n$ maka:

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \qquad (7)$$

$$u_{jk} = \frac{a_{jk} - \sum_{i=1}^{j-1} l_{ji} u_{ik}}{l_{jj}} \quad \text{untuk}$$

$$k = j+1, j+2, \cdots, n \qquad (8)$$

dan

$$l_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn} \qquad (9)$$

3. METODOLOGI PENELITIAN

Sesuai dengan kebutuhan dalam penelitian ini langkah-langkahnya adalah sebagai berikut:

3.1 Menurunkan persamaan model

Persamaan (1) sebagai sistem persamaan linier berukuran besar diselesaikan dengan 4 buah metode, yaitu metode Gauss-Jordan, metode Gauss-Seidel, Dekomposisi LU dan dekomposisi Crout. Hasil penyelesaian berupa $x_1, x_2, x_3, \cdots x_n$ dan waktu eksekusi beserta pembahasan kompleksitas algoritma.

3.2 Menentukan analisis kebutuhan

Sistem yang baik adalah suatu sistem yang benar, efisien dan mudah pengoperasiannya serta menarik. Agar tercapai tujuan membangun sistem yang baik, maka proses perlu disusun analisis kebutuhan yang meliputi:

- ***Kebutuhan input***
  Input yang diperlukan yang sesuai dengan persamaan (1) adalah ukuran matriks dan nilai koefisien-koefisien matriks A dan vektor b.
- ***Kebutuhan proses***
  Dari pers (1) dan (2) dapat disusun algoritma metode Gauss-Jordan, persamaan (4) algoritma metode Gauss-Seidel, persamaan (6) dan $[A] = [L][U]$ algoritma dekomposisi LU dan persamaan (7), (8), (9) algoritma Dekomposisi Crout. Adapun algoritma keempat prosedur menurut [2] beserta teknik pivoting untuk metode Gauss adalah sebagai berikut.

Algoritma 1. Metode Gauss-Jordan

```
Do k = 1 to n
     Dummy = a(k,k)
     Do j = 1 to n+1
          a(k,j) = a(k,j) / dummy]
     Enddo
     Do i = 1 to n
          If ( i ≠ k )
               Dummy = a(I,k)
               Do j = 1 to n+1
               a(i,j) = a(I,j) - Dummy
* a(k,j)
               Enddo
          Enddif
     Enddo
Enddo
```

Algoritma 2. Metode Gauss-Seidel

```
Do i = 1 to n
     Dummy = a(i,i)
     Do j = 1 to n
          a(i,j) = a(i,j) / Dummy
     Enddo
     c(i) = c(i) / Dummy
Enddo
Sentinel = 0
Iter = 0
DoWhile (iter < maxiterasi) and (
Sentinel =o)
     Sentinel = 1
     Iter = Iter +1
     Do i = 1 to n
          Old = x(i)
          Sum = c(i)
          Do j = 1 to n
               If (i ≠ j)
                    Sum = sum -
a(i,j)*x(j)
               Endif
          Enddo
     X(i) = Lamda * Sum + (1 -
Lamda)*Old
               If (Sentinel = 1) and
(x(i) ≠ 0)
     Eps = abs((x(i) - Old)/x(i))*100%
               If (Eps > Tol) Then
Sentinel = 0
               Endif
     Enddo
Enddo
```

Algoritma 3. Pivoting Metode Gauss

```
Pivot = k
Big = abs(a(k,k))
Do ii = (k + 1) to n
     Dummy = abs(a(ii,k))
     If (Dummy > Big)
          Big = Dummy
          Pivot = ii
     Endif
Enddo
If (Pivot  k)
     Do jj = k to n
          Dummy = a(Pivot,jj)
          a(Pivot,jj) = a(k,jj)
          a(k,jj) = Dummy
     Enddo
     Dummy = b(Pivot)
     b(k) = Dummy
Endif
```

