

Analisa Kinerja Protokol iSCSI Melalui Jaringan Virtual

Ahmad Sabiq¹, Setiadi Yazid²

¹Program Studi Teknik Informatika Politeknik Purbaya Tegal

²Fakultas Ilmu Komputer Universitas Indonesia Depok

E-mail : ahmad.sabiq@purbaya.ac.id, setiadi@cs.ui.ac.id

Abstrak

Internet Small Computer System Interface (iSCSI) merupakan protokol yang berjalan di lapisan transport pada jaringan TCP/IP yang bekerja untuk mengirimkan perintah-perintah SCSI agar memungkinkan suatu komputer dapat mengakses block-level dari media penyimpanan pada komputer yang berbeda. Selain digunakan pada jaringan LAN, MAN ataupun MAN, iSCSI juga dapat digunakan pada jaringan virtual yang menghubungkan mesin virtual dan mesin yang secara fisik memang ada atau nyata. Pada penelitian ini, dilakukan pengujian terhadap kinerja dari iSCSI pada jaringan virtual yang menghubungkan mesin virtual dan mesin nyata agar mendapatkan nilai throughput dan kecepatan baca/tulis untuk kemudian dianalisa lebih lanjut.

Kata kunci: iSCSI, Mesin Virtual, SAN.

Abstract

Internet Small Computer System Interface (iSCSI) is a protocol that runs at the transport layer in the TCP/IP network that works to send SCSI commands that allow a computer to access block-level storage media on different computers. Besides being used on a LAN, MAN or MAN, iSCSI can also be used on a virtual network connecting the virtual machines and physical machines. In this research, testing the performance of iSCSI in virtual network connecting the virtual machine and the real machine in order to get the value of throughput and speeds up read/write and then analyzed further.

Keyword: iSCSI, Virtual Machine, SAN.

I. PENDAHULUAN

Storage Area Network (SAN) merupakan suatu sistem media penyimpanan yang terpusat dalam suatu jaringan, dimana komputer *client* maupun *server* dimungkinkan dapat menggunakan media penyimpanan tersebut seperti menggunakan media penyimpanan lokal.

Dengan adanya teknologi *Local Area Network* (LAN) berkecepatan tinggi seperti 1Gigabit *Ethernet* dan 10Gigabit *Ethernet*, iSCSI yang berbasis pada jaringan *Internet Protocol* (IP) yaitu IP-SAN menjadi lebih umum digunakan dibandingkan dengan yang menggunakan *Fiber Channel* yaitu SAN-FC yang membutuhkan biaya lebih tinggi. Hingga saat ini, studi tentang analisa terhadap pengukuran kinerja iSCSI melalui jaringan yang nyata secara fisik telah dilakukan baik melalui LAN maupun *Wide Area Network* (WAN) [1], selain itu dilakukan juga studi analisa kinerja iSCSI menggunakan perangkat lunak simulasi [2][3] untuk mempermudah pengaturan dalam pengujian dan meminimalisir biaya dalam penelitian, serta dapat mensimulasikan sistem iSCSI secara luas.

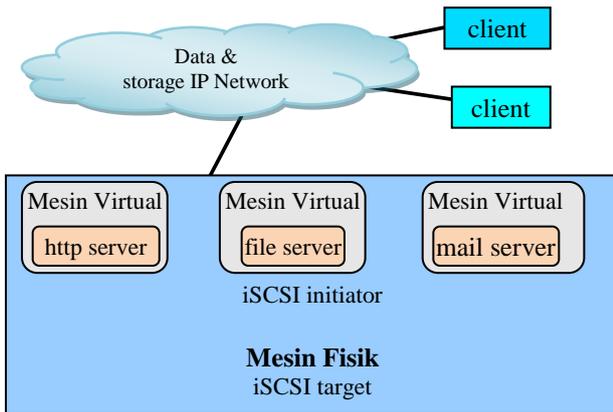
Teknologi virtualisasi memungkinkan beberapa sistem operasi dapat dipasang dan dijalankan pada satu mesin PC yang dinamakan Sistem Operasi tamu pada mesin virtual yang dikelola oleh *Virtual Machine Monitor* (VMM). Penggunaan virtualisasi pada suatu instansi atau perusahaan dapat memungkinkan instansi atau perusahaan tersebut untuk memiliki beberapa aplikasi, layanan, dan sistem operasi dalam satu hardware yang sama, hal ini dapat digunakan untuk menghemat biaya dan anggaran dalam pembelian ataupun pemeliharaan. Dari sisi keamanan perangkat keras, penggunaan mesin virtual juga dapat meminimalisir kerusakan yang terjadi pada perangkat keras akibat dari gangguan atau serangan dari luar.

iSCSI juga dapat diterapkan pada mesin-mesin virtual yang berjalan pada mesin server [4][5][6], dimana mesin virtual tersebut menggunakan media penyimpan dari mesin lain melalui jaringan IP ataupun media penyimpan pada mesin yang sama melalui protokol iSCSI seperti yang ditunjukkan pada Gambar 1.

