

WINDOWS - LINUX TERMINAL SERVER PROJECT

Rico Agung Firmansyah
STMIK AMIKOM Yogyakarta

Abstraksi

Pembangunan jaringan komputer internet, intranet maupun ekstranet memang membutuhkan resources yang besar, apalagi jika ada aplikasi server dan router-nya. Ada beberapa developer jaringan yang mengabaikan kalkulasi biaya demi mendapatkan performa yang baik. Ada juga yang mengabaikan performa karena pengaruh biaya yang minim. Dua kasus diatas sebenarnya bisa dijembatani dengan membuat sistem terminal server atau pc cloning, sehingga kita tetap bisa mengedepankan performa yang baik dengan biaya yang tidak besar. Namun, dengan terminal server, masalah lain muncul lagi, yaitu tidak semua user bisa familier dengan sistem ini (Linux), dimana kebanyakan user terbiasa dengan windows. Oleh karena itu, solusi cerdasnya adalah pengembangan aplikasi Terminal server yang berbasis Linux dan Windows sekaligus.

Kata Kunci: *Jaringan komputer, PC Cloning, Linux Terminal Server, Windows Terminal Server.*

1. Pendahuluan.

Sesuai dengan abstraksi diatas, pengembangan jaringan komputer dengan banyak *client/user* memang akan membutuhkan banyak *resources* mulai dari *hardware, software, brainware*, dan lainnya. Hal ini otomatis akan membutuhkan biaya yang cukup besar. Jika kita tekan biaya seminim mungkin, performa sistem akan menurun, Tapi jika kita kedepankan performa, maka biaya yang akan membengkak. Memang hal ini sangat dilematis dan perlu dicari solusinya tanpa mengorbankan salah satunya.

Akhirnya para ahli IT bersama-sama membangun sebuah sistem tersentral yang diberi nama terminal *server* atau cloning sebagai jawaban masalah tersebut. Ada bermacam-macam versi aplikasi terminal *server* yang dibangun sesuai dengan kebutuhan masing-masing. Misalnya *Novel Netware*, *Linux Terminal Server Program* (LTSP), dan banyak lagi yang lainnya. Kebanyakan diantaranya berbasis Unix. Tetapi, dengan solusi tersebut, muncul kembali masalah yang sangat mendasar, yaitu tidak semua user bisa familier dengan sistem ini (Linux), dimana kebanyakan *user* terbiasa dengan windows. Oleh karena itu, solusi cerdasnya adalah pengembangan aplikasi *Terminal server* yang berbasis Linux dan Windows sekaligus. Maka dari itu para ahli mendevlop sistem tersebut agar bisa menampung beberapa basis sistem operasi sekaligus. Penulis kali ini hanya akan menjelaskan *interoperability* antara dua buah sistem operasi, yaitu Linux dan Windows dengan menggunakan aplikasi *Linux Terminal Server*.

Aplikasi terminal *server* identik dengan pengurangan *resources* pada *client* dengan anggapan seluruh/sebagian aktifitas *client* akan diberi oleh *Server*nya. Alasan inilah yang digunakan sebagai pengurangan *resources* pada sisi *client*, yakni berupa penghematan *hardware* dan *software* (proses). Penghematan *hardware*nya diantaranya tidak diperlukannya lagi hardisk, cd-rom, atau dvd-rom sebagai media storage di sisi *client*, hanya cukup membutuhkan sebuah *floppy disk* sebagai *sistem loader* PC *client* ke *server*nya.

