

Komputasi Paralel Menggunakan Model Message Passing Pada SIM RS (Sistem Informasi Manajemen Rumah Sakit)

I Putu Adi Pradnyana Wibawa¹, I.A. Dwi Giriantari², Made Sudarma³

Abstract— The development of increasingly rapid the data can go beyond the capabilities of the device database management. One example of a system that has a high data complexity is the SIM RS (hospital management information system). In this study a patient's data on a SIM RS to be used as an object model using Parallel implementations of message-passing. Parallel computing is designed in a way to divide the execution data to a number of computer/CPU. The testing will be done by comparing the data processing time between sequential and parallel. In addition, parallel will be tested using the calculation Speed Up and Efficiency. The test results proved that the data processing time patients using parallel programs faster than using sequential topology. On speed up testing showed an increased speed until the use of 3 computers/CPU. While in testing the efficiency of the highest efficiency values contained on the use of 2 and 3 computers/CPU. The decline in the value of the speed up and efficiency due to the amount of data that belongs to a little when handled by 7 computers/CPU. So the greater number of computer/CPU is involved, it is not directly proportional to the time taken in processing/data processing. This is due to a job processing/data processing in terms of the amount of data handled has a limited number of ideal computer/CPU to handle the job.

Intisari— Perkembangan data yang semakin pesat dapat melampaui batas kemampuan perangkat manajemen database. Salah satu contoh sistem yang memiliki kompleksitas data yang tinggi adalah SIM RS (Sistem Informasi Manajemen Rumah Sakit). Dalam penelitian ini data pasien pada SIM RS akan digunakan sebagai objek implementasi Komputasi Paralel menggunakan model *message-passing*. Komputasi paralel didesain dengan cara membagi eksekusi data ke sejumlah komputer/CPU. Pengujian akan dilakukan dengan membandingkan waktu pengolahan data antara sekuensial dan paralel. Selain itu komputasi paralel akan diuji menggunakan perhitungan *Speed Up* dan Efisiensi. Hasil pengujian membuktikan bahwa waktu pengolahan data pasien menggunakan program paralel lebih cepat dibandingkan menggunakan topologi sekuensial. Pada pengujian *speed up* menunjukkan peningkatan kecepatan sampai pada penggunaan 3 komputer/CPU. Sedangkan pada pengujian efisiensi nilai efisiensi tertinggi terdapat pada penggunaan 2 dan 3 komputer/CPU. Penurunan nilai *speed up* dan efisiensi diakibatkan oleh jumlah data yang tergolong sedikit apabila ditangani oleh 7 komputer/CPU. Jadi semakin banyak jumlah komputer/CPU yang dilibatkan, tidak berbanding lurus dengan waktu yang dibutuhkan dalam pemrosesan/ pengolahan data tersebut. Hal ini dikarenakan sebuah pekerjaan

pemrosesan/pengolahan data dalam hal jumlah data yang ditangani memiliki batas ideal jumlah komputer/CPU yang menangani pekerjaan tersebut.

Kata Kunci— Komputasi Paralel, *Message-passing*, Metode Foster, SIM RS

I. PENDAHULUAN

Perkembangan teknologi informasi meliputi perkembangan infrastruktur pendukungnya, yaitu *hardware*, *software*, teknologi komunikasi dan teknologi penyimpanan data. Semakin berkembangnya teknologi akan berdampak pada pertumbuhan data yang berlipat ganda dari waktu ke waktu. Data menjadi semakin luas dan kompleks hingga melampaui batas kemampuan aplikasi pemroses ataupun database manajemen *tools* yang ada, dilatarbelakangi hal tersebut munculah istilah *Big Data*. Kemunculan *big data* seolah menjadi permasalahan yang cukup kompleks dihadapi oleh organisasi pengguna sistem dengan tingkat pertumbuhan data yang cepat dan pengolahan data yang besar.

Salah satu sistem yang memiliki jumlah dan jenis data yang cukup besar adalah SIM RS. Berdasarkan ketentuan Pasal 52 ayat (1) Undang- Undang Nomor 44 Tahun 2009 tentang Rumah Sakit, setiap rumah sakit wajib melakukan pencatatan dan pelaporan semua kegiatan penyelenggaraan rumah sakit dalam bentuk SIM RS. Pemanfaatan SIM RS yang telah berjalan optimal akan menjadikan jumlah data semakin besar terutama data dengan jenis teks berupa angka (nomor rekam medis pasien), dan huruf (biodata pasien). Untuk memunculkan *output* dari data-data tersebut dibutuhkan proses komputasi dan pengolahan data dalam jumlah besar, yang sudah barang tentu kinerja server akan sangat berat sehingga tidak menutup kemungkinan terjadinya *hang* akibat kelebihan beban. *HPC (High Performance Computing)* merupakan metode untuk mengatasi permasalahan yang memiliki kompleksitas tinggi terkait dengan beban pekerjaan dan penggunaan banyak data [9]. Salah satu teknik yang digunakan dalam metode *HPC* adalah Komputasi Paralel. Komputasi paralel adalah salah satu teknik melakukan komputasi secara bersamaan dengan memanfaatkan beberapa komputer independen secara bersamaan.