Pada penelitian ini, akan didesain dan diimplementasikan beberapa pengujian untuk

melakukan analisa dari kinerja iSCSI pada mesin virtual.

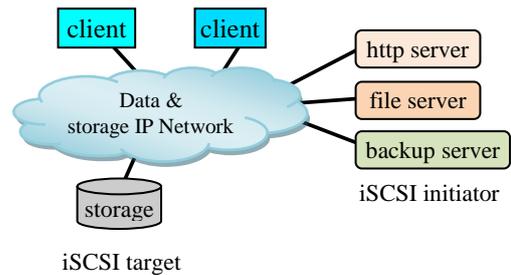
Pada penelitian ini, penulis membatasi permasalahan untuk menganalisa kinerja kecepatan baca dan tulis dari sistem iSCSI yang dibangun melalui jaringan virtual antara PC Linux yang secara fisik nyata sebagai iSCSI *target* dan PC Linux yang merupakan mesin virtual yang dipasang pada PC fisik yang dijadikan iSCSI *target* sebagai iSCSI *initiator* terhadap beberapa parameter ukuran blok.



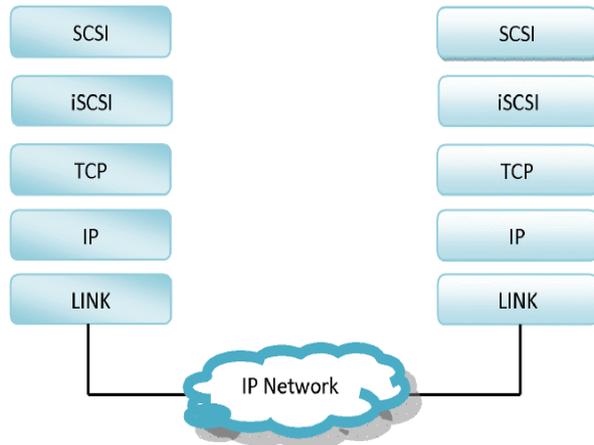
Gambar 1 Model iSCSI dengan iSCSI *initiator* pada mesin-mesin virtual yang masing-masing memiliki layanan yang berbeda dan terhubung pada iSCSI target yang sama pada satu mesin fisik yang sama

1.1 SAN dan iSCSI

SAN merupakan sistem media penyimpanan terpusat yang dapat diterapkan dengan menggunakan jaringan *Fiber Channel* (FC) yang disebut dengan FC-SAN ataupun menggunakan *Internet Protocol* (IP) yang disebut IP-SAN. FC-SAN merupakan sistem SAN yang mempunyai performa sangat tinggi, tetapi FC-SAN membutuhkan biaya yang mahal dan bersifat kompleks. IP-SAN dikenal dengan sebuah jaringan yang dibangun dengan menggunakan iSCSI sebagai protokolnya. iSCSI [7] merupakan protokol yang mengenkapsulasi perintah SCSI (*Small Computer System Interface*) dan data kemudian mengirimkannya melalui jaringan IP yang digunakan untuk mengirimkan data melalui *internet* ataupun *intranet* dan mengelola media penyimpanan dari jarak jauh melalui internet. Secara umum gambaran dari desain IP-SAN ditunjukkan pada Gambar 2, dimana mesin-mesin server (seperti *file server*, *web server* dan sebagainya) yang dijadikan sebagai iSCSI *initiator* menggunakan media penyimpan dari mesin lain yang berfungsi sebagai iSCSI *target* melalui jaringan IP.



Gambar 2 Model Storage Area Network (SAN) berbasis IP atau internet Small Computer System Interface (iSCSI)



Gambar 3 Protokol iSCSI pada lapisan jaringan TCP/IP

Protokol iSCSI merupakan protokol pada lapisan pertama dan kedua dari arsitektur jaringan TCP/IP, gambar 3 menunjukkan iSCSI bekerja pada lapisan *application* dan *session*. Pada bagian client dinamakan dengan iSCSI *initiator* dan pada server dinamakan iSCSI *target*. iSCSI *initiator* dan *target* membentuk Protocol Data Unit (PDU) yang terdiri dari perintah SCSI, data, dan respon. PDU iSCSI dimasukan pada lapisan transport (TCP) pada bagian *segment data*, yaitu bagian dari lapisan transport yang berisi data yang akan dikirim, dari lapisan transport PDU dikirimkan melalui jaringan.

