STUDI KOMPARASI KINERJA ALGORITMA REDUKSI SIKLIS DENGAN ALGORITMA PEMISAHAN REKURSIF PADA SISTEM MULTIPROSESOR BERBASIS PVM

Tri Prabawa

Program Studi Teknik Informatika, STMIK AKAKOM Yogyakarta
Jl. Raya Janti 143, Yogyakarta 55198
E-mail : tprabawa@akakom.ac.id, tprabawamsoetrisno@gmail.com

ABSTRACT

This study discusses the comparison of the performance of the settlement system of linear equations Au = d, where the coefficient matrix A tridiagonal, with a cyclic reduction method and recursive decoupling on multiprocessor systems. The basic idea reduction cyclic reduction method is lowered rows independently by reduction row indexed odd or even, while for the recursive decoupling is lowered rows independent based on a strategy of rank-one updating and the partitioning process repeated on the matrix system in order to obtain the form of a matrix diagonal blocks of 2x2.

Solving problems in parallel system is to perform the decomposition problems in algorithmic or geometric, in order to identify the characteristics of parallelism. The performance characteristics of parallel algorithms can be seen from the measurement execution time, and communication, speed-up, and efficiency. To determine these characteristics will be carried out a comparative study of the results of two studies have been carried out. The first study discusses the results of a cyclic reduction algorithm performance and performance results of a second study on the recursive decoupling algorithms are implemented on a system of parallel virtual machine (PVM), which is a model of a single distributed parallel processor.

From the test results it appears that there is an increase of acceleration as the number of processors used. Acceleration for cyclic reduction algorithms ranged from 1.61 (2 processors) to 4.22 (8 processors), Whereas for the recursive decoupling algorithms separation between 1.61 (2 processors) to 5.90 (8 processors). But on the contrary, by increasing the number of processors used a drop in efficiency. Level of efficiency for cyclic reduction algorithm 88.38% (2 processors) and the lowest was 35.58% (8 processors), whereas for the recursive decoupling algorithm 80.43% (2 processors) and the lowest was 36.25% (8 processors), It is heavily influenced the higher the communication time caused the synchronous process occurs repeatedly.

Key words

tridiagonal system, cyclic reduction, recursive decoupling, speed-up, efficiency, and PVM.

1. PENDAHULUAN

Dalam beberapa aplikasi, penggunaan model matematika menjadi amat populer, karena teknik ini banyak dipakai dalam pemodelan dari pelbagai persoalan nyata. Dengan model matematika bentuk persoalannya menjadi jelas dan sederhana, serta metode dan analisisnya lebih dapat dipertanggung jawabkan. Banyak pemodelan dari suatu fenomena fisik --- seperti mekanika fluida, penjalaran panas dan lain sebagainya --- biasanya memberikan berbagai bentuk persamaan diferensial parsial (PDP).

Pada umumnya, kecuali dalam hal yang amat sederhana, penyelesaian secara analitik dari suatu PDP sulit diperoleh, sehingga perlu dicari solusi numeriknya sebagai alternatif jawaban. Meskipun solusi numerik memerlukan banyak perhitungan, namun pada perkembangannya metode numerik telah memberikan hasil yang berarti, terutama setelah didukung dengan pemakaian perangkat komputer digital. Kebutuhan akan kecepatan penyelesaian masalah menjadi amat penting terutama untuk persoalan yang cukup besar dan kompleks, serta informasinya segera diperlukan.

Diberikan suatu sistem persamaan linier dari sistem tridiagonal Au = d, dengan A matriks tridiagonal dan dapat ditulis sebagai berikut:

$$\begin{pmatrix} b1 & c1 & & & & & 0 \\ a2 & b2 & c2 & & & & \\ & a3 & b3 & c3 & & & \\ & & a4 & b4 & c4 & & \\ & & & & \ddots & \ddots & \\ & & & & & & \\ 0 & & & & & bn-1 & cn-1 \\ & & & & & an & bn \end{pmatrix} \begin{pmatrix} u1 \\ u2 \\ \\ . \\ \\ . \\ un-1 \\ un \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ \\ . \\ \\ . \\ dn-1 \\ dn \end{pmatrix} \qquad (1)$$

Persoalannya adalah mencari vektor u yang memenuhi sistem persamaan Au = d, yang banyak dijumpai sebagai hasil diskritisasi suatu PDP dengan memakai metode beda hingga atau elemen hingga dengan kondisi syarat batas tertentu. Solusi analitis sistem persamaan tersebut ada dan dapat ditulis $u = A^{-1} d$, jika matriks A nonsingular [9]. Sistem persamaan (1) dapat diselesaikan baik secara langsung maupun iteratif. Penyelesaian secara langsung memerlukan $\theta(n^2)$ *flops* sedangkan iteratif $\theta(n)$ / langkah dan menjadi mahal jika terjadi konvergensi yang lamban [7]. Nampak bahwa jika untuk n besar, kedua cara ini memerlukan waktu komputasi yang besar dan kurang efisien. Penyelesaian yang diinginkan dapat dikerjakan secara cepat, dengan kontribusi komputer, dan jika memungkinkan diproses secara paralel.

