BAB III

PEMBAHASAN

Bab ini akan menguraikan proses analis dan perancangan dengan langkah-langkah sebagai berikut :

1. Analisis sistem *Linux Terminal Server Project* (LTSP)
2. Perancangan sistem *Linux Terminal Server Project* (LTSP)
3. Analisis Pengguna
4. Analisis Perangkat Keras
5. Analisis Jaringan Komputer
6. Analisis Perangkat Lunak
7. implementasi dan pengujian terhadap pembangunan Linux *Terminal Server Project* (LTSP) . pengujian pada kinerja dari Linux *Terminal Server Project* (LTSP).
8. pengujian pada kinerja dari Linux *Terminal Server Project* (LTSP).

3.1 Analisis Sistem *LINUX SERVER TERMINAL PROJECT* (LTSP)

Dalam analisis sistem dilakukan penguraian dari sistem yang telah ada sehingga dapat di identifikasi dan diketahui kebutuhan untuk pengembangannya.

3.1.1 *Linux Terminal Server Project* (LTSP)

Linux Terminal Server Project (LTSP) adalah sebuah *add-on package* yang bebas dan bersifat *open source* untuk Linux yang memungkinkan banyak orang

untuk secara simultan menggunakan satu komputer dengan konsep terminal. Aplikasi dijalankan pada server dengan sebuah terminal yang dikenal sebagai sebuah *thin client* atau juga dikenal sebagai X terminal menangani *input* dan *output*. Secara umum, terminal hanya memerlukan sumber daya yang rendah, dan menuntut ruang yang lebih sedikit untuk tempat fisik.

Teknologi *Linux Terminal Server Project* (LTSP) menjadi populer karena dapat menyediakan akses komputer dengan memanfaatkan komputer yang memiliki spesifikasi rendah. Komputer-komputer berspesifikasi rendah tersebut dapat digunakan sebagai *client* untuk mengakses berbagai fungsi dan aplikasi yang disediakan oleh server. Selain itu fungsi yang banyak dipergunakan adalah kontrol atas akses *thin client* yang dapat dikonfigurasi menurut keinginan dengan cara melakukan konfigurasi pada *thin client*. Beberapa contoh distribusi yang menggunakan LTSP adalah AbulÉdu, Cutter project, Deworks, Edubuntu, K12LTSP dan Skolelinux

Ada empat layanan dasar yang dibutuhkan untuk melakukan *boot* pada komputer kerja LTSP. Layanan-layanan tersebut adalah:

1. DHCP
2. TFTP
3. NFS
2. XDMCP

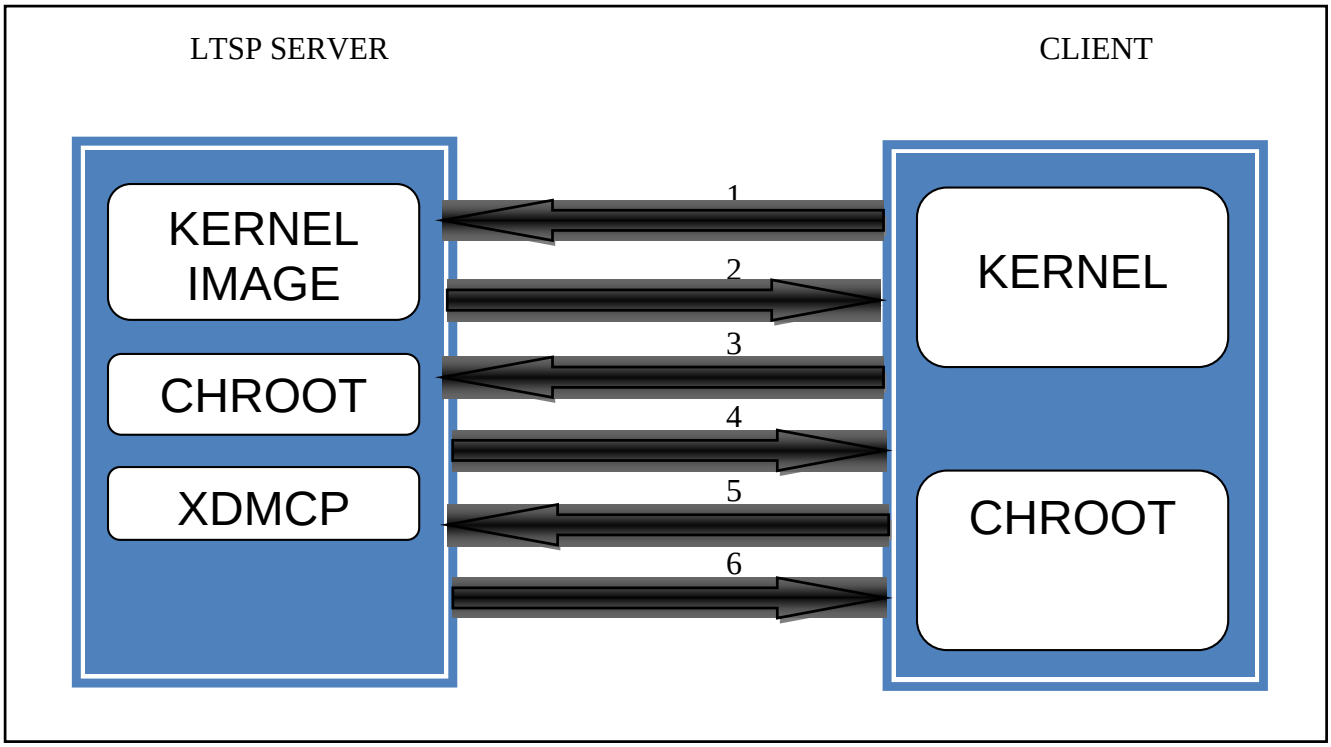
Pada Proses kerjanya, LTSP menggunakan *port-port* komunikasi untuk melakukan pertukaran informasi maupun permintaan-permintaan dari *client* kepada *server*. Berikut adalah daftar *port-port* yang dipergunakan dalam LTSP:

Tabel III.1 Daftar *Port* pada LTSP

Port Spesifik LTSP		
Port	**Tipe**	**Dekripsi**
9100 9102	TCP	Printer, dilayani oleh lp_server
9200	TCP	ltspinfo – digunakan untuk query informasi dari client
9202	TCP	lbuscd – berjalan pada thin client, menunggu koneksi dan pendaftaran session
9210	TCP	Port default untuk ltspswapd (NBD Swap Server)
Standard services		
67	UDP	DHCP (menunggu request DHCP dari clients)
68	UDP	DHCP (dhcpd mengirim jawaban kepada client)
69	UDP	TFTP (client terhubung ke port 69 pada server untuk mengunduh kernel)
111	UDP/TCP	Portmapper
177	UDP	XDMCP (XDM,GDM,KDM)
1067	UDP	Port alternatif untuk port DHCP-67
1068	UDP	Port alternatif untuk port for DHCP-68
6000	TCP	X Protocol menggunakan port ini untuk menghubungkan aplikasi client dengan Xserver

3.1.2 Proses Kerja Umum

Tahapan atau proses kerja umum dari sebuah *linux terminal server project* dapat dilihat pada gambar berikut :



Gambar III.1 Proses Kerja Umum LTSP

Penjelasan dari proses dapat dilihat pada tabel berikut :

Tabel III.2 Tabel penjelasan proses

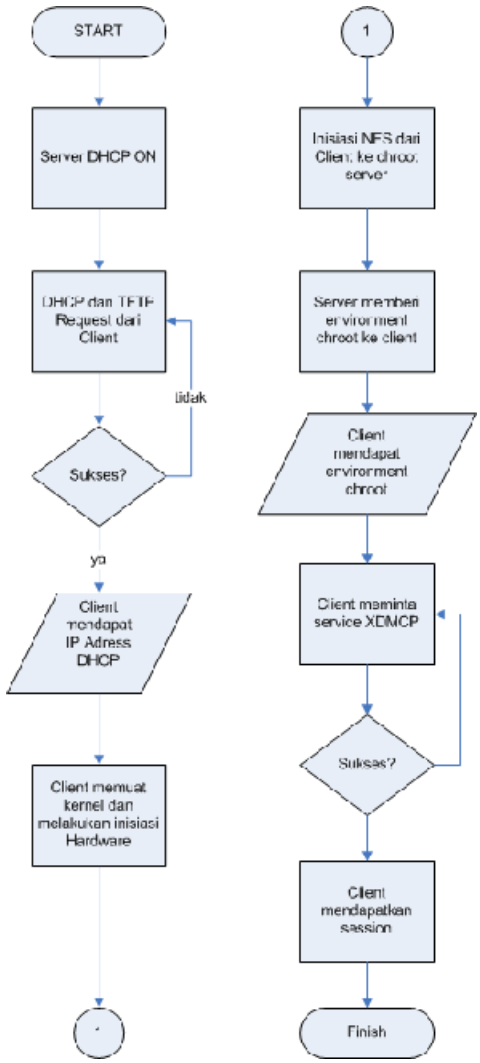
No	Deskripsi
1	Client mengirim permintaan DHCP dan TFTP kepada LTSP server
2	LTSP server memberikan alamat IP dan kernel image
3	Client melakukan komunikasi NFS untuk mendapatkan Chroot
4	LTSP server memberikan environment Chroot pada client