1.2 Mesin Virtual

Mesin virtual [8] merupakan sebuah perangkat lunak yang dipasang pada *Virtual Machine Manager* (VMM) di suatu sistem operasi yang dapat menjalankan sistem operasi dan aplikasi sendiri seperti halnya komputer fisik. Mesin virtual mempunyai perilaku yang sama persis dengan mesin fisik, dan komponen dari mesin virtual tersebut terdiri dari CPU, RAM, HardDisk dan Network Interface Card (NIC) virtual yang sebenarnya adalah perangkat lunak.

Sistem operasi, aplikasi ataupun komputer lain pada jaringan tidak akan dapat membedakan apakah suatu mesin komputer berwujud fisik ataupun virtual. Bahkan mesin virtual sendiri memiliki pemikiran yang sama seperti halnya komputer fisik, walaupun seluruh komponennya merupakan perangkat lunak dan tidak terdapat komponen perangkat keras sedikitpun, sehingga mesin virtual memiliki sejumlah keunggulan berbeda dari perangkat keras fisik.

1.3 Jaringan Pada Mesin Virtual

Pada Mesin virtual, dapat dibangun jaringan yang menghubungkan suatu mesin virtual dengan mesin virtual lain, mesin host ataupun mesin fisik lain yang terhubung dengan mesin host menggunakan pilihan mode *network address translation* (NAT), *bridge*, *internal network* ataupun *host only adapter*.

1.3.1 Network Address Translation (NAT)

NAT merupakan cara yang paling sederhana untuk mengakses jaringan luar atau jaringan nyata dari mesin virtual. Pada mode ini mesin host dan mesin virtual tidak diperlukan konfigurasi pada jaringan.

Sebuah mesin virtual yang menggunakan NAT dianalogikan seperti sebuah mesin komputer yang terhubung dengan sebuah *router*, dimana *router* yang digunakan adalah Pengelola Virtual Mesin / *Virtual Mechine Manager* (VMM) yang ditempatkan di antara mesin virtual dan mesin host yang memetakan lalu-lintas jaringan dari dan ke mesin virtual.

1.3.2 Bridge Adapter

Pada tipe jaringan *bridge*, Pengelola Virtual Mesin menggunakan *device driver* pada sistem host yang dinamakan "*net filter*" untuk menyaring data dari adapter jaringan fisik. Jaringan tipe *bridge* ini mengizinkan VMM menangkap data dari jaringan fisik dan memasukan data kedalamnya, VMM secara efektif menciptakan antarmuka jaringan dalam bentuk perangkat lunak.

Ketika mesin virtual menggunakan antarmuka jaringan bertipe *bridge*, pada sistem host seolah-olah mesin virtual terhubung secara fisik ke mesin host, baik melalui kabel jaringan maupun nirkabel (tergantung pengaturan yang dilakukan).

1.3.3 Internal network

Jaringan *internal* pada dasarnya hampir sama dengan tipe jaringan *bridge*, hanya saja pada jaringan internal ini, mesin virtual tidak dapat

berkomunikasi langsung dengan mesin-mesin fisik diluar mesin host. Suatu mesin virtual yang menggunakan jaringan internal hanya dapat berkomunikasi langsung dengan mesin host dan mesin-mesin virtual pada host yang sama.

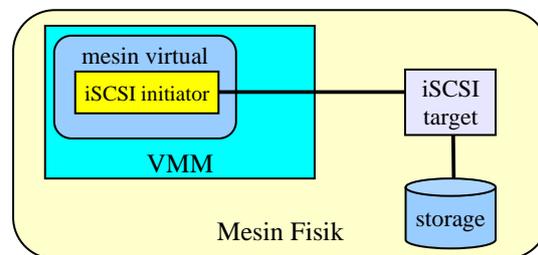
1.3.4 Host Only Adapter

Pilihan jaringan jenis Host only adapter pada mesin virtual membuat mesin virtual tersebut dapat berkomunikasi hanya dengan mesin-mesin virtual lain dalam host yang sama melalui switch yang dapat disediakan dan dikelola secara virtual pada pengelola mesin virtual, jaringan tipe ini tidak dapat terhubung secara lokal dengan mesin fisik.

II. METODE PENELITIAN

2.1 Rancangan Arsitektur Sistem

Sistem iSCSI yang akan dibangun terdiri dari dua buah PC yang berperan sebagai iSCSI *target* dan iSCSI *initiator*. PC untuk iSCSI *initiator* merupakan mesin virtual yang berjalan di atas *Virtual Machine Monitor* yang berada pada PC untuk iSCSI *target* seperti yang ditunjukkan pada Gambar 4. Kedua mesin ini juga dihubungkan secara virtual melalui jaringan menggunakan *Gigabit Ethernet Local Area Network* (GE-LAN).



