BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Seringkali pengadaan infrastruktur komputer terbentur pada masalah optimalisasi penggunaan perangkatnya yang harus menyesuaikan perkembangan teknologi yang diadopsi seperti sistem operasi atau perangkat lunak yang digunakan.

Hal ini disebabkan penggunaan arsitektur yang umumnya mengharuskan kompabilitas semua perangkat yang digunakan. Solusi dari masalah tersebut adalah dengan menerapkan arsitektur *thin-client,* sehingga tidak semua perangkat komputer harus di *upgrade* atau diganti. Hanya perangkat di sisi *server* saja yang disesuaikan agar dapar mengadopsi teknologi sistem operasi dan perangkat lunaknya, dalam artian memenuhi kompabilitas yang diinginkan.

Dengan sistem LTSP komputer *client* tidak diinstal suatu Sistem Operasi. Semuanya memanfaatkan teknologi kloning, yaitu sebuah *Server* yang besar diakses oleh banyak *client* melalui jaringaan LAN agar merasakan kecepatan yang hampir sama dengan Server tersebut.

Teknologi ini populer di lingkungan sekolah dan kantor yang memiliki keterbatasan dana dalam pengembangan jaringan komputernya. Peningkatan performa serta fitur dilakukan hanya pada server sehingga dapat menekan biaya

investasi jangka panjang. Disamping itu sistem yang terpusat telah mengurangi beban pemeliharaan dan penanganan masalah (*error handling*), sehingga pengembangan dapat lebih ditekankan pada pengembangan fitur dan performa dari server dan *client*.

3.2 Analisis Masalah

Linux Terminal Server Project (LTSP) menyediakan suatu cara yang bisa dipelajari dengan mudah untuk mengoptimalkan komputer jaringan yang murah (*low cost workstation*) sebagai *workstation* (*client*) dengan tampilan grafik ataupun terminal yang berbasiskan teks dengan memanfaatkan GNU/Linux server. Teknologi yang mirip sistem *mainframe* ini semakin populer karena dapat menghemat sumber daya perangkat keras (*hardware*) tanpa perlu mengurangi kinerja.

Platform Linux yang kian berkembang dan memasyarakat telah terbukti memberikan banyak solusi alternatif yang mendorong efisiensi, penghematan biaya dan kemudahan kerja. Sifatnya yang *open source* telah melahirkan berbagai kombinasi baru di bidang teknologi informasi dan dapat dimanfaatkan oleh siapa saja yang mau memanfaatkannya. Cukup dengan menyediakan satu *harddisk*, LTSP dapat menjalankan banyak terminal komputer. Dengan demikian teknologi *lowend* PC yang menggunakan LTSP ini dapat mengoptimalkan komputer lama yang sudah tidak terpakai untuk dapat digunakan kembali dengan mode GUI. LTSP merupakaan penggunaan PC secara bersama - sama oleh dua *user*/*client* atau lebih dengan memanfaatkan komputer berspesifikasi minimal tanpa

