

Penggunaan PXE (Preboot Execution Environment) untuk Mengontrol Pemilihan Operating System Saat Booting

Christian Adi Widjaja¹, Henry Novianus Palit², Agustinus Noertjahyana³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: renmuachci@gmail.com¹, hnpalit@petra.ac.id², agust@petra.ac.id³

ABSTRAK

Salah satu laboratorium Informatika di Universitas Kristen Petra yaitu laboratorium Multimedia. Di laboratorium ini terdapat banyak komputer dimana satu komputer telah ter-install beberapa sistem operasi. Banyak kegiatan belajar mengajar yang menggunakan sistem operasi Windows dan di luar jam belajar mengajar *user* biasanya menggunakan sistem operasi Ubuntu. Mengubah sistem operasi dari sistem operasi Windows 7 menjadi sistem operasi Ubuntu membutuhkan waktu yang cukup lama karena sistem operasi harus diganti secara manual.

Mengubah sistem operasi secara manual tidak masalah kalau jumlah komputer hanya 1 buah. Jika komputer berjumlah sekitar 20 buah, apalagi mengubah sistem operasi sekitar 2 – 4 kali dalam sehari, maka banyak sekali waktu yang diperlukan hanya untuk mengganti sistem operasinya saja. Hal ini dapat merepotkan asisten lab dan menghambat proses belajar mengajar.

Kata Kunci: Sistem Operasi, Windows, Ubuntu, PXE.

ABSTRACT

One of the Informatic laboratories in Petra Christian University is Multimedia laboratory. In this laboratory, every computer has multiple operating systems. Lot of class activities use Windows operating systems and outside class activities users usually use Ubuntu operating system. Changing the operating system from Windows 7 to Ubuntu takes a long time because the operating system must be changed manually.

Changing the operating system manually may not cause a problem if the change is just applied to a single computer. If the number of computers is 20, further more, the operating system is changed 2 – 4 times a day, then changing the operating system may take a lot of time. This will trouble the lab assistant too much and delay the class activities.

Keywords: Operating System, Windows, Ubuntu, PXE.

1. PENDAHULUAN

Salah satu laboratorium komputer di Teknik Informatika UK Petra adalah laboratorium Multimedia. Utilitas komputer-komputer di laboratorium tersebut kurang dari 25%. Untuk meningkatkan utilitasnya, suatu *platform cloud computing* berbasis *Open Stack* dijalankan di komputer-komputer tersebut. *Platform* tersebut dijalankan hanya saat laboratorium Multimedia tidak digunakan untuk kegiatan belajar mengajar. Sistem operasi yang digunakan untuk pelajaran dan praktikum adalah Windows 7, sedangkan sistem operasi yang digunakan untuk *cloud computing* adalah Linux Ubuntu.

Pergantian sistem operasi dari Windows 7 menjadi Linux Ubuntu perlu waktu cukup lama karena sistem operasi harus diganti manual seperti Gambar 1. Pergantian sistem operasi dengan cara tersebut tidaklah merepotkan jika hanya terdapat 1 unit komputer, namun jika terdapat 20 unit komputer dan pergantian dilakukan sekitar 2 – 4 kali dalam sehari, tentu banyak waktu yang terbuang sia-sia hanya untuk mengganti sistem operasi saja. Hal ini merepotkan asisten laboratorium dan menghambat proses belajar mengajar.



Gambar 1. Proses Pemilihan Sistem Operasi dengan GRUB

2. DASAR TEORI

2.1 Pengertian Jaringan

Jaringan Komputer [2] merupakan kumpulan dari perangkat keras dan lunak di dalam suatu sistem yang memiliki aturan tertentu untuk mengatur seluruh anggotanya dalam melakukan aktivitas komunikasi. Satu komputer yang terkoneksi ke jaringan menjadi satu node dari jaringan tersebut. Sedangkan host secara umum diartikan sebagai komputer yang terkoneksi ke jaringan yang dapat memberikan layanan jaringan (*network service*).

2.2 PXE

PXE (*Preboot Existing Environment*) adalah sistem *interface klien/server* yang digunakan untuk mengkonfigurasi komputer-komputer dalam satu jaringan. Komputer-komputer tersebut belum memiliki sistem operasi dan di *booting* secara *remote* oleh *administrator*. Kode PXE biasanya terletak di dalam *chip ROM* (*Read-only memory*) atau *boot disk* yang memungkinkan komputer *client* berkomunikasi dengan *server* jaringan sehingga komputer *client* tersebut dapat dikonfigurasi dan sistem operasinya juga dapat di-*booting* secara *remote* [1]. Komponen pendukung PXE :

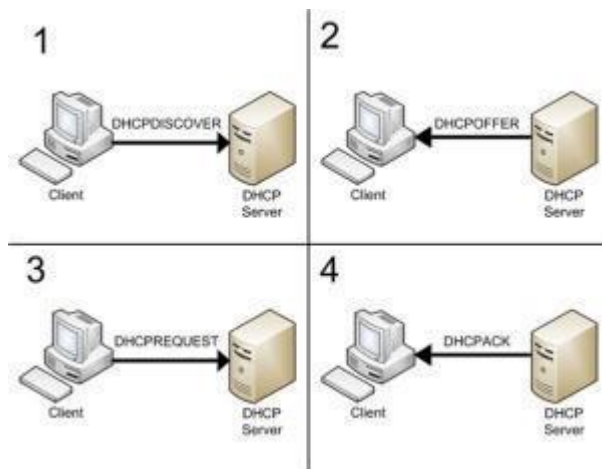
- DHCP (*Dynamic Host Configuration Protocol*), dimana klien menerima alamat IP untuk mendapatkan akses menuju ke server jaringan.