Pemanfaatan komputasi paralel banyak digunakan dalam berbagai pemecahan masalah pada berbagai bidang ilmu. Pada [10] melakukan penelitian mengenai pengolahan gambar digital menggunakan komputasi paralel. Pendekatan yang digunakan adalah menggunakan CUDA sebagai alat pemrograman paralel pada GPU untuk mengambil *resources* dari semua *core* yang tersedia. Pada [18] melakukan proses enkripsi citra menggunakan algoritma *Advanced Encryption Standard* (AES). Untuk mempercepat waktu enkripsi diterapkan komputasi paralel. Perangkat komputasi paralel

¹Mahasiswa, Program Studi Magister Teknik Elektro, Jalan Dewi Supraba VI. No.23 ,Denpasar Bali INDONESIA (telp:081916153335; e-mail: adipwgirondins@gmail.com)

^{2, 3} Dosen, Program Studi Magister Teknik Elektro, Jalan Panglima Besar Sudirman, Denpasar Bali Indonesia (telp: 0361-239599; fax: 0361-239599; e-mail: dayu.giriantari@unud.ac.id, imasudarma@gmail.com)



yang digunakan adalah *Java Parallel Programming Framework (JPPF)* yang berarsitektur *master/slave*. Selain membahas mengenai pemanfaatan komputasi paralel, terdapat pula penelitian yang mengkaji perbandingan perangkat bantu pemrograman paralel. Pada [5] dilakukan perbandingan serta analisis kemampuan komputasi paralel antara perangkat bantu *OpenMP* dan *MPI (Message Passing Interface)*. Berdasarkan penelitian yang dilakukan tersebut, disimpulkan bahwa komputasi paralel menggunakan *MPI* memiliki kemampuan yang baik saat digunakan dalam metode paralel terhadap sistem dengan kompleksitas data yang besar.

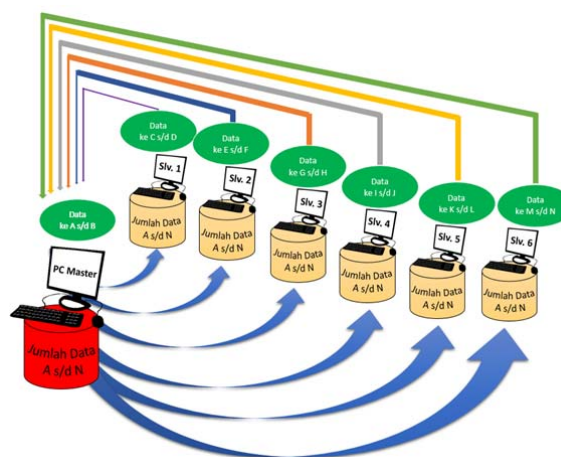
Dalam Penelitian ini pemanfaatan komputasi paralel menggunakan *MPI* diharapkan mampu meningkatkan kemampuan komputasi sehingga mampu menjadi alternatif solusi bagi permasalahan beban data dan kecepatan waktu komputasi yang terjadi pada SIM RS. Hal ini memberikan motivasi dalam perancangan komputasi paralel dengan melakukan pembagian eksekusi data kepada beberapa komputer/*CPU* yang saling terhubung. Proses komputasi paralel bagi data dilakukan karena jumlah data pada SIM RS cukup besar dan antara satu data dengan data yang lainnya tidak saling berkaitan. Dengan menggunakan model *message-passing* pada *library MPI* pengolahan data difokuskan pada proses pencarian data manajemen rumah sakit. Sebuah komputer/*CPU* akan difungsikan sebagai *master* dimana proses komputasi akan dibantu oleh beberapa komputer/*CPU* yang akan berfungsi sebagai *slave*. Pembagian dan eksekusi data (paralel bagi data) kepada masing-masing komputer/*CPU slave* akan dilakukan oleh komputer/*CPU master*. Dengan demikian sumber daya komputer yang tersedia dapat dimanfaatkan semaksimal mungkin untuk membantu kinerja komputer/*CPU master*. *Speedup* dan efisiensi digunakan untuk mengukur seberapa baik arsitektur paralel yang diterapkan.

