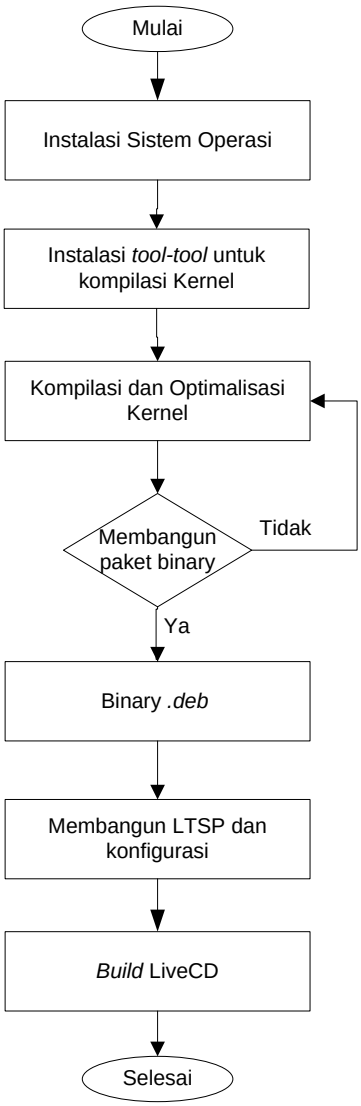
BAB III
PERANCANGAN SISTEM

Bab ini akan memaparkan bagaimana *source* kernel linux dibangun hingga menjadi sebuah paket *binary* kernel linux yang berkstensi **.deb** yang optimal serta membangun jaringan LTSP dan membangunnya ke dalam sebuah *Live* CD, dapat dilihat dari diagram dibawah ini.



Gambar 3.1 Diagram Perancangan Secara Umum

Diagram diatas menjelaskan bahwa proses perancangan secara umum dari mulai instalasi sistem operasi yang akan digunakan, dan setelah itu instalasi *tool* yang akan digunakan untuk proses kompilasi dan membangun paket binary **.deb** selanjutnya membangun environment *client*, LTSP *server,* dan membangunnya kedalam sebuah *Live* CD/DVD.

3.1 Instalasi Sistem

Instalasi sistem adalah proses intalasi secara keseluruhan yang dibutuhkan dalam perancangan membangun paket binary **.deb**, membangun LTSP, dan membangun *Live* CD/DVD.

3.1.1 Perangkat Keras yang Digunakan

Tabel 3.1 Perangkat Keras yang Digunakan

Hardware	Jumlah	Spesifikasi
Server	1	a. AMD Athlon™ 64 X2 Dual Core Processor 4600+ b. RAM DDR2 1024 MB c. Harddisk 160 GB d. 100 Mbps NIC
Client	3	a. Sekelas AMD Athlon™ 64 X2 b. RAM DDR2 1024 MB
Switch	1 Buah	Switch 100 Mbps D-Link
Kabel UTP	4 Meter	Belden Australia

3.1.2 Perangkat Lunak yang Digunakan

1. Sistem Operasi Ubuntu Desktop 9.10

2. *Kernel Package*
3. LTSP Server versi 4.2
4. UCK (*Ubuntu Customization Kit*)

3.1.3 Instalasi Ubuntu Desktop

Ubuntu Desktop sebagai sistem operasi berbasis GNU/Linux yang dijadikan penelitian tugas akhir ini adalah distribusi Ubuntu Desktop 9.10 (*Karmic Koala*), proses instalasi berada pada halaman Lampiran A.

3.2 Konfigurasi, Kompilasi, dan Instalasi Kernel

Proses konfigurasi dan optimasi kernel dengan dua metode, metode manual dan metode aturan ubuntu untuk kompilasi kernel karena menggunakan dua versi kernel yang berbeda. Untuk fitur-fitur akan diaktifkan maka akan disesuaikan dengan kebutuhan dari sistem jaringan LTSP karena dengan demikian kernel akan menjadi optimal dan stabil.