- TFTP sebagai tempat untuk menyimpan *image* sistem operasi dan menyimpan konfigurasi untuk *booting* melalui PXE.
- DNSMASQ sebagai program yang berperan sebagai DHCP Server dan TFTP Server.

2.3 DHCP

DHCP (*Dynamic Configuration Protocol*) [3] adalah layanan yang secara otomatis memberikan nomor *IP* kepada komputer yang memintanya. DHCP terdiri dari :

- DHCP *Client* yaitu komputer yang meminta nomor *IP*
 - DHCP *Server* yaitu komputer yang memberikan nomor *IP*.
- Dengan demikian *administrator* tidak perlu lagi harus memberikan *IP address* secara manual pada saat konfigurasi TCP/IP, tapi cukup dengan memberikan referensi kepada DHCP server. Cara kerja DHCP dapat dilihat pada Gambar 2.



Gambar 2. Cara Kerja DHCP

Berikut penjelasan untuk Gambar 2:

1. IP Lease Request
Client meminta *IP address* ke server (*Broadcast* mencari DHCP Server). (DHCP DISCOVER).
2. IP Lease Offer
DHCP Server (jumlah lebih dari satu) yang mempunyai *IP address*, memberikan penawaran ke *client* tersebut. (DHCPOFFER).
3. IP Lease Selection
Client menerima penawaran DHCP Server yang pertama dan kembali melakukan *broadcast* dengan *message* menyetujui peminjaman tersebut kepada DHCP Server. (DHCPREQUEST).
4. IP Lease Acknowledge
DHCP Server yang menang memberikan jawaban atas pesan tersebut berupa konfirmasi *IP address* dan informasi lain kepada Client dengan sebuah *ACKnowledgement*. Kemudian *client* melakukan inisialisasi dengan mengikat (*binding*) *IP address* tersebut dan *client* dapat bekerja pada jaringan tersebut. Sedangkan DHCP Server yang lain menarik tawarannya kembali. Jika komputer Client memiliki NIC (*Network Interface Card*) lebih dari satu dan perlu *IP address* lebih dari satu maka proses DHCP dijalankan untuk setiap adaptor secara sendiri-sendiri. (DHCPACK).

2.4 TFTP

TFTP (*Trivial File Transfer Protocol*) [4] merupakan sebuah protokol FTP yang disederhanakan untuk saling bertukar file antar komputer yang sama maupun berbeda jaringan. TFTP

menggunakan port 69. TFTP dirancang khusus dengan ukuran kecil karena mudah diimplementasikan. TFTP bertugas untuk membaca dan menulis *file* atau *mail* yang berasal dari komputer server maupun *file* yang menuju ke komputer server. Protokol ini membutuhkan jumlah memori yang sedikit untuk menjalankan kodenya, sehingga dapat dengan mudah dipasang pada *bootROM* komputer. Bagi administrator jaringan, TFTP [6] merupakan jalan utama untuk membackup *file* konfigurasi router dan switch. TFTP juga dapat digunakan untuk memberikan *file boot* PXE ke *thin-clients* atau sistem instalasi jaringan workstation.

2.5 DNSMASQ

DNSMASQ [5] berperan sebagai infrastruktur jaringan untuk jaringan skala kecil. DNSMASQ dapat berperan sebagai DNS, DHCP, *router advertisement* dan *network boot*. DNSMASQ didesain dengan ukuran ringan dan meninggalkan jejak *security* yang minimal. Cocok untuk *resource* yang memiliki kendala dengan batasan yang berasal dari router dan firewall. DNSMASQ berperan sebagai DHCP mendukung DHCPv4, DHCPv6, BOOTP dan PXE. DNSMASQ berperan sebagai sistem PXE yang memiliki fitur yang sama layaknya server PXE yang nyata. Fitur tersebut antara lain mendukung menu *netboot* dengan banyak arsitektur. Ada juga mode *proxy* dimana sistem PXE bekerja sama dengan server DHCP yang lain.

2.6 Komponen Remote Shutdown & Reboot Windows

Komponen *Remote Shutdown & Reboot* Windows merupakan komponen yang berfungsi untuk mengirimkan perintah *remote shutdown* dan perintah *remote reboot* ke komputer klien Windows. Komponen tersebut terdiri dari:

- SAMBA merupakan program untuk mengirimkan perintah *remote shutdown* dan *remote reboot* ke komputer klien Windows.
- Apache sebagai *localhost* untuk mengakses program *remote shutdown* dan *remote reboot* melalui web.
- GCC Compiler sebagai *compiler* untuk meng-*compile* program *remote shutdown* dan *remote reboot* sehingga program tersebut siap untuk dieksekusi.