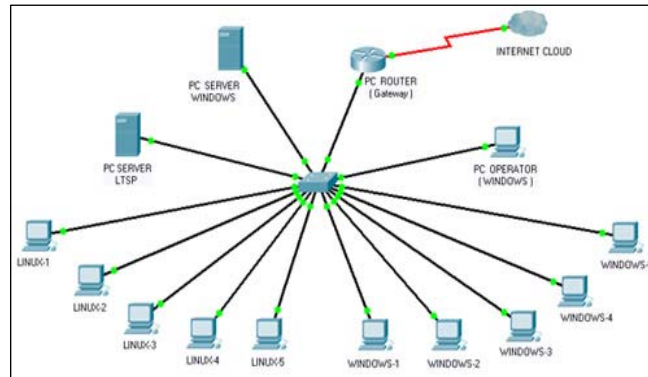
2. Pembahasan

2.1 Sistem Requirement

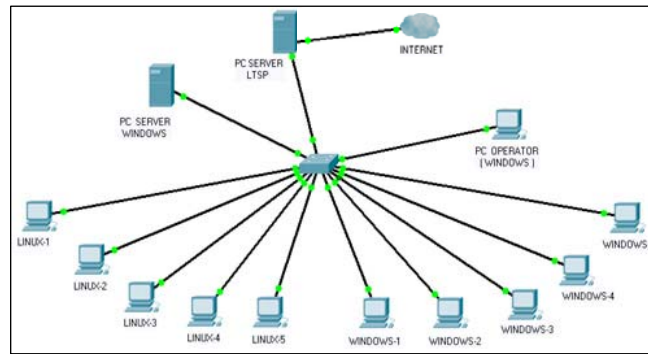
Untuk membangun sebuah sistem, kita perlu mengkalkulasi *resources* yang akan dibutuhkan (*sistem requirement*). Kali ini kita asumsikan sebuah warnet yang akan dibangun dengan aplikasi *Windows Linux Terminal Server Project* (WLTSP). Implementasi pada contoh kasus yang berbeda juga akan serupa dengan penjelasan pada contoh kasus kali ini.

Misalkan kita akan membangun sebuah warnet WLTSP dengan jumlah *client* 10 PC, maka PC yang diperlukan sebanyak 14 buah. 10

PC untuk *client* (5 PC Linux dan 5 PC Windows), 1 PC untuk operator, 1 PC untuk router, serta 2 buah PC untuk *server* aplikasi WLTS (gambar 1). Jika anggarannya masih tidak memungkinkan juga untuk mengadakan 14 PC, maka PC Linux *Server* bisa difungsikan ganda, sekaligus sebagai *Router* jika perbandingan jumlah *client* dan kemampuan *server* masih dalam batas-batas normal (gambar 2). Skenarionya bisa dilihat pada gambar dibawah ini.



Gambar 1. Skenario Kebutuhan standar Sistem WLTS.



Gambar 2. Skenario Kebutuhan minimalis (alternatif) Sistem WLTS.

Dari dua buah gambar diatas, terlihat sebuah LAN yang terhubung ke internet melalui Router (*Gateway*). LTSP *Server* bersistem operasi Linux, berfungsi sebagai *terminal server*. windows *server* nantinya akan berfungsi sebagai pemberi layanan untuk *client*nya yang akan berbasis windows. Penulis tidak akan membahas pembuatan PC *Router*, namun hanya akan membahas aplikasi Terminal *Server*nya saja. Penulis mengasumsikan *Gateway* sudah dapat difungsikan, baik dengan skenario pada gambar 1 maupun gambar 2.

Dari penjelasan singkat, *requirement* untuk masing-masing mesin berbeda-beda karena disesuaikan dengan kebutuhan, fungsionalitas serta seberapa berat kerja (proses) yang akan dikerjakan. Berikut ini *sistem requirement* minimal untuk masing-masing mesinnya.