5	Client mengirim permintaan service XDMCP kepada LTSP server
6	Server memberi session kepada client

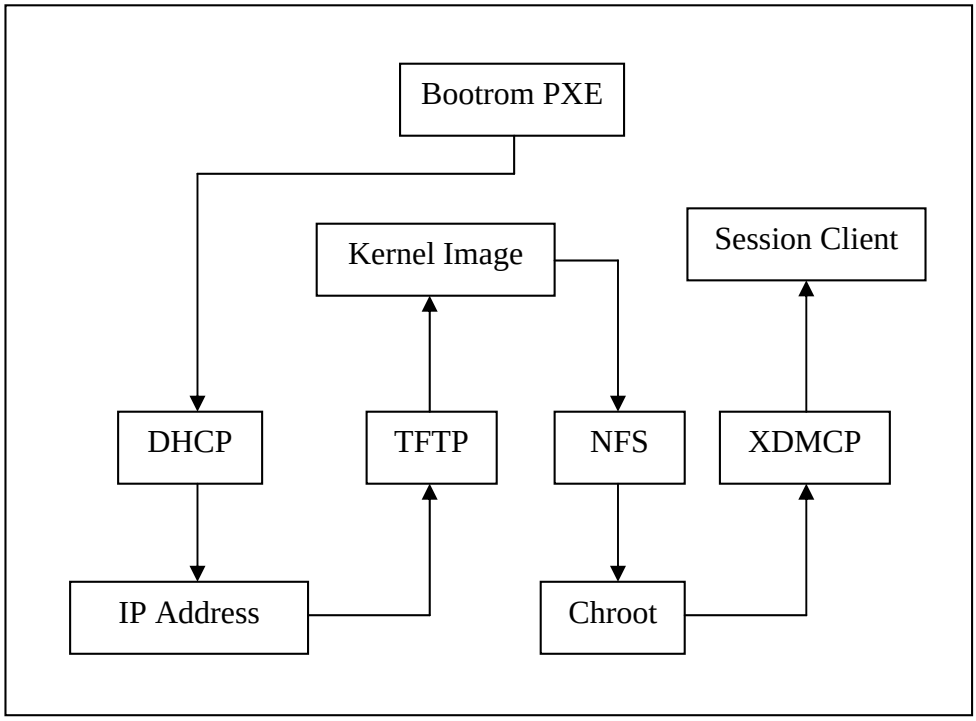
3.1.3 Flowchart dan Gambar Kerja LTSP

Flowchart atau diagram alir dari proses *Linux Terminal Server Project* (LTSP) adalah sebagai berikut

:



Gambar III.2 Flowchart Kerja LTSP



Gambar III.3 Gambar kerja LTSP

3.1.4 Tahapan Proses Rinci sistem LTSP

Berikut adalah tahapan proses rinci dari sistem LTSP:

1. Muat *kernel* linux ke dalam memori komputer kerja. Langkah ini dapat dilakukan dengan cara yang berbeda, termasuk:
    a. Bootrom (Etherboot,PXE,MBA,Netboot)
    b. Floppy
    c. Cakram keras
    d. CD-ROM
    e. USB

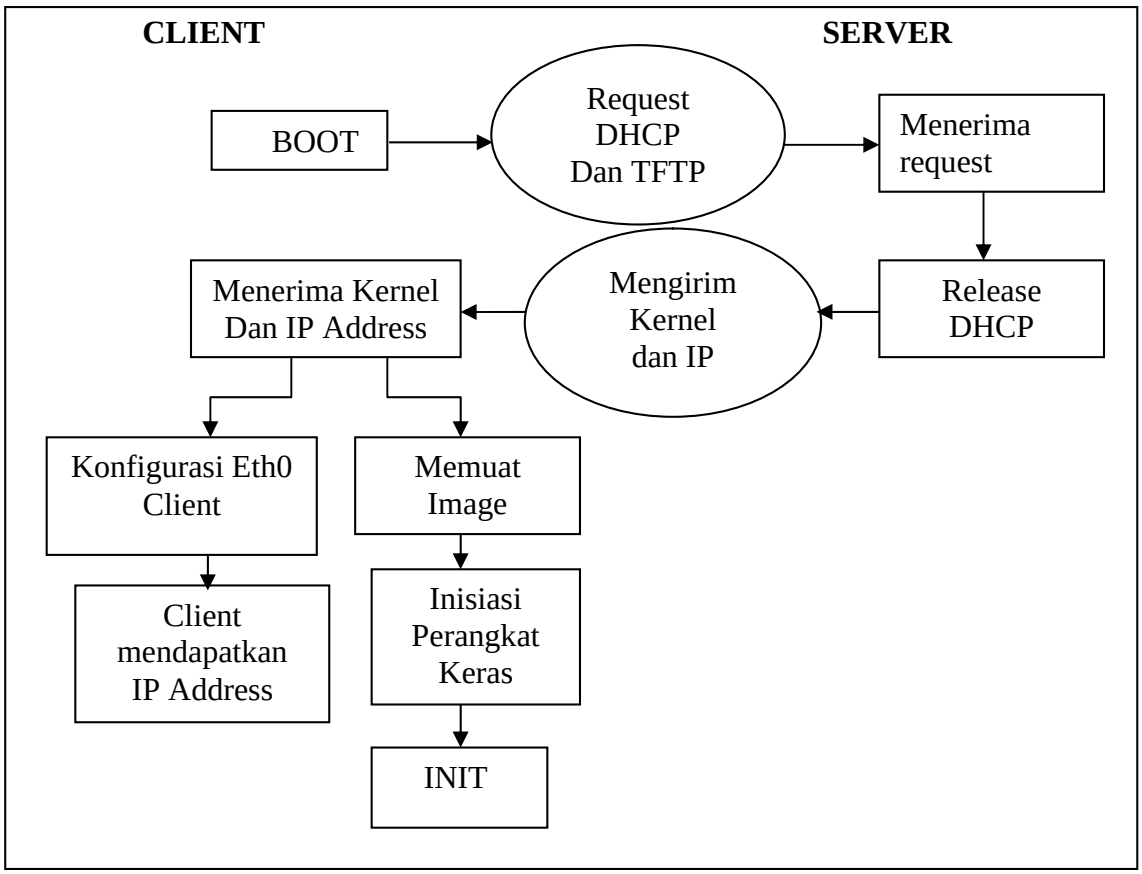
    Selanjutnya kita akan menggunakan PXE bootrom.

2. Setelah *kernel* dimuat ke memori, ia akan mulai mengeksekusi.

3. *Kernel* akan menginisialisasi keseluruhan sistem dan semua *periferal* yang dikenalinya.
4. Ini awal dari semuanya. Selama proses pemuatan *kernel*, citra ramdisk juga akan dimuat ke dalam memori. Argumen baris perintah *kernel* **root=/dev/ram0** memberitahu kernel untuk menambatkan citra tersebut sebagai direktori root.
5. Secara normal, jika *kernel* selesai melakukan *boot*, ia akan meluncurkan program **init**. Namun, dalam kasus ini, telah di instruksikan *kernel* untuk memuat skrip *shell* kecil sebagai gantinya. Hal ini dilakukan ini dengan melewatkan **init=/linuxrc** pada perintah baris *kernel*.
6. Skrip **/linuxrc** dimulai dengan memindai bus PCI, mencari kartu jaringan, Untuk tiap peranti PCI yang ditemukannya, skrip akan dicari di berkas /etc/niclist, untuk melihat apakah ditemukan yang bersesuaian. Setelah yang ditemukan sesuai, nama modul *driver* NIC dikembalikan, dan modul *kernel* itu akan dimuat. Untuk kartu ISA, modul *driver* harus ditetapkan pada baris perintah *kernel*, bersama dengan IRQ atau parameter-parameter alamat yang mungkin diperlukan.
7. Sebuah klien DHCP kecil yang disebut **dhclient** kemudian akan dijalankan untuk membuat permintaan dari *server* DHCP. Hal ini perlu dilakukan untuk memisahkan permintaan ruang- pengguna, karena dibutuhkan lebih banyak informasi dibandingkan yang dibutuhkan bootrom dari permintaan dhcp pertama.

8. Ketika **dhclient** mendapat jawaban dari server ia akan menjalankan berkas **/etc/dhclient-script**, yang akan mengambil informasi yang diperoleh, dan mengkonfigurasi antarmuka eth0.

   Gambaran langkah satu sampai dengan delapan dapat dilihat pada gambar di bawah ini :



Gambar III.4 Proses DHCP dan TFTP

9. Sampai pada tahap ini, sistem berkas *root* telah berada di *ram disk*. Sekarang, skrip /linuxrc akan ditambatkan pada sitem berkas *root* baru via NFS. Direktori tersebut diekspor dari server yang secara khas berada di **/opt/ltsp/i386**. Ia tidak dapat menambatkan sistem berkas baru sebagai /.