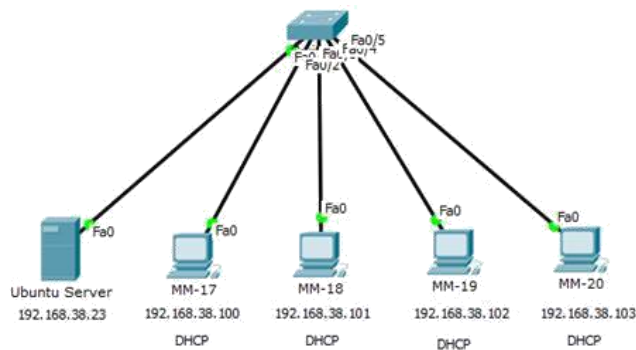
2.7 Teknologi Client-Server

Teknologi *Client-Server* menggambarkan hubungan antara dua program komputer atau lebih, dimana *client* akan meminta *request* sedangkan server akan membalas *request* tersebut. Proses ini berlangsung secara terus menerus. Teknologi *Client-Server* pada biasanya memiliki satu Server yang kadang-kadang disebut dengan “*daemon*”, yang berfungsi untuk bekerja di belakang layar dan menunggu *request* dari Client.

3. DESAIN SISTEM

3.1 Jaringan di Laboratorium Universitas Kristen Petra

Network yang tersebar melalui kabel LAN (*Local Area Network*) dengan koneksi 100Mbps Ethernet di Universitas Kristen Petra adalah network 192.168.32.0/21. Pada network ini terdapat DHCP (*Dynamic Host Configuration Protocol*) dan tersebar ke seluruh infrastruktur bangunan melalui sejumlah switch. Beberapa switch tersebut terhubung ke sejumlah laboratorium komputer, dan satu diantaranya terhubung dengan laboratorium komputer yang digunakan (Lab. Multi Media).



Gambar 3. Jaringan Laboratorium MM

Penjelasan pada Gambar 3 yaitu konfigurasi *IP address* untuk

- **Komputer Server**
Pengaturan pada komputer *server* dengan menggunakan *IP address* 192.168.38.23 dengan *subnet mask* 255.255.248.0, *gateway* 192.168.32.1, *network* 192.168.32.0, *broadcast address* 192.168.39.255, *DNS server* 203.189.120.4 dan 203.189.120.7.
- **Komputer Klien**
Pengaturan pada komputer klien menggunakan *IP DHCP* untuk setiap komputer dimulai dari 192.168.38.100 untuk komputer pertama, 192.168.38.101 untuk komputer kedua dan seterusnya secara berurutan hingga komputer terakhir mendapatkan *IP address* 192.168.38.103. Keseluruhan komputer mendapatkan pengaturan *subnet mask* 255.255.248.0, *gateway* 192.168.32.1, *DNS server* yaitu 203.189.120.4 dan 203.189.120.7 secara otomatis dari *DHCP server*.

3.2 Lingkungan Dasar

Lingkungan Dasar (*Basic Environment*) adalah pengaturan yang perlu disediakan sebelum melakukan instalasi *PXE*. Lingkungan dasar yang diperlukan adalah node-node dengan sistem operasi *Ubuntu Server 14.04 LTS*. Untuk setiap node spesifikasi *hardware* dapat dilihat pada Tabel 3.1.

Tabel 3.1. Node Spesifikasi *hardware*

Jenis Hardware	Keterangan
Processor	Intel Core i5-3340 @ 3.1 GHz (4 cores / 4 threads)
RAM	16 GB
Disk	250 GB
Koneksi	1 interface 100 Mbps Ethernet
Network Interface Card	Support PXE-Network-Bootting & Wake-On-LAN

Selain itu, dilakukan instalasi komponen-komponen dasar yang diperlukan oleh *PXE*. Komponen-komponen dasar tersebut dijelaskan pada subseksi-subseksi berikut ini.

3.2.1. DNSMASQ

Untuk menjalankan fungsi *DHCP* di komputer *server* sehingga komputer klien mendapatkan *IP address* sesuai *MAC address* masing-masing.

3.2.2. TFTP

Berperan sebagai tempat untuk menyimpan kernel dan *initrd* sistem operasi *Ubuntu* yang hendak *dibooting* secara *remote*. Juga menyimpan konfigurasi *PXE* untuk menentukan sistem operasi yang akan *dibooting* berdasarkan *MAC address*.

3.2.3. SYSLINUX

Bersifat wajib karena terdapat *file pxelinux.0* di dalamnya. *File pxelinux.0* merupakan *file boot* untuk *PXE*.

3.2.4. SAMBA

Merupakan program untuk mengirimkan perintah *remote shutdown* dan *remote reboot* ke komputer klien *Windows*.

3.2.5. SSH

Bersifat wajib karena berperan sebagai program untuk mengirimkan perintah *remote shutdown* dan *remote reboot* ke komputer klien *Ubuntu*.

3.2.6. Etherwake

Merupakan program untuk mengirimkan perintah *remote wakeonlan* ke komputer klien.

3.2.7. Apache

Merupakan program untuk menjalankan komputer *server* sebagai *localhost* sehingga program *remote* dapat diakses melalui *web*.