Gambar 4 Desain rancangan dari arsitektur sistem iSCSI yang akan digunakan pada penelitian

2.2 Perangkat Keras dan Sistem Operasi

Perangkat keras yang digunakan pada eksperimen adalah sebuah PC dengan spesifikasi sebagai berikut :

- processor intel Pentium Dual Core 2.03GHz,
- RAM (*Random Access Memory*) DDR2 dengan kapasitas 3GB,
- *Harddisk* SATA dengan kapasitas 350 GB. yang dijadikan sebagai iSCSI *target*.

Sedangkan PC *client* yang dijadikan sebagai iSCSI *initiator* merupakan mesin virtual yang dipasang pada mesin iSCSI *target* menggunakan virtualbox versi 4.0.4r70112 Sistem operasi yang

digunakan pada kedua mesin tersebut adalah Linux Fedora 14 dengan kernel versi 2.6.35.6-45.

2.3 Perangkat Lunak iSCSI

Untuk pengimplementasian iSCSI, digunakan dua buah perangkat lunak iSCSI yang merupakan perangkat lunak kode sumber terbuka, yaitu open-iSCSI yang berfungsi sebagai iSCSI *client* dan dipasang pada mesin iSCSI *initiator* dan iSCSI *target* yang berfungsi sebagai iSCSI server dan dipasang pada mesin iSCSI *target*.

2.4 Alat Pengukuran

Untuk mendapatkan hasil penilaian yang dapat dianalisa, digunakan beberapa perangkat lunak, antara lain:

- *Iperf benchmarking*. Merupakan utilitas yang digunakan untuk mendapatkan nilai kinerja dari jaringan TCP. *Iperf* dapat digunakan untuk melakukan penyetelan terhadap beberapa parameter seperti TCP_Window dan TCP_Segment_size, dan juga mampu untuk memberikan hasil pengukuran *bandwidth*, *delay jitter* dan gambaran dari kehilangan datagram.
- *dd*, perangkat utilitas yang digunakan untuk pengukuran kinerja *block device*, secara umum sudah terdapat pada distribusi Linux, digunakan untuk mengumpulkan data dari perpindahan *low-level* pada *block device*.
- *IOZone filesystem benchmarking*, digunakan untuk mengukur kinerja sistem berkas pada pengoperasian berkas secara luas.

2.5 Pengujian

Agar kinerja dari iSCSI dapat dianalisa, diperlukan skenario percobaan untuk mendapatkan data yang sesuai. Pada tahap pertama akan dilakukan uji coba untuk mendapatkan data dari kemampuan akses baca dan tulis *local drive* pada *pc target* atau mesin fisik sebagai pembanding, yaitu dengan menggunakan utilitas *dd* untuk membuat data hingga sebesar 5GB agar proses penulisan dan pembacaan data tidak selesai terlalu cepat, sehingga diperoleh hasil pengukuran dengan nilai yang tidak terlalu banyak berubah jika dilakukan pengukuran kembali untuk ukuran blok yang sama.

Kemudian untuk mendapatkan data dari kemampuan baca dan tulis iSCSI, dilakukan pengujian dengan membuat data hingga 5GB dari *pc initiator* yang terhubung dengan *pc target* melalui jaringan virtual.

Pengukuran kecepatan baca dan tulis baik untuk *local drive* ataupun iSCSI dilakukan terhadap ukuran blok yang bervariasi mulai dari 4 KB sampai 1024 KB.

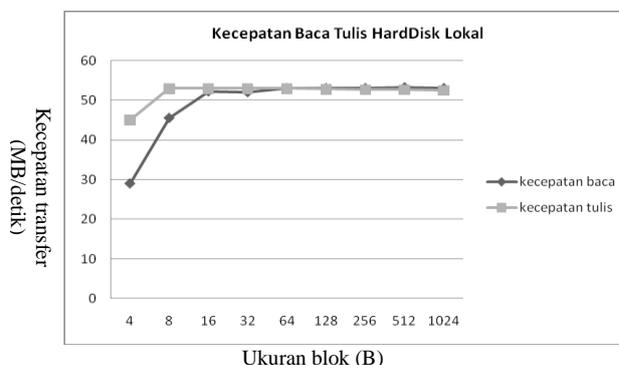
Pada setiap percobaan diatas dilakukan pengukuran kecepatan transfer data melalui jaringan pada saat melakukan pembacaan dan penulisan data dari sistem iSCSI menggunakan perangkat lunak yang sudah dipersiapkan untuk mendapatkan hasil berupa nilai *throughput* pada saat melakukan pembacaan data dan penulisan data dari dan ke *drive* yang ada pada PC target untuk kemudian dianalisa lebih lanjut.

Pada percobaan berikutnya dilakukan *benchmarking* terhadap sistem berkas secara luas dengan menggunakan perangkat lunak *IOZone* untuk mendapatkan nilai dari kinerja *filesystem*.