- a. PC *Server* LTSP : P-4 1.2 Ghz *Processor*, 512 MB *Internal Memory*, 8 GB *Hard drive*, 42X CD-Rom, 1 *Ethernet Card*, Linux OS (RedHat, Mandrake, Mandriva, Suse, Ubuntu, dsb).
- b. PC *Server* Windows : P-4 1.2 Ghz *Processor*, 128 MB *Internal Memory*, 5 GB *Hard drive*, 42X CD-Rom, 1 *Ethernet Card*, Win2000 *Server* OS / Win2003 *Server* / Win XP Professional.
- c. PC Router/*Gateway* : P-2 300 MHz *Processor*, 64 MB *Internal Memory*, 4 GB *Hard drive*, 2 *Ethernet Card*, Linux OS, Squid (*Routing dan Proxy Server*).
- d. PC Operator : PC operator sama seperti PC *Client* jika dikonfigurasi sebagai *Client* dari *WLTSP*. Namun jika ingin dikonfigurasi independen, spek PC bisa disesuaikan dengan keinginan, kenyamanan kerja si operator atau sesuai *budget*. Sistem

operasi yang akan digunakan pada pc ini tergantung user, bisa linux atau windows.

- e. *PC Client* : P-2 300 MHz Processor, 32 MB *Internal Memory*, 1 *Ethernet Card*, 1 *Floppy Drive*.
- f. 1 *buah* Hub atau Switch 24 port. Penulis menyarankan untuk menggunakan switch untuk performa jaringan yang lebih baik.

Selain *requirement* diatas, kita juga masih harus mempersiapkan beberapa hal, diantaranya:

- a. Siapkan File *Installer* untuk aplikasi LTSP (<http://www.ltsp.org>). File yang dibutuhkan antara lain *ltsp-4.1-0.iso* (104MB), *ltsp-utils-0.1.0.tgz* (40KB). Kemudian *Burn* file tersebut ke CD untuk instalasi.
- b. Siapkan file *Boot ROM Loader* (<http://rom-o-matic.net>). File ini digunakan untuk sistem loader dari *client* ke *server* melalui NIC yang terkoneksi ke jaringan. File *Boot Rom* ini harus dikonfigurasi berdasarkan parameter pada NIC *client* sebelum di *download*. Kemudian *burn* file tersebut ke *Boot sector* pada *floppy disk*.
- c. Siapkan file *Virtual Network Connection (VNC* : <http://www.realvnc.com>) untuk mentransfer GUI dari *server* ke *clientnya*. Sebenarnya di Linux juga ada fasilitas serupa, yaitu *remote desktop*. Jika anda tidak mendownload *realvnc*, maka cukup gunakan *remote desktop*.
- d. Siapkan Aplikasi *DHCP server*, *TFTP server*, *NFS server*, dan *XDMCP server*, untuk servis di *LTSP server*. Biasanya file tersebut sudah terdapat dalam paket distro Linux versi *enterprise* atau *advance server*. Penulis menggunakan *Red Hat* versi *Enterprise Advance Server*.
- e. Siapkan file *Windows Linux Terminal Server Client* (www.linuxtsc.org) yang akan digunakan *Windows Server* untuk mentransfer GUI-nya ke *Linux Server* kemudian ke *client* yang meminta layanan GUI *Windows*.

2.2 Cara Kerja Sistem WLTP

Setelah kita lihat gambar dan *requirement* tiap mesin, maka terlihat bahwa kedua *server* yang nantinya memberikan layanan kepada *client*nya mulai dari sistem operasi, GUI, servis serta fasilitas-fasilitas yang akan dijalankan di *client*. *Server* Linux akan memberikan layanan berupa GUI ke *client* yang akan berbasis Linux, serta menjembatani *client* yang akan berbasis windows untuk meminta layanan ke *server* windowsnya.

Jika kedua *server* sudah siap bekerja, saat ada *client* yang menghidupkan komputernya kemudian *booting* dengan disket yang berisi *Boot ROM*, maka setelah inisialisasi *hardware* di *client* selesai, maka NIC *client* akan menghubungi *server* LTSP (*Linux Server*). *Linux Server* akan merespon dengan memberi konfigurasi NIC *Client* dengan servis DHCPnya. Kemudian *LTSP Server* akan mentransfer Sistem Operasi yang di-*share* (*NFS daemon*) melalui servis TFTP. Setelah sistem operasi di *client* siap, selanjutnya *server* mentransfer GUI-nya menggunakan servis XDMCP. Dengan demikian *client* seperti memiliki sistem operasi sendiri (linux) padahal sistem dan GUI-nya adalah sistem dan GUI-nya *LTSP Server*.