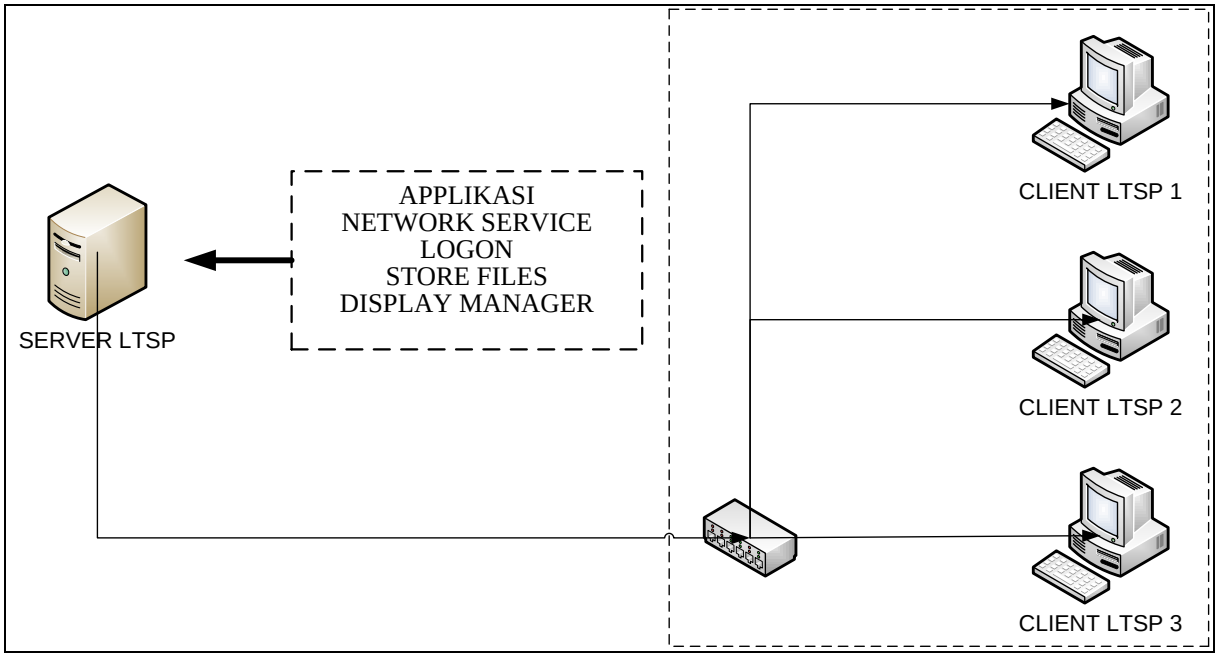
menggunakan *harddisk,* dan dapat digunakan untuk solusi alternatif jika tidak menggunakan komputer berkekuatan tinggi.

Dengan *diskless system,* PC lama bisa digunakan dengan lebih cepat. Hal ini bisa terjadi karena proses kerjanya sebagai berikut :

1. Sebuah *client* yang disambungkan ke server tanpa menggunakan sistem operasi yang tersimpan di harddisk. Sebagai gantinya, menggunakan *bootrom* yang terpasang di kartu untuk menghubungkan *client* ke *server*.
2. Penggunaan metode *diskless computer* berbeda dengan metode *dump terminal*. Karena pada *diskless system,* meskipun *client* menggunakan *resource* pada server namun tidak semuanya, karena tetap dibantu oleh *processor* dan memori pada *client*.
3. Tentunya semakin baik *client* akan membantu kinerja server. Demikian pula halnya server, semakin tinggi spesifikasi dan optimalnya aplikasi pada server yang digunakan akan semakin baik pula kinerja diskless system.

3.3 Deskripsi Sistem

LTSP adalah kepanjangan dari *Linux terminal Server Project* yang dikembangkan oleh James McQuillan (www.ltsp.org). LTSP adalah paket add-on dari linux yang memungkinkan banyak user bekerja pada sebuah komputer. Seluruh aplikasi berjalan pada server dan terminal hanya menangani input dan output.



Gambar 3.1 Deskripsi Sistem LTSP

Seperti yang terlihat pada Gambar 3.1, LTSP merupakan sebuah jaringan komputer dimana server memberikan fasilitas identifikasi, sistem operasi,aplikasi, penyimpanan data dan *display manager* kepada *client*, sehingga komputer *client* dapat memiliki sistem operasi dan aplikasi yang sama dengan servernya. Teknologi yang mirip sistem *mainframe* ini semakin populer karena dapat menghemat sumber daya hardware tanpa perlu mengurangi performanya.

Linux Terminal Server Project adalah sebuah aplikasi linux yang mampu menjawab kebutuhan akan sebuah sistem yang murah, free dan powerfull. Dengan membuat sebuah LTSP server, maka semua terminal yang terhubung ke server tidak harus mempunyai harddisk, dan bisa berbagi resource dengan sempurna.