Komputer paralel adalah suatu perangkat komputer yang mempunyai sejumlah alat pemroses (disebut prosesor) yang saling bekerja sama dalam suatu koordinasi program kendali [1]. Adanya arsitektur seperti ini memungkinkan suatu masalah diselesaikan secara paralel. Dalam menggunakan arsitektur komputer yang demikian maka kecepatan algoritma sangat ditentukan oleh jumlah prosesor yang dipakai serta pola hubungan interkoneksi antara prosesor yang satu dengan yang lain. Model komputer yang sering dipertimbangkan sebagai sistem multiprosesor didefinisikan sebagai SIMD (*single instruction multiple data*) dan MIMD (*multiple instruction multiple data*).

Dalam menggunakan arsitektur komputer yang demikian maka kecepatan algoritma sangat ditentukan oleh jumlah prosesor yang dipakai serta pola interkoneksi antara prosesor yang satu dengan yang lain. Mengingat bahwa komputasi pada sistem multi prosesor akan lebih cepat dibanding dari sistem komputer biasa (yang disebabkan adanya tambahan prosesor), maka perlu didefinisikan suatu besaran yang merupakan ukuran peningkatan kecepatan yang sebenarnya. Besaran ini antara lain adalah peningkatan kecepatan (*speed-up*) dan efisiensi sistem multiprosesor, yang didefinisikan sebagai berikut:

$$E_{(efisisensi)} = T(1) / p\, T(p)$$

$$S_{(speed\text{-}up)} = T(1) / T(p)$$

dimana *T(1)* = waktu yang diperlukan untuk menyelesaikan masalah dengan *1* prosesor, *p* = banyak prosesor yang dipakai, dan *T(p)* = waktu yang diperlukan untuk menyelesaikan masalah dengan *p* prosesor.

Suatu sistem paralel dapat digambarkan sebagai teknik pemrosesan secara simultan pada subproses yang independen. Menurut Hwang dan Briggs, pemrosesan paralel didefinisikan sebagai bentuk pemrosesan yang efisien dengan menitikberatkan pada ekploitasi kejadian-kejadian yang bersamaan [8]. Tujuan utamanya adalah mereduksi waktu proses yang dibutuhkan untuk penyelesaian masalah yang sebelumnya dipandang terlalu besar [6].

Untuk melihat proses-proses yang dapat dikerjakan secara simultan, langkah pertama adalah melakukan proses dekomposisi persoalan sehingga diperoleh bagian-bagian yang independen atau mencari letak paralelisasi dari suatu permasalahan. Ada dua macam cara melakukan dekomposisi masalah, yaitu: (i) dekomposisi secara algoritmis dan (ii) dekomposisi secara geometrik [2]. Dekomposisi secara algoritmis adalah dekomposisi algoritma sekuensial yang ada atas beberapa blok instruksi, dimana tiap blok instruksi akan dikerjakan oleh prosesor yang berbeda. Strategi ini biasanya cenderung melihat alur pemecahan masalah yang dihadapi. Sedangkan dekomposisi secara geometrik adalah dekomposisi masalah yang ada menjadi beberapa submasalah dengan aturan tertentu sehingga tiap submasalah bisa dipecahkan secara terpisah dan paralel. Strategi ini biasanya cenderung melihat struktur data dari persoalan yang dihadapi.

Untuk memecahkan permasalahan sistem tridiagonal, digunakan strategi dekomposisi secara geometrik, karena lebih banyak melihat pada masalah struktur data dari persoalan yang dihadapi, yaitu bagaimana mendekomposisi data atas beberapa kelompok data yang akan diproses oleh prosesor yang berbeda. Selain itu untuk mendapatkan model algoritma paralel dapat ditempuh dengan cara memodifikasi algoritma sekuensial, sehingga akan diperoleh bentuk algoritma paralel. Sedangkan untuk memperoleh gambaran kinerjanya, algoritma paralel perlu dibandingkan dengan algoritma sekuensial. Berdasarkan dari pertimbangan ini maka akan dibahas algoritma reduksi siklis sekuensial dan teknik dekomposisi.