Algoritma 4. Dekomposisi LU

```
Do k = 1 to (n – 1)
     Do i = k + 1 to n
          Faktor = a(k,i)/a(k,k)
          Do j = k to n
               a(j,i) = a(j,i) – Faktor
* a(j,k)
          Enddo
     Enddo
Enddo
y(1) = c(1)/a(1,1)
Do i = 2 to n
     Jumlah = 0
     Do j = 1 to (i-1)
          Jumlah = Jumlah + (a(I,j) *
y(j))
     Enddo
     Y(i) = (c(i) – Jumlah) / a(I,i)
Enddo
Do k = 1 to (n – 1)
     Do i = (k + 1) to n
          Faktor = a(I,k) / a(k,k)
          Do j = k to n
               A(I,j) = a(I,j) – Faktor
* a(k,j)
          Enddo
     Enddo
Enddo
Do i = 1 to n
     Faktor = a(i,i)
     Do j = 1 to n
          A(I,j) = a(I,j) / Faktor
     Enddo
Enddo
x(n) = y(n) / a(n,n)
Do I = n to 1 Step (-1)
     Jumlah = 0
     Do j = (i + 1) to n
          Jumlah = Jumlah + (a(i,j) *
x(j))
     Enddo
     x(i) = (y(i) – Jumlah) / a(i,i)
Enddo
```

Algoritma 5. Dekomposisi Crout

```
Do j = 2 to n
     a(i,j) = a(i,j) / a(i,1)
Enddo
Do j = 2 to (n – 1)
     Do I = j to n
          Sum = 0
          Do k = 1 to (j – 1)
               Sum = Sum + a(i,k) *
a(k,j)
          Enddo
          a(I,j) = a(i,j) – Sum
     Enddo
     Do k = (j + 1) to n
          Sum = 0
          Do i = 1 to (j – 1)
               Sum = Sum + a(j,i) *
a(i,k)
          Enddo
          a(j,k) = (a(j,k) – Sum) /
a(j,j)
     Enddo
Enddo
Sum = 0
Do k = 1 to (n – 1)
     Sum = Sum + (a(n,k) * a(k,n))
Enddo
a(n,n) = a(n,n) – Sum
```

- ***Kebutuhan output***
  Sesuai dengan prinsip membangun sistem, maka peranan output juga penting. Minimal output dapat memperlihatkan hasil akhir. Dalam penelitian ini, outputnya berupa $x_1, x_2, x_3, \cdots x_n$ dan waktu eksekusi.
- ***Kebutuhan perangkat lunak dan perangkat keras***
  Dalam membangun sistem ada 2 hal tentang perangkat keras yang perlu diperhatikan, yang pertama adalah dengan spesifikasi apa sistem itu dibangun dan dengan spesikasi apa sistem itu dapat dijalankan. Sistem ini dibangun dengan perangkat keras komputer pentium 100 Mhz dengan RAM 32 MB dan dapat dijalankan dengan komputer pentium setara ke atas. Adapun perangkat lunak yang digunakan untuk membangun sistem adalah Borland Delphi 5 dengan sistem operasi Windows 98.

3.3 Pembuatan perancangan sistem

Dalam penelitian ini karena yang menjadi tujuan adalah mencari $x_1, x_2, x_3, \cdots x_n$ dan waktu eksekusi, maka dalam perancangannya dibuat flowchart dengan basis keempat algoritma di atas. Dalam program ini sebagai pembanding yang dapat dilihat melalui eksekusi komputer adalah apakah hasilnya sama dan kedua program tersebut bagaimana perbandingan waktu eksekusinya.

3.4 Membangun program komputer

Program komputer yang digunakan untuk membangun sistem ini adalah perangkat lunak Borland Delphi 5 dengan alasan bahwa Borland Delphi 5 merupakan bahasa komputasi teknis yang sangat populer dan sangat mudah digunakan serta mudah pula untuk dipahami struktur bahasanya [4]. Dalam penelitian ini karena listing programnya panjang, maka tidak dapat ditampilkan dalam makalah ini.

3.5 Pengujian program

Setelah sistem selesai dibangun, maka harus diuji apakah sistem dapat berjalan dengan baik dan mudah dioperasikan. Pengujian dilakukan secara detail disampaikan pada bab hasil dan pembahasan berikut ini.

