

Studi Performa PC Cluster

A. S. Sakul, A. M. Rumagit, B. A. Sugiarto, ST, MT,

Jurusan Teknik Elektro-FT. UNSRAT, Manado-95115, Email: chill.24590@gmail.com

Abstract

The development of computer and network technologies led to the idea to combine multiple computers in a network, where the computer is able to cooperate into processing a problem. PC cluster is an amalgamation of several computers to process a job of processing, which way of submit the task / job to every computer and produces very high speed result. This study used PC clusters for cracking passwords. Password Cracking John The Ripper is a process of opening a password in this case Windows XP SP 2. The network topology used is a star topology and uses the internet protocol (IP) class C (192.168.1.0/24).

The number of nodes consists of three nodes, that are two slave nodes, one master node and one computer for sampling password. By doing so, the result can be seen the difference time between cracking passwords using single computer and PC cluster. After testing, it was found that the cracking using PC cluster is faster than single computer.

Keywords: BackTrack 4R 2, John the ripper mpi8, message passing interface (MPICH2), PC cluster.

Abstrak

Perkembangan teknologi komputer dan jaringan memunculkan ide untuk menggabungkan beberapa komputer dalam suatu jaringan dan dapat bekerjasama dalam memproses suatu program atau berkas yang berat. Aplikasi dari hal ini adalah PC (Personal Computer) cluster. PC Cluster merupakan penggabungan beberapa komputer untuk memproses suatu pekerjaan, pemrosesan dengan cara membagi-bagikan *task/job* ke tiap *computer* sehingga menghasilkan kecepatan yang sangat tinggi. Penelitian ini memanfaatkan PC cluster untuk melakukan *cracking password*. *Cracking Password John The Ripper* adalah suatu proses pembuka *password* dalam hal ini *windows XP SP 2*. Topologi jaringan yang digunakan adalah topologi *star* dan menggunakan *internet protocol (IP)* kelas C (192.168.1.0/24).

Banyaknya *nodes* berjumlah tiga *node*, yaitu dua *slave node*, satu *master node* serta satu komputer untuk pengambilan *sample password*. Setelah dilakukan pengujian maka didapatkan bahwa *cracking* menggunakan PC cluster lebih cepat dibandingkan dengan *single computer*.

Kata Kunci : BackTrack 4 R2, John the ripper mpi8, message passing interface (MPICH2), PC cluster.

I. PENDAHULUAN

Komputasi yang berkinerja tinggi (*high performance computing*) dapat dikaitkan dengan sebuah metode untuk meningkatkan kinerja dari sebuah aplikasi. Hal ini meliputi pembagian sebuah pekerjaan program aplikasi ke dalam beberapa unit paralel yang memungkinkan dan berkerja secara simultan untuk meningkatkan kecepatan dalam penyelesaian pekerjaan tersebut. Kebutuhan akan komputer berkinerja tinggi yang sangat erat hubungannya dengan super computer dan

massively parallel processors (MPP) telah dapat dipenuhi oleh komputer cluster. *Supercomputer* dan *massively parallel processors* (MPP) sangat kompleks untuk dikembangkan dan butuh biaya yang besar sedangkan komputer cluster dapat dibangun dari komputer-komputer sebagai *node* dengan harga yang lebih murah dengan jaringan berkecepatan tinggi. Komputer cluster dapat dibuat dari komputer-komputer *node* yang masing-masing terdiri dari satu atau lebih *processor*, *memory* yang dibagi oleh semua *processor* di dalam *node*, dan *device* lainnya seperti disk, dan terhubung dengan sebuah jaringan yang mengijinkan perpindahan data antar *node-node* tersebut.

Cluster saat ini telah menjadi sebuah *de-facto building block* untuk komputasi berkinerja tinggi. Dalam pengembangan infrastruktur *cluster*, sistem operasi sangat mempunyai peranan yang tidak bisa diabaikan. Sistem operasi harus bersifat *scalable* dan *cluster friendly*. Sistem operasi harus mendukung berbagai macam *platform* dan *device*. Selain itu sistem operasi yang dipilih juga harus mendukung virtualisasi sumberdaya dan fitur keamanan yang tangguh. Dalam tulisan ini akan dijelaskan langkah-langkah dan kebutuhan yang diperlukan dalam pembangunan komputer cluster dengan menggunakan *Backtrack 4 r2* merupakan linux distro berbasis ubuntu dan pengujian *cluster* pada *John the ripper* perangkat lunak untuk meretas *password*.

II. LANDASAN TEORI

A. Teknologi PC Cluster

Secara umum, saat orang membicarakan mengenai *clustering*, mereka akan mengacu pada suatu teknologi yang memungkinkan sejumlah komputer untuk bekerja sama menyelesaikan permasalahan komputasi biasa. Teknologi *cluster* ini dibutuhkan untuk meningkatkan kinerja beberapa komputer agar menjadi suatu sistem tunggal sumber daya komputasi yang melakukan pekerjaan besar. Dari sisi pengguna, ia tak merasa bahwa pekerjaan yang dia berikan telah dibagi ke mesin fisik yang berbeda.

B. Arsitektur Clustering

Suatu komputer *cluster* memiliki suatu arsitektur tertentu, dimana arsitektur tersebut memungkinkan suatu komputer dapat berkomunikasi antar komputer satu dengan lainnya. PC Cluster dirancang untuk meningkatkan kemampuan kinerja dari komputer-komputer yang berada pada suatu jaringan komputer, untuk dapat meningkatkan berikut hal-hal yang harus diperhatikan.

