

# Metode Eliminasi Gauss dengan Komputasi Paralel

Christian Anthony Setyawan 13514085<sup>1</sup>

Program Studi Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13514085@std.stei.itb.ac.id

**Abstrak**—Sistem persamaan linear banyak dipakai. Salah satu cara untuk menyelesaikan sistem persamaan linear adalah dengan metode eliminasi Gauss. Untuk mengolah data yang besar dibutuhkan bantuan komputer. Ada teknik paralel yang dapat mempercepat algoritmas eliminasi Gauss di komputer. Dari hasil analisis ditemukan salah satu iterasi dari algoritma metode eliminasi Gauss bisa dibuat paralel sehingga dapat mempercepat metode eliminasi Gauss ini.

**Kata Kunci**—Eliminasi Gauss, Komputasi paralel, Matriks, Sistem Persamaan Linear

## I. LATAR BELAKANG

Di dalam matematika, system persamaan linier adalah kumpulan persamaan-persamaan linier yang memiliki variabel-variabel yang sama. Banyak aplikasi di dunia nyata yang membutuhkan sistem persamaan linear. Salah satu aplikasi yang berguna adalah menggunakan system persamaan linear untuk extrapolasi dari stock market sehingga pasha pialang saham dapat mendapatkan keuntungan yang lebih pasti.

Sistem persamaan linear sendiri perlu diselesaikan dengan metode penyelesaian sistem persamaan linear seperti metode grafik, metode substitusi dan yang lainnya. Salah satu metode penyelesaian system persamaan linear yang paling terkenal ialah metode eliminasi Gauss.

Eliminasi Gauss adalah suatu metode untuk mengoperasikan nilai-nilai di dalam matriks sehingga menjadi matriks yang lebih sederhana lagi. Dengan melakukan operasi baris sehingga matriks tersebut menjadi matriks yang baris. Ini dapat digunakan sebagai salah satu metode penyelesaian persamaan linear dengan menggunakan matriks. Caranya dengan mengubah persamaan linear tersebut ke dalam matriks teraugmentasi dan mengoperasikannya. Setelah menjadi matriks baris, lakukan substitusi balik untuk mendapatkan nilai dari variabel-variabel tersebut.

Meskipun sudah memakai berbagai macam metode, namun menyelesaikan system persamaan linear tetaplah sulit apabila banyak variable dan persamaan yang perlu ditinjau. Oleh karena itu, kita memerlukan bantuan komputer untuk menyelesaikan system persamaan linear. Makalah ini hanya khusus membahas bagaimana cara menyelesaikan system persamaan linear dengan metode eliminasi Gauss dan juga bagaimana menyelesaikannya dengan menggunakan komputasi paralel.

## II. LANDASAN TEORI

### 2.1 Sistem persamaan linear

Suatu persamaan dalam matematika merupakan sebuah ekspresi kesamaan (memuat tanda sama dengan, “=”) yang melibatkan konstanta, variabel, dan operasi-operasi hitung/matematika. Di dalam sebuah persamaan, komponen-komponen yang dijumlahkan atau dikurangkan disebut persamaan linear

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Kuantitas-kuantitas  $a_{ij}$  (untuk  $i, j = 1, 2, \dots, n$ ) disebut koefisien. Nilai koefisien-koefisien dan ruas kanan bipada setiap persamaan diketahui. Kuantitas-kuantitas  $x_i$  disebut variabel, yang nilainya belum diketahui dan hendak dicari.

Sistem persamaan di atas dapat ditulis dalam bentuk matriks sebagai

$$AX = B$$

dengan  $A$  adalah sebuah matriks  $n \times n$ :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

dan  $X$  dan  $B$  adalah vector-vektor  $n$ -komponen:

$$\begin{aligned} X &= [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T \\ B &= [b_1 \ b_2 \ b_3 \ \dots \ b_n]^T \end{aligned}$$

dengan pangkat  $T$  menyatakan operasi transpose matriks, yakni mengubah baris menjadi kolom dan kolom menjadi baris.

Matriks  $A$  disebut matriks koefisien, vektor kolom  $B$  sering disebut vektor konstanta. Gabungan matriks  $A$  dan vektor kolom  $B$ , yakni matriks  $n \times (n + 1)(A|B)$ , disebut matriks augmented dari Sistem Persamaan Linear.