3.2.8. GCC + PHP

Memiliki fungsi untuk membuat program *remote shutdown*, *reboot* dan *wakeonlan* sehingga program tersebut siap untuk dieksekusi.

3.3 Sistem Otomasi

Sistem otomasi merupakan sistem yang dibutuhkan sehingga eksekusi program dapat dijalankan secara otomatis dan dari jarak jauh. Komponen untuk sistem otomasi terdiri dari:

3.3.1. Komponen Remote Shutdown Windows

Komponen *remote shutdown* *Windows* merupakan komponen yang diperlukan supaya proses *remote shutdown* di komputer klien sistem operasi *Windows* dapat berjalan secara lancar. Komponen tersebut meliputi instalasi program *SAMBA* di komputer *server*, pembuatan *software* di komputer *server* dan konfigurasi servis di komputer klien. Pembuatan *software* di komputer *server* seperti Tabel 3.2.

Tabel 3.2. Komponen Software Remote Shutdown Windows

Software Shutdown	Software Shutdown
Khusus PC MM-17	Khusus PC MM-18
mm17mati	mm18mati
mm17mati.c	mm18mati.c

Software *mm17mati* dan *mm18mati* merupakan program dari hasil *compile* *mm17mati.c* dan *mm18mati.c*. Program *mm17mati* dan *mm18mati* sudah siap dieksekusi. Konfigurasi servis di komputer klien dengan cara mengatur *Windows firewall* untuk mengijinkan program mengakses sistem krusial di *Windows*, pengaturan *file sharing* sehingga komputer *server* dapat menjalankan program di komputer klien tanpa *password*, konfigurasi *remote registry*

supaya selalu aktif, menonaktifkan UAC (*User Access Control*) jika ketiga langkah diatas belum berhasil membuat program *shutdown* berjalan lancar.

3.3.2. Komponen Remote Reboot Windows

Komponen *remote reboot* Windows adalah komponen yang dibutuhkan agar *reboot* Windows berjalan lancar. Cara implementasi program *reboot* di Windows mirip dengan konfigurasi *remote shutdown* Windows. Pembuatan *software remote reboot* Windows di komputer *server* seperti Tabel 3.3.

Tabel 3.3. Komponen Software Remote Reboot Windows

Software Reboot	Software Reboot
Khusus PC MM-17	Khusus PC MM-18
mm17reboot	mm18reboot
mm17reboot.c	mm18reboot.c

Software mm17reboot dan mm18reboot merupakan program dari hasil compile mm17reboot.c dan mm18reboot.c. Program mm17reboot dan mm18reboot sudah siap dieksekusi. Konfigurasi servis di komputer klien sama seperti Sub Bab 3.3.1. Komponen *Remote Shutdown* Windows.

3.3.3. Komponen Remote Shutdown Ubuntu

Komponen *remote shutdown* Ubuntu merupakan komponen yang diperlukan supaya proses *remote shutdown* di komputer klien sistem operasi Ubuntu dapat berjalan lancar. Komponen *Remote Shutdown* Ubuntu meliputi:

- Instalasi program di komputer *server*
Program yang diinstall yaitu Apache, GCC Compiler dan SSH.
- Pembuatan program *shutdown*
Program *shutdown* dibuat menggunakan *script* untuk mengeksekusi perintah *remote shutdown* Ubuntu melalui SSH.
- Konfigurasi servis di komputer *server*
Mengkonfigurasi servis di komputer *server* dengan cara menjalankan program *remote shutdown* Ubuntu sebagai *www-data*, melakukan koneksi SSH dengan komputer klien Ubuntu pertama kali sehingga komputer klien Ubuntu dikenali.
- Konfigurasi komputer klien Ubuntu
Mengkonfigurasi komputer klien Ubuntu dengan cara membuat *user* yang sama di semua komputer klien. Instalasi program SSH serta *permission* SSH supaya eksekusi program tidak memerlukan *password* saat melakukan koneksi.

3.3.4. Komponen Remote Reboot Ubuntu

Komponen *remote reboot* Ubuntu adalah komponen yang dibutuhkan agar *reboot* Ubuntu berjalan lancar. Komponen *Remote Reboot* Ubuntu meliputi:

- Instalasi program di komputer *server*
Program yang diinstall sama seperti sub bab 3.3.3.
- Pembuatan program *reboot*
Program *reboot* dibuat menggunakan *script* untuk mengeksekusi perintah *remote reboot* Ubuntu melalui SSH.
- Konfigurasi servis di komputer *server*
Mengkonfigurasi *servis* di komputer *server* dengan cara menjalankan program *remote reboot* Ubuntu sebagai *www-data*, melakukan koneksi SSH sama seperti sub bab 3.3.3.
- Konfigurasi komputer klien Ubuntu sama seperti sub bab 3.3.3.

3.3.5. Komponen Remote Wake-On-LAN

Komponen *remote Wake-On-LAN* merupakan komponen yang dibutuhkan agar komputer klien dapat dihidupkan secara *remote* sesuai keinginan. Komponen *Remote Wake-On-LAN* meliputi instalasi program di komputer *server*, pembuatan program di komputer *server*, konfigurasi program di komputer *klien*. Instalasi program di komputer *server* menggunakan *etherwake*. Pembuatan program di komputer *server* menggunakan program C yang diakses dengan PHP. Konfigurasi program di komputer klien meliputi *setting* BIOS supaya fitur *deep power off* selalu dalam posisi tidak aktif. Mengaktifkan fitur *Wake-On-LAN* pada *Network Interface Card Properties* melalui sistem operasi Windows.