3.4 Analisis Non Fungsional

3.4.1 Analisis Pengguna

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

1. Sebagian besar sudah terbiasa menggunakan komputer
2. Memiliki ketertarikan pada perangkat lunak berbasis kode sumber terbuka dan bebas *(open source)*
3. Pernah menggunakan sistem operasi Linux ataupun minimal pernah mengetahui sistem operasi Linux

3.4.2 Analisis Perangkat Keras

3.4.2.1 Server

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

1. Prosesor : AMD Turion 2,2 GHz
2. RAM : 2 GB DDR2
3. Ethernet : Realtek Ethernet
4. HDD : SATA 160 Gb

3.4.2.2 *Client*

Spesifikasi perangkat keras yang digunakan sebagai komputer *client* sebagai berikut :

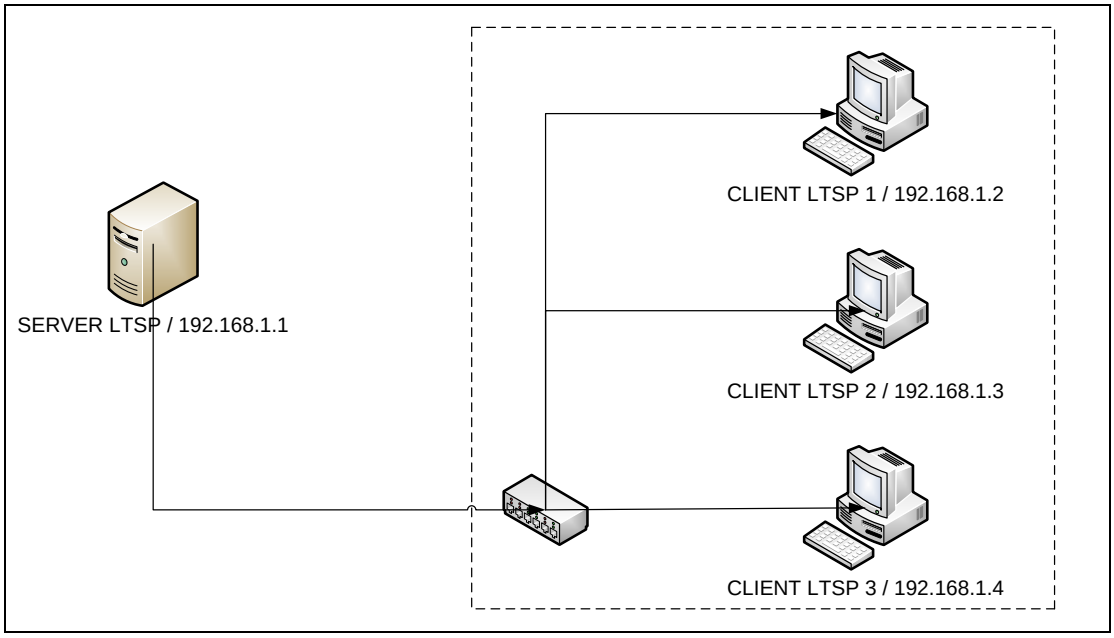
1. Prosesor : Minimal 533 MHz
2. RAM : 128 MB
3. Ethernet : Ethernet card dengan dukungan PXE boot

3.4.3 Analisis Jaringan Komputer

Dalam pengerjaan skripsi ini, tipe dan arsitektur jaringan yang digunakan adalan LAN berdasarkan jangkauan areanya, kemudian berdasarkan fungsinya digunakan sistem jaringan *client* server, serta menggunakan Topologi Star.

Tipe ini digunakan karena jaringan dibangun di satu ruangan dengan jangkauan kecil. LAN merupakan salah satu arsitektur jaringan yang paling sederhana dan dapat dikembangkan menjadi arsitektur jaringan yang cukup luas cakupannya. Luas cakupan LAN itu sendiri tidak melebihi dari satu area yang terdiri dari beberapa terminal yang saling dihubungkan, sehingga menambahkan fungsi dari terminal itu sendiri.

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



Gambar 3.2 Topologi star LTSP

1. Menggunakan topologi *star* dengan satu *switch.*
2. Terdapat satu *server* LTSP yang melayani 3 *client*.
3. Server menggunakan IP 192.168.1.1
4. *Range* IP *client* mulai 192.168.1.2 sampai 192.168.1.20

Lebih lanjut mengenai jaringan yang akan dipakai untuk server dan *client* LTSP dapat dilihat pada tabel berikut :

Tabel III.1 Konfigurasi jaringan LTSP

No	Jenis Konfigurasi	Analisis Konfigurasi
1	Subnet	192.168.1.0
2	Netmask	255.255.255.0
3	Range IP DHCP	192.168.1.2 – 192.168.1.20
4	Root path	/opt/ltsp/i386