Toleransi kesalahan (*fault tolerance*), yang dapat menyebabkan *node* lainnya (misal komputer B) akan mengambil alih kerja *node* utama (sebutan untuk *node* yang melakukan eksekusi program tertentu, misal komputer A) ketika *node* 1 tersebut mengalami kegagalan. *Client* tidak akan melihat pergantian peran ini. Dengan begitu, *down time* pun dapat dikurangi secara drastis. Penyerataan beban (*load-balancing*), yang dapat mendistribusikan beban satu *node* ke semua *node* anggota *cluster*. Bagian terpenting dari komputer *cluster* adalah adanya sebuah aplikasi *middleware* yang dapat menggabungkan seluruh anggota dalam *cluster* sehingga dapat bekerja sama. Tugas utama dari aplikasi *middleware* ini adalah untuk komunikasi dan sinkronisasi antar komputer.

C. Komputasi Paralel

Komputasi *parallel independent* merupakan beberapa komputer independen yang digabungkan menjadi satu sistem dengan menggunakan jaringan dan *software*. Elemen pemroses dapat berupa komputer tunggal dengan banyak prosesor, beberapa komputer yang terhubung dalam satu jaringan, perangkat keras yang dikhususkan untuk melakukan komputasi paralel, ataupun kombinasi dari perangkat-perangkat yang telah disebutkan.

D. Jenis-jenis Cluster Secara Mendasar Terdapat Tiga macam cluster :

High availability cluster (HA)

High availability cluster, sering disebut sebagai *failover cluster* pada umumnya diimplementasikan untuk tujuan meningkatkan ketersediaan layanan yang disediakan oleh *cluster* tersebut. Elemen *cluster* akan bekerja dengan memiliki *node-noderedundant*, yang kemudian digunakan untuk menyediakan layanan saat salah satu elemen *cluster* mengalami kegagalan. Implementasi *cluster* jenis ini akan mencoba untuk menggunakan redundansi komponen *cluster* untuk menghilangkan kegagalan di satu titik (*Single Point of Failure*).

High performance computing (HPC)

High performance computing (HPC) adalah *cluster* yang di dedikasikan untuk melakukan proses komputasi secara paralel sehingga dapat menghasilkan performa komputasi yang tinggi dengan kecepatan yang tinggi pula. Mekanisme yang digunakan untuk melakukan pemrosesan secara paralel adalah dengan menggunakan *Message passing interface (MPI)*. Proses yang dijalankan oleh sebuah aplikasi dapat dibagi untuk dikirimkan ke masing-masing *computer node* yang kemudian masing-masing *computer node* tersebut mengolah dan mengembalikan hasilnya ke *computer head node*.

Load balancing cluster

Cluster kategori ini beroperasi dengan mendistribusikan beban pekerjaan secara merata melalui beberapa *node* yang bekerja di belakang (*back-end node*). Umumnya *cluster* ini akan dikonfigurasi sedemikian rupa dengan beberapa *front-end load balancing redundant*. Karena setiap elemen dalam sebuah *cluster load balancing* menawarkan layanan

penuh, maka dapat dikatakan bahwa komponen *cluster* tersebut merupakan sebuah *cluster active/cluster high availability (HA) active*, yang bisa menerima semua permintaan yang diajukan oleh klien.

E. Network File System (NFS)

Salah satu *protocol* yang dipergunakan pada komputasi paralel adalah *Network File System (NFS)*, *NFS* adalah protokol yang dapat membagi sumber daya melalui jaringan. *NFS* dibuat untuk dapat independent dari jenis mesin, jenis system operasi, dan jenis protokol transport yang digunakan. Hal ini dilakukan dengan menggunakan *RPC*, *NFS* memperbolehkan pengguna yang telah diijinkan untuk mengakses *file-file* yang berada di remote host seperti mengakses *file* yang berada di lokal. Protokol yang digunakan protokol mount menentukan host remote dan jenis *file* sistem yang akan diakses dan menempatkan di suatu direktori, protokol *NFS* melakukan I/O pada *remote file system*. Protokol *mount* dan protokol *NFS* bekerja dengan menggunakan *RPC* dan mengirim dengan protokol *TCP* dan *UDP*. Kegunaan dari *NFS* pada komputasi paralel adalah untuk melakukan sharing data sehingga setiap *nodeslave* dapat mengakses program yang sama pada *node master*.

F. Secure Shell (SSH)

Secure shell (SSH) adalah protokol standar yang membentuk jalur yang aman pada komunikasi antar komputer. *SSH* menggunakan teknik enkripsi *public key* pada sistem autentikasi pengguna untuk mengakses komputer yang lain. *SSH* memberikan sistem enkripsi pada jalur yang digunakan, sehingga memberikan tingkat keamanan data yang tinggi. *SSH* biasa digunakan untuk melakukan *remote login* dan menjalankan perintah pada komputer remote, tetapi *SSH* juga dapat digunakan sebagai *tunnel* jaringan, melakukan penerusan pada *port TCP*, dan koneksi X11. Selain itu dapat juga digunakan untuk mentransfer suatu *file* dengan protokol *Secure file transfer protocol (SFTP)* atau *Secure copy protocol (SCP)*. *SSH server* bekerja pada *port 22*. Sebuah *client program Secure shell (SSH)* digunakan untuk membangun koneksi ke *Secure shell (SSH) daemon* untuk dapat diremote.

G. Secure Copy Protocol (SCP)

Adalah sebuah *tools* yang sangat berguna dalam hal mengirim dan menerima *file* melalui *secure shell (SSH)* antara komputer satu dengan komputer yang lain dalam satu jaringan.

H. Cryptography

Cryptography adalah teknik penulisan dan pembacaan kode atau sandi rahasia (*cipher*). *Cryptography* digunakan untuk pengamanan informasi agar informasi tetap privasi dan untuk mengautentikasi identitas pengirim atau penerima informasi. *Cryptography* dapat juga menyediakan keutuhan (*integrity*) informasi, sebab ia hanya mengijinkan orang atau proses yang berhak yang bisa mengakses informasi, dan dapat mendeteksi kerusakan atau perubahan informasi asli. *Key* adalah sebuah deretan dari matematik penulisan kode (*enchipering*) dan penguraian kode (*denchipering*) informasi.