3.4 Website Schedule

Website schedule merupakan *website* untuk menentukan jadwal *booting* komputer klien. *Booting* menuju sistem operasi Windows atau *booting* menuju sistem operasi Ubuntu. *Website* ini terhubung dengan *database pxeschedule*. Komponen *website schedule* terdiri dari *Index.php* sebagai halaman utama *website* tampilan program *schedule*. *Add.php* merupakan skrip php yang terhubung dengan form *add* dan berfungsi untuk menambahkan jadwal baru. *Edit.php* merupakan skrip php yang terhubung dengan fitur *edit* dan berfungsi untuk meng-*edit* jadwal yang sudah ditambahkan pada *database pxeschedule*. *Saveedit.php* merupakan fungsi untuk menyimpan hasil perubahan pada *database pxeschedule*. *Delete.php* merupakan skrip php yang terhubung dengan fitur *delete* dan berfungsi untuk men-*delete* jadwal pada *database pxeschedule*. *Checkschedule.php* merupakan skrip php yang berfungsi untuk memeriksa *database pxeschedule*. Jika tanggal dan waktu eksekusi sudah sama dengan tanggal dan waktu sekarang maka lakukan perintah eksekusi.

3.5 Website Realtime

Website Realtime merupakan *website* yang berfungsi melakukan perintah *power on*, *shutdown*, *reboot*, *change* Windows *os* dan *change* Ubuntu *os* pada 2 jenis komputer. Jenis komputer pertama adalah komputer individu sedangkan jenis komputer kedua adalah komputer kelompok. Eksekusi program secara satu per satu pada komputer individu sedangkan eksekusi program secara bersamaan pada komputer kelompok. Tampilan *website realtime* berisi gambar komputer berjumlah 23 buah dan letaknya dibuat mirip dengan denah ruang laboratorium MM di Universitas Kristen Petra.

3.6 Connector

Connector merupakan semua komponen sebagai penghubung antara *website schedule*, *website realtime* dan *website file explorer*. Komponen konektor yaitu koneksi.php sebagai penghubung antara *website schedule* dengan *database pxeschedule*. *Default.php* sebagai form *login* untuk dapat masuk ke halaman utama semua fitur. *Cek_login.php* sebagai skrip untuk memeriksa *username* dan *password* yang diberikan oleh *user* saat *input login* sudah sama dengan *database* atau belum sama. *Logout.php* sebagai fitur untuk *logout* atau keluar dari sesi. *Utama.html* sebagai *website* halaman *home* untuk berpindah dari satu *website* sistem menuju *website* sistem yang lain.

3.7 Website Explorer

Website Explorer merupakan *website* yang digunakan untuk mengedit, mendelete, mendownload isi file dari komputer *server*. Pada *website* ini komputer klien bisa melihat isi *file* komputer *server*. Dengan fitur *edit* pada *website explorer* komputer klien

juga bisa mengedit isi *file* yang lama dan menyimpan isi *file* yang baru. Untuk fitur *delete*, komputer klien bisa menghapus isi *file* yang tidak sesuai dengan konfigurasi awal. Sedangkan untuk fitur *download* komputer klien bisa mendownload *file* dengan syarat nama *file* yang asli otomatis diubah menjadi *filexx.php*.

4. IMPLEMENTASI SISTEM

4.1 Penempatan Peran Komputer di Laboratorium Komputer

Peran komputer di laboratorium MM sebagai berikut: 1 komputer *server* yang terhubung dengan 4 komputer klien. Dari 4 komputer klien ada 1 komputer klien yang *request* eksekusi program di komputer klien yang lain. Komputer MM-17 sebagai komputer yang meminta perintah, komputer MM-18, MM-19 dan MM-20 sebagai komputer target.

4.2 Network

Komputer *server* dan komputer klien harus berada dalam satu *network*. Komputer *server* memberikan *IP address* DHCP statis secara berurutan dimulai dari *IP address* 192.168.38.100 untuk komputer pertama hingga *IP address* 192.168.38.105 untuk komputer terakhir. Konfigurasi *IP address* pada komputer *server* dengan cara mengetikkan perintah seperti berikut

```
sudo nano /etc/network/interfaces
```

Pastikan isi konfigurasi */etc/network/ interfaces* sama dengan isi konfigurasi sebagai berikut:

```
auto eth0
interface inet static
address 192.168.38.23
netmask 255.255.248.0
gateway 192.168.32.1
network 192.168.32.0
broadcast 192.168.39.255
dns-nameservers 203.189.120.4 203.189.120.7
```

Jika sudah selesai mengkonfigurasi, klik tombol *save*.