Pertama- tama harus ditambatkan sebagai /mnt. Kemudian, ia akan melakukan **pivot_root**. **pivot_root** akan menukar sistem berkas *root* sekarang dengan sistem berkas baru. Ketika selesai, sistem berkas NFS akan ditambatkan pada /, dan sistem berkas *root* lama akan ditambatkan pada /oldroot.

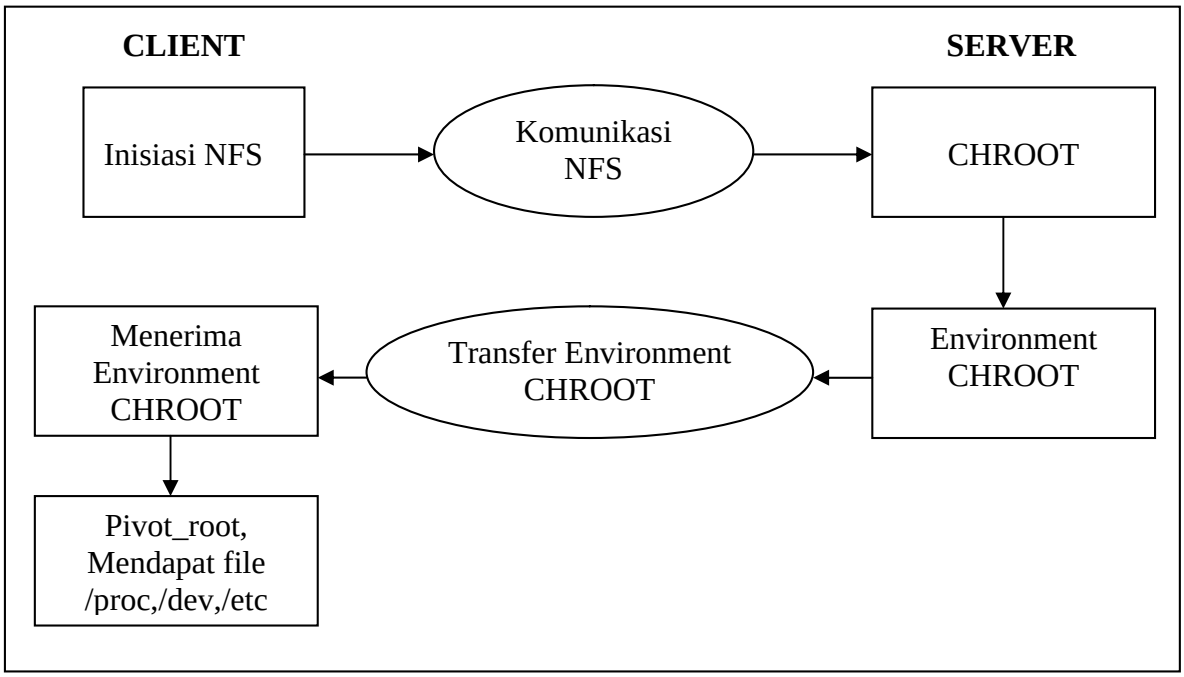
10. Setelah proses penambatan dan *pivot* pada sistem berkas *root* baru selesai, kita selesai dengan skrip *shell* /linuxrc dan kita perlu memanggil program **/sbin/init** yang sebenarnya.
11. Init akan membaca berkas /etc/inittab dan mulai menyiapkan lingkungan komputer kerja.
12. Salah satu dari butir pertama dalam berkas inittab adalah perintah **rc.sysinit** yang akan dijalankan selagi komputer kerja berada dalam keadaan '***sysinit***'.
13. Skrip rc.sysinit akan membuat ramdisk 1mb untuk memuat semua hal yang dibutuhkan untuk ditulis atau diubah dalam berbagai cara.
14. Ramdisk akan ditambatkan sebagai direktori /tmp. Berkas apapun yang perlu ditulis akan berada di direktori /tmp, dan ada tautan simbolis yang menunjuk kepada berkas-berkas ini.
15. Sistem berkas /proc ditambatkan.
16. Berkas lts.conf akan diuraikan, dan semua parameter dalam berkas itu yang berkenaan dengan komputer kerja ini akan di atur sebagai variabel lingkungan untuk digunakan oleh skrip rc.sysinit.

17. Jika komputer kerja tersebut dikonfigurasi untuk melakukan pertukaran melalui NFS, direktori **/var/opt/ltsp/swapfiles** akan ditambatkan sebagai /tmp/swapfiles. Kemudian, jika belum ada berkas tukaran (*swapfile*) untuk direktori ini, maka akan dibuatkan secara otomatis. Ukuran berkas tukaran akan dikonfigurasi dalam berkas lts.conf.

    Berkas tukaran kemudian akan dimungkinkan, menggunakan perintah **swapon**.

18. Antarmuka jaringan ***loopback*** dikonfigurasi. Ini adalah antarmuka yang memiliki alamat *127.0.0.1*.
19. Jika aplikasi lokal dimungkinkan, maka direktori **/home** akan ditambatkan, sehingga aplikasi tersebut dapat mengakses direktori home pengguna.

    Gambaran langkah Sembilan sampai dengan Sembilan belas dapat dilihat pada gambar di bawah ini :



Gambar III.5 Proses CHROOT

20. Beberapa direktori akan dibuat dalam sistem berkas /tmp untuk menyimpan berkas- berkas sementara yang dibutuhkan selagi sistem berjalan. Akan dibuat Direktori-direktori tersebut seperti:
    a. /tmp/compiled
    b. /tmp/var
    c. /tmp/var/run
    d. /tmp/var/log
    e. /tmp/var/lock
21. Berkas /tmp/syslog.conf akan dibuat. Berkas ini akan mengandung informasi yang memerintahkan daemon **syslogd** *host* yang mana dalam jaringan yang akan dikirimi informasi *log*. *Host* syslog ditetentukan dalam berkas lts.conf. Ada hubungan simbolik yang disebut /etc/syslog.conf yang menunjuk ke berkas /tmp/syslog.conf.
22. *Daemon* **syslogd** dimulai, menggunakan berkas konfigurasi yang dibuat dalam langkah sebelumnya.
23. Setelah skrip rc.init selesai, kendali dikembalikan ke program /sbin/init, yang akan mengubah *runlevel* dari ***sysinit*** ke ***5***. Ini akan menyebabkan entri-entri dalam /etc/inittab dijalankan.
24. Secara bawaan, ada entri dalam inittab untuk menjalankan skrip **/etc/screen_session** pada tty1, tty2 dan tty3. Itu artinya bahwa sudah dapat menjalankan 3 sesi sekali jalan, dan tipe sesi dikendalikan oleh entri ***SCREEN_01***, ***SCREEN_02*** dan ***SCREEN_03*** dalam lts.conf.

Lebih banyak entri dapat ditatasiapkan dalam inittab untuk sesi yang lebih banyak, jika diinginkan.

25. Jika ***SCREEN_01*** diatur ke nilai dari ***startx*** maka skrip **/etc/screen.d/startx** akan dijalankan, yang akan meluncurkan sistem X Windows, yang memberi anda antarmuka pengguna grafis.

    Dalam berkas **lts.conf**, ada sebuah parameter yang disebut **XSERVER**. Jika parameter ini tidak ada, atau diatur ke "**auto**", maka deteksi otomatis terhadap kartu video akan dilakukan. Jika kartu tersebut berupa PCI atau AGP, maka ia akan mendapatkan id *vendor* dan peranti PCI, dan akan mencari dalam berkas **/etc/vidlist**.