3.4.4 Analisis Perangkat Lunak

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

1. Sistem operasi : Ubuntu 10.04 (distro turunan Debian)
2. Ruang Hard disk : 120 GB
3. SWAP : 1Gb
4. Tipe partisi : EXT3
5. Kernel : Paket Kernel versi 3.0

3.5 Analisis Fungsional

Untuk menjalankan sistem LTSP dibutuhkan beberapa sevice yang harus telah terinstall dan berjalan dengan baik, service – service itu sendiri adalah :

1. DHCP (*Dynamic Host Configuration Protocol*), adalah layanan yang secara otomatis memberikan nomor IP kepada komputer yang memintanya. Komputer yang memberikan nomor IP disebut sebagai DHCP server, sedangkan komputer yang meminta nomor IP disebut sebagai DHCP *Client*. Dengan demikian tidak perlu lagi harus memberikan nomor IP secara manual pada saat konfigurasi TCP/IP untuk *client* LTSP, tapi cukup dengan memberikan referensi kepada DHCP Server.
2. TFTP (*Trivial File Transfer Protocol*), memiliki fungsionalitas dasar dari protokol *File Transfer Protocol* (FTP). TFTP digunakan untuk melakukan booting komputer dalam proses LTSP.
3. NBD (*Network block device*) merupakan sebuah teknik remote data storage, yang memungkinkan sebuah komputer *client* untuk dapat mengakses penyimpanan data (data store) pada LTSP. NBD digunakan seolah-olah terdapat sebuah disk drive sebenarnya pada *client*.
4. SSH (*Secure Shell*), protokol jaringan yang memungkinkan pertukaran data melalui saluran aman antara dua perangkat jaringan. Digunakan untuk login ke mesin *remote* dalam X11 connections.

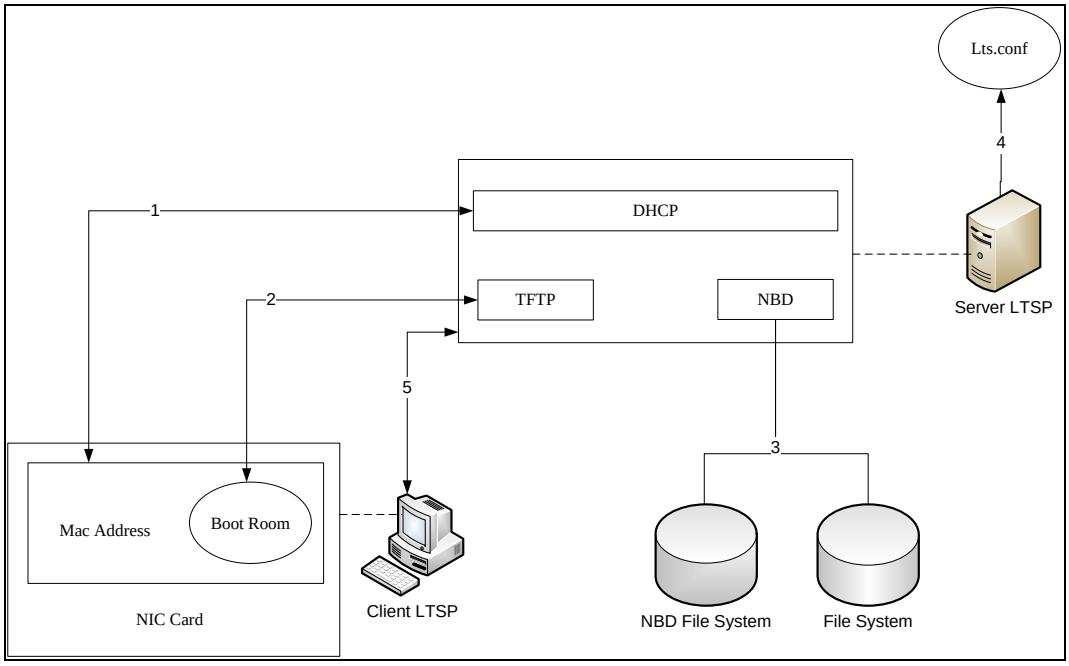
Pada Proses kerjanya, LTSP menggunakan *port-port* komunikasi untuk melakukan pertukaran informasi maupun permintaan-permintaan dari *client* kepada server. Berikut adalah daftar *port-port* yang dipergunakan dalam LTSP :