4.3 DHCP

Langkah pertama untuk pengaturan DHCP adalah dengan mengatur konfigurasi DNSMASQ dengan cara instalasi DNSMASQ terlebih dahulu. Masukkan perintah seperti contoh dibawah.

```
sudo apt-get install dnsmasq
```

Jika sudah selesai instalasi DNSMASQ, ubah isi *file* DNSMASQ dengan cara *sudo nano /etc/dnsmasq.conf*. Setelah masuk pada konfigurasi DNSMASQ, ketikkan *dhcp-range=192.168.38.100, 192.68.38.105,12h*. Konfigurasi tersebut akan membuat komputer *server* meminjamkan *IP address* 192.168.38.100 – 192.168.38.105 ke komputer klien selama 12 jam. Settingan berikutnya ketikkan *dhcp-host=74:27:ea:5a:1d:04,192.168.38.100*. Konfigurasi *dhcp-host* akan membuat komputer klien dengan *MAC address* 74:27:ea:5a:1d:04 selalu mendapatkan *IP address* 192.168.38.100. Konfigurasi *dhcp-host=74:27:ea:5a:1e:4a,192.168.38.101* khusus untuk komputer klien MM-18. Untuk komputer MM-19, konfigurasi *dhcp-host=74:27:ea:55:5f:8a, 192.168.38.102* sedangkan konfigurasi *dhcp-host* untuk komputer klien MM-20 adalah *dhcp-host=74:27:ea:5a:21:75,192.168.38.103*.

5. PENGUJIAN DAN ANALISIS SISTEM

5.1 Windows Client

Pencatatan waktu pada Windows *client* menggunakan *stopwatch*. Pengujian dilakukan dengan cara membandingkan waktu *shutdown* PC MM-17. Waktu *shutdown* PC MM-17 *disshutdown* sendiri dibandingkan dengan waktu *shutdown* PC MM-17 digabung dengan *shutdown* komputer yang lain. Lebih tepatnya *shutdown* dua komputer secara bersamaan apakah waktu *shutdown* lebih cepat daripada *shutdown* satu komputer.

1. Shutdown MM-17 vs (MM-17&MM-18).

Waktu *shutdown* PC MM-17 *disshutdown* sendiri dibandingkan dengan waktu *shutdown* PC MM-17 digabung dengan *shutdown* PC MM-18. Untuk PC = MM-17. *Shutdown* dengan memanggil perintah *mm17mati* dicatat dengan menggunakan *stopwatch*. File *mm17mati* seperti kotak dibawah

```
#include <stdio.h>
int main()
{
system("net rpc shutdown -t 1 -I 192.168.38.100 -U Admin%fettuccinicarbonara");
return 0;
}
```

Untuk PC=MM-17 & MM-18, *shutdown* dengan memanggil perintah *mm1718mati*. Pencatatan waktu menggunakan *stopwatch*. File *mm1718mati* seperti kotak dibawah

```
#include <stdio.h>
int main()
{
system("net rpc shutdown -t 1 -I 192.168.38.100 -U Admin%fettuccinicarbonara");
system("net rpc shutdown -t 1 -I 192.168.38.101 -U Admin%fettuccinicarbonara");
return 0;
}
```

Hasil Perbandingan waktu MM-17 vs MM-17&MM-18 seperti Gambar 4.

PC MM-17 Berjalan Sendiri				
		I	II	III
MM-17	Waktu	1,98 detik	1,98 detik	1,49 detik
PC MM-17 & PC MM-18				
		I	II	III
MM-17	Waktu	1,74 detik	1,91 detik	1,85 detik
MM-18				
Hasil Perbandingan				
		I	II	III
MM-17	Waktu	0,24 s Bersama	0,07 s Bersama	0,36 Sendiri
	Average	0,155 s Bersama		

Gambar 4. Hasil Perbandingan Waktu Windows Shutdown

Berikut penjelasan Gambar 4:

• Percobaan I

Waktu *Shutdown* Windows *client* di PC MM-17 digabung dengan MM-18 lebih cepat 0,24 detik daripada waktu *shutdown* Windows di PC MM-17.

• Percobaan II

Waktu *Shutdown* Windows *client* di PC MM-17 digabung dengan MM-18 lebih cepat 0,07 detik daripada waktu *shutdown* Windows di PC MM-17.

- Percobaan III
Waktu *Shutdown* Windows *client* di PC MM-17 lebih cepat 0,36 detik daripada waktu *shutdown* Windows di PC MM-17 digabung dengan MM-18.
- *Average* merupakan rata-rata waktu *shutdown* pada dua hasil percobaan yang lebih cepat. Dalam hal ini $((0,24 \text{ detik} + 0,07 \text{ detik})/2) = 0,155 \text{ detik}$.
- Kesimpulan
Dari 3 kali percobaan waktu *Shutdown* Windows *client* di PC MM-17 digabung dengan MM-18 lebih cepat rata-rata 0,155 detik daripada PC MM-17 berdasarkan hasil percobaan I dan II. Sedangkan pada percobaan yang ketiga, waktu *shutdown* Windows *client* di PC MM-17 lebih cepat 0,36 detik daripada waktu *shutdown* Windows *client* di PC MM-17 digabung dengan MM-18.