I. John The Ripper

John The Ripper adalah suatu *tools cracker password* yang cepat yang tersedia pada banyak platform, antara lain UNIX, Windows, DOS, BeOS dan OpenVMS. Tujuan utama dari *John The Ripper* adalah untuk mendeteksi kelemahan *password* pada sistem UNIX (termasuk *Linux*). *John The Ripper* merupakan program yang dapat membantu administrator menentukan kelayakan suatu *password*. Namun *John The Ripper* juga digunakan oleh pihak-pihak yang tidak berkepentingan untuk mendapatkan *password* seorang *user*. *John The Ripper* mampu menangani berbagai tipe enkripsi yang dikenakan terhadap *password*, dan *John The Ripper* juga menyediakan fasilitas untuk membuat makalah IF3058 KRIPTOGRAFI tahun 2010, permutasi dari setiap kata yang ada di dalam *wordlist* atau kamusnya.

Berikut ini teknik enkripsi yang didukung oleh *John* antara lain *Data Encryption Standard (DES)*, *Message Digest (MD5)* dan lain-lain. Secara umum, *John The Ripper* mendukung dan dapat mendeteksi tipe *crypt* dari sistem UNIX:

Traditional and double-length berbasis *Data Encryption Standard (DES)*

Serial digital interface (SDI) extended berbasis *Data Encryption Standard (DES)*

Free BSD berbasis *Message Digest5 (MD5)* (yang sekarang juga digunakan di dalam *Linux* dan *CiscoIOS*)

OpenBSD berbasis *Blowfish* (sekarang juga digunakan dalam beberapa distribusi *Linux*)

John The Ripper juga mendukung *Kerberos/AFS* dan *hash Windows LM* berbasis *Data Encryption Standard (DES)*

J. Pemrograman Message Passing Interface Chameleon (MPICH)

Message Passing Interface (MPI) adalah suatu spesifikasi *library* pemrograman untuk meneruskan pesan (*message-passing*), yang diajukan sebagai standar oleh berbagai komite dari *vendor*, pelaksana dan pemakai MPICH2 merupakan pengembangan dari MPICH1 yang menggunakan MPI1 untuk standar *message passing library*. Pada MPICH2 mampu mengimplementasikan baik MPI1 maupun MPI2 sebagai pengembangan dari MPI1. MPI sendiri merupakan singkatan dari *Message Passing Interface* sedangkan kata CH berasal dari *Chameleon*. Dalam pemodelan menggunakan *message passing*, suatu *process* adalah sebuah pencacah program dan ruang alamat. Proses dapat memiliki banyak *thread* (pencacah program dan *memory* lokal) yang saling berbagi ruang alamat.

MPI dalam hal ini berfungsi sebagai alat komunikasi di antara proses, yang saling memiliki ruang terpisah. Komunikasi ini terutama berupa sinkronisasi dan perpindahan data antar proses. Informasi dari domain komunikasi seluruh proses disimpan di sebuah variabel yang disebut *communicators*, misalnya *MPI_COMM_WORLD* yang mencakup keseluruhan proses. Paralel dalam MPI bersifat *Multiple Instruction Multiple Data (MIMD)*. Pemrograman paralel menggunakan MPI bersifat eksplisit, yaitu ditentukan dalam program secara jelas. Disediakan beberapa fungsi dasar untuk keperluan ini, yaitu *MPI_Send* (mengirim), *MPI_Recv* (menerima), dan *MPI_Sendrecv* (mengirim dan menerima sekaligus).

Fungsi-fungsi ini masuk dalam kelompok komunikasi titik-ke-titik (*point-to-point*). Komunikasi dalam MPI bersifat kooperatif, yaitu tiap-tiap proses saling bekerjasama. Operasi dari fungsi –fungsi komunikasi titik ke titik memiliki *2mode*, yaitu *blocking* dan *non-blocking*.

K. Transmission Control Protocol/Internet Protocol (TCP/IP)

TCP/IP adalah standar komunikasi data yang digunakan untuk saling tukar-menukar informasi dalam suatu jaringan. Protokol *TCP/IP* dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas *Wide Area Network (WAN)*. *TCP/IP* merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme transport jaringan fisik yang digunakan, sehingga dapat digunakan dimana saja. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti *Microsoft Windows* dan keluarga *UNIX*) untuk membentuk jaringan yang heterogen. *TCP/IP* dibentuk dalam beberapa lapisan (*layer*).

L. Topologi Jaringan

Topologi jaringan adalah suatu aturan atau cara untuk menghubungkan komputer yang satu dengan komputer yang lainnya sehingga membentuk suatu jaringan. Topologi jaringan juga dapat didefinisikan sebagai gambaran secara fisik dari pola hubungan antara komponen jaringan, yang meliputi *Server*, *Workstation*, *Hub* dan pengkabelannya.

M. Kabel Unshielded Twisted Pair (UTP)

Kabel *Unshielded Twisted Pair (UTP)* terdiri atas 8 *wire* yang ditandai dengan 8 warna berbeda. Kabel *UTP* ini menggunakan bahan dasar tembaga, tanpa selubung pembungkus luar. Didalamnya terdapat 4 pasang kabel yang setiap pasangannya dipilin, masing-masing pasangan kabel tersebut saling melilit (*twisted*) antara satu *wire* dengan *wire* lainnya. Jenis kabel ini biasanya digunakan untuk menghubungkan beberapa unit komputer melalui perantara *HUB/Switch* yang berfungsi sebagai konsentrator maupun *repeater*. Penggunaan kabel *Unshielded Twisted Pair (UTP)* model *Straight Through* pada jaringan lokal biasanya akan membentuk topologi *Star* atau *Treed* dengan *HUB/switch* sebagai pusatnya.