Tabel III.2 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 *client*s)
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

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



Gambar 3.3 Proses kerja umum LTSP

Berikut penjelasan langkah - langkah proses kerja umum LTSP dari gambar diatas :

1. Pada saat komputer *client* melakukan booting dari Lancard, *Client* akan meminta IP Address dari LTSP Server melalui protokol DHCP (server), DHCP akan membaca file konfigurasi dhcp.conf yang berada pada /etc/ltsp/dhcp.conf.
2. Setelah *client* mendapatkan IP address dari protocol DHCP, Server kemudian memuat Linux kernel dari *preconfigured* Linux Image (pxelinux.0 atau nbi.img) kedalam BOOT Rom *client* dengan menggunakan protokol TFTP (Trivial File Transfer Protokol) yang telah berjalan pada services LTSP Server.
3. Pada saat bersamaan Server akan melakukan mount filesystem baru yang dilakukan oleh protokol NBD (*Network File System*) khas berada di /opt/ltsp/i386. Ia tidak dapat menambatkan sistem berkas baru sebagai /. Pertama−tama harus dimount sebagai /mnt. Kemudian, ia akan melakukan pivot_root. pivot_root akan menukar sistem berkas root sekarang dengan sistem berkas baru. Ketika selesai, sistem berkas NBD akan dimount pada /, dan sistem berkas root lama akan dimount pada/oldroot. dan meload image dari /opt/ltsp/images/i386.img.
4. Server akan membaca file konfigurasi lts.conf, jika pada file lts.conf terdapat konfigurasi sound maka daemon pulseaudio akan dijalankan, untuk mengizinkan *client* melakukan koneksi ke server dan menjalankan file audio. Server juga menjalankan service ltspfsd untuk memungkinkan *client* menggunakan USB Flash Drive dan me*load* screen session yang digunakan pada file lts.conf. default session screen adalah session_7.

5. *Client* akan membangun saluran SSH (Secure Shell tunnel) untuk berhubungan ke Server LTSP yang kemudian memulai LDM (LTSP Display Manager) login manager pada *client*.

3.6 Perancangan Sistem Linux Terminal Server Project (LTSP)

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

1. Menggunakan Distribusi Linux Ubuntu 10.04 (Lucid Lynx)

   Sistem operasi yang akan digunakan pada *server* LTSP adalah Linux distribusi Ubuntu 10.04 (Lucid Lynx). 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 10.04 dapat ditemui di situs resmi ubuntu, maupun pada *mirror-mirror* lokal yang terdapat di Indonesia.

2. Menggunakan kernel versi 3.0

   Untuk rilis versi 3.0 ini merupakan “loncatan” dari versi sebelumnya yaitu versi 2.6.39 tidak dimaksudkan sebagai suatu perubahan fitur yang mendasar. Fitur terpenting yang diikutkan disamping perampingan penomoran menjadi Linux 3.0, antara lain adalah termasuk: Btrfs Data scrubbing dan defragmentasi otomatis, dukungan

XEN Dom0, ICMP_ECHO biasa, wake on WLAN, penyaringan Berkeley Packet Filter JIT, syscall yang memungkinkan penanganan lebih baik dari sistem virtualisasi ringan seperti kontainer. Dukungan hardware baru telah ditambahkan: misalnya, Microsoft Kinect, AMD Fusion Llano APUS, Intel iwlwifi 105 dan 135, Intel C600 serial-attached-scsi controller, Ralink RT5370 USB, beberapa perangkat Realtek RTL81xx atau Apple iSight webcam.

3. 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.

4. 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 10.04 dan dilakukan setelah *server* LTSP terinstall dengan baik. 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.

5. 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).*

6. Efisiensi sistem LTSP agar kinerja *client* dan server semakin optimal, salah satunya dengan menggunakan dengan menggunakan *localapps* .