Apabila semua nilai  $b_i = 0$  untuk  $i = 1, 2, \dots, n$ , maka Sistem Persamaan Linear tersebut disebut sistem homogen. Jika terdapat  $b_k \neq 0$  untuk suatu  $1 \leq k \leq n$ , maka Sistem Persamaan Linear tersebut disebut

sistem tak homogen. Sistem homogen memegang peranan penting untuk mengetahui ada tidaknya penyelesaian Sistem Persamaan Linear tersebut. Teorema berikut meringkaskan beberapa hasil penting tentang sistem-sistem persamaan linier.

## 2.2 Eliminasi Gauss

Metode eliminasi Gauss digunakan untuk menyelesaikan sebuah sistem persamaan linier dengan mengubah Sistem Persamaan Linear tersebut ke dalam bentuk sistem persamaan linier berbentuk segitiga atas, yakni yang semua koefisien di bawah diagonal utamanya bernilai nol. Bentuk segitiga atas ini dapat diselesaikan dengan menggunakan substitusi (penyulihan) balik. Untuk mendapatkan bentuk Sistem Persamaan Linear segitiga dari Sistem Persamaan Linear yang diketahui, metode eliminasi Gauss menggunakan sejumlah operasi baris elementer (OBE):

1. Menukar posisi dua buah persamaan (dua baris matriks augmented)
2. Menambah sebuah persamaan (baris matriks augmented) dengan suatu kelipatan persamaan lain (baris lain)
3. Mengalikan sebuah persamaan (baris matriks augmented) dengan sebarang konstanta tak nol.

Pemakaian operasi-operasi baris elementer di atas pada sebuah Sistem Persamaan Linear tidak akan mengubah penyelesaian Sistem Persamaan Linear yang bersangkutan. Jelas bahwa penyelesaian sebuah Sistem Persamaan Linear tidak tergantung pada susunan penulisan persamaan, sehingga operasi baris nomor 1 dapat dipakai. Dalam setiap persamaan, kedua ruas menyatakan nilai yang sama, sehingga operasi baris nomor 2 dapat digunakan. Demikian pula, operasi baris nomor 3 menghasilkan persamaan yang ekuivalen.

Nilai  $a_{kk} \neq 0$  pada baris ke- $k$  di mana  $a_{kj} = 0$  untuk  $j < k$ , disebut elemen pivot pada langkah ke- $k$ . Baris yang memuat elemen pivot disebut baris pivot. Eliminasi juga dapat menyebabkan hasil yang jelek apabila pada beberapa langkah digunakan pengali  $m_{ik}$  yang nilainya lebih besar daripada 1. Hal ini dikarenakan pada langkah ke- $k$ , galat pembulatan pada koefisien-koefisien  $a_{k,k+1}$ ,  $a_{k,k+2}$ ,  $a_{k,k+3}$ , ..., dan  $a_{kn}$ , serta  $b_k$  diperbesar oleh faktor  $m_{ik}$ . Apabila nilai-nilai  $m_{ik}$  pada langkah-langkah berurutan hampir sama besar, maka galat pembulatannya akan terakumulasi secara cepat, menyebabkan metode yang tidak stabil. Angka-angka signifikan mungkin juga akan hilang pada proses penyulihan mundur apabila terdapat elemen-elemen pivot yang bernilai sangat kecil. Salah satu metode untuk mengatasi masalah-masalah ini disebut metode penentuan pivot parsial.

Kata parsial digunakan untuk membedakan prosedur ini dengan metode penentuan pivot total, yang menggunakan pertukaran baris dan kolom. Penentuan pivot total menghasilkan reduksi tambahan yang mempengaruhi galat-galat pembulatan dan hal ini sangat penting demi keakuratan penyelesaian sistem-sistem tertentu.

Apabila elemen pivot pada langkah ke- $k$  bernilai nol, maka terdapat tiga kemungkinan:

1. Sistem Persamaan Linear mempunyai solusi tunggal.

Dalam hal ini baris pivot ditukar dengan salah satu baris di bawahnya sedemikian hingga diperoleh elemen pivot yang tak nol.

2. Sistem Persamaan Linear yang bersangkutan tidak bebas.

Apabila elemen pivot nol tidak dapat dihindari dan Sistem Persamaan Linear-nya bersifat konsisten, maka Sistem Persamaan Linear tersebut mempunyai tak berhingga penyelesaian.

3. Sistem Persamaan Linear yang bersangkutan tidak konsisten.

Dengan mengganti ruas kanan persamaan pertama Sistem Persamaan Linear di atas akan diperoleh Sistem Persamaan Linear lain yang bersifat tidak konsisten.