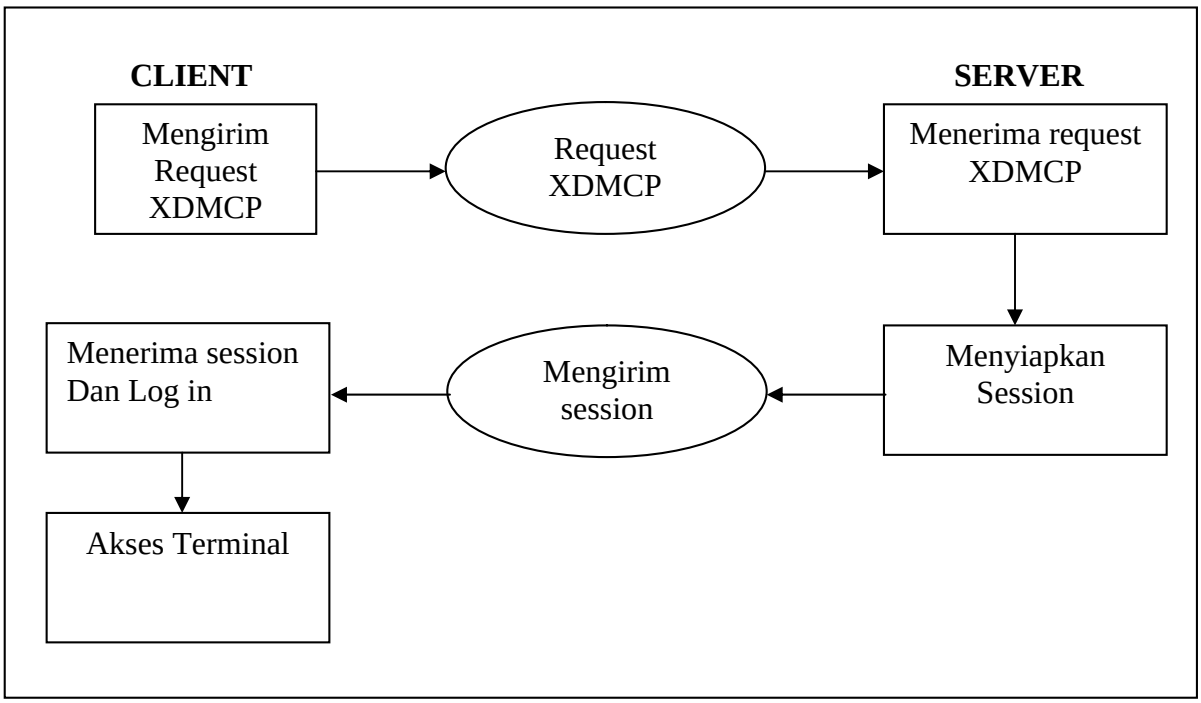
    Jika kartu tersebut didukung oleh Xorg 6.7, rutin pci_scan akan mengembalikan nama modul *driver*. Jika ia hanya didukung oleh XFree86 3.3.6, pci_scan mengembalikan nama dari server X yang akan digunakan. Skrip startx dapat memberi tahu perbedaan tersebut karena nama server 3.3.6 yang lama dimulai dengan 'XF86_', sedangkan modul server X dari Xorg yang lebih baru secara khusus merupakan nama-nama dengan huruf kecil, seperti *ati* atau *trident*.

26. Jika Xorg digunakan, maka skrip **/etc/build_x4_cfg** akan dipanggil untuk membangun berkas XF86Config. Jika XFree86 3.3.6 digunakan, maka **/etc/build_x3_cfg** akan dipanggil untuk membangun berkas XF86Config. Berkas- berkas ini diletakkan dalam direktori /tmp. Yang merupakan ramdisk, yang hanya dikenali oleh komputer kerja.

Berkas XF86Config akan dibangun, berdasarkan entri-entri dalam berkas **/etc/lts.conf**.

27. Setelah berkas XF86Config dibangun, kemudian skrip **startx** akan meluncurkan server X dengan berkas konfigurasi yang baru.
28. Server X akan mengirim permintaan ***XDMCP*** kepada server LTSP, yang akan memunculkan kotak dialog login.
29. Pada titik ini, pengguna dapat melakukan log masuk. dan akan memperoleh sesi di *server.*

Gambaran langkah dua puluh sampai dengan dua puluh sembilan dapat dilihat pada gambar di bawah ini :



Gambar III.6 Proses XDMCP dan session

3.1.5 Parameter Pengujian *Linux Terminal Server Project* (LTSP)

Untuk melihat kinerja dari *Linux Terminal Server Project* (LTSP), mula-mula dilakukan tahap benchmark untuk melihat parameter-parameter data yang dibutuhkan. Parameter-parameter tersebut meliputi :

1. Kinerja prosessor.
2. Kinerja memori dan paket data.

Dari benchmark tersebut akan diperoleh data-data yang dapat menunjukan kinerja *dari linux terminal server project* (LTSP) yang menunjukan secara jelas mengenai penggunaan prosessor dan memori. Adapun tools yang digunakan untuk melakukan benchmark adalah dengan menggunakan dua buah perangkat atau tools sebagai berikut :

1. TOP

   Program Top menyediakan sebuah monitoring dinamis waktu nyata (dynamic real-time view) sebuah dari sistem yang sedang berjalan. Top juga dapat memperlihatkan informasi sistem dan juga sebagai sebuah daftar tugas-tugas yang dikerjakan oleh kernel linux.

2. IPTRAF

   IPTraf adalah sebuah tool statistik jaringan yang berbasis konsol untuk Linux. IPTraf mengumpulkan berbagai parameter seperti paket koneksi dan TCP dan hitungan byte (byte counts), statistik antar muka dan indikator aktifitas, traffic TCP/UDP, dan paket stasiun LAN.

Setelah tahap benchmark dilakukan barulah dapat dilakukan langkah optimasi untuk *Linux Terminal Server Project*.

3.1.6 Analisis Chroot

Chroot pada sistem operasi Unix adalah sebuah operasi yang mengubah direktori root untuk proses yang sedang bekerja dan proses turunannya. Sebuah program yang di re-rooted ke direktori lain tidak dapat mengakses atau mengenali berkas-berkas diluar direktori tersebut, dan direktori tersebut disebut sebuah chroot *jail* atau chroot prison. Istilah chroot dapat merujuk kepada panggilan sistem chroot(2) atau *chroot*(8) wrapper program.

Pada umumnya *Chroot* yang digunakan pada client Linux Terminal Server Project (LTSP) memiliki arsitektur dan distribusi linux yang sama yang sama dengan yang digunakan pada *server* LTSP, artinya bila sebuah server LTSP menggunakan system operasi dengan tipe arsitektur 32 bit (i386) dan distribusi semisal Ubuntu 8.04, maka *chroot* yang digunakan menggunakan system dengan tipe arsitektur 32 bit (i386) dan dengan distribusi ubuntu 8.04. Secara praktikal pemasangan *chroot* pada *linux terminal server project* (LTSP) dilakukan setelah dilakukan *install server linux terminal server project* (LTSP). Perintah yang digunakan untuk *install chroot* adalah **ltsp-build-client**.

Untuk melakukan optimasi ataupun pengembangan pada system *linux terminal server project* (LTSP), dapat dilakukan dengan cara melakukan modifikasi pada system *chroot* yang digunakan. Hal tersebut

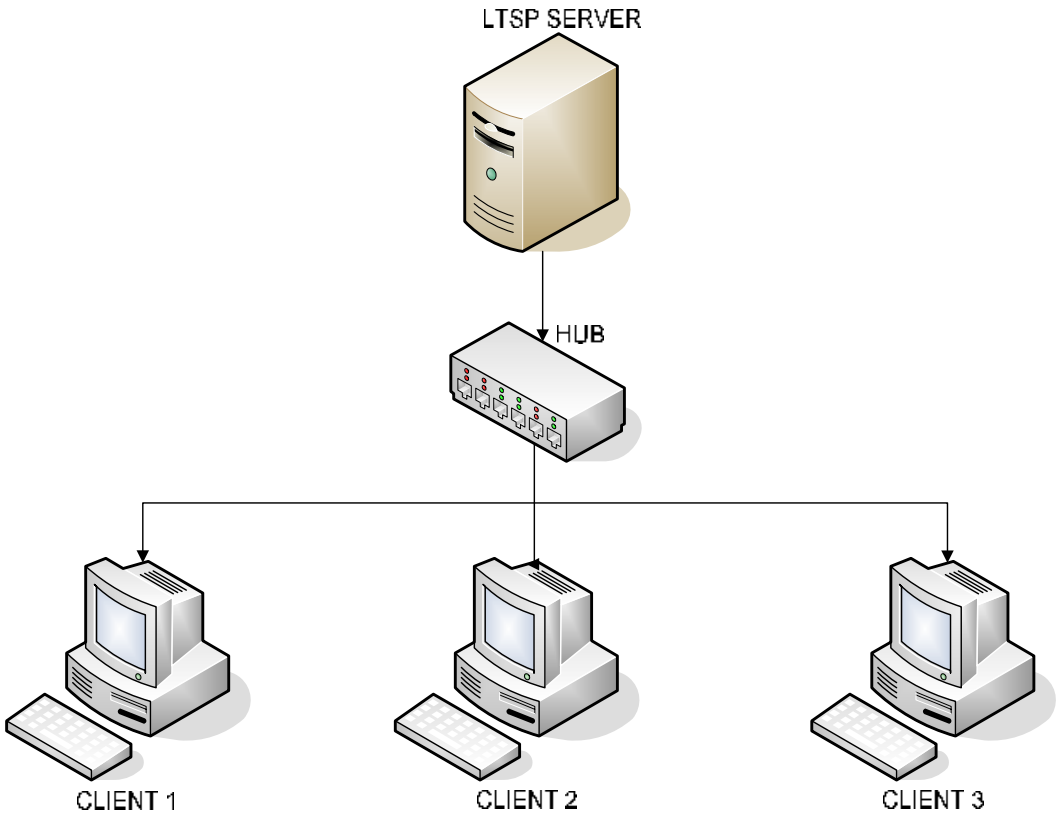
berarti kita dapat menggunakan system arsitektur yang berbeda pada chroot. Selanjutnya akan dilakukan pembandingan kinerja system LTSP yang lebih efektif dan optimal dengan menggunakan *chroot* dengan arsitektur 32 bit (i386) juga dengan melakukan kombinasi distribusi Linux Ubuntu yang digunakan, yaitu dengan Ubuntu 8.04 (Hardy Heron), Ubuntu 8.10 (Intrepid Ibex),dan Ubuntu 9.04 (Jaunty Jackalope).