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.***

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

3.6.1 Perancangan paket kernel versi 3.0 untuk server LTSP

Persiapan sebelum melakukan pembangunan dan instalasi paket kernel dengan metode manual, dilakukan dalam *mode user root*, yaitu *login* sebagai *root* dan mempersiapkan mesin untuk mengkompilasi kernel versi 3.0 semua dilakukan di *Terminal*. Paket-paket yang dibutuhkan diantaranya :

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 *gcc (GNU Compiler Collection),* yakni program untuk mengkompilasi source code C. Kernel Linux sendiri ditulis untuk dapat dikompilasi oleh GCC, sehingga paket ini dibutuhkan untuk mengkompilasi paket kernel.
3. Paket *libncurses5-dev*, digunakan agar saat melakukan kompilsai pada *kernel source* dapat menampilkan modus grafik (GUI) dari kernel *source* yang akan dikompilasi sehingga memudahkan dalam melakukan kompilasi kernel linux.
4. Paket *fakeroot,* berfungsi untuk menyamarkan user yang akan digunakan, seolah olah menjadi *user root.*
5. Paket *wget* dan *bzip2,* aplikasi download manager yang ada pada sistem operasi linux, biasanya aplikasi ini telah menjadi paket

default dari linux. Sedangkan paket *bzip2* digunakan untuk mengekstrak atau menguraikan file terkompresi berbentuk *tar.gz2* dan sama seperti *wget,* paket ini sudah terinstal secara *default.*

Untuk menginstal dependensi dari paket-paket yang dibutuhkan diatas, jalankan perintah berikut dalam terminal dan ketik password untuk user, ketika diminta:

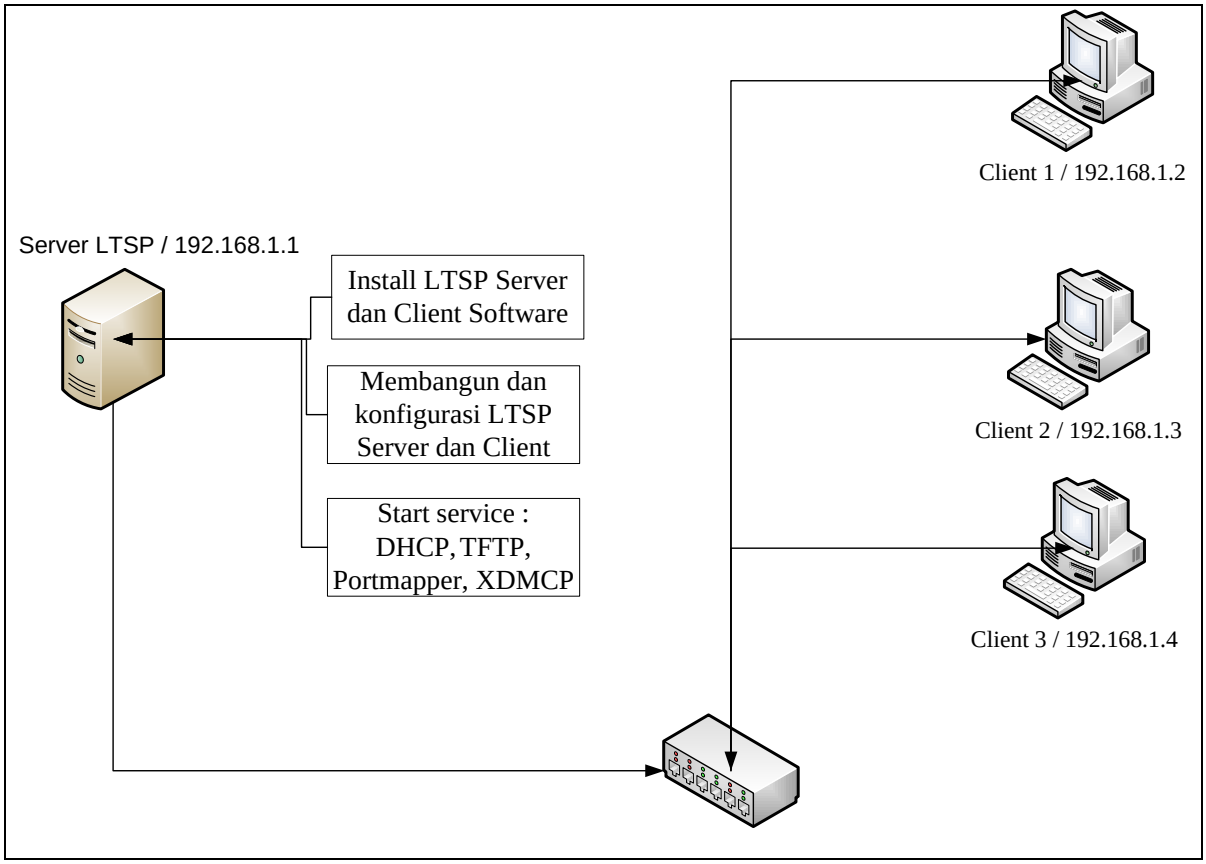
```
$ sudo apt-get install gcc

$ sudo apt-get install libncurses5-dev

$ sudo apt-get install wget bzip2
```