II. METODE PENELITIAN

Komputasi Paralel merupakan metode komputasi yang membagi beban komputasi ke dalam beberapa bagian kecil sub proses komputasi, dimana sub komputasi tersebut dijalankan pada *processor* yang berbeda secara bersamaan dan saling berinteraksi satu sama lain dalam menyelesaikan satu permasalahan komputasi. Salah satu protokol dalam pemrograman paralel adalah *MPI* yang dikembangkan dalam skema *distributed memory* [12]. *MPI* mengijinkan pertukaran data (*message*) antara *processor*. Dalam penelitian ini komputasi paralel difokuskan pada pengolahan/pencarian data pasien pada data base SIM RS. Komputasi paralel didesain untuk mengurangi waktu komputasi saat melakukan proses pengolahan data manajemen rumah sakit pada SIM RS.

Pada Gambar 1. digambarkan bahwa proses pengolahan data yang sebelumnya dilakukan secara sekuensial (satu data dalam satu waktu dan satu komputer/*CPU*) akan dikerjakan secara bersamaan dengan beberapa komputer/*CPU*. Dalam proses kerja Komputasi Paralel menggunakan *MPI* membutuhkan Komputer/*CPU* yang memiliki peranan berbeda yaitu sebagai *master* dan *slave*. Komputer/*CPU* yang berfungsi sebagai *master* akan menginstruksikan komputer/*CPU slave* untuk mengolah data dan mengirim kembali *output* yang telah diperoleh ke komputer/*CPU master* sebagai hasil dari pengolahan data menggunakan komputasi

paralel. Dalam perancangan penelitian terdapat beberapa tahapan yang digunakan yaitu sebagai berikut:



Gambar 1: Topologi Pembagian Beban Kerja Komputasi Paralel SIM RS

A. Pengumpulan Data dan Analisis Kebutuhan

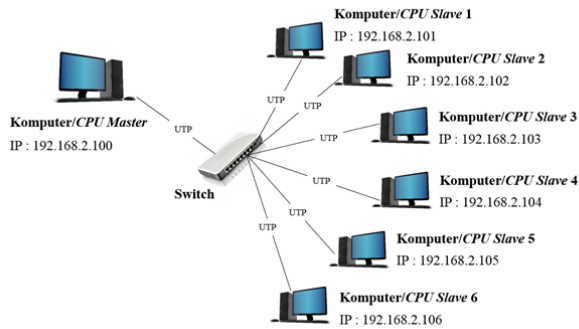
Teknik pengumpulan data yang digunakan adalah observasi, dan studi literatur. Observasi ini dimaksudkan untuk mengamati SIM RS yang telah berjalan dan proses pengolahan data dalam memenuhi kebutuhan manajemen rumah sakit. Sedangkan teknik Studi Literatur yaitu mencari referensi teori yang relevan dengan penelitian yang dilakukan.

Dalam tahap ini dilakukan pengumpulan data berupa Proses/alur kerja SIM RS yang telah berjalan dan Pengambilan data pasien yang ada pada data base SIM RS. Sesuai dengan hasil analisis kebutuhan yang dilakukan bersama pihak administrator SIMRS, tabel kunjungan_id dengan jumlah data sebanyak 40.036 digunakan sebagai tabel objek pengolahan data menggunakan program *MPI*. Tabel kunjungan_id merupakan *Primary Key* pada database his_db SIM RS, dimana semua tabel yang terdapat pada data base his_db mengambil data dan mengirimkan data pada tabel tersebut. Data pada tabel kunjungan_id akan dibagi menjadi beban kerja ke masing-masing komputer/*CPU slave* sesuai jumlah komputer/*CPU slave* yang ditentukan dalam mengeksekusi *query* pengolahan data.

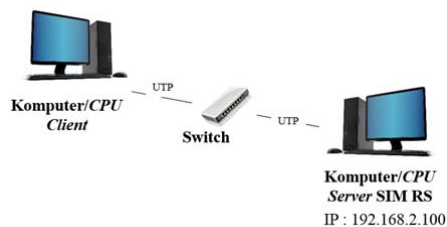
B. Perancangan dan Konfigurasi Local Area Network (LAN)

Berdasarkan konsep arsitektur jaringan komputasi paralel yang menggunakan lebih dari 1 komputer/*CPU* dan ketersediaan perangkat komputer/*CPU* sesuai ijin dari pihak rumah sakit, maka pada penelitian ini topologi LAN memanfaatkan 7 unit komputer/*CPU* dimana 1 unit komputer/*CPU* difungsikan sebagai master dan 6 unit komputer/*CPU* difungsikan sebagai *slave*. Sedangkan untuk dapat membandingkan kinerja komputasi dengan arsitektur *parallel* dan sekuensial, maka diperlukan perancangan arsitektur sekuensial dengan menggunakan 1 unit komputer/*CPU* difungsikan sebagai master dan 1 unit

komputer/CPU difungsikan sebagai *slave*. Untuk kedua arsitektur tersebut dapat digambarkan pada Gambar 2 dan Gambar 3.