2. SISTEM PERSAMAAN LINIER

a. Metode Reduksi Siklis Untuk Sistem Tridiagonal

Karena matriks koefisien memiliki struktur yang spesifik, maka dimungkinkan pemakaian metode lain yang penyelesaiannya dapat dikerjakan secara paralel. Metode yang dibahas adalah metode reduksi siklis. Untuk menyederhanakan persoalan pada kasus ini akan ditinjau elemen matriks berupa tridiagonal konstan simetri. Maka dengan tidak menghilangkan sifat umum sistem tersebut, sistem (1) dapat ditulis sbb :

$$\begin{pmatrix} b & a & & & & & \\ a & b & a & & & 0 & \\ & a & b & a & & & \\ & & a & b & a & & \\ & & & & \ddots & \ddots & \\ 0 & & & & a & b & a \\ & & & & & a & b \end{pmatrix} \begin{pmatrix} u1 \\ u2 \\ \cdot \\ \cdot \\ un-1 \\ un \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ \cdot \\ \cdot \\ dn-1 \\ dn \end{pmatrix} \qquad (2)$$

ide dasar metode reduksi siklis adalah menurunkan baris-baris independen dengan cara reduksi pada baris yang berindeks ganjil atau genap. Metode ini cocok dikembangkan untuk matriks berukuran $n = 2^m-1,\ 2^m,\ 2^m+1$.

Secara umum prosedur metode reduksi siklis dapat diturunkan sebagai berikut:

Perhatikan 3 baris yang berdekatan, yaitu baris ke i-1, i, dan i+1

$$\begin{array}{lccccc} i-1 & a & b & a & & \\ i & & a & b & a & \\ i+1 & & & a & b & a \end{array} \qquad (3)$$

secara singkat bentuk (3) dapat dinyatakan dengan (…… a, b, a, ……). Dengan melakukan operasi baris elementer terhadap baris tengahnya, yaitu baris i dikalikan dengan konstanta b, kemudian ditambahkan dengan –a kali jumlahan baris ke i-1 dan i+1, maka diperoleh bentuk :

$$\begin{array}{lcccccc} i-1 & a & b & a & & & \\ i & a^{[1]} & 0 & b^{[1]} & 0 & a^{[1]} & \\ i+1 & & & a & b & a & \end{array} \qquad (4)$$

dengan memperhatikan baris tengah (4) diperoleh sistem persamaan baru yang dapat dinyatakan (............. $a^{[1]}$, 0, $b^{[1]}$, 0, $a^{[1]}$ ....).

Bentuk ini merupakan sistem tridiagonal dengan $n = 2^m-1,\ 2^m,\ 2^m+1$, yang unsur-unsurnya hanya terdiri dari baris-baris berindeks ganjil atau genap saja dari matriks awal A. Operasi ini juga dikerjakan terhadap elemen-elemen pada vektor ruas kanan d.

Misalkan dari bentuk (4) sistem tridiagonal baru yang diperoleh terdiri dari baris genap. Sedangkan baris ganjil dapat dieliminasi oleh baris genap dengan substitusi balik. Jika proses tersebut dilakukan berulang-ulang, akhirnya didapat sistem persamaan baru yang minimal, yang solusinya segera dapat dihitung. Untuk lebih jelasnya perhatikan contoh berikut

$$\begin{array}{l} 1 \\ 2 \\ 3 \end{array} \begin{pmatrix} b & a & \\ a & b & a \\ & a & b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \qquad (5)$$

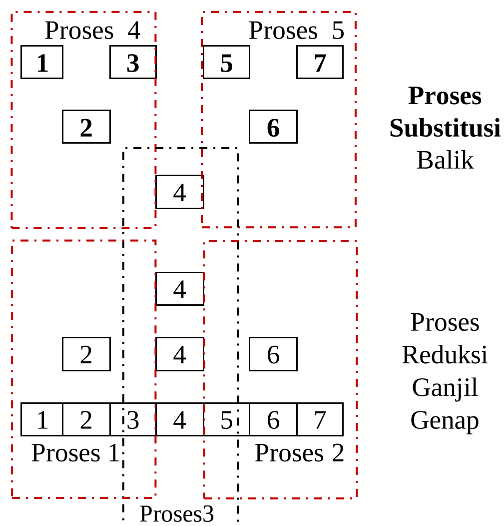
Lakukan proses operasi baris elementer pada persamaan (5), sehingga diperoleh hasil

$$\begin{array}{l} 1 \\ 2 \\ 3 \end{array} \begin{pmatrix} b & a & \\ 0 & b' & 0 \\ & a & b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} d_1 \\ d'_2 \\ d_3 \end{pmatrix} \qquad (6)$$

Dari persamaan ini diperoleh hasil $u_2 = d'_2/b'$, selanjutnya nilai u_1 dan u_3 dapat dihitung (secara paralel) dengan substituís balik, yaitu $u_1 = (d_1 - a\ u_2)/b$ dan $u_3 = (d_3 - a\ u_2)/b$.