Serupa dengan cara kerja *client* Linux, *client* Windows juga me-load *Boot ROM*-nya ke *LTSP Server* (*Linux Server*), lalu direspon dengan memberikan layanan DHCP ke NIC *client*. Namun kemudian *LTSP Server* tidak akan mentransfer sistem operasinya ke *client*, tapi *LTSP Server* akan mengakses file sistem dari *Windows Server* yang sudah di *share*, kemudian ditransfer ke *client*nya. Dengan demikian *client* akan me-load sistem operasi windows ditambah beberapa file konfigurasi LTSPnya.

Setelah sistem operasi windows di *load*, maka servis XDMCP di *Linux Server* akan mentransfer GUI dari *Windows Server* ke *Client*nya dengan *remote desktop* atau dengan VNC, sehingga di sisi *client* akan tampak GUI Windows. Dengan demikian, *LTSP Server* bekerja ganda, bahkan *triple*, yaitu bekerja untuk dirinya sendiri, bekerja untuk

melayani *client* yang ber-GUI Linux, serta sebagai jembatan antara *windows server* dengan *client* yang ber-GUI Windows. Oleh karena itu, LTSP *Server* haruslah dikonfigurasi dengan sebaik mungkin dan dijaga performanya agar tetap bisa melayani banyaknya proses dan servis terhadapnya.

2.3 Instalasi dan Konfigurasi

Jika *software* yang akan dibutuhkan telah siap, cara kerja sistem juga sudah diketahui, maka langkah selanjutnya adalah tahap instalasi dan konfigurasi sistem. Dalam artikel ini hanya akan dibahas instalasi sistem WLTPS-nya saja. Penulis mengasumsikan pada *Linux server* dan *Windows Server* sudah terinstal *software requirement* seperti dijelaskan pada bab sebelumnya secara *default* dengan penambahan paket (install-lah) *DHCP Server*, *TFTP Server*, *NFS Server*, dan *XDMCP Server*. Jika anda tidak mendapatkan paket tersebut di distro anda, anda bisa *download* dari situs yang sesuai dengan distronya. Berikutnya konfigurasi ip address dan netmask pada NIC *Linux Server*nya, misalnya “ **# ifconfig eth0 192.168.0.1 netmask 255.255.255.240** “. Konfigurasi *netmask* dan *IP Address* bisa dirubah sesuai dengan kebutuhan.

Begitu juga dengan *windows server*, instalasi sistem operasinya sesuai dengan prosedur/*default* saja, kemudian konfigurasi ip address dan netmask-nya, misalkan ip address 192.168.0.2, netmask 255.255.255.240. lalu instal aplikasi didalam *windows server* anda, misalnya aplikasi perkantoran, *media player*, *photo editor*, *browser*, dan sebagainya yang akan digunakan. Jika prosedur diatas telah dilalui, langkah berikutnya adalah sebagai berikut, kita mulai dari *windows server*, kemudian *linux server*.

1. Instal aplikasi yang ingin digunakan di *windows server*, lalu konfigur *ip address* dan netmasknya, misalnya 192.168.0.2 255.255.255.240. Berikutnya instal *windows linux terminal server client (Linux TSC)*. Aplikasi ini akan bekerja sama dengan *remote desktop* dan protokol XDMCP untuk mentransfer GUI di *windows server* ke *client* yang menginginkan layanan Windows (Windows OS).