Gambar 2: Arsitektur Parallel



Gambar 3: Arsitektur Sekuensial

C. Instalasi Software

Software yang dibutuhkan dalam penelitian ini adalah :

1. XAMPP (MYSQL) sebagai *data base* data pasien pada SIM RS.
2. SQLyog, *tools* yang digunakan untuk melakukan eksekusi *query*.
3. Bahasa pemrograman C. Untuk memfasilitasi proses perancangan *coding* komputasi paralel menggunakan bahasa pemrograman C dibutuhkan instalasi *software* Microsoft Visual Studio .
4. *Library MPI*. Dalam penggunaan model *message-passing* dalam perancangan komputasi paralel dibutuhkan *library MPI* yang berjalan sesuai dengan sistem operasi yang digunakan. Pada penelitian ini sistem operasi yang digunakan adalah Windows, untuk *library MPI* menggunakan *MPICH2* .

D. Konfigurasi Komputasi Paralel

Konfigurasi komputasi paralel pada komputer/CPU *master* dan *slave* menggunakan metode Foster dengan langkah sebagai berikut :

1. Langkah pertama metode Foster untuk mendesain komputasi paralel adalah proses partisi. Pembagian dilakukan dengan membagi dan menyebarkan data pasien ke sejumlah p . Sehingga dalam kondisi ideal, komputer/CPU akan mengolah sebanyak N/p data,

dengan N merupakan jumlah data dan p merupakan jumlah komputer/CPU.

2. Langkah kedua adalah menentukan komunikasi dalam desain paralel. Komunikasi terjadi pada saat proses pembagian data dari komputer/CPU *master* ke sejumlah komputer/CPU *slave*. Komunikasi yang terjadi adalah *collective communication*. *Collective communication* merupakan komunikasi yang terjadi antara komputer/CPU *master* dengan semua komputer/CPU *slave* yang ada
3. Langkah ketiga adalah proses mengelompokkan *task* ke dalam *task* yang lebih besar guna meningkatkan kinerja maupun menyederhanakan pemrograman atau yang disebut dengan proses aglomerasi
4. Langkah terakhir adalah memaksimalkan kemampuan komputer/CPU dan meminimalkan komunikasi antar komputer/CPU atau yang disebut proses pemetaan. Kemampuan komputer/CPU adalah persentasi rata-rata waktu komputer/CPU dalam mengeksekusi suatu *task* untuk menyelesaikan suatu masalah dan memberi suatu solusi penyelesaian.

E. Input Perintah

Tahap ini dilakukan untuk mengetahui jumlah data dari eksekusi *query* menggunakan *tools* SQLyog. Dalam penelitian ini digunakan *query* yang membutuhkan waktu eksekusi cukup kompleks yaitu *query* dalam mencari data Laporan Tampil Kunjungan Pasien berdasarkan Jenis Bayar (JKBM) dan Nama Tindakan Per Poliklinik (MATA) dalam periode waktu dari tanggal 1 bulan Januari tahun 2014 sampai dengan tanggal 1 bulan Januari tahun 2015. Dari hasil eksekusi *query* tersebut pada database *his_db* akan didapat data *records* sebanyak 9.493.

F. Pengujian Sistem

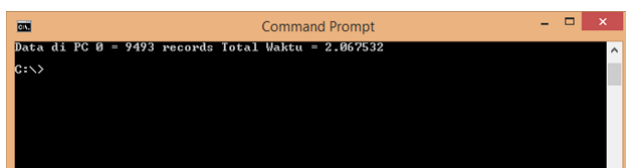
Pengujian terhadap penelitian ini difokuskan pada dua aspek yaitu *speedup* dan efisiensi. Proses pengujian dengan menguji *Speedup* adalah nilai yang diperoleh dari perbandingan antara waktu proses komputasi serial dengan waktu proses komputasi paralel. Pengujian efisiensi pada pengujian ini adalah perbandingan antara nilai *speed up* yang dihasilkan dari banyaknya komputer/CPU yang digunakan untuk mencapai nilai tersebut.

III. HASIL DAN PEMBAHASAN

A. Program MPI Pada Topologi Jaringan Sekuensial dan Topologi Paralel

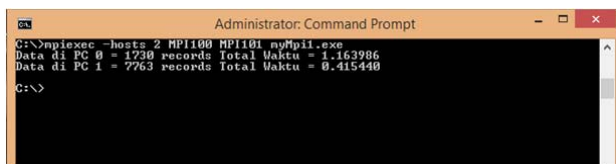
Tahap berikutnya adalah dengan menguji apakah program paralel telah berjalan sesuai konsep penelitian yang telah direncanakan yaitu dengan mengolah data SIM RS menggunakan topologi jaringan sekuensial (1 komputer/CPU) dan mengolah data SIMRS menggunakan program paralel *MPI* dengan topologi jaringan paralel yaitu menggunakan 1 komputer/CPU berfungsi sebagai *master* dan 6 komputer/CPU sebagai *slave* dengan total 7 komputer/CPU baik *master* ataupun *slave* dapat berfungsi mengolah data.