3.2.1 Membangun Paket Kernel Linux 2.6.33

Persiapan sebelum melakukan membangun dan instalasi paket kernel dengan metode manual, dilakukan dalam *mode user root* yaitu *login* sebagai *root* dan mempersiapkan mesin untuk mengkompilasi kernel linux versi 2.6.33 semua dilakukan di *Terminal.* [7][8][9]

1. Paket `kernel-package` digunakan sebagai paket kernel yang akan menjadi pedoman saat akan melakukan pengkonfigurasian kernel *source,* sebelum kernel *source* akan di kompilasi sesuai dengan kebutuhan dan menyediakan kemampuan untuk membuat paket *kernel-image* debian.
2. Paket `libncurses5-dev` ini digunakan agar saat melakukan kompilasi pada kernel *source,* dapat menampilkan modus grafik (GUI) dari kernel

source yang akan dikompilasi sehingga memudahkan kita dalam melakukan kompilasi kernel linux.

3. Paket `fakeroot` ini digunakan sebagai paket yang akan berfungsi untuk menyamarkan *user* yang kita gunakan menjadi seolah-olah menjadi *user root.*
4. Paket `wget` dan `bzip2` aplikasi download manager yang ada pada sistem operasi linux, dan biasanya aplikasi ini telah menjadi *default* dari linux, sehingga kita tidak perlu menginstall aplikasi ini setelah melakukan instalasi sistem operasi linux. Sedangkan `bzip2` aplikasi ini digunakan untuk mengekstrak atau mengurai file terkompresi berbentuk file *tar.bz* dan biasanya sudah terinstall secara *default.*.
5. Menyunting *file script shell* untuk membangun paket kernel.

```
~$ sudo –i <password root>
~# apt-get install kernel-package
~# apt-get install libncurses5-dev
~# apt-get install fakeroot
~# apt-get install wget
~# apt-get install bzip2
```

Sebelum melakukan konfigurasi ada baiknya *file script shell* disunting terlebih dahulu agar pada saat selesai kompilasi, paket *binary* akan di bangun secara otomatis. Script tersebut berada didalam `/usr/share/kernel-package/ruleset/misc/version_vars.mk` dengan menggunakan *text editor vim,* untuk menyuntingnya masih dalam *mode root*, setelah itu disimpan dan pindah direktori kerja ke `/usr/src` untuk melakukan konfigurasi kernel download kernel *source* yang dibutuhkan, kernel yang digunakan versi 2.6.33, script dan proses konfigurasi secara lengkap ada pada halaman Lampiran B.

```
~# vim /usr/share/kernel-package/ruleset/misc/version_vars.mk
```

3.2.2 Membangun Paket Kernel Linux 2.6.32-22.33

Persiapan untuk melakukan membangun dan instalasi paket kernel linux dengan metode aturan ubuntu, paket-paket untuk menkompilasi, konfigurasi, dan install kernel *source* versi 2.6.32-22.33 sebagai berikut, dilakukan dalam *mode root*. [6][7]

1. Paket `fakeroot` ini digunakan sebagai paket yang akan berfungsi untuk menyamarkan *user* yang kita gunakan menjadi seolah-olah menjadi *user root.*
2. Paket `build-essential` ini berfungsi untuk membuat sebuah *file* binary berkstensi **.deb**.
3. Paket `crash` ini untuk mengetahui apkah terjadi *crash* pada system pada saat kompilasi.
4. Paket `kexec-tools` alat untuk membantu proses pembuatan paket binary.
5. Paket `makedumpfile` ini untuk memperkecil suatu *file* atau menkompresi data.
6. Paket `kernel-wedge` ini untuk membagi paket udeb dan mengeluarkan untuk menjasi paket `kernel-image` **.deb** untuk digunakan oleh paket `debian-installer.`
7. Paket `git-core` ini menyediakan komponen inti git.
8. Paket `libncurses5` dan `libncurses5-dev` ini digunakan agar saat melakukan kompilasi pada kernel *source*, dapat menampilkan modus grafik (GUI) dari kernel *source* yang akan dikompilasi sehingga memudahkan kita dalam melakukan kompilasi kernel linux.
9. Paket `libelf-dev` ini digunakan untuk menyediakan `libelf1` menyediakan *library* bersama yang memungkinkan membaca dan menulis *file* ELF pada tingkat tinggi, paket ini berisi pengembangan *library* dan *file header* untuk `libelf1`.
10. Paket `asciidoc` untuk menulis dokumentasi *text* manual UNIX.