2. Instal aplikasi LTSP pada *Linux Server*.

✦ Sebelum instalasi LTSP, servis DHCP, TFTP, NFS, dan Display Manager harus sudah aktif. Caranya ketikkan perintah :

```
# service dhcpd restart    —>    (mengaktifkan servis DHCP)
# service nfsd restart     —>    (mengaktifkan servis NFS)
# service tftpd restart    —>    (mengaktifkan servis TFTP)
# service dm restart      —>    (mengaktifkan servis display manager)
```

✓ Servis DHCP, digunakan untuk mengkonfigurasi NIC *client* dari *server*. Konfigurasilah isi dari file **/etc/dhcpd.conf** sesuai dengan konfigurasi jaringan anda (*ip server/ip address* sendiri, *netmask*, *ip router* ke internet, DNS, dan *range ip clientnya*). Sebagai tambahan, baris yang bertuliskan “filename” pada file **/etc/dhcpd.conf**, harus diisi nama filenya.

Misalkan : filename “/lts/vmlinuz-2.4.26-ltsp-2” ;

✓ Servis NFS (*network file sistem*), digunakan untuk mengatur file sistem dalam jaringan. Konfigurasilah parameter-parameter pada isi file berikut :

- /etc/host dan /etc/host.allow (berisi konfigurasi *host* mana saja yang boleh mengakses dan bagaimana cara atau aturan-aturan *host* untuk konek ke *servernya*).
- /etc/export (berisi konfigurasi file/*resources* apa saja yang akan di-*share* ke jaringan).

✓ Servis TFTP (*Trivial File Transfer Protocol*), digunakan untuk mentransfer file yang berukuran besar serta *file* yang sedang digunakan/*file sistem* (OS) dari *server* ke *client*. Tambahkan beberapa parameter dalam file **/etc/xinetd.d/tftp**, yaitu edit parameter “disable” dengan “no”, serta parameter “ -s /tftpboot “ pada poin **server_args**.

- ✓ Servis *Display Manager* digunakan untuk memberikan tampilan *server* ke *clientnya*, meskipun servis ini nanti harus berjalan dengan servis XDMCP nantinya. Anda tidak mengkonfigurasi servis ini, karena servis display manager hanya dikonfigurasi secara default saja.
- ✦ Aktifkan dan konfigurasi servis XDMCP di PC Linux *Server* dengan cara:
 - ✓ Ubah *Run Level* untuk meng-*enable*-kan *Display Manager* kdm/gdm dengan perintah : `# vi /etc/inittab` , kemudian ubah *run level*-nya menjadi : “ **id:5:initdefault:** “.
 - ✓ Aktifkan xdmcp-nya dengan mengedit *file*: `/etc/X11/gdm/gdm.conf` (jika anda menggunakan gdm) atau *file*: `/etc/X11/xdm/xdm-config` (jika anda menggunakan kdm). Kemudian masuk ke [xdmcp], lalu set “Enable” dengan “1”. Jika anda menggunakan gdm, maka cari baris “DisplayManager.requestPort:0”. Kemudian Ubah baris tersebut menjadi: “**!DisplayManager.requestPort:0**”.
 - ✓ Edit *file* Xaccess supaya bisa dibuka oleh semua *work station*. Caranya editlah *file* `/etc/X11/xdm/Xaccess` dengan *editor* pico atau vi. Kemudian cari baris “ `## Any host can get a login window` ”, lalu hilangkan satu buah tanda # di awal baris tersebut.
 - ✓ Edit *file* `/etc/X11/xdm/kdmrc` atau `/usr/share/config/kdm/kdmrc` atau `/etc/opt/kde2/share/config/kdm/kdmrc`. Kemudian ubah nilai agar “Enable” menjadi “True”. Setelah itu *restart* komputer anda untuk mendapatkan efek dari konfigurasi yang telah dilakukan.
- ✦ Instal ltsp-utils dengan cara:


```
# cp ltsp-utils.0.1.0.tgz/usr/local/src
# tar zxvf ltsp-utils.0.1.0.tgz
# cd ltsp-utils
# ./install.sh/
```