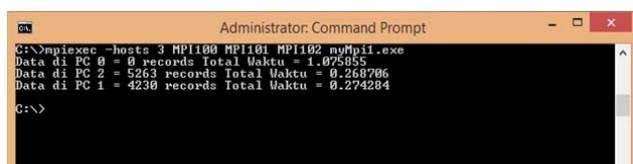
Gambar 4: Hasil Program MPI Pada Topologi Jaringan Sekuensial (1 Komputer/CPU)

Pada Gambar 4 menunjukkan hasil pengolahan data pada topologi jaringan sekuensial (1 komputer/CPU) dengan nama *host MPI100*. Dari total 40.036 records data yang ada pada tabel kunjungan_id eksekusi *query* dalam mencari data Laporan Tampil Kunjungan Pasien berdasarkan Jenis Bayar (JKBM) dan Nama Tindakan Per Poliklinik (MATA) periode waktu dari tanggal 1 bulan Januari tahun 2014 sampai dengan tanggal 1 bulan Januari tahun 2015 menggunakan program MPI menghasilkan 9.493 records data. Jumlah records data ini sesuai dengan jumlah data yang didapat pada uji eksekusi *query* pada tahap input perintah sebelumnya, hal ini membuktikan bahwa program MPI berjalan sesuai hasil yang diinginkan. Waktu yang dibutuhkan dalam proses eksekusi adalah 2,067532 detik. Pada tahap selanjutnya akan dilakukan pengujian menggunakan program paralel MPI pada topologi jaringan paralel menggunakan masing-masing 2,3,4,5,6, dan 7 komputer/CPU untuk kemudian waktu pengolahan data masing-masing akan dibandingkan dengan waktu pengolahan data pada topologi jaringan sekuensial/1 komputer/CPU.



Gambar 5: Hasil program MPI pada topologi jaringan paralel menggunakan 2 komputer/CPU

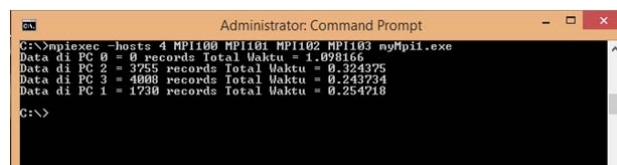
Gambar 5 menunjukkan pengujian program paralel MPI pada topologi jaringan paralel menggunakan 2 komputer/CPU, masing-masing komputer/CPU menghasilkan records data sebanyak 1.730 dan 7.763 dengan total jumlah records data sebanyak 9.493. Hasil pengujian tersebut jika dibandingkan dengan pengujian pada topologi jaringan sekuensial (1 komputer/CPU), terdapat peningkatan kecepatan pengolahan data selama 0,903546 detik.



Gambar 6: Hasil program MPI pada topologi jaringan paralel menggunakan 3 komputer/CPU

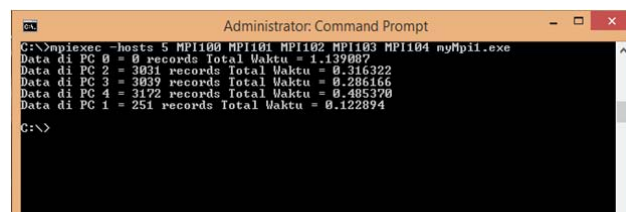
Gambar 6 menunjukkan pengujian program paralel MPI pada topologi jaringan paralel menggunakan 3 komputer/CPU

(*host MPI100*, *MPI101* dan *MPI102*) total waktu yang dibutuhkan dalam proses pengolahan data yaitu selama 1,075855 detik. Dari hasil pengujian tersebut jika dibandingkan dengan pengujian pada topologi jaringan paralel 2 komputer/CPU, terdapat peningkatan kecepatan pengolahan data selama 0,088131 detik.