III. HASIL DAN PEMBAHASAN

Dari percobaan yang sudah dilakukan, didapatkan beberapa hasil dari pengukuran terhadap kinerja dari *harddisk* yang digunakan pada sistem iSCSI pada penelitian.

Pada percobaan pertama dilakukan pengujian kecepatan baca dan tulis terhadap *localdisk* yang digunakan sebagai *harddisk* iSCSI target pada mesin yang digunakan menggunakan perangkat lunak *dd*. Pengukuran ini digunakan agar dapat dilakukan perbandingan kinerja *harddisk* yang digunakan saat digunakan sebagai *harddisk* lokal ataupun *harddisk* yang digunakan secara remote melalui protokol iSCSI. Dari pengukuran yang telah dilakukan diperoleh hasil yang ditampilkan dalam bentuk grafik pada Gambar 5. Dari percobaan tersebut didapatkan bahwa kecepatan baca pada *disk* lokal meningkat hingga 53 MB/s pada ukuran blok data 16 KB, untuk kecepatan tulis peningkatan sudah dimulai pada ukuran blok data 8 KB. Kecepatan baca dan tulis terlihat stabil pada 53 MB/s hingga pada ukuran blok data 1024 KB.

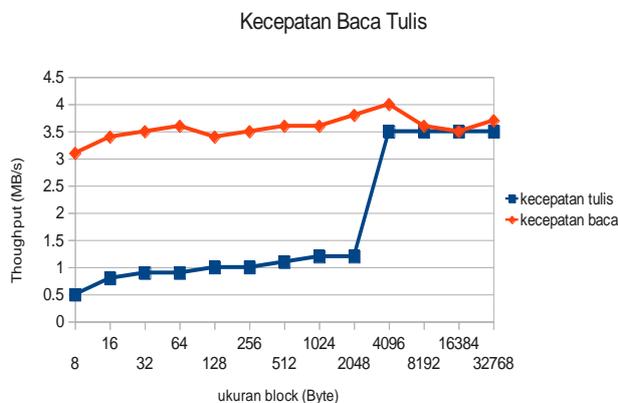


Gambar 5 Kecepatan pembacaan dan penulisan pada *localdisc* di mesin iSCSI target

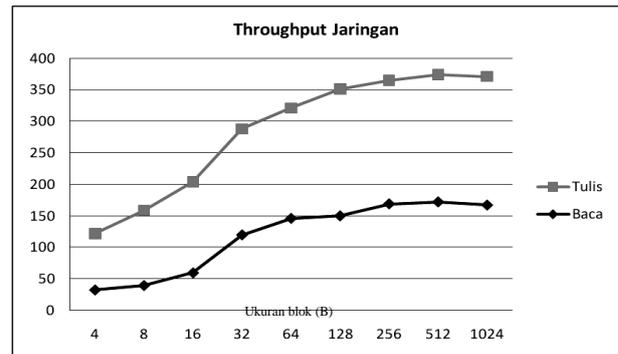
Dari mesin iSCSI *initiator* dilakukan juga pengukuran kecepatan baca dan tulis pada *harddisk* iSCSI yang digunakan dari mesin iSCSI target serta pengukuran nilai *throughput* pada jaringan virtual ketika melakukan pembacaan dan penulisan data dari dan ke iSCSI target. Percobaan dilakukan pada iSCSI dengan nilai parameter standar dari perangkat lunak iSCSI target yaitu:

MaxRecvDataSegmentLength = 8192 Bytes
 MaxXmitDataSegmentLength = 8192 Bytes
 FirstBurstLength = 65536 Bytes
 MaxBurstLength = 262144 Bytes

Dari pengukuran yang telah dilakukan diperoleh hasil yang ditampilkan dalam bentuk grafik pada Gambar 6 dan Gambar 7 dari hasil pengukuran didapatkan kecepatan baca dari mesin *initiator* mengalami perubahan naik dan turun serta mengalami nilai tertinggi pada ukuran block 4096 Bytes. Pada kecepatan pembacaan terjadi peningkatan kecepatan hingga 3 kali lipat untuk ukuran block 4096 Bytes. Untuk kinerja jaringan didapatkan nilai *throughput* jaringan semakin meningkat ketika dilakukan pembacaan dan penulisan data pada iSCSI ukuran blok data yang lebih besar.