N. Konektor Registered Jack (RJ) 45

Konektor RJ 45 adalah konektor kabel *Ethernet* yang biasa digunakan dalam topologi jaringan komputer *Local Area Network (LAN)* maupun jaringan komputer tipe lainnya. Singkatan dari RJ ialah *Register Jack* yang merupakan standar peralatan pada jaringan yang mengatur tentang pemasangan kepala konektor dan urutan kabel, yang digunakan untuk menghubungkan 2 atau lebih peralatan telekomunikasi (*Telephone Jack*) ataupun peralatan jaringan (*Computer Networking*). RJ-45 merupakan kode seri dari *Registered Jack*, suatu interface fisik dari jaringan kerja *Network*, untuk kegunaan telekomunikasi dan komunikasi data.

O. Switch

Switch merupakan perluasan dari konsep *bridge*, sehingga pada dasarnya prinsip kerja *bridge* dan *switch* sama. Switch adalah hub pintar yang mempunyai kemampuan untuk menentukan tujuan *MAC address* dari *packet Port-port* pada switch bisa berupa bagian tetap (*fixed part*) atau berupa modul-modul ekspansi yang harus dipasangkan terlebih dahulu pada *slot* yang tersedia pada switch. Switch berfungsi untuk menghubungkan kabel-kabel *Unshielded Twisted Pair(UTP)* (Kategori 5/5e) komputer yang satu dengan komputer yang lain.

Dalam mengolah data switch dapat digolongkan dalam tiga jenis *Store and Forward switch* akan meneruskan *frame* setelah data di terimasecara lengkap, *CutThrough Switch* Meneruskan *Frame* tanpa menunggu penerimaan *frame* secara lengkap. *Fragment Free (Hybrid)* merupakan kompromi dari kedua jenis switch diatas

III. METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

Dalam pelaksanaan tugas akhir ini penulis mengambil tempat penelitian pada Ruang Laboratorium Sistem Komputer (LSK), Jurusan Teknik Elektro, Fakultas Teknik Universitas Sam Ratulangi (UNSRAT), dan rumah penulis.

B. Bahan dan Peralatan

Dalam perancangan *PC cluster* ini, penulis menggunakan 3 buah PC yang terdiri dari 1 *master computer*, dan 2 *slave computers* serta satu buah komputer untuk pengambilan *sample password* dengan sistem operasi *Linux Backtrack 4r2* ter-install didalamnya dengan beberapa program pendukung antara lain *John The Ripper mpi8*, *MPICH2*, *Secure Shell* dan *Network File System*. Dan peralatan yang digunakan adalah switch *ethernet 8 port*, kabel *Unshielded Twisted Pair* (5 meter), konektor *RJ-45* (6 buah) dan tang *crimper*.

C. Prosedur Penelitian

Dalam perancangan tugas akhir ini mula-mula penulis mencari referensi-referensi yang berhubungan dengan judul tugas akhir serta mencari pokok permasalahan yang harus diselesaikan. Kemudian dilakukan prancangan sistem baik *software* maupun *hardware* lalu sistem di implementasikan. Setelah itu penulis melakukan penelitian.

D. Desain Sistem

Pada pembuatan tugas akhir ini, penulis merancang *PC cluster* sederhana yang bertujuan untuk melihat performa *PC cluster* dalam hal ini membuka *password windows XP SP 2* dengan menggunakan *john the ripper*. Topologi jaringan yang digunakan adalah topologi *star* (bintang) dan menggunakan *internet protocol (IP)* kelas C (192.168.1.0/24). Jumlah *nodes* berjumlah 3 *node*, 2 *slaves nodes* dan satu *master node* serta satu buah komputer untuk pengambilan *sample password*. Pada penulisan tugas akhir ini penulis merancang lingkungan jaringan dengan menggunakan *Linux Backtrack 4 r2* untuk menjalankan *MPICH2* dan *john the ripper mpi8*.

John the ripper mpi8 merupakan pemroses yang akan di-install pada tiap *slaves nodes* dan pada *nodemaster*. Kemudian ambil *sample password* di *PC XP SP 2* menggunakan *flash disc* yang sudah ter-install *Linux Backtrack 4r2* kemudian *copy hash.txt* ke *master computer* di folder */home/root/hash*, setelah itu gunakan *Secure Copy Protocol(SCP)* untuk *sharefile* ke *slave computer (node 1 dan node 2)*.

E. Perancangan Jaringan

Perancangan jaringan yang dibuat menggunakan topologi *star* dimana semua *nodes* terhubung pada switch yang menjadi titik pemersatu antara satu sama lain. IP yang digunakan adalah 192.168.1.0/24, alokasi IP pada tiap *node* dapat dilihat pada gambar 1.

F. Instalasi Backtrack 4 r2

Dalam hal ini, peran Backtrack yaitu sebagai mesin yang melakukan proses *cracking password* (membuka *password*). Backtrack harus di instal ke *master computer* dan *slave computer*. Untuk menginstal Backtrack masukkan *DVD drive* kemudian pilih opsi kedua *Start BackTrack FrameBuffer (800x600)*. Pada saat pembuatan tugas akhir ini, Backtrack yang digunakan adalah *backtrack 4 r2*.

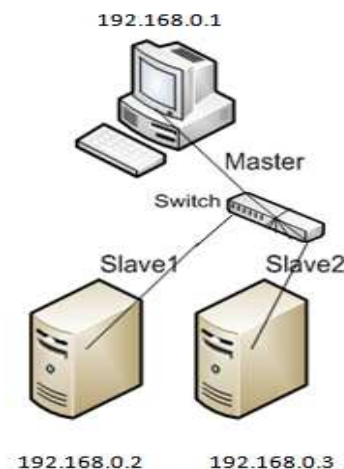
G. Install message passing interface Chameleon2 (MPICH2)

Untuk menginstal paket MPICH2 cukup mudah karena sudah tersedia di repository *Linux Backtrack*. Pada terminal masukan command menggunakan perintah *apt-get install mpich2* yang dieksekusi pada aplikasi konsol linux. Install MPICH2 pada setiap *personal computer (PC)*.