4. HASIL DAN PEMBAHASAN

Dalam penelitian ini, untuk masukan matriks berukuran besar digunakan masukan yang bersifat random. Dalam menganalisis hasil Penelitian ini ada 3 aspek, yaitu:

4.1 Aspek Kompleksitas Algoritma

Dilihat dari aspek kompleksitas dengan teori Big-O menurut [3], maka untuk metode Gauss – Jordan mempunyai bentuk fungsi $f(n) = n^3 + 2n^2 + n$, dimana *n* sebagai fungsi putaran (looping) sedangkan untuk metode Gauss – Seidel bentuk fungsinya adalah $f(n) = (C+1)n^2$ dimana C adalah suatu konstanta yang bergantung dengan tingkat toleransi penghentian iterasinya. Semakin kecil toleransi error, semakin besar nilai C. Kebalikannya semakin besar toleransi error semakin kecil nilai C. Sehingga fungsi putaran pada metode Gauss – Seidel agak tidak pasti. Untuk dekomposisi LU bentuk fungsinya adalah $f(n) = 3n^2 + 2n - 5$ dan dekomposisi Crout bentuk fungsinya adalah $f(n) = 2n^2 - 5n + 4$.

Dengan melihat keempat bentuk fungsi putaran, maka dekomposisi Crout terlihat paling sedikit nilainya. Hal ini dapat dilihat pada tabel kompleksitas algoritma berikut:

Tabel 1. Perbandingan Jumlah Putaran Masing-Masing Metode

Ukuran Matriks	Metode Penyelesaian Sistem Persamaan Linier			
	Gauss- Jordan	Gauss-Seidel	Dekomposisi LU	Dekomposisi Crout
100 x 100	1.020.100	1.010.000	30.195	19.504
500 x 500	125.500.500	25.250.000	750.995	497.504
1000 x 1000	1.002.001.000	101.000.000	3.001.995	1.995.004
1500 x 1500	3.379.501.500	227.250.000	6.752.995	4.492.504
2000 x 2000	8.008.002.000	404.000.000	12.003.995	7.990.004
2500 x 2500	15.637.502.500	631.250.000	18.754.995	12.487.504

Catatan: Untuk Metode Gauss–Seidel, diasumsikan C (maksimum iterasi) = 100.

Dari tabel 1. Di atas, nampak bahwa banyak putaran pada Dekomposisi Crout: Dekomposisi LU: Metode Gauss Jordan adalah 2: 3: 500. Dalam perbandingan tersebut, metode Gauss–Seidel tidak diikutsertakan karena ada faktor ketidak pastian, yaitu toleransi error (maksimum iterasi). Untuk membuktikan dan memperkuat analisis bahwa Dekompoisi Crout lebih baik dibandingkan dengan metode-metode yang lain, maka perlu dlakukan pengujian dengan melihat waktu eksekusi. Pengujian tersebut dituliskan pada sub bab 4.2 berikut.

Dalam pengujian pada Sub bab 4.2 tersebut dengan alasan yang sama, metode Gauss–Seidel tidak termasuk yang diujikan. Karena ketidak pastian jumlah iterasi.

4.2 Aspek Waktu Eksekusi

Untuk pengujian waktu eksekusi dengan menggunakan perangkat lunak Delphi 6, hasilnya disusun dalam bentuk tabel dan ditampilkan pada tabel 2. berikut ini.