Gambar 6 Kecepatan pembacaan dan penulisan pada *harddisk* iSCSI di mesin iSCSI *initiator*



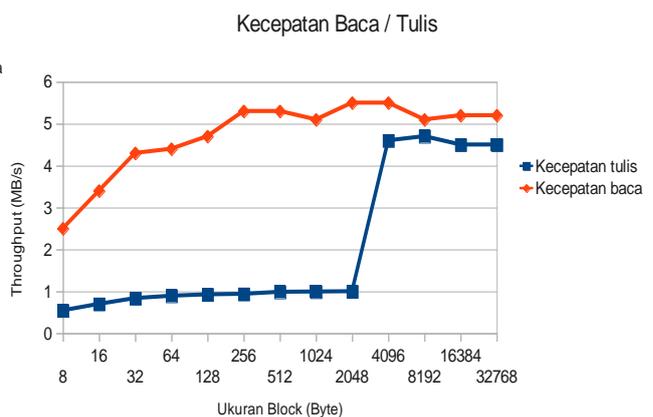
Gambar 7 Nilai *throughput* pada saat dilakukan pembacaan dan penulisan pada *harddisk* iSCSI di mesin iSCSI *initiator*

Percobaan berikutnya dilakukan pengujian dengan mengubah nilai parameter, hasil yang didapatkan pada Gambar 8 diperoleh dari perubahan nilai parameter berikut

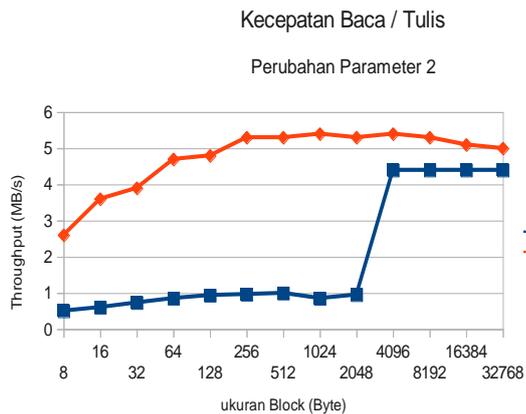
MaxRecvDataSegmentLength = 16384 Bytes
 MaxXmitDataSegmentLength = 16384 Bytes
 FirstBurstLength = 131072 Bytes
 MaxBurstLength = 524288 Bytes

Hasil yang didapatkan pada Gambar 3.5, diperoleh dari perubahan nilai parameter berikut

MaxRecvDataSegmentLength = 32768 Bytes
 MaxXmitDataSegmentLength = 32768 Bytes
 FirstBurstLength = 262144 Bytes
 MaxBurstLength = 1048576 Bytes



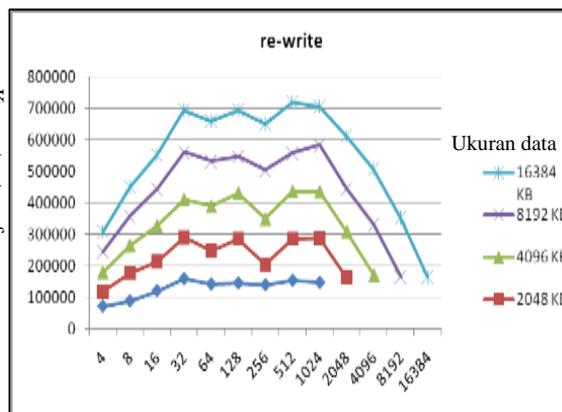
Gambar 8 Kecepatan pembacaan dan penulisan pada *harddisk* iSCSI di mesin iSCSI *initiator* dari hasil perubahan parameter I



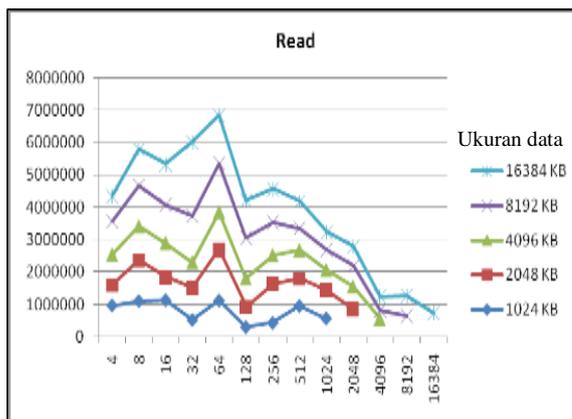
Gambar 9 Kecepatan pembacaan dan penulisan pada *harddisk* iSCSI di mesin iSCSI *initiator* dari hasil perubahan paramter II

Pada pengukuran berikutnya, dilakukan *benchmarking* sistem berkas pada partisi *harddisk* iSCSI yang dipasang pada mesin iSCSI *initiator* menggunakan perangkat lunak *iozone*. Hasil yang didapatkan dari *benchmarking* tersebut ditampilkan dalam bentuk grafik pada Gambar 10 untuk data 1024 B hingga 16384 B. Pada *benchmarking* tersebut dilakukan pengoperasian penulisan, penulisan ulang, pembacaan, pembacaan ulang, pembacaan secara acak serta penulisan berkas secara acak. Proses baca dan tulis dilakukan pada ukuran blok mulai 4 B hingga 16384 B.