11. Paket `binutils-dev` ini termasuk *file header* dan *library* statis yang diperlukan untuk membangun program-program yang menggunakan *library* GNU BPH, yang merupakan bagian dari `binutils`. Paket ini di pergunakan untuk membangun paket debian dan yang ketergantungan pada paket `libbfd` secara bersamaan maka tidak boleh dilakukan.

```
~$ sudo su - <password root>
~# apt-get install fakeroot
~# apt-get install build-essential
~# apt-get install crash
~# apt-get install kexec-tools
~# apt-get install makedumpfile
~# apt-get install kernel-wedge
~# apt-get build-dep linux
~# apt-get install git-core
~# apt-get install libncurses5
~# apt-get install libncurses5-dev
~# apt-get install libelf-dev asciidoc
~# apt-get install binutils-dev
```

Proses kompilasi, konfigurasi, dan installasi kernel versi 2.6.32-22.33, ada pada halaman Lampiran B.

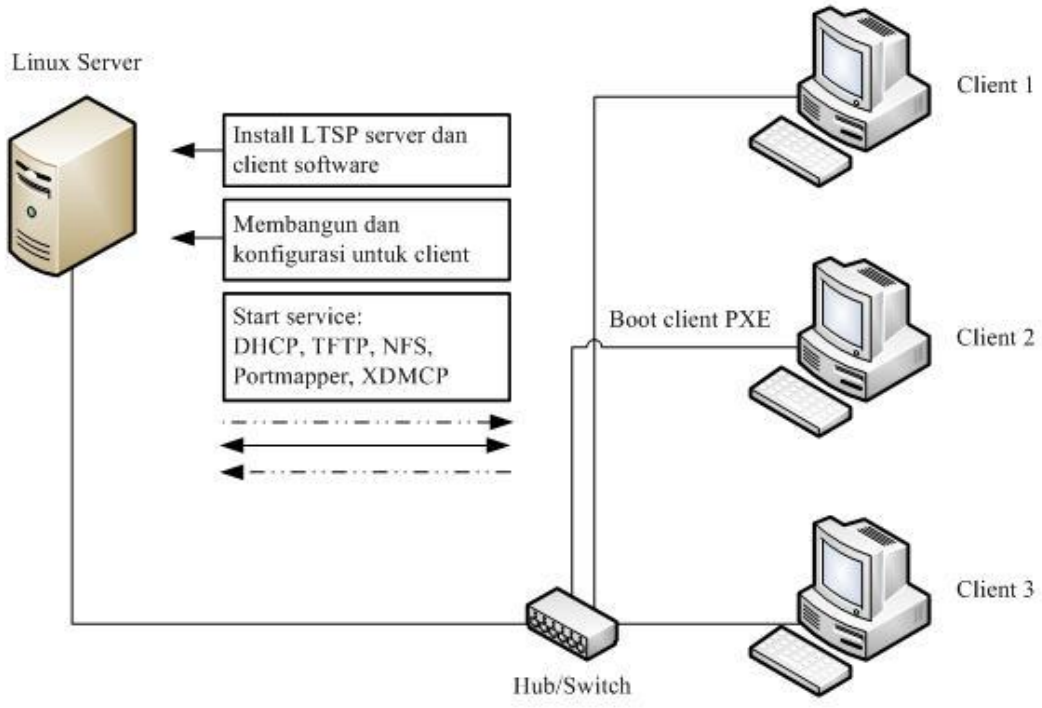
3.3 Perancangan Linux Terminal Server Project (LTSP)