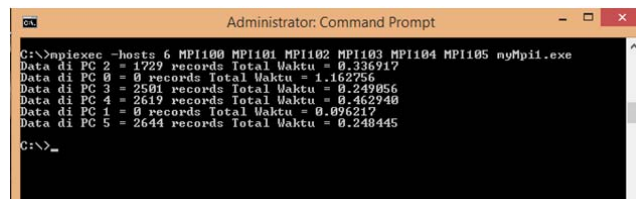
Gambar 7: Hasil program MPI pada topologi jaringan paralel menggunakan 4 komputer/CPU

Gambar 7 menunjukkan pengujian program paralel MPI pada topologi jaringan paralel menggunakan 4 komputer/CPU (*host MPI100*, *MPI101*, *MPI102* dan *MPI103*) waktu yang dibutuhkan adalah 1,098166 detik. Dari hasil pengujian tersebut jika dibandingkan dengan pengujian pada topologi jaringan paralel 3 komputer/CPU, terdapat penurunan kecepatan pengolahan data selama 0,022311 detik.



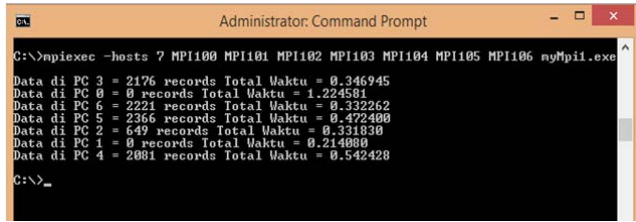
Gambar 8: Hasil program MPI pada topologi jaringan paralel menggunakan 5 komputer/CPU

Gambar 8 menunjukkan pengujian program paralel MPI pada topologi jaringan paralel menggunakan 5 komputer/CPU (*host MPI100*, *MPI101*, *MPI102*, *MPI103* dan *MPI104*) waktu yang dibutuhkan adalah 1,139087 detik. Dari hasil pengujian tersebut jika dibandingkan dengan pengujian pada topologi jaringan paralel 4 komputer/CPU, terdapat penurunan kecepatan pengolahan data selama 0,040921 detik



Gambar 9: Hasil program MPI pada topologi jaringan paralel menggunakan 6 komputer/CPU

Gambar 9 menunjukkan pengujian program paralel MPI pada topologi jaringan paralel menggunakan 6 komputer/CPU (*host MPI100*, *MPI101*, *MPI102*, *MPI103*, *MPI104* dan *MPI105*) waktu yang dibutuhkan adalah 1,162756 detik. Dari hasil pengujian tersebut jika dibandingkan dengan pengujian pada topologi jaringan paralel 5 komputer/CPU, terdapat penurunan kecepatan pengolahan data selama 0,023669 detik.



Gambar 10: Hasil program MPI pada topologi jaringan paralel menggunakan 7 komputer/CPU

Gambar 10 menunjukkan pengujian program paralel MPI pada topologi jaringan paralel menggunakan 7 komputer/CPU (*host MPI100, MPI101, MPI102, MPI103, MPI104, MPI105* dan *MPI106* waktu yang dibutuhkan adalah 1,224581 detik. Dari hasil pengujian tersebut jika dibandingkan dengan pengujian pada topologi jaringan paralel 6 komputer/CPU, terdapat penurunan kecepatan pengolahan data selama 0,061825 detik.

B. Pengujian Speed Up

Speed up (*S*) merupakan hasil perbandingan antara waktu sekuensial dengan waktu paralel. Pada hasil pengujian tahap sebelumnya hasil pengolahan data pada topologi jaringan sekuensial/1 komputer/CPU membutuhkan waktu 2,067532 detik. Untuk mengetahui nilai *speed up* maka waktu hasil pengolahan data topologi jaringan sekuensial (1 komputer/CPU) akan dibagi waktu pengolahan data menggunakan program paralel MPI sesuai dengan jumlah komputer/CPU yang digunakan.

TABEL I
HASIL PENGUJIAN SPEED UP

Jumlah komputer/CPU (<i>p</i>)	Waktu paralel (<i>T_p</i>) (detik)	<i>Speed Up</i> (<i>S</i>)	Efisiensi (<i>E</i>)
2	1,163986	1,78	0,89
3	1,075855	1,92	0,64
4	1,098166	1,88	0,47
5	1,139087	1,81	0,36
6	1,162756	1,78	0,30
7	1,224581	1,69	0,24

Hasil yang didapat dari pengujian *speed up* pada Tabel I menunjukkan adanya peningkatan kecepatan sampai pada penggunaan komputasi paralel pada 3 komputer/CPU. Pengujian menggunakan 2 komputer/CPU memerlukan waktu selama 1,163986 detik dengan *speed up* mencapai 1,78 kali lebih cepat dari waktu sekuensial/1 komputer/CPU. Pengujian menggunakan 3 komputer/CPU memerlukan waktu selama 1,075855 detik dengan *speed up* mencapai 1,92 kali lebih cepat dari waktu sekuensial/1 komputer/CPU. Sedangkan pada pengujian menggunakan 4,5,6 dan 7 komputer/CPU *speed up* sudah mulai menurun seperti ditunjukkan pada Gambar 11.