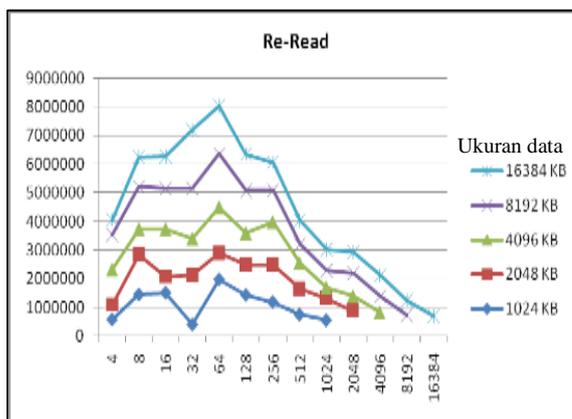
Hasil yang didapatkan terlihat bahwa pada saat melakukan penulisan berkas baru (*write*), kecepatan paling tinggi didapatkan pada ukuran blok 1024 B untuk ukuran data yang digunakan pada percobaan, sedangkan pada penulisan ulang pada berkas yang sudah ada (*re-write*) kecepatan yang paling tinggi untuk ukuran data yang digunakan didapatkan pada ukuran blok 32 B, 128 B, 512 B dan 1024 B. Saat dilakukan pembacaan berkas yang sudah ada (*read*) dan juga pembacaan ulang (*re-read*), kecepatan paling tinggi didapat pada ukuran blok 64 B.



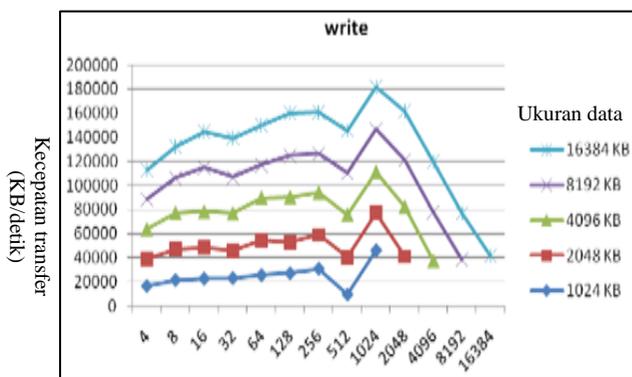
Ukuran blok (B)



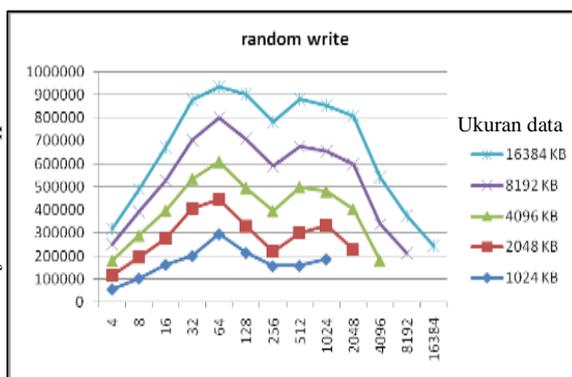
Ukuran blok (B)



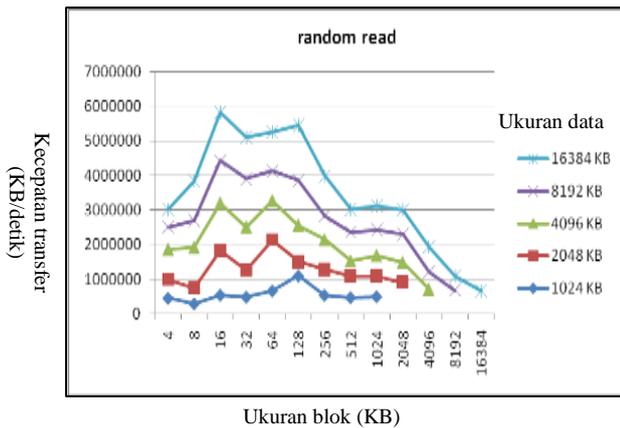
Ukuran blok (B)



Ukuran blok (B)



Ukuran blok (B)



Gambar 10 Kecepatan Pembacaan dan Penulisan *harddisk* iSCSI dari hasil *benchmark* menggunakan *iozone*

Pada saat dilakukan percobaan penulisan berkas pada lokasi yang acak (*random write*), yaitu penulisan file yang tidak berurutan, kecepatan penulisan paling tinggi didapatkan pada ukuran blok 64 B untuk setiap ukuran data yang diujikan. Sedangkan pada pembacaan informasi secara acak dari berkas (*random read*), yaitu pembacaan berkas secara tidak berurutan, didapatkan kecepatan paling tinggi pada blok data berukuran 16 B pada untuk berukuran 16384 KB dan 8192 KB, pada blok data berukuran 64 B untuk data berukuran 4096 KB dan 2048 KB serta pada ukuran blok data 128 B untuk data berukuran 1024 KB.