Suatu SPL yang bersifat bahwa tidak ada satu persamaan pun yang dapat dinyatakan sebagai kombinasi linier persamaan-persamaan yang lain disebut bebas linier. Dari teorema dasar dalam aljabar linier diketahui bahwa setiap SPL bebas linier yang terdiri atas  $n$  persamaan dalam  $n$  variabel mempunyai penyelesaian tunggal. Akan tetapi di dalam komputasi numerik, karena digunakan pendekatan hampiran dengan menggunakan perhitungan-perhitungan aritmetika oleh komputer, pernyataan tersebut diartikan secara kurang persis. Khususnya, jika pada suatu langkah eliminasi Gauss diperoleh elemen pivot yang tidak tepat bernilai nol, namun sangat kecil (mendekati nol) dibandingkan dengan koefisien-koefisien lain dalam baris pivot, pembagian oleh elemen pivot tersebut mengakibatkan penyelesaian numerik yang memuat galat pembulatan yang mungkin cukup berarti, sehingga penyelesaian yang diperoleh tidak valid.

## 2.3 Komputasi Paralel

Paralel Processing adalah kemampuan menjalankan tugas atau aplikasi lebih dari satu aplikasi dan dijalankan secara simultan atau bersamaan pada sebuah komputer. Secara umum, ini adalah sebuah teknik dimana sebuah masalah dibagi dalam beberapa masalah kecil untuk mempercepat proses penyelesaian masalah.

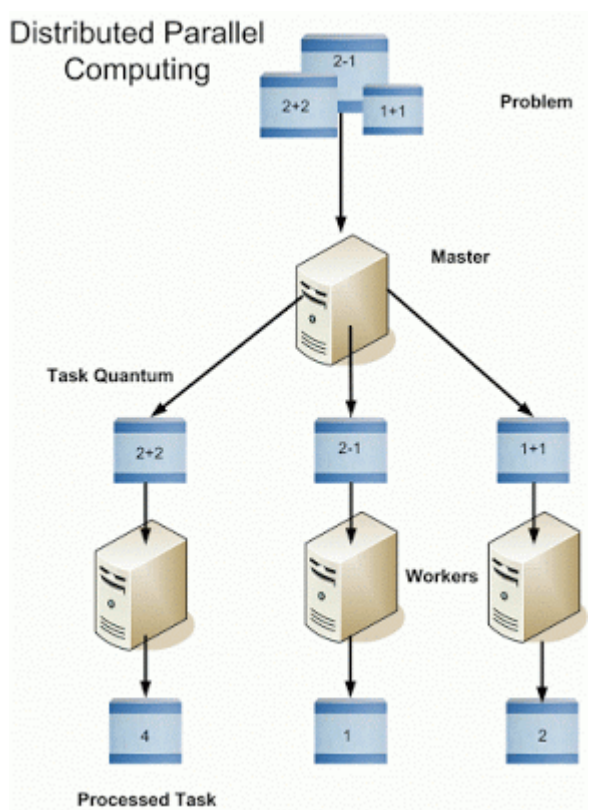
Terdapat dua hukum yang berlaku dalam sebuah parallel processing, yaitu:

- Hukum Amdahl  
Amdahl berpendapat, "Peningkatan kecepatan secara paralel akan menjadi linear, melipatgandakan kemampuan proses sebuah komputer dan mengurangi separuh dari waktu proses yang diperlukan untuk menyelesaikan sebuah masalah."
- Hukum Gustafson  
Pendapat yang dikemukakan Gustafson hampir sama dengan Amdahl, tetapi dalam pemikiran Gustafson, sebuah komputasi paralel berjalan

dengan menggunakan dua atau lebih mesin untuk mempercepat penyelesaian masalah dengan memperhatikan faktor eksternal, seperti kemampuan mesin dan kecepatan proses tiap-tiap mesin yang digunakan.

Merupakan sebuah perangkat lunak yang mampu mensimulasikan pemrosesan paralel pada jaringan.

Model komputasi Paralel.



Sumber: <https://andri102.wordpress.com/game/soft-skill/konsep-komputasi-paralel-processing/>

Gambar diatas merupakan contoh dari sebuah komputasi paralel, dimana pada gambar diatas terdapat sebuah masalah, dari masalah tersebut dibagi lagi menjadi beberapa bagian agar sebuah masalah dapat dengan cepat diatasi.

Tujuan dari komputasi paralel adalah meningkatkan kinerja komputer dalam menyelesaikan berbagai masalah. Dengan membagi sebuah masalah besar ke dalam beberapa masalah kecil, membuat kinerja menjadi cepat.