Metode reduksi siklis memberikan komputasi yang cepat jika elemen subdiagonal dan superdiagonal bernilai satu. Hal ini dapat diperoleh dengan melakukan normalisasi, sehingga sistem persamaan berbentuk (.......... 1, b, 1, ...).

Secara umum tahapan solusi metode reduksi siklis terdiri dari dua tahapan utama yaitu (i) proses reduksi baris ganjil atau genap, dan (ii) proses substitusi balik.



Gambar 1. Teknik Dekomposisi Metode Reduksi Siklis ($n = 2^m-1$, m=3)

b. Metode Pemisahan Rekursif Untuk Sistem Tridiagonal

Metode pemisahan rekursif berdasarkan pada strategi perubahan rank-satu (*rank-one updating*) dan proses partisi berulang pada sistem matriks sehingga didapat bentuk matriks diagonal blok yang masing-masing blok berukuran 2x2. Metode ini akan dipakai untuk menyelesaikan bentuk matriks tridiagonal sembarang, dan untuk sederhananya dipilih matriks dengan ukuran $n = 2^k$ dengan k = bulat positif.

Proses Partisi

Pandang sistem persamaan (1) yang selanjutnya dinyatakan sebagai :

$$\begin{pmatrix} b1 & c1 & & & & \\ a2 & b2 & c2 & & 0 & \\ & a3 & b3 & c3 & & \\ & & a4 & b4 & c4 & \\ & & & & \ddots & \\ 0 & & & & bn-1 & cn-1 \\ & & & & an & bn \end{pmatrix} \begin{pmatrix} u1 \\ u2 \\ \cdot \\ \cdot \\ un-1 \\ un \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ \cdot \\ \cdot \\ dn-1 \\ dn \end{pmatrix} \qquad (7)$$

Dengan $|bi| \geq |ai| + |ci|$ untuk setiap i = 1, 2, 3, ...n, syarat ini menjamin proses reduksinya berjalan stabil. Proses partisi dilakukan terhadap matriks koefisien A sedemikian sehingga diperoleh bentuk blok matriks berukuran 2x2 sebagai berikut

$$\begin{pmatrix} e1 & c1 & & & & \\ a2 & e2 & & & 0 & \\ & & e3 & c3 & & \\ & & a4 & e4 & & \\ & & & & & \\ 0 & & & & en-1 & cn-1 \\ & & & & an & en \end{pmatrix} + \sum_{j=1}^{m-1} \begin{pmatrix} x1 \\ x2 \\ \cdot \\ \cdot \\ xn-1 \\ xn \end{pmatrix}^{(j)} x \begin{pmatrix} y1 \\ y2 \\ \cdot \\ \cdot \\ yn-1 \\ yn \end{pmatrix}^{(j)T} \qquad (8)$$

Dengan $e_1 = b_1$, $e_{2j-1} = b_{2j-1} - c_{2j-2}$, j = 2, 3, ...m

$e_n = b_n$, $e_{2j} = b_{2j} - a_{2j+1}$, j =1, 2, ...m-1, dan m = n/2

Sedangkan vektor $x^{(j)}$ dan $y^{(j)}$merupakan vektor dengan elemen tidak nol hanya pada elemen ke 2j dan 2j+1 yang didefinisikan sebagai

$$x_k = \begin{cases} 1, untuk\ k = 2j, 2j+1 \\ 0,\ untuk\ yang\ lain \end{cases} \qquad (9)$$

$$y_k = \begin{cases} a2j+1,\ untuk\ k = 2j \\ c2j, untuk\ k = 2j+1 \\ 0,\ untuk\ yang\ lain \end{cases} \qquad (10)$$

Atau dapat ditulis sebagai

$x(j) = (0, 0, ..., 1, 1, ...0)^T$ (11)

$y(j) = (0, 0, ..., a_{2j+1}, c_{2j}, ...0)^T$ (12)

Persamaan (3) secara sederhana dapat ditulis menjadi $A = J + x^{(j)} y^{(j)T}$

Dengan J berbentuk

$$J = \begin{pmatrix} J1 & & & & \\ & J2 & & & \\ & & J3 & & \\ & & & Jm & \\ & & & & Jm \end{pmatrix} \qquad (13)$$

Sedangkan J1, J2, ...Jm terdiri dari submatrik 2x2 berbentuk

$\begin{pmatrix} e2j-1 & c2j-1 \\ a2j & e2j \end{pmatrix}$ dengan j = 1, 2, ...m dan m = n/2