H. Instalasi John the ripper mpi8

Cara instalasi *John the ripper* versi *mpi8*

Buka aplikasi konsol dan eksekusi beberapa perintah berikut ini di *master computer* serta di *slave computer* `oot@server:~#wgethttp://www.bindshell.net/tools/john_the_ripper/John-1.7.3.1-all-mpi8.tar.gz`.



Gambar 1
Perancangan Jaringan

```

root@server:~#tar -xvfz John-1.7.3.1-all-mpi8.tar.gz
root@server:~#cd / John-1.7.3.1-all-mpi8/src
root@server:~# / John-1.7.3.1-all-mpi8/src make clean linux-
x86-sse2
root@server:~#cd
root@server:~#mv -f John-1.7.3.1-all-mpi8 /pentest/password
/john-mpi

```

I. Instalasi Open SSH (Secure Shell)

Otentikasi *Open SSH (Secure shell)* diperlukan agar dapat menjalankan *Message Passing Interface 2 (MPICH2)* dalam jaringan dengan aman. Karena *openssh* mempunyai jalur yang aman dengan proteksi *password* yang terenkripsi. Setiap *user* harus dapat mengakses akun *user* melalui protokol *Secure shell (SSH)* pada semua *host* tanpa menginput ulang *password*, yaitu dengan cara konfigurasi *ssh* pada *master* dan *slave* komputer.

Berikut ini cara instalasi *Secure shell (SSH)* master dan *Secure shell (SSH) client*

```

root@server:~#apt-get install openssh-client
root@server:~#apt-get install openssh-server
root@server:~#. /etc/init.d/ssh start

```

J. Install Network File System (NFS)

Network File System (NFS) berfungsi untuk direktori *sharing* yang akan di gunakan secara bersama-sama oleh semua komputer. Hal ini di karenakan untuk menjalankan *MPICH2* semua *personal computer (PC)* menggunakan *source program* yang sama sehingga memerlukan mekanisme *sharing file*.

Install *NFS* pada *master* dan *slave computer* dengan cara masukan perintah ke *command prompt*

```

root@server:~#apt-get install nfs-kernel-server
root@server:~#apt-get install nfs-common

```

K. Konfigurasi alamat internet protocol (IP) pada master dan slave node.

Server Computer

```

root@server:~# ifconfig eth0 192.168.0.1 netmask
255.255.255.0

```

Node Computer

```

root@node1:~# ifconfig eth0 192.168.0.2 netmask
255.255.255.0

```

```

root@node2:~# ifconfig eth0 192.168.0.3 netmask
255.255.255.0

```

L. Konfigurasi host name pada master dan slave

Konfigurasi server computer

```

root@server:~#echo "server" > /etc/hosts

```

Konfigurasi node computer

```

root@node1:~#echo "node1" > /etc/hosts

```

```

root@node2:~#echo "node2" > /etc/hosts

```

Untuk melihat jaringan master computer dan slave computer sudah terhubung, ketikkan *script* berikut ini ke *master computer*

```

root@server:~#nano /etc/hosts

```

M. Konfigurasi Open Secure Shell (SSH)

Konfigurasi *ssh* terdiri dari dua tahap yaitu

Konfigurasi *ssh* di *server computer*, dengan perintah

```

root@server:~#start-sshd
root@server:~#ssh-keygen -t rsa
root@server:~#ssh root@192.168.0.1 mkdir -p /root/.ssh
root@server:~#cat/root/.ssh/id_rsa.pub|ssh root@ 192.168.0.1
"cat>>.ssh/authorized_keys"

```

Konfigurasi *ssh* di *node computer*, dengan perintah

```

root@server:~#cat/root/.ssh/id_rsa.pub|ssh root @192.168.0.2
"cat>>.ssh/authorized_keys"
root@server:~#cat/root/.ssh/id_rsa.pub|ssh root@192.168.0.3
"cat>>.ssh/authorized_keys"
root@server:~#scp/root/.ssh/authorized_keys2
root@192.168.0.2:/root/.ssh/
root@server:~#scp/root/.ssh/authorized_keys2
root@192.168.0.3:/root/.ssh/

```

N. Konfigurasi MPICH2

Konfigurasi *MPICH* di server computer dan slave computer dengan perintah

```

root@server:~#touch /etc/mpd.conf
root@server:~#chmod 600 /etc/mpd.conf
root@server:~#echo "secretword=silcluster" >> /etc/mpd.conf
root@server:~#touch mpd.hosts
root@server:~#chmod 600 mpd.hosts
root@server:~#echo "192.168.0.1:1" >> mpd.hosts
root@server:~#echo "192.168.0.1:2" >> mpd.hosts
root@server:~#echo "192.168.0.1:3" >> mpd.hosts
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
root@server:~#scp mpd.hosts root@192.168.0.3:/root/

```

O. Pengecekan Mesin List untuk parallel environment

Cara pengecekan mesin list dengan cara mengetikkan *script* dibawah ini ke dalam *master computer*.

```

root@server:~#mpdboot -v -files=/root/mpd.hosts -n 3

```

```

runing mpdallexit on server

```

```

LAUNCHED mpd on server via

```

```

RUNNING: mpd on server

```

```

LAUNCHED mpd on 192.168.0.2 via server

```

```

LAUNCHED mpd on 192.168.0.3 via server

```

```

RUNNING: mpd on 192.168.0.2

```

```

RUNNING: mpd on 192.168.0.3

```

Cara untuk melihat hubungan jaringan

```

root@server:~#mpdtrace

```

output dari *mpdtrace* :

```

server

```

```

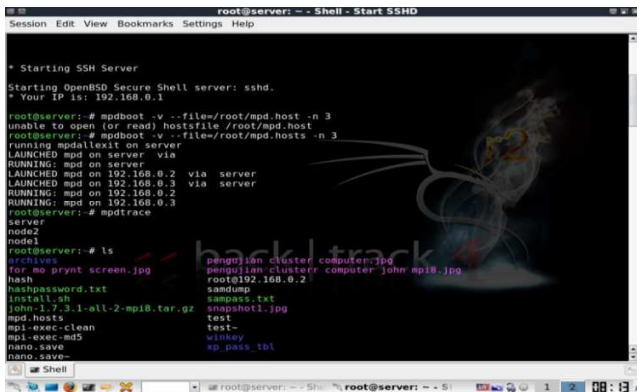
node1

```

```

node2

```



Gambar 2. Tampilan pengecekan jaringan *parallel environment*

Gambar 2 menunjukkan cara pengecekan mesin list pada jaringan *parallel environment*