Formula komputasi paralel yang diajukan pada hukum Amdahl

$$S = \frac{1}{\alpha}$$

dimana  $\alpha$  adalah banyaknya paralel yang terjadi. Secara teori, artinya proses penyelesaian masalah menjadi lebih cepat dengan menggunakan komputasi paralel.

Salah satu jenis penggunaan komputasi paralel adalah:

PVM(Parallel Virtual Machine)

1. Embarrassingly Parallel adalah pemrograman paralel yang digunakan pada masalah-masalah yang bisa diparalelkan tanpa membutuhkan komunikasi satu sama lain. Sebenarnya pemrograman ini bisa dibilang sebagai pemrograman paralel yang ideal, karena tanpa biaya komunikasi, lebih banyak peningkatan kecepatan yang bisa dicapai.
2. Taksonomi dari model pemrosesan paralel dibuat berdasarkan alur instruksi dan alur data yang digunakan:
  - SISD (Single Instruction Single Datapath) merupakan prosesor tunggal, yang bukan paralel.
  - SIMD (Single Instruction Multiple Datapath) alur instruksi yang sama dijalankan terhadap banyak alur data yang berbeda. Alur instruksi di sini kalau tidak salah maksudnya ya program komputer itu. trus datapath itu paling ya inputnya, jadi inputnya lain-lain tapi program yang digunakan sama.
  - MIMD (Multiple Instruction Multiple Datapath) alur instruksinya banyak, alur datanya juga banyak, tapi masing-masing bisa berinteraksi.
  - MISD (Multiple Instruction Single Datapath) alur instruksinya banyak tapi beroperasi pada data yang sama.

### III. METODE ELIMINASI GAUSS DENGAN KOMPUTASI PARALEL

#### 3.1 Algoritmas Eliminasi Gauss Linear

Eliminasi Gauss bertujuan untuk mengubah sistem persamaan linear menjadi matriks segitiga atas yang nantinya akan dapat digunakan untuk mengetahui solusi dari sistem persamaan linear tersebut. Sebuah kolom pivot digunakan untuk menyederhanakan baris sebelumnya; kemudian setelah transformasi, substitusi balik akan digunakan. Setelah kita mendapatkan matriks augmentednya, maka kita akan melakukan operasi baris elementer untuk mengubahnya menjadi

$$\left[ \begin{array}{cccc|c} a'_{11} & a'_{12} & \cdots & a'_{1k} & b'_1 \\ 0 & a'_{22} & \cdots & a'_{2k} & b'_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a'_{kk} & b'_k \end{array} \right]$$

Selesaikan persamaan pada baris ke- $k$  untuk menyelesaikan  $x_k$ , kemudian substitusi balik persamaan pada baris ke  $(k - 1)$  untuk mengetahui  $x_{k-1}$

Langkah-langkah diatas dapat diselesaikan dengan algoritma berikut:

```
void gaussianElimination(int N, int **A, int *b)
{
    for (int i = 0; i < N-1; i++) {
        for (int j = i; j < N; j++) {
            double ratio = A[j][i]/A[i][i];
            for (int k = i; k < N; k++) {
                A[j][k] -= (ratio*A[i][k]);
                b[j] -= (ratio*b[i]);
            }
        }
    }
}
```

Algoritma ini juga mengasumsikan nilai  $A[k,k]$  akan menjadi 0 (nol) ketika elemen tersebut sudah dijadikan pivot.

Untuk nilai  $0 < k < n - 1$ , prosedur eliminasi Gauss secara sistematis mengeliminasi variable  $x[k]$  dari persamaan ke-  $k+1$  hingga  $n-1$  sehingga koefisien matrix menjadi upper triangular. seperti terlihat pada algoritma diatas, pada iterasi ke -  $k$  dari loop terluar, persamaan ke-  $k$  yang dikalikan dengan pivot-nya kemudian dikurangi dengan persamaan ke  $k+1$  hingga persamaan  $n-1$  berikutnya. Perkalian persamaan ke- $k$  (atau baris matrix  $A$  ke - $k$ ) dipilih sedemikian hingga koefisien ke- $k$  pada persamaan ke- $k+1$  hingga persamaan ke- $n-1$  bernilai 0 (nol) dan meng hilangkan nilai elemen  $x[k]$  dari persamaan tersebut.