Penurunan kecepatan tersebut sangat mungkin terjadi berdasarkan Hukum Amdahl yang tercantum dalam buku Kurniawan, A (2010) mengenai Pemrograman Paralel MPI & C, bahwa semakin banyak prosesor maka kecepatan akan sampai pada titik jenuh. Secara keseluruhan pengujian *speedup* pada penelitian ini jika dibandingkan dengan hasil pengolahan data menggunakan topologi jaringan sekuensial/1 komputer/CPU, pengolahan data menggunakan topologi jaringan paralel menggunakan 7 komputer/CPU terbukti memiliki kecepatan yang lebih baik.

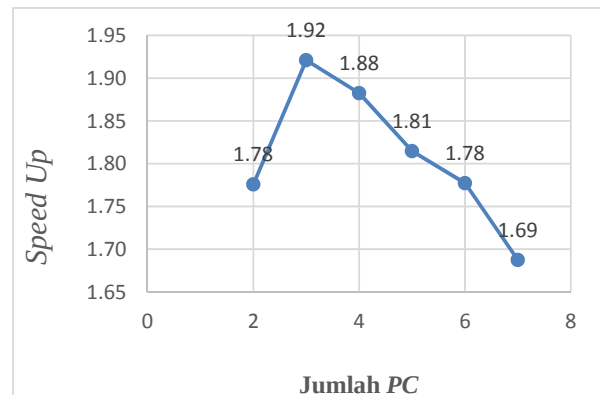
C. Pengujian Efisiensi

Efisiensi (*E*) merupakan perbandingan antara *speed up* (*S*) dengan jumlah komputer/CPU yang digunakan (*p*). Efisiensi mengukur seberapa efisien penggunaan sejumlah komputer/CPU di dalam topologi jaringan paralel yang dibangun. Penelitian ini mengukur nilai efisiensi pada topologi jaringan paralel menggunakan 7 komputer/CPU seperti pada Tabel II berikut.

TABEL II
HASIL PENGUJIAN EFISIENSI

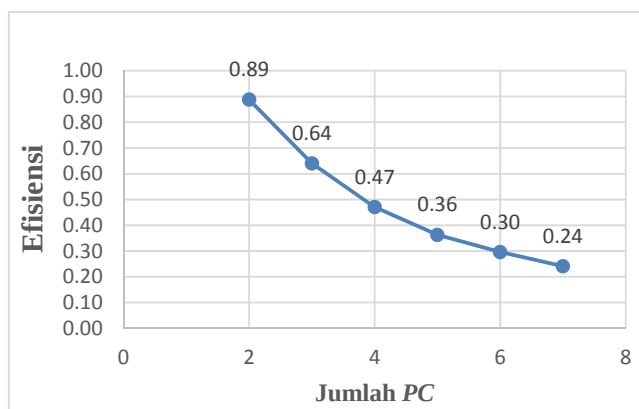
Jumlah komputer/CPU (<i>p</i>)	Waktu paralel (<i>T_p</i>) (detik)	<i>Speed Up</i> (<i>S</i>)
2	1,163986	1,78
3	1,075855	1,92
4	1,098166	1,88
5	1,139087	1,81
6	1,162756	1,78
7	1,224581	1,69

Nilai efisiensi yang dihasilkan pada topologi jaringan paralel dengan nilai tertinggi adalah 0.89 pada saat penggunaan 2 komputer/CPU. Sedangkan untuk selanjutnya pada pengujian menggunakan 3, 4, 5, 6 dan 7 komputer/CPU secara berurutan mengalami penurunan nilai efisiensi.



Gambar 11: Grafik Pengujian Speed Up





Gambar 12: Grafik Pengujian Efisiensi

Pada Gambar 12 terlihat penurunan nilai efisiensi pada pengujian menggunakan 3 komputer/CPU yaitu dengan nilai efisiensi 0,64. Pengujian menggunakan 4 komputer/CPU dengan nilai efisiensi 0,47. Pengujian dengan menggunakan 5 komputer/CPU dengan nilai efisiensi 0,36. Pengujian dengan menggunakan 6 komputer/CPU dengan nilai efisiensi 0,30 dan Pengujian dengan menggunakan 7 komputer/CPU dengan nilai efisiensi terendah yaitu 0,24. Penurunan nilai efisiensi yang terjadi pada setiap peningkatan jumlah komputer/CPU disebabkan oleh terdapat komputer/CPU yang tidak bekerja (*idle*). Proses komunikasi yang kompleks menyebabkan terjadinya kondisi dimana komputer/CPU tidak bekerja dan menunggu untuk mendapat perintah. Hasil nilai efisiensi menunjukkan semakin banyak penggunaan komputer/CPU maka semakin menurun nilai efisiensi.