Ide dasar partisi ini mengacu pada metode inversi Sherman-Morrison yang bertujuan untuk mereduksi kompleksitas komputasi A^{-1} secara langsung. Menurut Golub [7] metode inversi Sherman-Morrison memberikan formula perhitungan invers matrik A berbentuk $A = (J + uv^T)$ dengan A elemen R^{nxn} , Sedangkan u dan v vektor berukuran nx1, Invers matrik A didefinisikan sebagai

$$A^{-1} = (J + uv^T)^{-1} = J^{-1} - (J^{-1} u\ v\ J^{-1}) / (1 + v^T J^1 u) \qquad (14)$$

Dengan memakai persamaan tsb, maka invers matrik $A = J + xy^T$ adalah

$$A^{-1} = (J + xy^T) = J^{-1} - \alpha (J^{-1} x) (y^T J^1) \qquad (15)$$

Dengan $\alpha = 1/ (1 + y^T J^{-1} x)$

Kemudian dengan menghitung invers A, solusi persamaan (2) adalah

$$\begin{aligned} U &= A^{-1} d = (J + xy^T)^{-1} d \\ &= \{J^{-1} - \alpha (J^{-1} x) (y^T J^1)\} d \\ &= J^{-1} d - \alpha\ J^{-1} x\ y^T J^1 d \\ &= J^{-1} d - \alpha\ (J^{-1} x)\ y^T (J^{-1} d) \qquad (16) \end{aligned}$$

dengan $\alpha = 1/ (1 + y^T J^{-1} x)$

Dengan menghitung nilai-nilai (11), yaitu $J^{-1}x$, J^{-1} d dan α diperoleh solusi dar sistem persamaan (2).

Proses Pemisahan Rekursif.

Proses ini bertujuan mengubah vektor u dan d sedemikian hingga elemen-elemennya menjadi berpasangan (*couple*) yang dapat ditulis

$$u = \begin{pmatrix} \left.\begin{matrix} u1 \\ u2 \end{matrix}\right\} \\ \left.\begin{matrix} u3 \\ u4 \end{matrix}\right\} \\ \cdot \\ \cdot \\ \left.\begin{matrix} un-1 \\ un \end{matrix}\right\} \end{pmatrix} = \begin{pmatrix} u1 \\ u2 \\ \cdot \\ \cdot \\ dm \end{pmatrix} \qquad (17)$$

$$dan\ \ d = \begin{pmatrix} \left.\begin{matrix} d1 \\ d2 \end{matrix}\right\} \\ \left.\begin{matrix} d3 \\ d4 \end{matrix}\right\} \\ \cdot \\ \cdot \\ \left.\begin{matrix} dn-1 \\ dn \end{matrix}\right\} \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ \cdot \\ \cdot \\ dm \end{pmatrix} \qquad (18)$$

dari persamaan (16) solusi Au = d dapat dinyatakan

$$u = (J + \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1} d$$

$$= (J^{-1} - \alpha J^{-1} \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1}) d$$

$$= (Jd - \alpha J^{-1} \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1} d \qquad (19)$$

dengan

$$\alpha = 1/(1 + \sum_{j=1}^{m-1} y^{(j)T})J^{-1} \sum_{j=1}^{m-1} x^{(j)})$$

Dengan menyatakan J^{-1} d’ = u’ dan $J^{-1} x^{(j)} = g^{(j)}$ maka bentuk (19) dapat disederhanakan menjadi

$$u = (u' - \alpha g^{(j)} y^{(j)T} u') = (1 - \alpha g^{(J)} (y^{(j)T}) u' \quad \ldots(20)$$

dengan $\alpha = 1/(1 + y^{(j)T} g^{(j)})$ dan j = 1, 2, ….m-1

persamaan (15) merupakan bentuk formula perubahan rank satu, maka tahapan solusi untuk persamaan diatas adalah

(a) Mencari solusi vektor u’ dari persamaan $u' = J^{-1} d$

(b) Mencari solusi vektor $g^{(j)}$ dari persamaan $g^{(j)} = J^{-1} x^{(j)}$

(c) melakukan perubahan rank satu (20)

Sedangkan vektor $x^{(j)}$ dan $y^{(j)}$merupakan vektor dengan elemen tidak nol hanya pada elemen ke 2j dan 2j+1 yang didefinisikan sebagai

$$x_k = \begin{cases} 1, \text{untuk } k = 2j, 2j+1 \\ 0, \text{ untuk yang lain} \end{cases} \qquad (9)$$

$$y_k = \begin{cases} a2j+1, \text{ untuk } k = 2j \\ c2j, \text{untuk } k = 2j+1 \\ 0, \text{ untuk yang lain} \end{cases} \qquad (10)$$

Atau dapat ditulis sebagai

$x(j) = (0, 0, \ldots\ldots, 1, 1, \ldots\ldots\ldots 0)^T$ (11)