3.2 Perancangan Sistem *Linux Terminal Server Project* (LTSP)

Sistem LTSP yang akan dibangun akan terdiri dari satu buah *server* dan tiga buah *client* yang akan dihubungkan dengan satu *hub* dan menggunakan topologi *star*. Aplikasi yang akan dijalankan oleh *client* akan terdiri dari tiga aplikasi yaitu :

1. Open Office
2. GIMP
3. Firefox

Gambar dari perancangan sistem LTSP yang akan dibangun dapat dilihat pada gambar di bawah ini :


LTSP SERVER
HUB
CLIENT 1
CLIENT 2
CLIENT 3

Gambar III.7 Rancangan sistem LTSP

3.2.1 Perancangan Server LTSP

Spesifikasi *service* dan konfigurasi *server* LTSP yang akan dibangun adalah sebagai berikut :

1. Menggunakan Distribusi Linux Ubuntu 8.04 (Hardy Heron)

   Sistem operasi yang akan digunakan pada *server* LTSP adalah Linux distribusi Ubuntu 8.04 (Hardy Heron). Distribusi linux ini digunakan karena merupakan versi LTS (*long term support*), yang berarti adanya support waktu sepanjang 3 tahun baik dari segi penyediaan program dan *plugin-plugin* tambahan yang diperlukan. *Repository* atau sumber program untuk linux Ubuntu 8.04 dapat ditemui di situs resmi ubuntu, maupun pada *mirror-mirror* lokal yang terdapat di Indonesia.

2. Menggunakan *LTSP-Server-Standalone*

   Terdapat dua opsi untuk instalasi LTSP pada linux distribusi Ubuntu, yaitu **ltsp-server** dan **ltsp-server-standalone**. Perbedaan dari kedua paket tersebut adalah kelengkapan *service* dan *environment* pendukungnya.

   Ltsp-server-standalone digunakan karena merupakan paket LTSP server yang lengkap dan memiliki *environment* dan program-program pendukung yang lengkap untuk *linux terminal server project* (LTSP) yang akan dibangun.

3. Melakukan modifikasi pada *chroot*

   Modifikasi pada *chroot* akan dilakukan dengan cara melakukan kombinasi penggunaan *environment* chroot yang digunakan pada sistem *linux terminal server project* (LTSP). Kombinasi

tersebut adalah dengan menggunakan *server* dengan paket LTSP linux distribusi Ubuntu 8.04 dan *chroot* dari tiga jenis distribusi Ubuntu, yaitu : Ubuntu 8.04 (Hardy Heron), Ubuntu 8.10 (intrepid Ibex), dan Ubuntu 9.04 (Jaunty Jackalope). Hal ini dilakukan setelah *server* LTSP terinstall dengan baik.

Implementasi dari pengggunaan *chroot* yang berbeda, yaitu *chroot* Ubuntu 8.04 (Hardy Heron), Ubuntu 8.10 (Intrepid Ibex), dan Ubuntu 9.04 (Jaunty Jackalope) selanjutnya akan melalui tahap *benchmark* yang pada akhirnya akan memberikan data yang dibutuhkan untuk mengetahui kombinasi yang paling memberikan efesiensi dan kinerja terbaik dari sistem *linux terminal server project* (LTSP).

Tools pengujian yang akan digunakan adalah program **top**

yang akan memberikan informasi yang dibutuhkan mengenai kinerja prosessor dan memori pada saat *linux terminal server project* (LTSP) bekerja.

4. Melakukan konfigurasi pada file /etc/ltsp/dhcpd.conf

   Konfigurasi yang dilakukan pada file ini adalah setting pada range DHCP, *root path*, serta *next server* yang akan digunakan pada *client linux terminal server project (LTSP).*

3.2.2 Analisis Pengguna

Pengguna Linux Terminal Server Project akan dibuat untuk user yang memiliki karakteristik pengguna sebagai berikut :

1. Berumur 14 sampai dengan 30 tahun
2. Sebagian besar sudah terbiasa menggunakan komputer
3. Memiliki ketertarikan pada perangkat lunak berbasis kode sumber terbuka dan bebas *(open source)*

4. Pernah menggunakan sistem operasi Linux ataupun minimal pernah mengetahui sistem operasi Linux

3.2.3 Analisis Perangkat Keras

3.2.3.1 *Server*

Spesifikasi perangkat keras yang digunakan sebagai komputer *server* memiliki spesifikasi sebagai berikut :

1. Prosesor : AMD Athlon - 2,11 GHz [64 X2]
2. RAM : 1 GB DDR2
3. VGA : NVidia GForce 6600, 256 MB
4. Sound Card : Intel SoundMax HD Audio
5. Ethernet : Nvidia Ethernet
6. HDD : SATA 120 Gb
7. Monitor : Samsung Digital 15”
8. Keyboard dan Optical Mouse Simbadda

Dengan spesifikasi tersebut perangkat untuk kebutuhan *server* LTSP sudah cukup terpenuhi.

3.2.3.2 *Client*

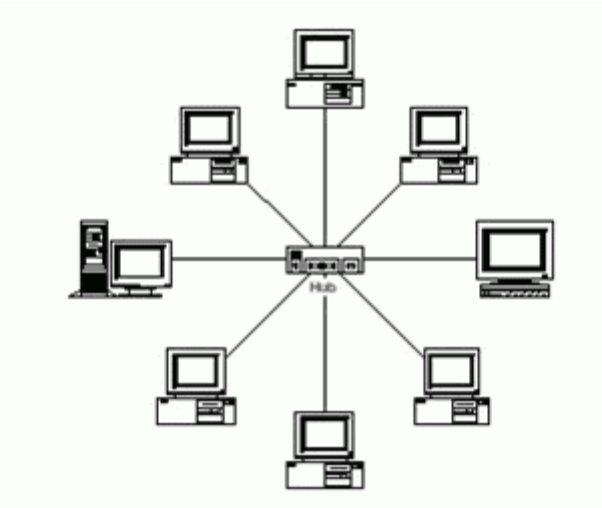
Spesifikasi perangkat keras yang digunakan sebagai komputer client memiliki spesifikasi sebagai berikut :

1. Prosesor : Intel Pentium III - 533 MHz
2. RAM : 512 GB DDR2
3. VGA : On Board

4. Sound Card : On Board
5. Ethernet : Compax RTL 8139
6. Monitor : Samsung Digital 15”
7. Keyboard dan Optical Mouse Simbadda

3.2.4 Analisis Jaringan Komputer

Jaringan komputer yang digunakan dalam sistem *Linux Terminal Server Project* (LTSP) ini adalah sbb :



Gambar III.8 Topologi star

1. Menggunakan topologi *star* dengan satu *hub.*
2.Terdapat satu *server* LTSP yang melayani 3 *client.*

3.2.5 Analisis Perangkat Lunak

3.2.5.1 Server

Spesifikasi perangkat lunak yang digunakan pada *server* adalah sebagai berikut :

1. Sistem operasi : Ubuntu 8.04 (distro turunan Debian)

2. Ruang Hard disk : 120 GB
3. SWAP : 1Gb
4. Tipe partisi : EXT3
5. Web server : XAMPP 1.6.1
6. Firewall : Iptables 1.3.5

3.3 Implementasi Penginstalasi *Server Linux Terminal Server Project*

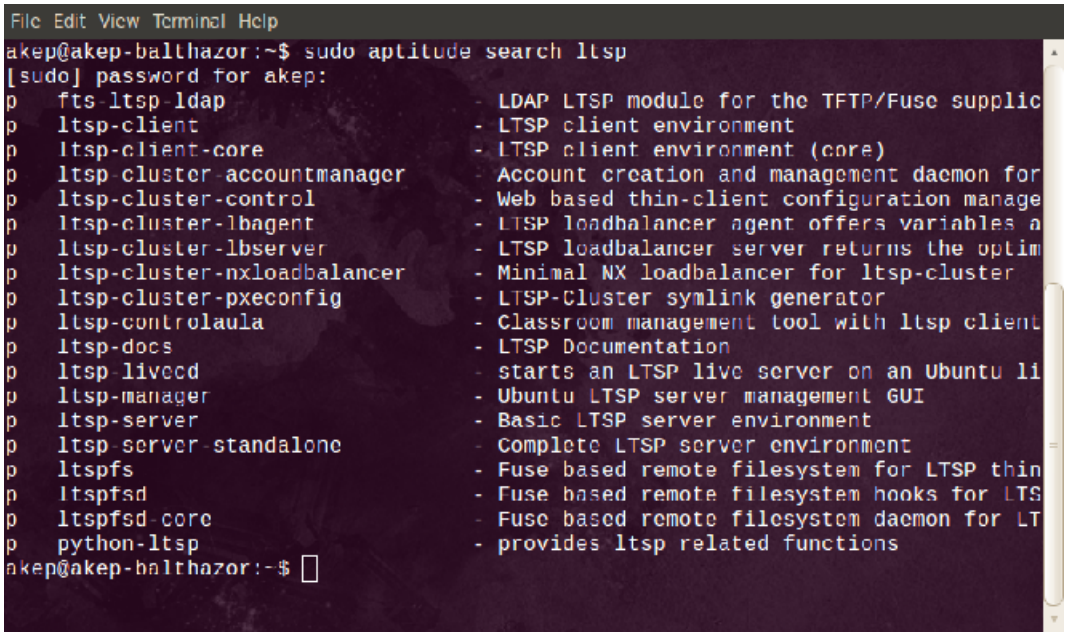
Untuk memulai instalasi *server* LTSP, langkah pertama yang dilakukan adalah mempersiapkan *repository*, yaitu *bundle* paket-paket program yang diperlukan oleh sistem operasi Linux Ubuntu. Sumber *repository* dapat diperoleh dari koneksi langsung dari internet dengan merujuk kepada *repository* resmi Ubuntu di alamat http://archive.canonical.com/ubuntu atau *mirror-mirror* lokal seperti http://kambing.ui.edu/repository/ubuntu.