- ✓ Konfigurasi agar instalasi diatas mengambil *source* ke cd-rom, caranya masukkan CD LTSP yang telah *download*, kemudian ketikkan `:# ltspadmin`
 - > configure the installer options
 - > where to retrieve packages from? [ketik <file:///mnt/cdrom>]
 - > use http proxy ? [Enter]
 - > use ftp proxy ? [Enter]
 - > Correct ? [Y]
 - # ltspadmin
 - > install / update LTSP Packages
 - > A
 - > Q

Instalasi akan berjalan otomatis setelah kita menekan tombol Q.

3. Konfigur *LTSP Server* (`/opt/ltsp/i386/etc/lts.conf`), dimana parameter didalam file ini akan mengatur langsung *rule* atau aturan-aturan pada *client*nya. Anda cukup sesuaikan dengan konfigurasi pada *client* nantinya, dengan isi parameternya dalam file tersebut. Misalkan *ip server* (isi sesuai *ip address server*), *XServer* (*screen resolution* pada *client* misalnya diisi 1024x768), *X_Mouse_Protocol* (= "PS/2" atau "Com1"), *Screen_01* (*tty1* atau *session1* akan diisi tampilan apa, misalkan = *startx*), serta parameter device lainnya yang akan difungsikan (*printer, keyboard, touch screen*, dan lain sebagainya). Penulis tidak memberi konfigurasi lengkapnya karena konfigurasi ini sangat bergantung pada kondisi sesungguhnya.
4. Konfigur *Screen scripts* (`/opt/ltsp/i386/etc/screen.d`) untuk layanan *session* terminalnya. Linux memberikan fasilitas *multiple screen session* atau beberapa sesi tampilan yang bisa dilihat dan dikonfigurasi berbeda-beda dalam satu proses. Misalkan pada *session 1* tampilan XWindow (*startx*) yang berfungsi, di *session 2* *shell* (*commend line interface*), pada *session 3* untuk *remote desktop* ke mesin lain, atau yang lain sebagainya.

Contohnya “SCREEN_01 = rdesktop -f w2k.mydomain.com, SCREEN_02 = telnet server2.mydomain.com, SCREEN_03 = startx”. Pada contoh diatas, *session* pertama pada *client* akan mengakses *remote desktop* dengan target windows 2000 dalam domain mydomain.com. akibatnya *client* akan menampilkan GUI Windows 2000 pada *session* pertama. Sedangkan pada *session* kedua akan mengakses telnet ke *LTSP server* dengan tampilan CLI. Serta pada *session* ketiga akan mengakses mode xwindow pada *LTSP server*,

Untuk *client* yang akan difungsikan dengan GUI Windows, maka cukup panggil *session* yang parameternya berisi *rdesktop* (*session* pertama atau tekan ALT+F1), dengan catatan *remote desktop* ke target *Windows Server* dan *Linux TSC*-nya sudah aktif. Begitu juga dengan *client* yang akan difungsikan dengan GUI Linux, maka cukup buka *session* baru dengan mengetikkan Alt + F2, maka tampilan langsung akan berpindah ke *session telnet*, begitu juga dengan *session* berikutnya. Bahkan anda bisa memanfaatkan dua *session* atau lebih dalam satu waktu.

Dengan *multiple session* yang dikonfigurasi sistem operasi yang berbeda-beda tiap *session*nya, memungkinkan untuk berpindah sistem operasi dengan sangat cepat dan mudah. Bahkan hal ini mustahil dilakukan dengan PC *Stand-alone* atau *workstation* non LTSP. Jika anda tidak menggunakan *remote desktop* atau anda menggunakan aplikasi Real VNC atau yang lainnya, konfigurasinya sama seperti penjelasan diatas.

3. Penutup

Dari penjelasan diatas, maka dapat disimpulkan bahwa sistem Windows Linux Terminal *Server* ini memiliki keuntungan dan kerugian sebagai berikut.