$y(j) = (0, 0, \ldots\ldots, a_{2j+1}, c_{2j}, \ldots\ldots\ldots 0)^T$ (12)

Persamaan (3) secara sederhana dapat ditulis menjadi $A = J + x^{(j)} y^{(j)T}$

Dengan J berbentuk

$$J = \begin{pmatrix} J1 & & & & \\ & J2 & & & \\ & & J3 & & \\ & & & Jm & \\ & & & & Jm \end{pmatrix} \qquad (13)$$

Sedangkan J1, J2, ...Jm terdiri dari submatrik 2x2 berbentuk

$\begin{pmatrix} e2j-1 & c2j-1 \\ a2j & e2j \end{pmatrix}$ dengan j = 1, 2, ...m dan m = n/2

Ide dasar partisi ini mengacu pada metode inversi Sherman-Morrison yang bertujuan untuk mereduksi kompleksitas komputasi A^{-1} secara langsung. Menurut Golub [7] metode inversi Sherman-Morrison memberikan formula perhitungan invers matrik A berbentuk $A = (J + uv^T)$ dengan A elemen R^{nxn}, Sedangkan u dan v vektor berukuran nx1, Invers matrik A didefinisikan sebagai

$$A^{-1} = (J + uv^T)^{-1} = J^{-1} - (J^{-1} u\, v\, J^{-1}) / (1 + v^T J^1 u) \qquad (14)$$

Dengan memakai persamaan tsb, maka invers matrik $A = J + xy^T$ adalah

$$A^{-1} = (J + xy^T) = J^{-1} - \alpha (J^{-1} x)(y^T J^1) \qquad (15)$$

Dengan $\alpha = 1/(1 + y^T J^{-1} x)$

Kemudian dengan menghitung invers A, solusi persamaan (2) adalah

$$\begin{aligned} U &= A^{-1} d = (J + xy^T)^{-1} d \\ &= \{J^{-1} - \alpha (J^{-1} x)(y^T J^1)\} d \\ &= J^{-1} d - \alpha\, J^{-1} x\, y^T J^1 d \\ &= J^{-1} d - \alpha\, (J^{-1} x)\, y^T (J^{-1} d) \qquad (16) \end{aligned}$$

dengan $\alpha = 1/(1 + y^T J^{-1} x)$

Dengan menghitung nilai-nilai (11), yaitu $J^{-1}x$, J^{-1} d dan α diperoleh solusi dar sistem persamaan (2).

Proses Pemisahan Rekursif.

Proses ini bertujuan mengubah vektor u dan d sedemikian hingga elemen-elemennya menjadi berpasangan (*couple*) yang dapat ditulis

$$u = \begin{pmatrix} \left.\begin{matrix} u1 \\ u2 \end{matrix}\right\} \\ \left.\begin{matrix} u3 \\ u4 \end{matrix}\right\} \\ . \\ . \\ \left.\begin{matrix} un-1 \\ un \end{matrix}\right\} \end{pmatrix} = \begin{pmatrix} u1 \\ u2 \\ . \\ . \\ dm \end{pmatrix} \qquad (17)$$

$$\text{dan } d = \begin{pmatrix} \left.\begin{matrix} d1 \\ d2 \end{matrix}\right\} \\ \left.\begin{matrix} d3 \\ d4 \end{matrix}\right\} \\ . \\ . \\ \left.\begin{matrix} dn-1 \\ dn \end{matrix}\right\} \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ . \\ . \\ dm \end{pmatrix} \qquad (18)$$

dari persamaan (16) solusi Au = d dapat dinyatakan

$$u = (J + \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1} d$$

$$= (J^{-1} - \alpha J^{-1} \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1}) d$$

$$= (Jd - \alpha J^{-1} \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1} d \qquad (19)$$

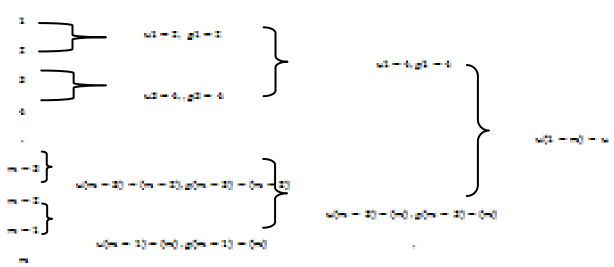
dengan

$$\alpha = 1/(1 + \sum_{j=1}^{m-1} y^{(j)T}) J^{-1} \sum_{j=1}^{m-1} x^{(j)})$$

Dengan menyatakan J^{-1} d' = u' dan $J^{-1} x^{(j)} = g^{(j)}$ maka bentuk (19) dapat disederhanakan menjadi

$$u = (u' - \alpha g^{(j)} y^{(j)T} u') = (1 - \alpha g^{(J)} (y^{(j)T}) u' \quad \ldots(20)$$

dengan $\alpha = 1/(1 + y^{(j)T} g^{(j)})$ dan j = 1, 2, ….m-1
persamaan (15) merupakan bentuk formula perubahan rank satu, maka tahapan solusi untuk persamaan diatas adalah
(d) Mencari solusi vektor u' dari persamaan $u' = J^{-1} d$
(e) Mencari solusi vektor $g^{(j)}$ dari persamaan $g^{(j)} = J^{-1} x^{(j)}$
(f) melakukan perubahan rank satu (20)