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* dengan tiga kernel linux yang berbeda dan terdiri dari tiga aplikasi yaitu :

1. *Booting*
2. Open Office

3. Web Browser Firefox

Gambar dari perencangan sistem LTSP yang akan dibangun dapat dilihat pada **Gambar 3.1**.



Gambar 3.1 Rancangan Sistem LTSP

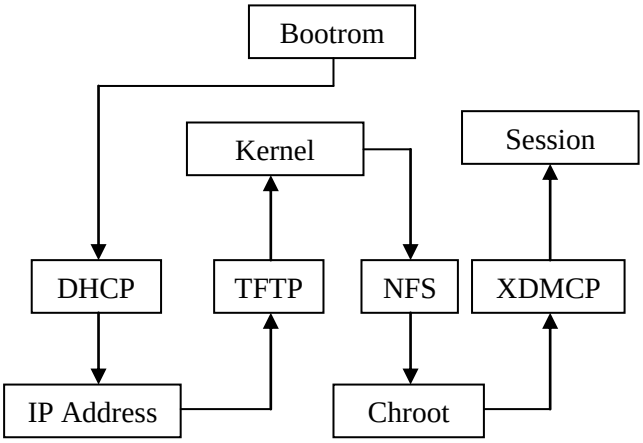
Ada empat layanan dasar yang dibutuhkan untuk melakukan *boot* pada komputer kerja LTSP. Layanan−layanan tersebut adalah:

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

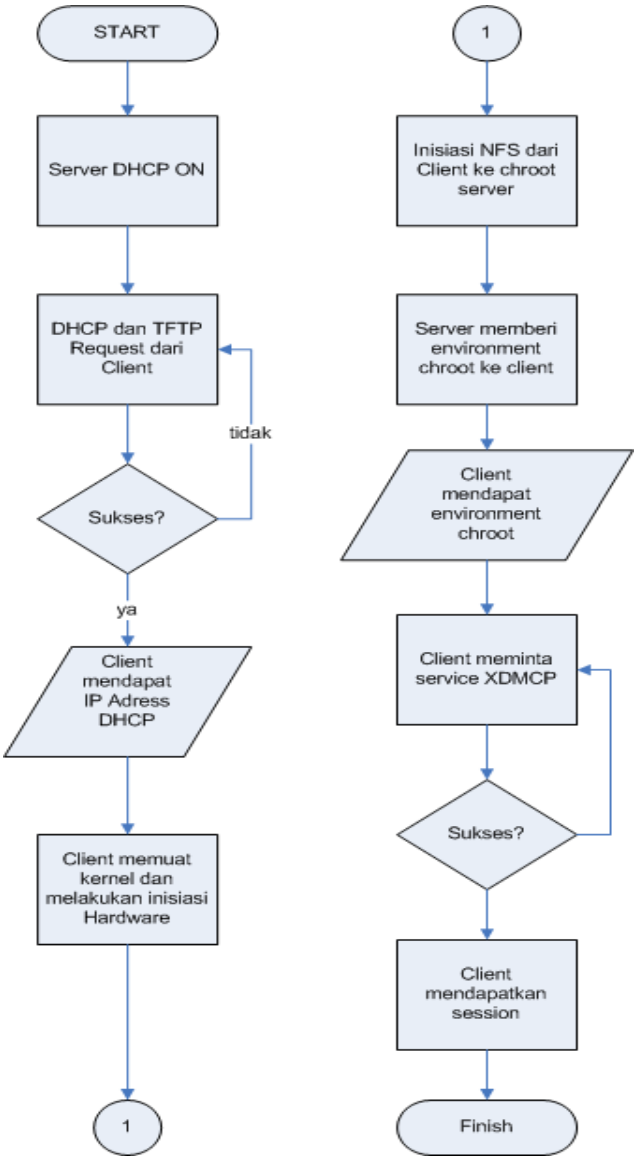
3.3.1 Proses Kerja Umum

Tahapan proses kerja umum dari sebuah sistem jaringan LTSP berikut penjelasannya sebagai berikut, beserta flowchart dan gambar kerja:

1. Ketika komputer *client* melakukan booting melalaui bootrom (Etherboot, PXE, MBA, Netboot) maka *client* akan mengirimkan permintaan DHCP dan TFTP kepada LTSP *server.*
2. LTSP *server* akan memberikan alamat IP dan kernel image.
3. Setelah *client* mendapatkan IP, *client* melakukan komunikasi NFS untuk mendapatkan Chroot.
4. LTSP *server* memberikan environment Chroot pada *client*.
5. Maka *client* akan mengirimkan perintah *service* XDMCP kepada LTSP *server*.
6. *Server* member *session* kepada *client.*



Gambar 3.2 Proses Kerja LTSP



Gambar 3.3 Flowchart Kerja LTSP

Dari diagram dan flowchart alir proses LTSP tersebut sudah dijelaskan cara kerja diatas dari mulai proses permintaan IP dari *client*, lalu *client* memuat kernel dan melakukan inisialisasi *hardware*, melakukan inisialisasi *client* ke Chroot *server* sehingga *server* member environment terhadap *client* dan *client* meminta *service* XDMCP sampai *client* mendapatkan *session* dan bisa *login* sebagai *user*.

3.3.2 Perancangan Server LTSP

Pada perancangan server LTSP dengan *service* dan konfigurasi yang akan dibangun adalah sebagai berikut:

1. Paket `ltsp-server` dan `ltsp-server-standalone` merupakan paket LTSP *server* yang lengkap untuk membangun system LTSP *server* yang memiliki environment dan program-program pendukung untuk LTSP yang akan dibangun.
2. Paket `openssh-server` merupakan program yang menyerupai telnet, tetapi berbeda dengan telnet karena SSH mempunyai pengendalian terhadap keamanan. Program ini menyediakan komunikasi aman ter-enkripsi antara dua *host* yang tidak saling mengenal melalui jaringan.
3. Melakukan konfigurasi *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* LTSP, semua proses ada pada halaman Lampiran C.

```
~$ sudo su - <password root>
~# apt-get install ltsp-server
~# apt-get install ltsp-server-standalone
~# apt-get install openssh-server
```

3.4 Membangun Live CD/DVD

Setelah semua sistem telah di uji dan berjalan dengan baik dan stabil, selanjutnya adalah membangun ulang untuk dijadikan sebuah sistem operasi GNU/Linux *Live* CD/DVD khususnya berbasis Ubuntu Desktop 9.10 (*Karmic Koala*). Untuk membangun sebuah *Live* CD/DVD menggunakan UCK (*Ubuntu Customization Kit*) adalah *tool* resmi untuk membangun ulang (*remaster*) sebuah distribusi yang sesuai kebutuhan, dengan menambahkan paket-paket apa pun ke dalam sistem yang akan dibangun, untuk paket misalnya bahasa, atau aplikasi.

```
~$ sudo apt-get install uck  <password root>
```

Setelah proses instalasi *tool* UCK lakukan perombakan, proses ada di Lampiran C. Pada saat *remastering* ada penambahan paket yang disertakan ke dalam *Live* CD/DVD nantinya, maka berkas **.iso** *size* akan bertambah dan tidak akan cukup dalam satu CD jadi menggunakan DVD yang ukuran kapasitasnya yang relatif lebih besar dibandingkan dengan CD.