P. Implementasi Sistem

Pengambilan *hash password* pada *windows xp sp2*, Download *Backtrack 4 r2 Boot CD ISO* dan burn ke dalam CD-R, semua alat-alat yang akan kita gunakan dalam tutorial ini terdapat pada *Backtrack 4 r2 Boot CD*. Masukkan *Backtrack 4 r2 Boot CD* ke dalam CD-ROM, *reboot* dan mengatur CD-ROM sebagai perangkat *boot* pertama di BIOS. *Backtrack 4 r2* akan mulai untuk *boot* dan menanyakan apa resolusi layar yang ingin anda gunakan. Pilih resolusi yang akan mendukung monitor (saya menggunakan 2 untuk 1024x768) kemudian tekan *enter*.

Ketika *Backtrack 4r2* selesai *booting*, ketik *startx* tekan *enter* untuk menjalankan *desktop mode* grafis *Mount hard disk* lokal, kemungkinan besar *hda1*, caranya *mount/dev/hda1*, sehingga mengubah direktori kerja hadir untuk *ramdisk* sehingga kita ruang untuk bekerja dengan *file* kita akan menciptakan *Linux Command cd/ramdisk/ Backtrack 4r2* dilengkapi perangkat lunak *bkhive*, *samdump2*, dan *pwdump7* yang bekerja sebagai editor yang berfungsi untuk mendapat *system key* pada *file system*. untuk mendapatkan kunci sistem kita perlu menggunakan *Bkhive* pada berkas *SYSTEM* yang terletak di *C: \ WINDOWS \ system32/config \ SYSTEM* dan *samdump2* pada berkas *SAM* yang terletak di *C: \ WINDOWS \ system32/config \ SAM*.

Sekarang kita memiliki kunci sistem yang dapat digunakan untuk membatalkan *Syskey*. yang berada di *SAM*, ekstrak *hash password* ke dalam *format file pwdump samdump2/mnt/hda1/WINDOWS /system32/config/sam* di simpan *syskey.txt* sandi *hashes.txt*. Pada titik ini kita telah mendapatkan *format file pwdump* disebut sandi *hashes.txt*. Untuk menampilkan *flash disk* menggunakan perintah pada *root@server:~#Mount /dev/sdc1 /mnt/usb1*, serta Untuk membuat *directory flash disk* *root@server:~#mkdir /mnt/usb1* Copy file ke *directory* yang telah dibuat *root@server:~#cp hashes.txt /mnt/usb*. Kemudian *copyfile* ke *master computer* *root@server:~#cp hashes.txt /root/hash* Kemudian *Share /root/hash* ke tiap *node (slave computer)* *root@server:~#scp /root/hash root@192.168.0.2:/root/* *root@server:~#scp /root/hash root@192.168.0.3:/root/* Menampilkan *hash pass* yang akan di eksekusi *root@server:~#cat hash* dari *cat hash.txt*

Q. Proses eksekusi cracking password /root/hash

Cracking Password /root/hash pada *PC cluster (cluster 3 node)*

Ketik *script* ini di *master computer*, selanjutnya *share file* *mpd.hosts* ke tiap *slave*

```
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
```

```
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
```

Kemudian ketik *script* ini untuk menghubungkan jaringan *paralell cluster*

```
root@server:~# mpdboot -v -files=/root/mpd.hosts -n 3
```

Cara menampilkan seluruh jaringan yang terhubung *root@server:~# mpdtrace*

```
Server
```

```
Node 2
```

```
Node 1
```

Proses eksekusi *PC Cluster (Cluster 3 node)* ketikan *script* berikut ini ke dalam *computer* *root@server:~# mpiexec -host 192.168.0.1 -np 3 /pentest/passwords/ john-mpi/run/john - -format: LM /root/hash*. Setiap eksekusi *sample pass* dalam hal ini */root/hash* jangan lupa menghapus rekam jejak dari hasil tersebut dengan mengetikan *script* kedalam sitem command prompt *root@server:~#mpdallexit*

Cracking password /root/hash pada *PC cluster (cluster 2 node)*

Ketik *script* ini di *master computer*, selanjutnya *share file* *mpd.hosts* ke tiap *slave*

```
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
```

```
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
```

Kemudian ketik *script* ini kedalam *master computer* untuk menghubungkan jaringan *paralell cluster*

```
root@server:~# mpdboot -v -files=/root/mpd.hosts -n 2
```

Cara menampilkan seluruh jaringan yang terhubung *root@server:~# mpdtrace*

```
Server
```

```
Node 1
```

Proses eksekusi *PC Cluster (Cluster 2 node)* ketikan *script* berikut ini ke dalam *computer master* *root@server:~#mpiexec -host 192.168.0.1 -np 2 /pentest/passwords/ john-mpi/run/john - -format: LM /root /hash .*

Cracking password /root/hash pada *Single Computer*

Ketik *script* ini di *master computer*, selanjutnya *share file* *mpd.hosts* ke tiap *slave*

```
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
```