Gambar 2. Teknik Dekomposisi Metode Pemisahan Rekursif ($n = 2^m$)

3. HASIL DAN ANALISIS

Untuk mengukur karakteristik kinerja algoritma, selanjutnya algoritma reduksi siklis dan pemisahan rekursif diimplementasikan pada mesin paralel berbasis PVM dengan melibatkan prosesor sebanyak 1, 2, 4, dan 8 prosesor. Kedua algoritma tersebut dipakai untuk menyelesaikan sistem Au = d, dengan elemen A matriks tridiagonal konstan simetri. Agar diperoleh hasil yang cukup baik dan signifikan dipakai beberapa ukuran data matriks yang relatif besar agar waktu komputasi dapat dihitung dengan jelas. Ada 7 ukuran data yang dipakai, yaitu n = 512, 1024, 2048, 4096, 8192, 16384, 32768 (ukuran data lebih besar dan banyak akan lebih baik).

Secara analitik komputasi, metode reduksi siklis cenderung *fine grain,* yaitu rasio komputasi dan komunikasi yang kecil. Hal ini disebabkan karena faktor granularitas, efek *fork and joint* dan proses sinkronisasi berulang-ulang, sedangkan pada pada metode pemisahan rekursif tidak demikian.

Berdasarkan hasil uji coba pada mesin paralel berbasis PVM, pengukuran waktu hasil eksekusi dan karakteristik kinerja algoritma reduksi siklis dan pemisahan rekursif dapat disajikan pada tabel 1, tabel 2, tabel 3, dan tabel 4.

Tabel 1 Presentasi Pengukuran Waktu Eksekusi Algoritma Reduksi Siklis (RS).

NO	Ukuran Data	Waktu eksekusi Algoritma RS (milidetik)			
		P=1	P=2	P=4	P=8
1	512	280	168	114	85
2	1024	540	317	206	149
3	2048	1057	610	388	274
4	4096	2089	1196	750	520
5	8192	4148	2362	1468	1006
6	16384	8260	4691	2901	1975
7	32768	16513	9342	5766	3911

Tabel 2 Presentasi Pengukuran Waktu Eksekusi Algoritma Pemisahan Rekursif (PR).

NO	Ukuran Data	Waktu eksekusi Algoritma PR (milidetik)			
		P=1	P=2	P=4	P=8
1	512	280	168	114	96
2	1024	712	379	249	191
3	2048	1817	938	575	421
4	4096	4639	2440	1421	989
5	8192	13886	6694	3766	2502
6	16384	33107	16554	9094	5821
7	32768	78281	39141	21180	13257

Tabel 3 Presentasi Perhitungan *Speed-up* Algoritma Reduksi Siklis dan Pemisahan Rekursif

NO	Ukur Data	*Speed-up* Algoritma RS (Sp = T1/Tp)				***Speed-up*** **Algoritma PR (Sp = T1/Tp)**			
		P=1	P=2	P=4	P=8	P=1	P=2	P=4	P=8
1	512	1	1,67	2,46	3,29	1	1,61	2,44	2,90
2	1024	1	1,70	2,62	3,62	1	1,88	2,86	3,73
3	2048	1	1,73	2,72	3,86	1	1,94	3,16	4,32
4	4096	1	1,75	2,79	4,02	1	1,98	3,41	4,89
5	8192	1	1,76	2,83	4,12	1	2,00	3,55	5,35
6	16384	1	1,76	2,85	4,18	1	2,00	3,64	5,69
7	32768	1	1,77	2,86	4,22	1	2,00	3,70	5,90

Tabel 4 Presentasi Perhitungan Efisiensi Algoritma Reduksi Siklis dan Pemisahan Rekursif

N	Ukur	Efisiensi Algoritma RS (Sp/p x 100%)				Efisiensi Algoritma PR (Sp/p x 100%)			
O	Data	P=1	P=2	P=4	P=8	P=1	P=2	P=4	P=8
1	512	100	83,33	61,40	41,18	100	80,43	61,00	36,25
2	1024	100	85,17	65,53	45,30	100	94,40	71,50	46,63
3	2048	100	86,64	68,11	48,22	100	98,17	79,00	54,00
4	4096	100	87,33	69,63	50,22	100	99,64	85,25	61,13
5	8192	100	87,81	70,64	51,54	100	100	88,75	66,88
6	16384	100	88,04	71,18	52,28	100	100	98,00	71,13
7	32768	100	88,38	71,60	52,78	100	100	92,50	73,75