Tabel 2. Perbandingan Waktu Eksekusi Masing-Masing Metode

Ukuran Matriks	Metode Penyelesaian Sistem Persamaan Linier		
	Gauss - Jordan	Dekomposisi LU	Dekomposisi Crout
100x100	00:00:160	00:00:050	00:00:010
500x500	00:05:320	00:04:170	00:01:700
1000x1000	00:39:490	00:35:590	00:12:030
1500x1500	02:05:060	01:30:440	00:38:390
2000x2000	05:33:730	05:29:890	02:45:380
2500x2500	10:56:030	10:29:890	06:38:370

Catatan: arti dari xx: yy: zz adalah menit: detik: mili detik

Dilihat dari tabel 2. Di atas, memang tidak terjadi perbadingan waktu eksekusi setara dengan perbandingan jumlah putaran. Hal ini disebabkan bahwa waktu eksekusi tidak hanya disebabkan oleh kompleksitas algoritma saja, tetapi juga menyangkut ukuran besar matriks. Jika penelitian ini dikembangkan terus dengan ukuran matriks yang lebih besar lagi, misalkan dengan ukuran matriks 1000000 x 1000000 niscaya ukuran perbandingan waktu eksekusi akan menunjukkan perbedaan yang signifikan.

Dalam penelitian ini, pengujian untuk ukuran matriks 1000000 x 1000000 ada kendala dalam perangkat keras yang digunakan. Walaupun demikian, untuk ukuran matriks berapapun juga, dekomposisi Crout menunjukkan kelebihannya. Pengujian dengan kompleksitas algoritma maupun dengan waktu eksekusi tidak akan ada gunanya jika tujuan pencarian nilai $x_1, x_2, x_3, \cdots x_n$ tidak tercapai, yaitu nilai $x_1, x_2, x_3, \cdots x_n$ yang berbeda-beda. Untuk pengujian tersebut dituliskan pada sub bab 4.3 berikut.

4.3 Aspek Hasil Pengujian

Sistem aplikasi dalam penelitian ini dapat digunakan untuk mencari .... untuk ukuran matriks berukuran besar maupun yang berukuran kecil. Untuk matriks yang berukuran besar, misalkan untuk matriks berukuran 100 x 100, maka nila matriks *A* dan vektor *b* dibangkitkan dengan bilangan random. Sedangkan untuk sistem yang berukuran kecil, misalkan untuk matriks berukuran 4 x 4, masukan dapat diisikan melalui menu yang disediakan. Sebagai contoh persoalan sistem persamaan linier berikut ini:

$$\begin{bmatrix} 2.630 & 5.210 & -1.694 & 0.938 \\ 3.160 & -2.950 & 0.813 & -4.210 \\ 5.360 & 1.880 & -2.150 & -4.950 \\ 1.340 & 2.980 & -0.432 & -1.768 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 4.230 \\ -0.716 \\ 1.280 \\ 0.419 \end{Bmatrix}$$

Dengan memilih menu Hitung, maka keempat metode menunjukkan hasil yang sama, yaitu:

$x_1 = 1.038$, $x_2 = 0.209$, $x_3 = 0.226$ dan $x_4 = 0.847$.

5. KESIMPULAN

Hasil penelitian menunjukkan bahwa:

a. Metode Crout dilihat dari waktu eksekusi dan analisis algoritma lebih baik dibandingkan dngan metode Gauss – Jordan dan dekomposisi LU.

b. Metode Gauss – Seidel dimungkinkan lebih baik dibandingkan dengan metode penyelesaian yang lain jika faktor toleransi error tidak begitu dipersoalkan. Artinya jika pengambilan toleransi error diperbesar, dimungkinkan metode Gauss–Seidel lebih baik dibandingkan dengan metode yang lain.

DAFTAR PUSTAKA

[1] Supriyono and Miyoshi, T., “A Modified Extrapolation Method for large System of Ordinary Differential Equations”, Japan Journal of Industrial and Applied Mathematics, Vol. 12, 1995, pp. 439 – 455.

[2] Chapra, S.T. and Canale, R.P., “Numerical Methods For Engineers”, McGraw-Hill, New York USA, 1998.

[3] Harijanto, B., “Struktur Data”, CV Informatika, Bandung, 2000.

[4] Martina, I., “36 Belajar Komputer Delphi 5.0”, PT. Elex Media Komputindo, Jakarta, 2000.

[5] Pamitrapati, D dan Siahaan, K., “Trik Pemrograman Delphi”, PT. Elex Media Komputindo, Jakarta, 2000.