5.2 Ubuntu Client

Pencatatan waktu pada Ubuntu *client* menggunakan perintah *time*. Pengujian dilakukan dengan cara membandingkan waktu *shutdown* via SSH yang lebih cepat pada Ubuntu *client*. Waktu *shutdown* Ubuntu *client* via SSH dijalankan dari skrip program C satu persatu dibandingkan dengan waktu *shutdown* Ubuntu *client* via SSH dijalankan dengan menggunakan fungsi *for*.

1. Perintah Program *Shutdown* di Ubuntu *Client*

Ubuntu *Shutdown* MM-17&MM-18 skrip program C vs MM-17&MM-18 dijalankan dengan fungsi *for*. Ubuntu *Client Shutdown* MM-17&MM-18 dijalankan lewat skrip program C. Pencatatan waktu pada PC MM-17&MM-18 dilakukan dengan perintah “time ./ubuntu1718mati”. File ubuntu1718mati berisi skrip program C seperti kotak dibawah

```
#include <stdio.h>
int main()
{
    system("ssh games@192.168.38.100 sudo shutdown -h now");
    system("ssh games@192.168.38.101 sudo shutdown -h now");
    return 0;
}
```

Ubuntu *Client Shutdown* MM-17&MM-18 dijalankan dengan fungsi *for*. Pencatatan waktu pada PC MM-17&MM-18 dilakukan dengan perintah “time for i in 100 101; do ssh games@192.168.38.\$i sudo shutdown -h now; done”. Hasil perbandingan Ubuntu *Client Shutdown* MM-17&MM-18 dijalankan lewat skrip program C vs Ubuntu *Client Shutdown* MM-17&MM-18 dijalankan dengan fungsi *for* dapat dilihat pada Gambar 5.

Berikut penjelasan berdasarkan hasil perbandingan di atas:

- Percobaan I
Waktu *Shutdown* Ubuntu *client* di PC MM-17&MM-18 dengan menggunakan fungsi *for* lebih cepat 0,33 detik daripada waktu *shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan skrip program C.
- Percobaan II
Waktu *Shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan menggunakan fungsi *for* lebih cepat 0,142 detik daripada waktu *shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan skrip program C.

UBUNTU SHUTDOWN SENDIRI				
PC		I	II	III
MM-17	Waktu	0m0.839s	0m0.834s	0m0.671s
MM-18				
UBUNTU SHUTDOWN BERSAMA				
PC		I	II	III
MM-17	Waktu	0m0.806s	0m0.692s	0m0.703s
MM-18				
HASIL PERBANDINGAN WAKTU				
PC		I	II	III
MM-17	Waktu	0m0.33s	0m0.142s	0m0.032s
MM-18	Average	0m0.0875s		

Gambar 5. Ubuntu Shutdown MM-17&MM-18

- Percobaan III
Waktu *Shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan skrip program C lebih cepat 0,032 detik daripada waktu *shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan menggunakan fungsi *for*.
- *Average* merupakan rata-rata waktu *shutdown* pada dua hasil percobaan yang lebih cepat. Dalam hal ini $((0,33 \text{ detik} + 0,142 \text{ detik})/2) = 0,0875 \text{ detik}$.
- Kesimpulan
Dari 3 kali percobaan waktu *Shutdown* Ubuntu *client* di PC MM-17&MM-18 dengan menggunakan fungsi *for* lebih cepat rata-rata 0,0875 detik daripada PC MM-17&MM-18 dengan menggunakan skrip program C berdasarkan hasil percobaan I, II. Sedangkan pada percobaan III, waktu *shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan menggunakan skrip program C lebih cepat 0,032 detik daripada waktu *shutdown* Ubuntu *client* di PC MM-17 & MM-18 dengan menggunakan fungsi *for*.

5.3 Power ON

Perintah Program *WakeOnLan* di semua komputer klien dijalankan lewat skrip program C vs perintah program *wakeonlan* PC MM-17&MM-18&MM-19&MM-20 dijalankan dengan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal. Program *WakeOnLan* PC MM-17&MM-18&MM-19&MM-20 dijalankan lewat skrip program C. Pencatatan waktu pada PC MM-17&MM-18&MM-19&MM-20 dilakukan dengan perintah “time ./semuapc”. File semuapc berisi skrip program C seperti kotak dibawah

```
#include <stdio.h>
int main()
{
    system("wakeonlan -i 192.168.39.255 -p 7 74:27:ea:5a:1d:04");
    system("wakeonlan -i 192.168.39.255 -p 7 74:27:ea:5a:1e:4a");
    system("wakeonlan -i 192.168.39.255 -p 7 74:27:ea:55:5f:8a");
    system("wakeonlan -i 192.168.39.255 -p 7 74:27:ea:5a:21:75");
    return 0;
}
```

Program *WakeOnLan* PC MM-17 & MM-18 & MM-19 & MM-20 dijalankan menggunakan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal. Pencatatan waktu pada PC MM-17 & MM-18 & MM-19 & MM-20 dilakukan dengan perintah

"time wakeonlan -i 192.168.39.255 -p 7 74:27:ea:5a:1d:04 74:27:ea:5a:1e:4a 74:27:ea:55:5f:8a 74:27:ea:5a:21:75".

Hasil Perbandingan perintah program *wakeonlan* PC MM-17 & MM-18 & MM-19 & MM-20 dijalankan lewat skrip program C vs perintah program *wakeonlan* PC MM-17 & MM-18 & MM-19 & MM-20 dijalankan dengan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal.