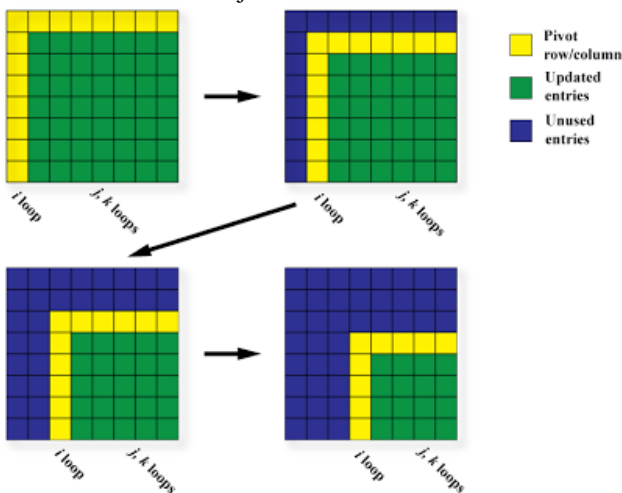
### 3.2 Algoritma Eliminasi Gauss Parallel

Iterasi manakah yang bisa diparallelkan?

- Iterasi di mana jumlah iterasi tetap pada saat masuk iterasi paralel.
- Iterasi di mana setiap iterasi independen dari semua iterasi lain.
- Iterasi yang tidak mengandung ketergantungan data.

Kita memiliki tiga kemungkinan calon iterasi yang bisa diparallelkan. Mana yang harus kita pilih?

Pertama, kita harus benar-benar memahami bagaimana Eliminasi Gauss bekerja. Gambar ini bisa membantu:



Sumber:

[http://riebecca.blogspot.com/2008/12/supercomputing-course-openmp-syntax-and\\_15.html](http://riebecca.blogspot.com/2008/12/supercomputing-course-openmp-syntax-and_15.html)

Eliminasi Gauss dapat dipahami melalui empat frame dari gambar ini, sekarang kita harus memutuskan mana iterasi yang bisa diparallelkan.

Iterasi I diwakili oleh baris dan kolom kuning. Entri pada baris kuning dan kolom yang digunakan untuk memperbarui sub matriks hijau sebelum pergi ke baris / kolom  $i + 1$ , yang berarti nilai-nilai entri ke -  $(i + 1)$  daerah kuning tergantung pada operasi apa yang dilakukan pada mereka pada nilai-nilai sebelumnya  $i$ . Oleh karena itu kita tidak dapat menggunakan iterasi I untuk memparallelkan iterasi ini karena ketergantungan data.

Iterasi J memiliki jumlah iterasi yang bervariasi dengan I, tapi kita tahu jumlah iterasi setiap kali kita akan memasuki iterasi. Tak satu pun dari iterasi kemudian tergantung pada iterasi-iterasi yang sebelumnya dan iterasi dapat dihitung dalam urutan apapun! Jadi iterasi  $j$  dapat diparallelkan.

Iterasi K, seperti iterasi J, memiliki jumlah iterasi yang bervariasi tetapi dihitung untuk setiap I. Tak satu pun dari iterasi kemudian bergantung pada yang sebelumnya, dan mereka semua dapat dihitung dalam urutan apapun. Oleh karena itu iterasi K juga dapat diparallelkan.

Pilihan yang terbaik adalah untuk memilih iterasi luar (J), karena dengan begitu kita akan memiliki paralelisme yang lebih tidak terganggu dan lebih sedikit pemanggilan fork dan join.