Dari tabel 1, tabel 2, tabel 3,dan tabel 4 dapat dijelaskan bahwa hasil uji coba terlihat bahwa terjadi kenaikan percepatan seiring dengan bertambahnya jumlah prosesor yang dipakai. Percepatan untuk algoritma reduksi siklis berkisar antara 1,61 (2 prosesor) sampai dengan 4,22 (8 prosesor), sedangan untuk algoritma pemisahan rekursif antara 1,61 (2 prosesor) sampai dengan 5,90 (8 prosesor).

Namun sebaliknya, dengan bertambahnya jumlah prosesor yang dipakai terjadi penurunan efisiensi. Tingkat efisiensi untuk algoritma reduksi siklis mencapai 88,38% (2 prosesor) dan terendah 35,58% (8 prosesor), sedangan untuk algoritma pemisahan rekursif mencapai 80,43% (2 prosesor) dan terendah 36,25% (8 prosesor). Hal ini banyak dipengaruhi waktu komunikasi semakin tinggi yang disebabkan proses sinkronsasi terjadi berulang-ulang.

4. KESIMPULAN

Berdasarkan uraian di atas dapat disimpulkan beberapa hal berikut:

(1) Dengan menggunakan sistem multiprosesor waktu penyelesaian sistem persamaan linier dari hasil diskritisasi suatu PDP dapat dipercepat.

(2) Pada metode reduksi siklis faktor granularitas, efek *fork* dan *join*, dan proses sinkronisasi akan berakibat dalam waktu komputasi, hal ini disebabkan karena memiliki unsur dependensi data yang kuat dan jumlah komputasi tiap level berikutnya menurun.

(3) Kinerja algoritama reduksi siklis mencapai percepatan tertinggi pada 8 prosesor (4,22) dan efisiensi terendahnya terjadi pada 8 prosesor (35,58%). Sedangkan Kinerja algoritma pemisahan rekursif mencapai percepatan tertinggi pada 8 prosesor (4,22) dan efisiensi terendahnya terjadi pada 8 prosesor (35,58%).

REFERENSI

[1] Akl, Selim G. 1989. *The Design and Analysis of Parallel Algorithms*, Prentice Hall International Inc.

[2] Askew, C.R., Carpenter, D.B., Chalker, J.T., Hey, A.J.G., Moore, M., Nicole, D.A, and Pritchard, D.J., 1988. *Monte Carlo Simulation on transputer arrays*. Parallel Computing 6, pp 247-258.

[3] Berstsekas and Tsitsiklis, 1989, *Parallel and Distributed Computation, Numerical Methods*, Prentice Hall New Jersey.

[4] Evans, DJ., 1990, *A Recursive Decoupling Method for Solving Tridiagonal Linier Systems*, International Journal Computer Mathematics.

[5] Evans, DJ., 1992, *Design of Parallel Numerical Algorithms*, Elsevier Science Publisher.

[6] Freman and Phillips, 1992, *Parallel Numerical Algorithms*, Prentice Hall, London

[7] Golub and Van Loan, 1989, *Matrix Computation*, Second Edition, The John Hopkins University Press

[8] Hwang, Kai and Briggs, FA., 1984. *Computer Architecture and Parallel Pocressing*. McGraw-Hill. Book Company

[9] Mitchell and Griffiths, 1989, *The Finite Difference Method in partial Differetial Equations*, John Wiley & Sons

[10] Tanembaum, 2002, *Structured Computer Organization*, Prentice Hall International Inc.

[11] Tri Prabawa, 2013, *Analisis Kinerja Algoritma Reduksi Siklis untuk Sistem Persamaan Linier dengan Matriks Tridiagonal berbasis PVM*. Proceeding Seminar Nasional Riset Teknologi Informasi STMIK Akakom Yogyakarta.

[12] Tri Prabawa, 2015, *Karakteristik Kinerja Algoritma Rekursive Decoupling pada Multiprosesor berbasis PVM*. Proceeding Seminar Nasional Teknologi Informasi dan Multimedia 2015, tanggal 6-8 Februari 2015. STMIK Amikom Yogyakarta.

[Biodata Penulis]

Tri Prabawa, menyelesaikan studi S1 bidang Matematika, Universitas Gadjah Mada (1986) dan S2 bidang Ilmu Komputer, Universitas Indonesia (1993). Dosen Prodi Teknik Informatika STMIK Akakom Yogyakarta (1994 – sekarang)