3.3.1 Instalasi *Server* LTSP

Langkah-langkah dalam instalasi LTSP adalah sbb:

1. Mencari paket LTSP pada *repository* dengan perintah sudo aptitude search ltsp

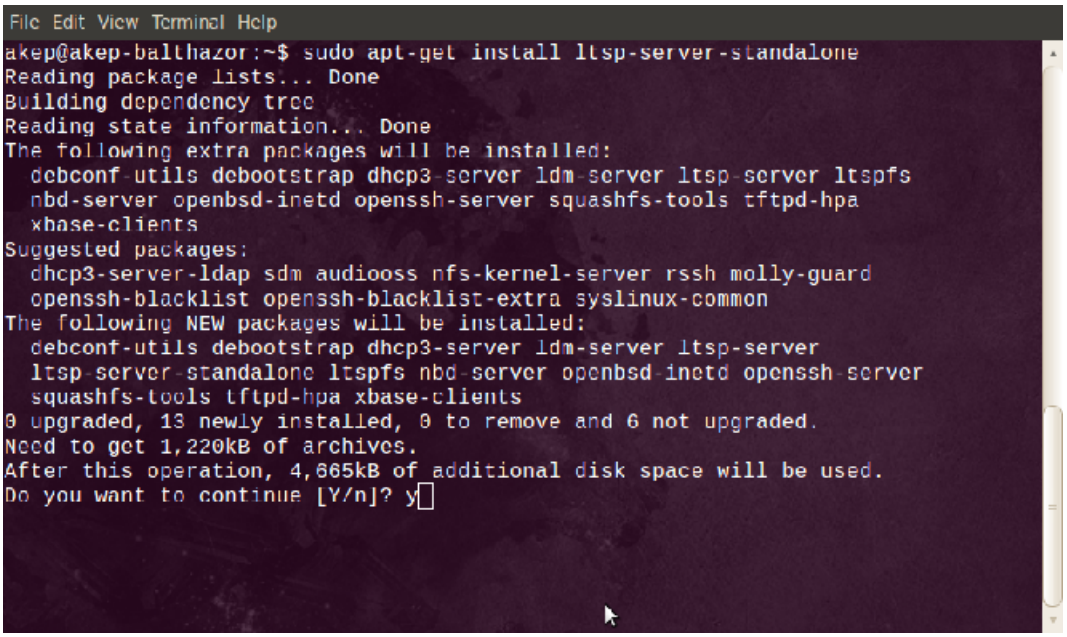
```
sudo aptitude search ltsp
```



Gambar III.8 Pencarian paket LTSP

2. Install ltsp-*server*-standalone pada *server*

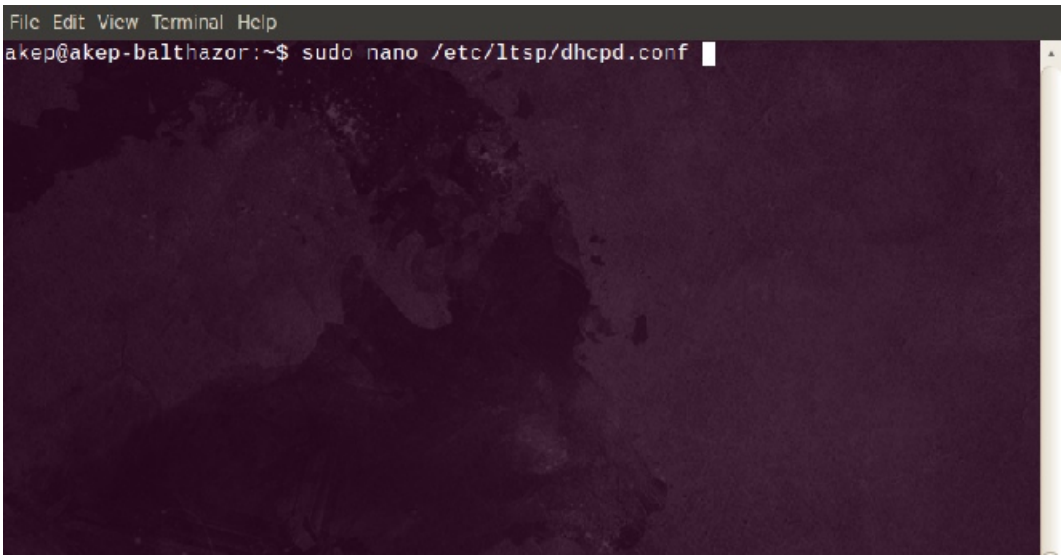
```
sudo apt-get install ltsp-server-standalone
```



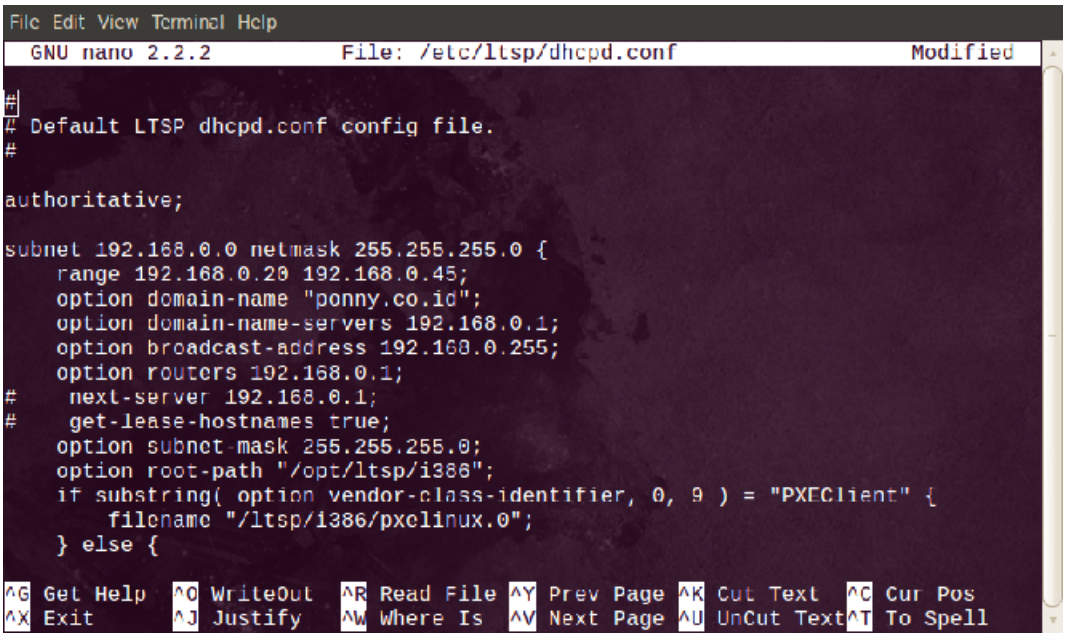
Gambar III.9 Instalasi ltsp-*server*-standalone

3. Konfigurasi file /etc/ltsp/dhcpd.conf

```
sudo nano /etc/ltsp/dhcpd.conf
```



Gambar III.10 Edit file /etc/ltsp/dhcpd.conf



Gambar III.11 File /etc/ltsp/dhcpd.conf

Tabel dibawah ini menunjukan konfigurasi pada file /etc/ltsp/dhcpd:

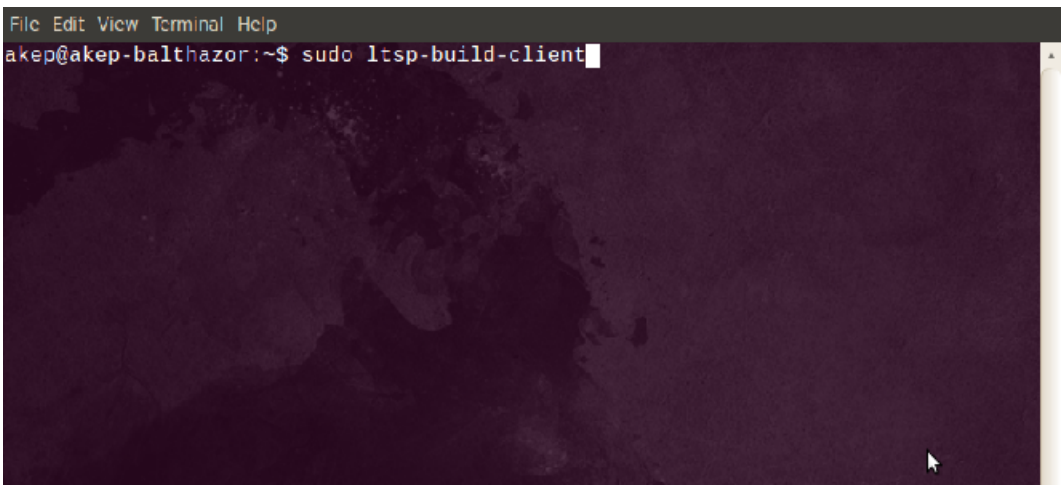
Tabel III.3 Konfigurasi file /etc/ltsp/dhcpd.conf

No	Jenis Konfigurasi	Setting

1	Subnet	10.0.0.0
2	Netmask	255.255.255.224
3	Range IP DHCP	10.0.0.16 – 10.0.0.31
4	Root path	/opt/ltsp/i386
5	Router	10.0.0.1
6	Next *server*	10.0.0.9

4. Install *environment chroot* LTSP Ditribusi Ubuntu 8.04

```
sudo ltsp-build-client
```



Gambar III.12 Instalasi chroot LTSP ditribusi Ubuntu 8.04

5. Install *environment chroot* LTSP Ditribusi Ubuntu 8.10

```
sudo ltsp-build-client –dist intrepid –arch i386
```



Gambar III.13 Instalasi chroot ltsp ditribusi Ubuntu 8.10

6. Membuat *user client* LTSP

Akan dibuat tiga *user client* pada sistem ltsp

```
sudo adduser ponny1
```



Gambar III.14 Pembuatan user ponny1

```
sudo adduser ponny2
```

```
File  Edit  View  Terminal  Help
akep@akep-balthazor:~$ sudo adduser ponny2
Adding user `ponny2' ...
Adding new group `ponny2' (1002) ...
Adding new user `ponny2' (1002) with group `ponny2' ...
Creating home directory `/home/ponny2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ponny2
Enter the new value, or press ENTER for the default
        Full Name []: ponny2
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Gambar III.15 Pembuatan user igos2

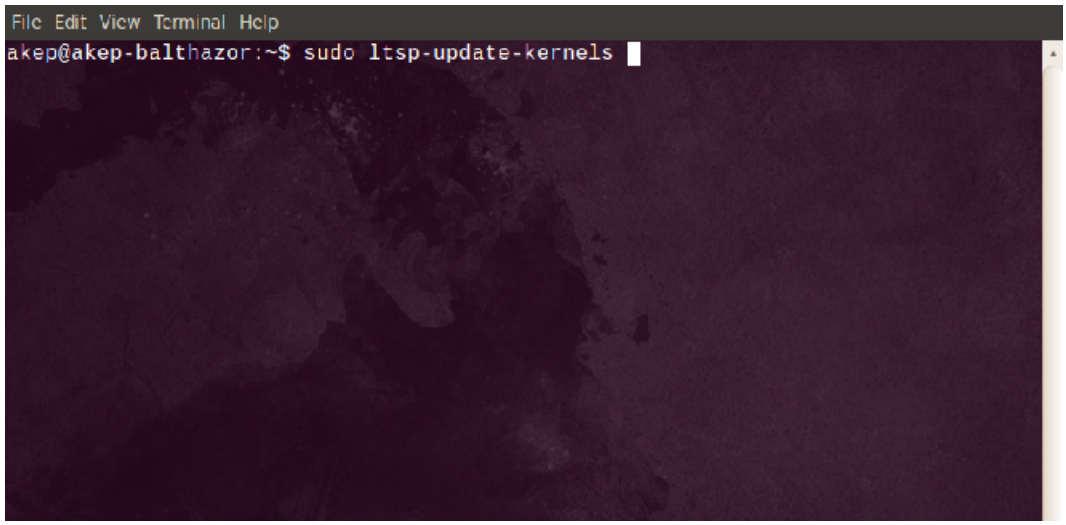
```
sudo adduser ponny3
```

```
File  Edit  View  Terminal  Help
akep@akep-balthazor:~$ sudo adduser ponny3
Adding user `ponny3' ...
Adding new group `ponny3' (1003) ...
Adding new user `ponny3' (1003) with group `ponny3' ...
Creating home directory `/home/ponny3' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ponny3
Enter the new value, or press ENTER for the default
        Full Name []: ponny3
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Gambar III.16 Pembuatan user igos3

7. Melakukan update-kernel

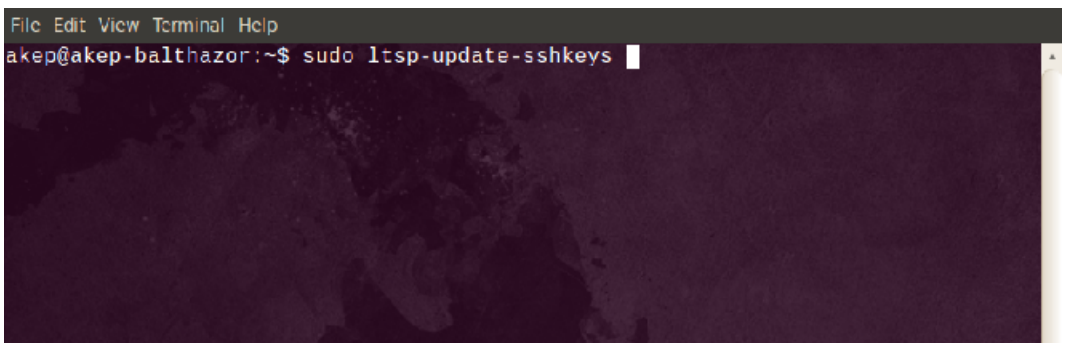
```
sudo ltsp-update-kernels
```



Gambar III.17 update kernel LTSP

8. Melakukan update-sshkey

```
sudo ltsp-update-sshkeys
```



Gambar III.18 update sshkey LTSP

3.3.2 Tampilan Login dan Desktop *Linux Terminal Server Project (LTSP)*

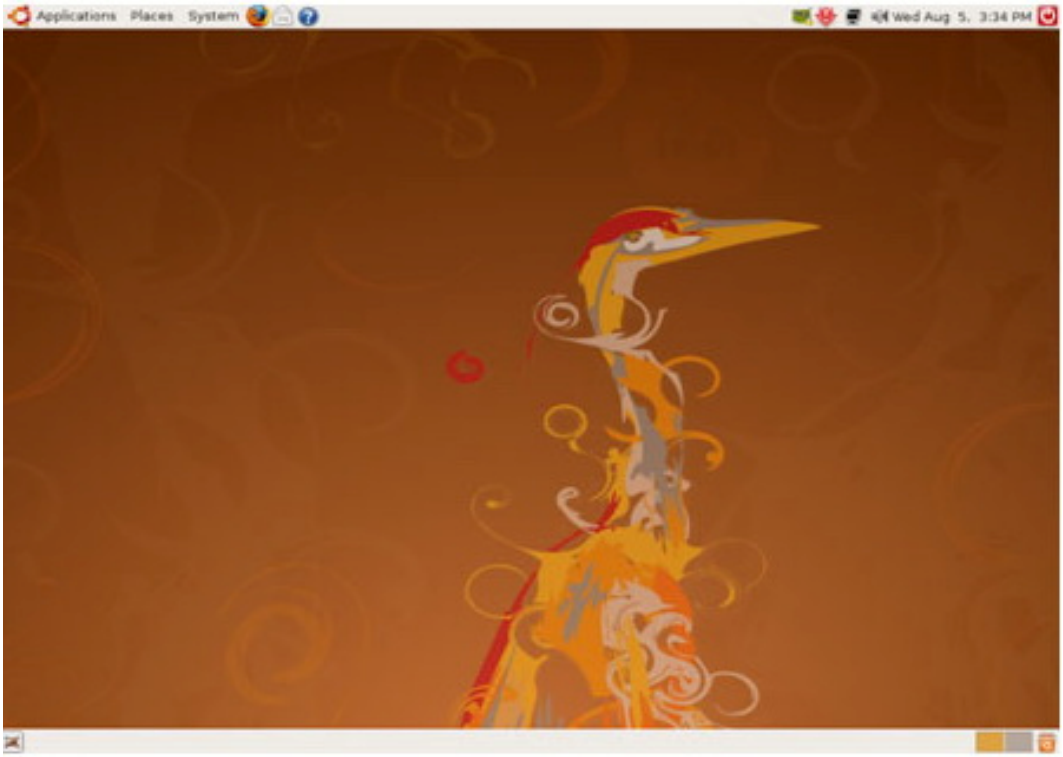
Setelah proses instalasi LTSP berhasil dilakukan, maka *Terminal Server Project (LTSP)* yang dibangun akan dapat digunakan. Tampilan antarmuka