POWER ON SENDIRI				
PC		I	II	III
MM-17	Waktu	0m0.081s	0m0.081s	0m0.081s
MM-18				
MM-19				
MM-20				
POWER ON BERSAMA-SAMA				
PC		I	II	III
MM-17	Waktu	0m0.032s	0m0.031s	0m0.029s
MM-18				
MM-19				
MM-20				
HASIL PERBANDINGAN POWER ON				
PC		I	II	III
MM-17	Waktu	0m0.049s	0m0.050s	0m0.052s
MM-18	Average	0m0.0503s		
MM-19				
MM-20				

Gambar 6. Hasil Perbandingan Power ON semua PC.

Berikut penjelasan hasil perbandingan pada Gambar 6:

- Percobaan I
Waktu *wakeonlan* di PC MM-17 & MM-18 & MM-19 & MM-20 menggunakan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal lebih cepat 0,049 detik daripada waktu *wakeonlan* di PC MM-17&MM-18&MM-19 & MM-20 dengan skrip program C.
- Percobaan II
Waktu *wakeonlan* di PC MM-17&MM-18&MM-19&MM-20 menggunakan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal lebih cepat 0,050 detik daripada waktu *wakeonlan* di PC MM-17&MM-18&MM-19&MM-20 dengan skrip program C.
- Percobaan III
Waktu *wakeonlan* di PC MM-17&MM-18&MM-19&MM-20 menggunakan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal lebih cepat 0,052 detik daripada waktu *wakeonlan* di PC MM-17&MM-18&MM-19&MM-20 dengan skrip program C.
- Average merupakan rata-rata waktu *shutdown* pada tiga hasil percobaan. Dalam hal ini $((0,049\text{detik} + 0,050\text{ detik} + 0,052\text{ detik})/3) = 0,0503\text{ detik}$.
- Kesimpulan
Dari 3 kali percobaan waktu *wakeonlan* di PC MM-17 & MM-18 & MM-19 & MM-20 dengan menggunakan aturan penulisan program *etherwake* untuk *wakeonlan* secara massal lebih cepat rata-rata 0,0503 detik daripada waktu *wakeonlan* di

PC MM-17 & MM-18 & MM-19 & MM-20 dengan menggunakan skrip program C.

6. KESIMPULAN

6.1 Kesimpulan

Setelah mengkonfigurasi komputer *server* yang menggunakan PXE (*Preboot Execution Environment*) untuk mengontrol pemilihan *Operating System* waktu proses *booting* di komputer klien ini dapat diambil beberapa kesimpulan yaitu:

- Sistem yang dibangun mampu memilih sistem operasi secara otomatis sesuai pengaturan per komputer. Layanan-layanan yang dibutuhkan untuk mendukung sistem ini antara lain DHCP, TFTP, PXE, SSH dan Etherwake.
- Aplikasi *web* yang dibangun memudahkan *administrator* (asisten lab) agar dapat menjadwalkan kapan suatu komputer harus di-*reboot* dan sistem operasi apa yang akan dijalankan.
- Tidak terdapat perbedaan waktu yang signifikan antara melakukan perintah *Shutdown* sistem operasi Ubuntu menggunakan program secara satu per satu dibandingkan dengan menggunakan skrip secara bersamaan.
- Tidak terdapat perbedaan waktu yang signifikan antara melakukan perintah *Wake-On-LAN* komputer klien menggunakan program secara satu per satu dibandingkan dengan menggunakan skrip secara bersamaan.
- Tidak terdapat perbedaan waktu yang signifikan antara melakukan perintah *Shutdown* sistem operasi Windows menggunakan program secara satu per satu dibandingkan dengan menggunakan program secara bersamaan.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan program lebih jauh adalah sebagai berikut :

- Setting password* untuk *database* sebaiknya dienkripsi.
- Untuk pengembangan ke depan khusus penjadwalan disarankan menggunakan *crontab*.
- Pengembangan sistem untuk ke depan sebaiknya ditambahkan fitur *user* dengan hak akses yang berbeda-beda.

7. REFERENSI

- [1] Doli, O. 2012. *Trivial File Transfer Protocol (TFTP)*. URI=<http://contohdanfungsi.blogspot.co.id/2012/08/trivial-file-transfer-protocol-tftp.html>.
- [2] Kelley, S. 2014. *Dnsmasq*. URI=<http://www.thekelleys.org.uk/dnsmasq/doc.html>.
- [3] Leskiw, A. 2012. *Four Free TFTP Servers for Windows*. URI=<http://www.networkmanagementsoftware.com/free-tftp-servers/>.
- [4] Prihanto, A. 2013. DHCP. URI=<https://cogierb201.wordpress.com/2013/11/24/dhcp/>
- [5] Rouse, M. 2005. *Preboot Execution Environment (PXE)*. URI=<http://searchnetworking.techtarget.com/definition/Preboot-Execution-Environment>.
- [6] Tanenbaum. 2002. *12 Pengertian Jaringan Komputer Menurut Para Ahli*. URI=<http://dosenit.com/jaringan-komputer/pengertian-jaringan-komputer-menurut-para-ahli>.