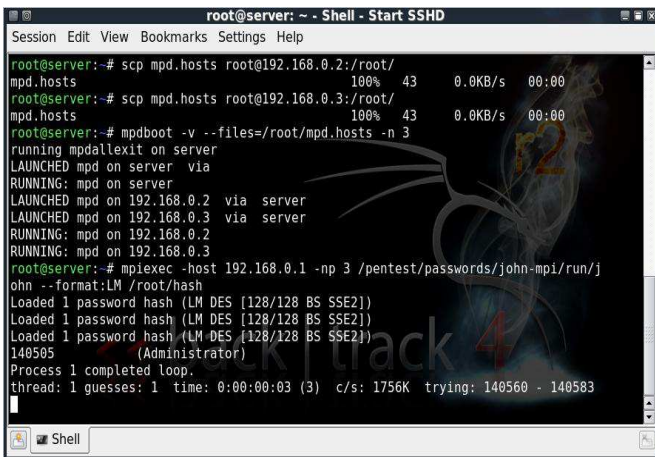
```
root@server:~#scp mpd.hosts root@192.168.0.2:/root/
```

Kemudian ketik *script* ini ke *master computer* untuk menghubungkan jaringan *paralell cluster* *root@server:~# mpdboot -v -files=/root/mpd.hosts -n 3*. Cara menampilkan seluruh jaringan yang terhubung *root@server:~# mpdtrace*

```
Server
```

Proses eksekusi *Single computer*

```
root@server:~# mpiexec -host 192.168.0.1 -np 1 /pentest/passwords/ john-mpi/run/john - -format: LM /root/ hash .
```



Gambar 3. Tampilan hasil eksekusi pada PC Cluster (Cluster 3 node)

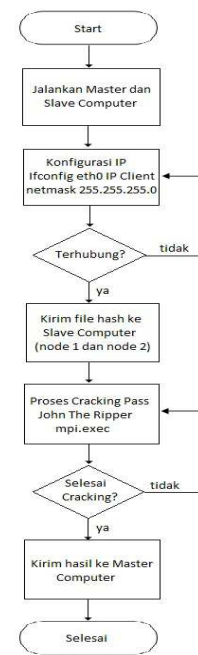


Gambar 4. Tampilan hasil eksekusi pada PC Cluster (Cluster 2 node)

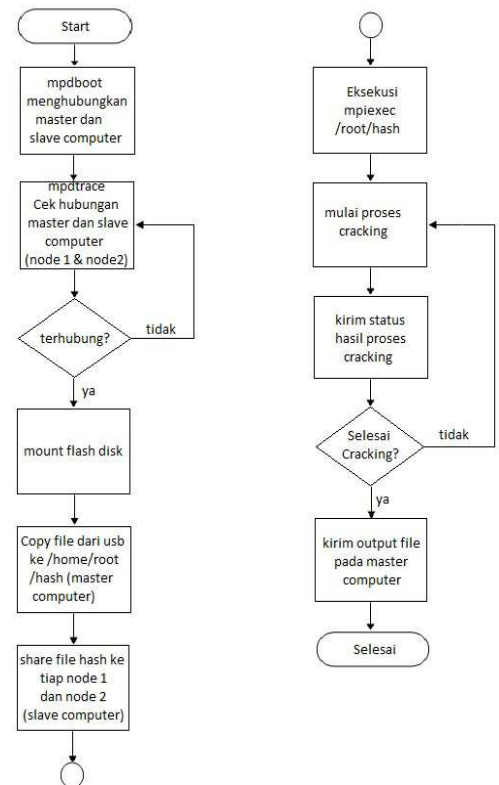


Gambar 5. Tampilan hasil eksekusi pada single computer

Gambar 3 merupakan hasil dari *cracking password 3 node*
 Gambar 4 merupakan hasil dari *cracking password 2 node*
 Gambar 5 adalah tampilan hasil eksekusi pada *single computer*



Gambar 6. Flowchart proses cracking password John The Ripper mpi8



Gambar 7. flowchart Cluster untuk proses Cracking Password

Gambar 6 adalah *Flowchart proses cracking password John The Ripper mpi8*

Secara keseluruhan proses yang dilakukan oleh cluster untuk proses *cracking* dapat dilihat pada gambar 7 *flowchart Cluster untuk proses Cracking Password*.

IV. PENGUJIAN DAN ANALISA

A. Pengujian Cracking Password

Pada bab ini penulis akan melakukan pengujian dan analisa menggunakan *cluster computer* untuk *crackingpasswordfile hash* yang berada di */root/hash*. Pengujian tersebut dimaksudkan agar dapat diketahui perbandingan waktu *cracking password* antara *single computer* dan *PC cluster* serta peningkatan *performance* pada *cluster* seiring bertambahnya jumlah *node*.

Kita akan mendapatkan peningkatan yang terjadi pada cluster dengan menggunakan persamaan $S = \frac{T1}{Tn}$ (1)

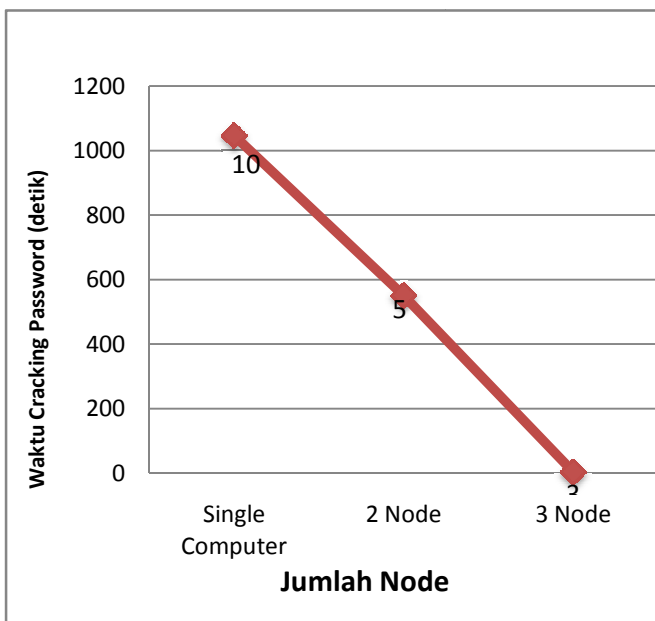
Dimana,

- S = Peningkatan kecepatan
- T1 = Waktu *crackingpassword* pada *single computer*
- Tn = Waktu *crackingpassword* pada *cluster n node*

Gambar 8 menunjukkan grafik dari *Cracking Password*, sedang pengujiannya dapat dilihat pada tabel I.