(*interface)* dari halaman *login* dan tampilan *desktop Linux Terminal Server Project (LTSP)* dapat dilihat pada gambar-gambar dibawah ini :



Gambar III.19 Halaman login LTSP

Pada halaman ini *user* memasukan *username* dan *password*,jika berhasil maka user akan mendapat sesi pada *server.* Tampilan *Desktop user* dapat dilihat pada gambar berikut :



Gambar III.20 *Desktop* LTSP

3.3.3 Pengujian kerja LTSP pada dua puluh *Client* dengan hak akses program

Sebelum melakukan pengujian pada tiga buah *client* dengan melakukan modifikasi pada chroot, untuk membuktikan penggunaan LTSP pada penggunaan banyak *client* (*multiple client*) dalam dokumentasi resminya di situs web http://www.ltsp.org menyebutkan, sebuah *server* tunggal LTSP dengan spesifikasi dual prosesor Intel Pentium III, 650MHz, dan RAM 1GB dapat melayani 40-120 client, yang semuanya menjalankan Netscape web browser dan OpenOffice.org. Parameter lain yang digunakan adalah adanya pembedaan hak akses dari setiap *user* untuk pengaksesan program pada LTSP, artinya dapat dilakukan pengaturan hak akses program yang digunakan pada setiap *user*. Pengaturan tersebut

dilakukan pada *server* dengan perintah **chmod,** sebagai contoh jika kita ingin melakukan restriksi program firefox maka perintah yang dilakukan adalah sebagai berikut :

Tahapan *benchmark* dilakukan dengan cara menyalakan *client* satu persatu sampai tiga *client* semuanya beroperasi dan diberikan tiga buah beban aplikasi, yaitu Openoffice,Gimp,dan Firefox web browser. Batasan ini dilakukan karena aplikasi-aplikasi tersebut adalah aplikasi yang dirancang untuk digunakan oleh *client* dan selain itu juga menuntut penggunaan sumberdaya yang cukup besar,sehingga layak untuk dijadikan aplikasi untuk pengujian *Linux terminal server project* (LTSP).

Tabel berikut memperlihatkan distro Linux yang digunakan yaitu distribusi Ubuntu 8.04, 8.10, 9.10 dan paket-paket utamanya.

Tabel III.4 Tabel Pengujian Distro Linux

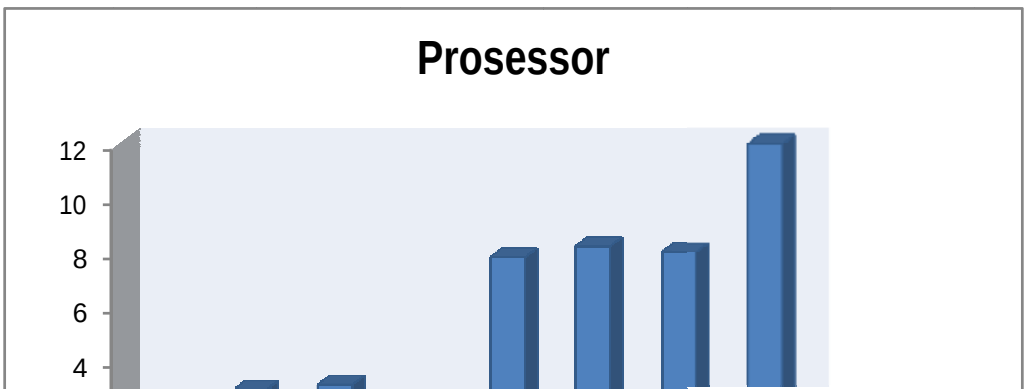
No	**Distro**	**Versi Kernel**	**Versi GCC**	**Versi X-Window**
1	Ubuntu 8.04	2.6.24-19-Generic	Gcc-4.2-base	Xorg-*server* 1.4.0.90
2	Ubuntu 8.10	2.6.27	Gcc-4.3.2	Xorg 7.4
3	Ubuntu 9.04	2.6.28-11.37	Gcc-4.3.3	Xorg-*server* 1.6

Tabel dibawah menunjukan penggunaan prossesor dan memory pada keadaan satu sampai tiga *client* aktif :

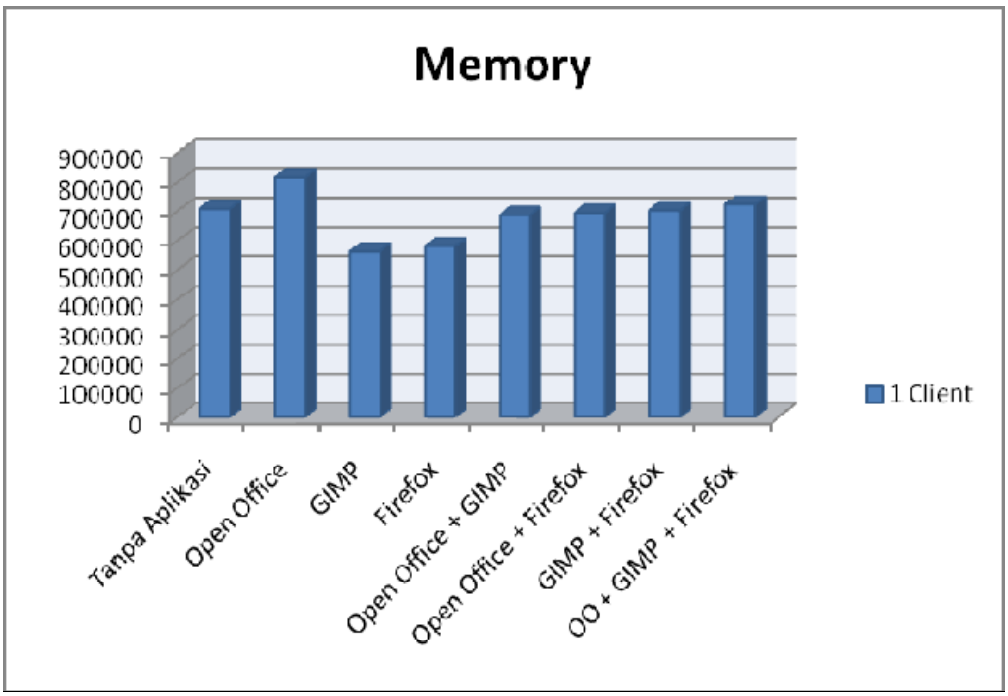
Tabel III.5 Tabel *Benchmark* pada satu *client* 8.04

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memory
1	Tanpa Aplikasi	1,0 %	707 576 k used
2	Open Office	2,9 %	817 136 k used
3	GIMP	3,1 %	562 340 k used
4	Firefox	1,6 %	580 204 k used
5	Open Office + GIMP	7,8 %	687 392 k used
6	Open Office + Firefox	8,2 %	694 300 k used
7	GIMP + Firefox	8,0 %	703 320 k used
8	OO + GIMP + Firefox	12,0 %	723 652 k used

Tabel diatas memperlihatkan hasil *benchmark* LTSP dengan satu *client* aktif mulai dari keadaan tidak menjalankan aplikasi sampai dengan menjalankan aplikasi Open Office,GIMP,dan firefox. Seperti yang terlihat pada tabel penggunan memory dan prossesor cukup stabil pada penggunaan satu *client*. Selanjutnya data yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar III.21 Grafik penggunaan prosessor pada satu *client* 8.04



Gambar III.22 Grafik penggunaan memory pada satu *client* 8.04

Dari hasil *benchmark* LTSP dengan sistem default yaitu dengan versi 8.04 dapat dilihat dengan jelas bahwa penggunaan memory masih cukup tinggi. Hal ini berkaitan dengan versi kernel dan GCC yang dimiliki oleh distribusi Linux

Ubuntu versi 8.04. Kekurangan dari kernel 2.6.24-19-Generic yang dimiliki oleh Linux distribusi Ubuntu 8.04 adalah kurangnya *library* untuk proses NFS dan *upgrade-upgrade* lainnya yang dapat meningkatkan kinerja LTSP.

Dari hasil data pengujian dan grafik yang didapatkan, dapat dilihat bahwa modifikasi LTSP menggunakan Ubuntu versi 9.04 adalah memiliki efesiensi dan kerja yang lebih baik dibandingkan dengan menggunakan chroot *default* Ubuntu 8.04 dan chroot Ubuntu 8.10 hal ini disebabkan oleh versi kernel Linux Ubuntu 9.04 yang merupakan versi kernel terbaru yang memiliki fitur-fitur tambahan untuk mendukung kinerja LTSP..

Kinerja LTSP yang lebih baik pada chroot Linux Ubuntu 9.04 juga disebabkan oleh fitur dan perbaikan yang ditambahkan pada Linux Ubuntu 9.04. Perbaikan dan fitur-fitur tambahan tersebut adalah :

1. *Local applications support*, yaitu fitur dimungkinkan untuk memilih aplikasi mana yang dijalankan oleh *server* dan yang dijalankan secara lokal oleh *client*.
2. *Integrated LTSP-Cluster support.*
3. *Optional fully-featured dhcp client integration (udhcpc).*
4. *Scripting xorg.conf* terbaru.