1. Keuntungannya antara lain :

- Mudah dalam pengoperasian, memonitor, mengontrol serta memenejemen aktifitas *client*. Hal ini dikarenakan keseragaman konfigurasi sistem operasi yang dijalankan dari

servernya (sistem yang tersentral), serta tidak perlu banyak konfigurasi di *client*-nya.

- Mengurangi penggunaan sumberdaya secara ilegal oleh orang yang tidak bertanggungjawab dengan adanya servis DHCP dan NFS.
 - Biaya pembangunan sistem dapat ditekan dengan adanya sistem *diskless* serta konsumsi *resources* yang minimalis yang digunakan di sisi *client*nya.
 - Dengan adanya *multi session*, tentunya memberikan kenyamanan lebih terhadap *user* yang ingin menggunakan dua sistem operasi sekaligus atau secara bergantian.
 - *Multi session* juga bermanfaat dalam hal fleksibilitas terhadap penggunaan sistem operasi yang bisa saling bergantian. Misalkan ketika pada suatu waktu terdapat banyak *client* yang minta layanan windows semua, maka seluruh *work station* akan bisa cepat diseting ke mode windows. Demikian juga sebaliknya, jika suatu saat banyak pengunjung yang minta layanan Linux, maka dengan segera bisa menggunakan Linux. Hal ini sangat menguntungkan bagi pemilik warnet/kantor, serta bagi user.
2. Kerugiannya antara lain :
- Dengan sistem tersentral, maka jika kondisi *Server* sedang labil, maka seluruh workstationnya akan merasakan hal yang serupa. Misal *hang*, terserang virus, *bottle neck* terhadap banyaknya proses dari *client*nya, *interrupt* dari *client*, serangan DoS, dan sebagainya.
 - Dengan fungsi kerja *Linux Server* yang berlipat ganda, rentan sekali terjadi tabrakan data, *bottle neck*, *hang*, *sistem crash*, *memory overload*, dan sebagainya sehingga mengakibatkan performa sistem menjadi lebih rendah dibandingkan dengan performa *PC Stand alone*.
 - Aktivitas *client* dengan *servernya* banyak memakan *resources* untuk aplikasi LTSP dan transfer GUI-nya, secara otomatis *bandwidth* ke *network* luar (*internet/upstream*-nya) menjadi

berkurang atau terasa lambat. Sangat tidak cocok untuk jaringan yang memiliki banyak *client*.

3. Saran penulis :

- Jangan implementasikan sistem ini dengan jumlah *client* berlebih. Sesuaikan kemampuan kedua *server* dengan jumlah *client*nya, terutama *Linux server*nya. Lakukan kalkulasi, analisa serta pengetesan sebelum mengimplementasikan sistem ini. Performa sangat bergantung pada *resources* yang anda pakai di sistem ini.
- Gunakan spek yang baik pada *server*nya, untuk mendapatkan performa baik, diatas rata-rata atau bahkan yang ekstrem.
- Selalu kontrol kondisi kedua *server* (*hardware* dan *softwarena*) karena baik/tidaknya performa sistem ini bergantung pada *server*.
- Pada konfigurasi DHCP-nya, berikan *range ip* yang sesuai kebutuhan, jangan samapai berlebih agar tidak banyak memakan *resources*.
- Untuk keamanan yang lebih baik lagi, tambahkan aplikasi/konfigurasi *proxy*, *firewall* atau autentikasi pada *Linux server*.

Daftar Pustaka

Basic LTSP How to URL : -

<http://www.ltsp.org/documentations/index.php>

- <http://sourceforge.net>

- <http://benpinter.net>

Windows Linux Terminal *Server* Project How to URL :

www.linuxtsc.org

Windows Cloning URL :

<http://www.microsoft.com/technet/arcive/termsrv/plan/tscap.msp>

<http://onno.vlsm.org/>