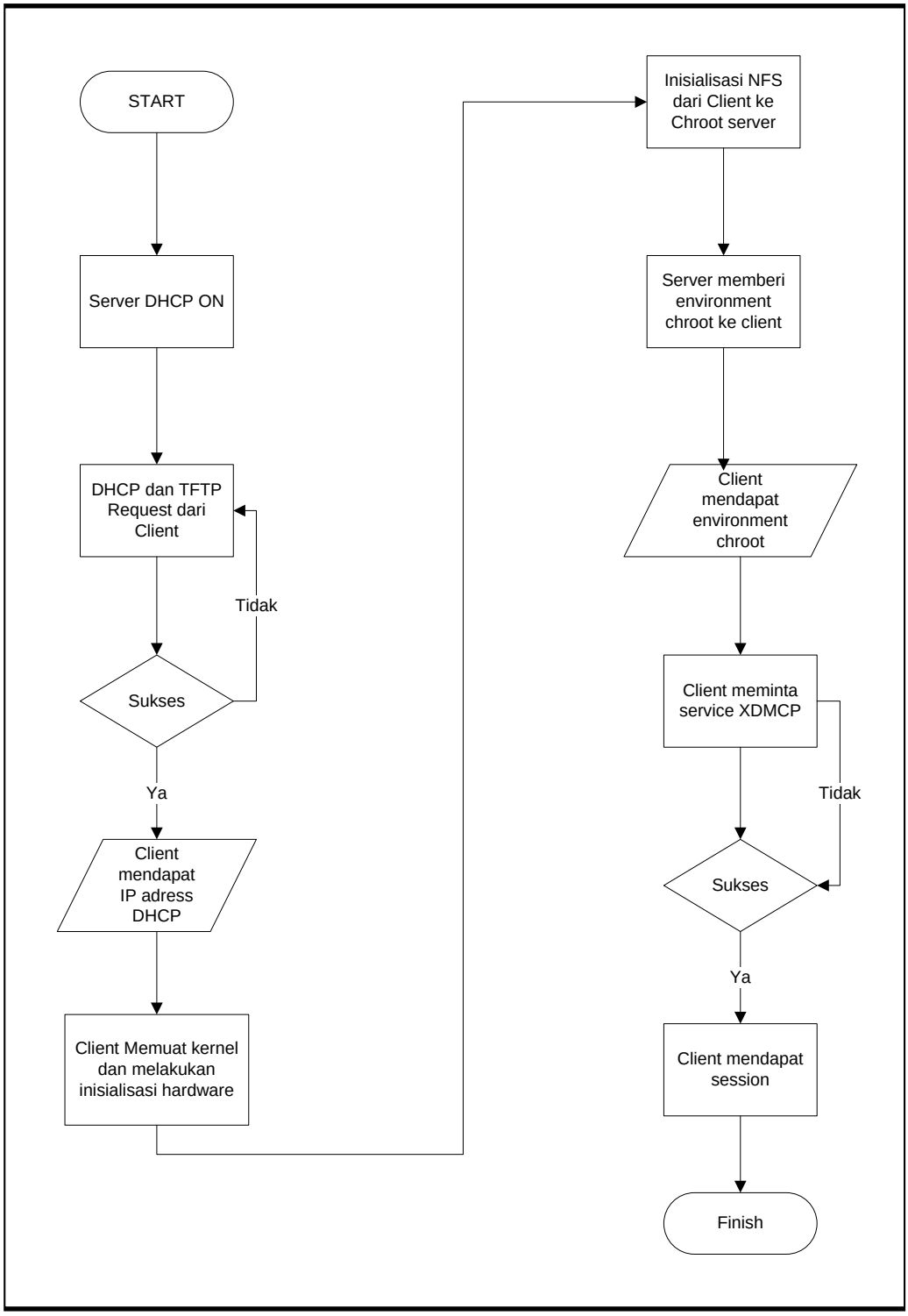
3.6.2 Perancangan Server Linux Terminal Server Project (LTSP)