IV. KESIMPULAN

Perancangan komputasi paralel pada penelitian ini telah berhasil berjalan sesuai dengan tahapan-tahapan yang telah direncanakan. Hasil yang didapat pada tahap pengujian menunjukkan pengolahan data pasien menunjukkan bahwa waktu pengolahan data pasien menggunakan program paralel *MPI* lebih cepat dibandingkan pengolahan data menggunakan topologi jaringan sekuensial/1 komputer/CPU. Pengujian *speed up* menunjukkan adanya peningkatan kecepatan sampai pada penggunaan komputasi paralel pada 3 komputer/CPU. Sedangkan pada pengujian efisiensi nilai efisiensi tertinggi terdapat pada penggunaan 2 dan 3 komputer/CPU. Hal ini menunjukkan bahwa untuk mengolah data sebanyak 40.036 records menggunakan program paralel *MPI* jumlah ideal komputer/CPU yang digunakan agar mendapatkan kecepatan maksimal adalah sebanyak 2 sampai 3 komputer/CPU. Hal tersebut menunjukkan bahwa komputasi paralel menggunakan model *Message Passing* tidak serta merta dapat diimplementasikan pada semua jenis data dan struktur database. Jenis data dan struktur database pada SIM RS dalam penelitian ini menunjukkan bahwa semakin banyak komputer/CPU yang ikut melakukan eksekusi data dan melewati titik idealnya (2 sampai 3 komputer/CPU) mengakibatkan waktu tunggu dalam melakukan proses komunikasi semakin panjang, sehingga mempengaruhi panjang waktu eksekusi data secara keseluruhan.

REFERENSI

- [1] Arta, Y. 2013. Analisa Kinerja Paralel Computing Dengan Menggunakan Perhitungan Hukum Amdahl Berbasis Linux.
- [2] Dumbill, E. 2012. *Big Data Now Current Perspective*. O'Reilly Media.
- [3] Dzacko, H. 2007. *Basis Data (Data Base)*. Mangosoft.
- [4] Grama A, Gupta A, Karypis G, Kumar V. 2003. *Introduction to Parallel Computing*. Pearson Education Limited: England.
- [5] Hua S, Zhang Y. 2013. Comparison and Analysis of Parallel Computing Performance Using OpenMP and MPI. *The Open Automation and Control Systems Journal*.
- [6] Kurniawan, A. 2010. *Pemrograman Paralel dengan MPI & C*. ANDI Yogyakarta
- [7] Laudon, K.C., Jane P. Laudon. 2006. *Management Information Systems. 9th edition*. New Jersey: Prentice- Hall, Inc.
- [8] Nashar AIE. 2011. Paralel Performance Of MPI Sorting Algorithms On Dual-Core Processor Windows-Based Systems. *International Journal of Distributed and Parallel Systems (IJDPSS)* Vol.2, No.3.
- [9] Nasir AFA, Rahman MNA, Mamat AR. 2012. A Study of Image Processing in Agriculture Application under High Performance Computing Environment. *International Journal of Computer Science and Telecommunications*.
- [10] Olmedo E, Calleja Jdl, Benitez A, Medina MA, 2012. Point to point processing of digital images using paralel computing
- [11] Petryniak R. 2008. Analysis Of Efficiency Of Paralel Computing In Image Processing Task. *Cracow University of Technology*.
- [12] Prajapati HB, Vij SK. 2011. Analytical Study of Paralel and Distributed Image Processing. *International Conference on Image Information Processing (ICIIP)*.
- [13] Quinn MJ. 2004. *Paralel Programming in C with MPI and OpenMP*. McGraw-Hill Education: Singapore.
- [14] Qureshi K, 2012. Perbandingan Praktis Kinerja Algoritme Sorting Paralel pada Jaringan Komputer Homogen.
- [15] R. Kelly Rainer, C. 2011. *Introduction to Information Systems*. John Wiley & Sons (Asia) Pte Ltd.
- [16] Suprpto, 2008. *Bahasa Pemrograman*. Direktorat Pembinaan Sekolah Menengah Kejuruan Direktorat Jenderal Manajemen Pendidikan Dasar dan Menengah Departemen Pendidikan Nasional.
- [17] Syukur A. 2013. Paralel Processing Untuk Meningkatkan Kinerja Server E-Learning Dengan Menggunakan Message Passing Interface (MPI) Studi Kasus Sma Negeri 1 Pekanbaru. *Jurnal Teknologi Informasi & Pendidikan* Vol. 6.
- [18] Zarkasi M, Wibisono W, Arunanto FX, 2013. Implementasi Komputasi Paralel Untuk Enkripsi Citra Berbasis AES Menggunakan JPPF.