TABEL I
PENGUJIAN CRACKING PASSWORD

Jumlah Node	Waktu Cracking Password (detik)	Peningkatan Kecepatan
Single Computer	1047	1x
Cluster 2 node	551	1.9x
Cluster 3 node	3	349x



Gambar 8 Grafik *Cracking Password*

B. Analisa

Setelah dilakukan pengujian pada *cluster*, penulis akan memperlihatkan hasil perbandingan *cracking password john the ripper mpi8* pada *single computer* dan *PC cluster* serta menganalisa waktu serta *output* data yang telah didapat dari pengujian tersebut. Panjang suatu karakter *password* yang akan dibuka (*cracking password*) sangat berpengaruh pada waktu pemrosesan. Begitu pula pada penambahan *node (PC cluster)*, semakin banyak *node* di tambah maka semakin cepat pula *output* yang di proses didapatkan. Pada pengujian sebelumnya telah didapatkan data dari hasil *cracking password* pada *1 password (/root/hash)* yang sama dengan menggunakan *single computer, 2node, dan 3node*.

Dari pengujian serta data yang di dapatkan maka perancangan *PC Cluster* dinyatakan berhasil dengan persentase peningkatan *performance* saat *cracking password* dari *single computer* ke *PC Cluster* dapat dilihat pada tabel I dan II serta grafiknya dapat dilihat pada gambar 9.

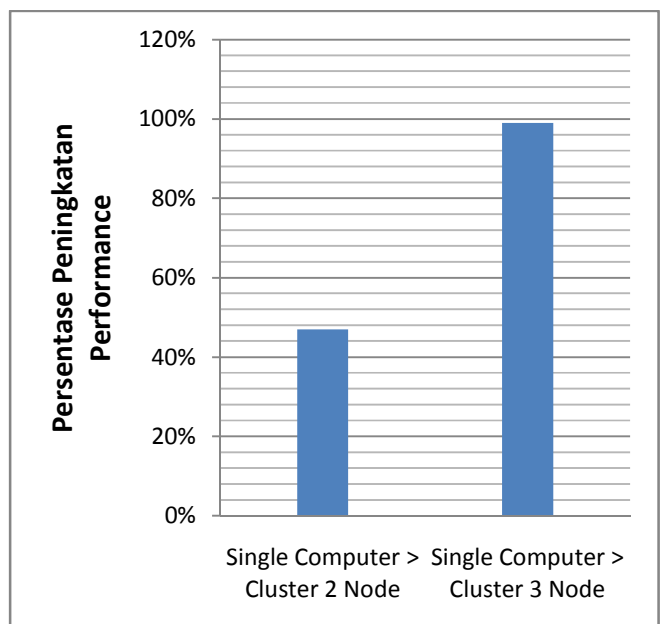
Rumus peningkatan performance $\frac{TSC - TCn}{TSC} \times 100\% = PP (\%)$

dimana,

- TSC = waktu *cracking password* dengan *single computer*
- TCn = waktu *cracking* dengan *cluster n node*
- PP (%) = persentase peningkatan *performance* dari *single computer ke PC Cluster*

TABEL II
PENINGKATAN PERFORMA

Peningkatan Performance (%)	Hasil
Single Computer > Cluster 2 Node	47%
Single Computer > Cluster 3 Node	99%



Gambar 9 Persentase Peningkatan *Performance*

V. KESIMPULAN DAN SARAN

C. Kesimpulan

Berdasarkan data hasil penelitian dan analisis data yang telah dilakukan maka dalam penyelesaian tugas akhir ini dapat diambil kesimpulan yang kan dijabarkan dibawah ini.

Terjadi peningkatan *performance* pada *pc cluster* seiring bertambahnya jumlah *node*.

Setiap komputer pada sistem *cluster* MPICH2 akan menjalankan proses pada *file* yang sama, apabila tidak maka *pc cluster* tidak akan bisa jalan.

Sistem *pc cluster* akan lebih cocok apabila digunakan untuk penyelesaian *task/job* besar

D. Saran

Perlu adanya penelitian dan pengembangan lebih lanjut tentang *PC Cluster* dengan topik dan judul yang berbeda, mengingat teknologi

Keterbatasan alat dalam membuat *system cluster* sangat penting, karena dalam penelitian ini komputer yang digunakan memiliki spesifikasi yang berbeda.

Sistem ini dapat digunakan untuk *penetration testing* untuk menguji kelemahan konfigurasi suatu sistem dan membuka jalan untuk penelitian lainnya. *cluster* semakin diperlukan untuk beban kerja yang lebih besar.

DAFTAR PUSTAKA

- [1] B. Wilkinson, & M. Allen , *Parallel Programming Teknik Dan Aplikasi Menggunakan Jaringan Workstation Dan Komputer Paralel* Edisi 2, Cetakan Pertama, Penerbit ANDI, Yogyakarta, 2005.
- [2] B.W.K Malubaya, Perancangan PC Cluster Untuk Render Animasi 3D, *Skripsi* Program Studi Teknik Elektro Universitas Sam Ratulangi, Manado, 2014
- [3] D.C. Suhendra, Implementasi Sistem PC Cluster Pada Operasi Perkalian Matriks, *Skripsi* Program Studi Teknik Elektro Universitas Sam Ratulangi, Manado, 2010
- [4] Download Tools John The Ripper, tersedia di: <http://www.bindshell.net /tools/johntheripper/John-1.7.3.1-all-mpi8.tar.gz>.
- [5] H. D. Mark, Amdahl's law in multicore era, University Wisconsin Madison, 2008.