Implementasi algoritma dalam MPI:

```
root = 0
chunk = n**2/p
! main loop
do pivot = 1, n-1
    ! root maintains communication
    if (my_rank.eq.0) then
        ! adjust the chunk size
        if (MOD(pivot, p).eq.0) then
            chunk = chunk - n
        endif

        ! calculate chunk vectors
        rem = MOD((n**2-(n*pivot)), chunk)
        tmp = 0
        do i = 1, p
            tmp = tmp + chunk
            if (tmp.le.(n**2-(n*pivot))) then
                a_chnk_vec(i) = chunk
                b_chnk_vec(i) = chunk / n
            else
                a_chnk_vec(i) = rem
                b_chnk_vec(i) = rem / n
                rem = 0
            endif
        continue
        ! calculate displacement vectors
        a_disp_vec(1) = (pivot*n)
        b_disp_vec(1) = pivot
        do i = 2, p
            a_disp_vec(i) = a_disp_vec(i-1)
                + a_chnk_vec(i-1)
            b_disp_vec(i) = b_disp_vec(i-1)
                + b_chnk_vec(i-1)
        enddo
    endif
enddo
```

```

continue

! fetch the pivot equation
do i = 1, n
  pivot_eqn(i) = a(n-(i-1),pivot)
  continue
  pivot_b = b(pivot)
endif ! my_rank.eq.0

! distribute the pivot equation
call MPI_BCAST(pivot_eqn, n,
  MPI_DOUBLE_PRECISION,
  root, MPI_COMM_WORLD, ierr)
call MPI_BCAST(pivot_b, 1,
  MPI_DOUBLE_PRECISION,
  root, MPI_COMM_WORLD, ierr)

! distribute the chunk vector
call MPI_SCATTER(a_chnk_vec, 1, MPI_INTEGER,
  chunk, 1, MPI_INTEGER,
  root, MPI_COMM_WORLD, ierr)

! distribute the data
call MPI_SCATTERV(a, a_chnk_vec, a_disp_vec,
  MPI_DOUBLE_PRECISION,
  local_a, chunk,
  MPI_DOUBLE_PRECISION,
  root, MPI_COMM_WORLD, ierr)

call MPI_SCATTERV(b, b_chnk_vec, b_disp_vec,
  MPI_DOUBLE_PRECISION,
  local_b, chunk/n,
  MPI_DOUBLE_PRECISION,
  root, MPI_COMM_WORLD, ierr)

! forward elimination
do j = 1, (chunk/n)
  xmult = local_a((n-(pivot-1)),j) /
pivot_eqn(pivot)
  do i = (n-pivot), 1, -1
    local_a(i,j) = local_a(i,j)
- (xmult * pivot_eqn(n-(i-1)))
  continue
  local_b(j) = local_b(j) - (xmult * pivot_b)
continue

! restore the data to root
call MPI_GATHERV(local_a, chunk,
  MPI_DOUBLE_PRECISION,
  a, a_chnk_vec, a_disp_vec,
  MPI_DOUBLE_PRECISION,
  root, MPI_COMM_WORLD, ierr)
call MPI_GATHERV(local_b, chunk/n,
  MPI_DOUBLE_PRECISION,
  b, b_chnk_vec, b_disp_vec,
  MPI_DOUBLE_PRECISION,
  root, MPI_COMM_WORLD, ierr)
continue ! end of main loop

```

Waktu yang diperlukan untuk mengeliminasi matriks  $n \times n$ :

n	Total waktu (detik)
400	2.88
800	24.03
1200	90.66

#### IV. KESIMPULAN

Metode penyelesaian sistem persamaan linear menggunakan metode eliminasi Gauss haruslah menggunakan komputer untuk mengolah data yang

banyak. Implementasi algoritma metode eliminasi Gauss bisa secara linear dan juga parallel. Apabila parallel maka iterasi J yang dapat di parallelkan sehingga metode eliminasi Gauss dengan komputasi parallel dapat lebih efektif dari metode komputasi linear.

#### V. UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa oleh karena anugerah-Nya penulis dapat menyelesaikan semua tulisan di makalah ini. Penulis ingin berterima kasih kepada dosen IF 2123 yaitu Pak Rinaldi Munir dan Pak Jundi. Serta penulis juga mengucapkan terima kasih yang tidak terhingga kepada semua teman-teman seperjuangan yang membantu penulis untuk menyelesaikan tulisan ini. Penulis pun tidak lupa mengucapkan terima kasih kepada semua pembaca tulisan ini dan semoga tulisan ini dapat bermanfaat bagi para pembacanya.

#### REFERENSI

- [1] <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2015-2016/Sistem%20Persamaan%20Linier%20-%20sebuah%20ilustrasi.pptx> 15 Desember 2015 pada jam 20:16
- [2] <http://www.sciencedirect.com/science/article/pii/S1319157813000086> 14 Desember 2015 pada jam 19:45
- [3] [http://hpds.ee.kuas.edu.tw/download/parallel\\_processing/96/96present/20071212/Gaussian.pdf](http://hpds.ee.kuas.edu.tw/download/parallel_processing/96/96present/20071212/Gaussian.pdf) 15 Desember 2015 pada jam 21:46
- [4] [http://www.cs.rutgers.edu/~venugopa/parallel\\_summer2012/ge.html](http://www.cs.rutgers.edu/~venugopa/parallel_summer2012/ge.html) 15 Desember 2015 pada jam 19:41
- [5] <https://andri102.wordpress.com/game/soft-skill/konsep-komputasi-parallel-processing/> 15 Desember 2015 pada jam 23:10

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 16 Desember 2015



Christian Anthony Setyawan  
13514085