Dari hasil percobaan diatas, terlihat adanya penurunan kinerja pembacaan *disk* dari pengaksesan *disk* secara lokal terhadap pengaksesan *disk* melalui portokol iSCSI baik pembacaan maupun penulisan, penurunan kinerja *disk* ini terjadi karena iSCSI melakukan proses pembacaan dan penulisan melalui jaringan yang memungkinkan adanya antrian pembacaan ataupun penulisan data pada jaringan. Dari hasil percobaan, didapatkan bahwa kecepatan pembacaan mengalami perubahan kecepatan untuk setiap ukuran block data yang berbeda dan mengalami nilai maksimal pada ukuran blok 4096B (4KB), pada kecepatan baca peningkatan kecepatan hingga tiga kali lipat pada ukuran blok 4096 B (4KB) dari ukuran blok yang lebih kecil. Perubahan kecepatan juga terjadi ketika beberapa nilai parameter-paramter iSCSI lainnya dilakukan, dua percobaan perubahan parameter yang dilakukan menunjukkan adanya peningkatan kecepatan dari nilai parameter standard yang digunakan oleh iSCSI, dari dua percobaan tersebut juga didapatkan nilai maksimal kecepatan baca dan tulis didapatkan pada ukuran

blok data 4096 B (4KB). Sehingga dapat dikatakan bahwa penggunaan ukuran blok data 4096 Bytes dapat meningkatkan performa dari iSCSI yang secara standard menggunakan ukuran blok data 512B.

Dari hasil *benchmarking* menggunakan *iozone* juga menunjukkan bahwa ukuran blok data yang digunakan mempengaruhi kinerja baca/tulis dari iSCSI, selain itu ukuran data yang digunakan juga dapat mempengaruhi kinerja baca/tulis dari protokol iSCSI. Hal ini dapat digunakan untuk menentukan pengaturan ukuran blok data pada server iSCSI atau iSCSI target sesuai dengan kebutuhan dari iSCSI *initiator* yang akan dijadikan sebagai mesin *server* untuk layanan tertentu. Untuk layanan yang sering membutuhkan pembacaan atau penulisan data secara berurutan dapat menggunakan ukuran blok data antara 512 B atau 1024B, sedangkan untuk layanan yang sering melakukan pembacaan dan penulisan berkas secara acak (tidak berurutan) dapat menggunakan ukuran blok data 64 B.

IV. KESIMPULAN

Pada penelitian ini telah dilakukan pembangunan sistem iSCSI melalui jaringan virtual antara iSCSI target pada mesin fisik dan iSCSI *initiator* pada mesin virtual. Dari hasil analisa didapatkan kinerja dari protokol iSCSI ini dapat dipengaruhi oleh ukuran blok data yang digunakan, ukuran data yang dikirim, serta nilai parameter *MaxRecvDataSegmentLength*, *MaxXmitDataSegmentLength*, *FirstBurstLength* dan *MaxBurstLengt* dalam melakukan pengiriman data dari dan ke *server* (iSCSI target) baik pada saat penulisan maupun pembacaan berkas.

DAFTAR PUSTAKA

- [1] Wilson Yong H.W., Heng N.Y., Yao L.Z., Tow C.C., "Design and Development of Ethernet-Based Storage Area Network Protocol", *12th IEEE International Conference on Networks*, IEEE, 2004.
- [2] M. Bakke, J. Hafner, J. Hufferd, V. K., dan M. Krueger, "iSCSI Naming and Discovery," *RFC 3721*, April 2004, <http://www.ietf.org/rfc/rfc3721.txt>.
- [3] Meth, K.Z., Satran, J., "Design of the iSCSI Protocol", *Proceeding, 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies*, pp.116 – 122, 2003.
- [4] Gongye Zhou, Peng Chen, "RH-SCSI: A Reliable HyperSCSI Protocol for Networking

- Storage”. *International Conference on Networking, Architecture and Storage NAS 2007*, IEEE, 2007.
- [5] Swee liang A.W., Saleh, M. Aly Alsalihiy W.A.H, “Distributed iSCSI protocol over Hypervisor Storage on Local Area Network”, *9th Malaysia International Conference on Communications*, IEEE, 2009.
- [6] Jiang Guo-song, “Design and Implementation of iSCSI Out-of-band Storage Virtualization”, *International Conference on Intelligence Science and Information Engineering*, IEEE, 2011.
- [7] Satran, J., dkk, “iSCSI”, <http://www.ietf.org/rfc/rfc3720.txt>.
- [8] Hyotaek Lim, Saebom Choi. “Design and Implementation of iSCSI-based Virtual Storage System for Mobile Health Care”, *Enterprise networking and Computing in Healthcare Industry HEALTHCOM 2005*, IEEE, 2005.