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



Gambar 3.4 Rancangan sistem LTSP

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



Gambar 3.5 Flowchart Kerja LTSP

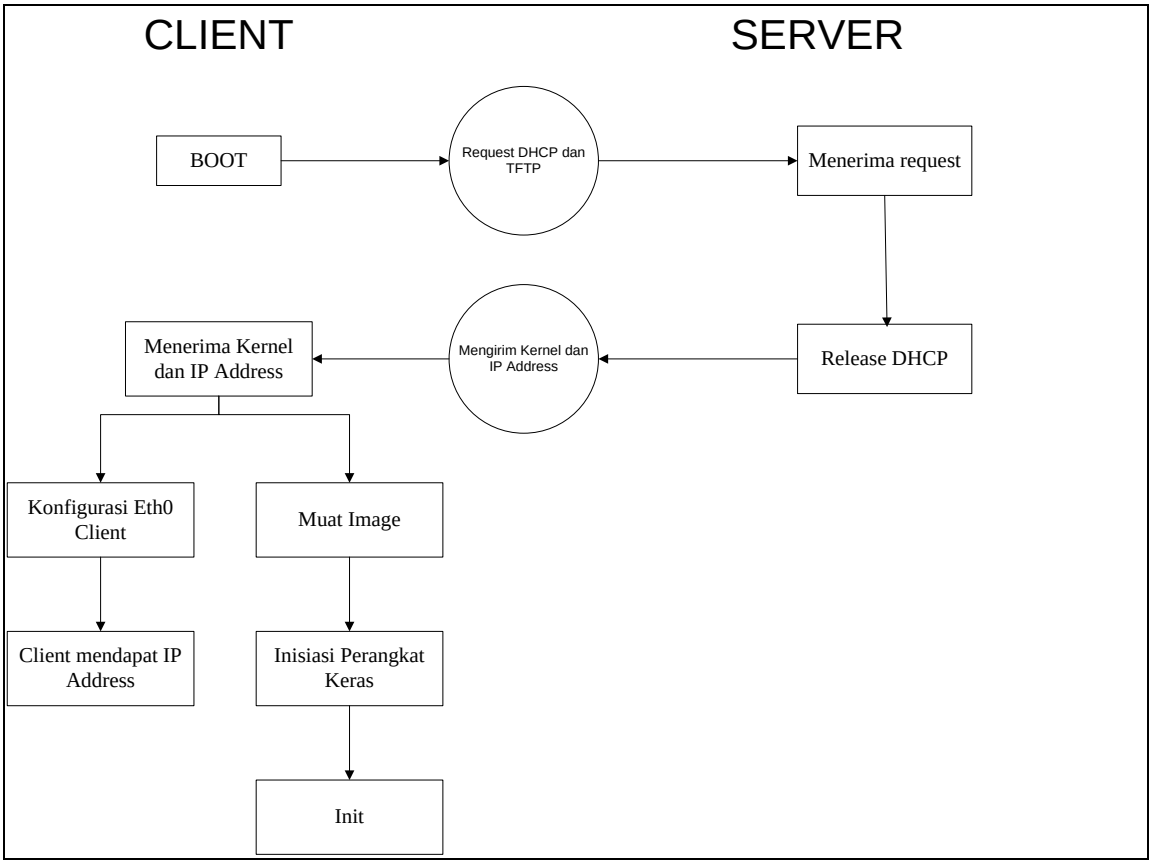
Penjelasan dari flowchart diatas adalah sebagai berikut :

1. Muat *kernel* linux ke dalam memori komputer kerja. Langkah ini dapat dilakukan dengan 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. 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 *client* DHCP kecil yang disebut **dh*client*** 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 **dh*client*** mendapat jawaban dari server ia akan menjalankan berkas **/etc/dh*client*−script**, yang akan mengambil informasi yang diperoleh, dan mengkonfigurasi antarmuka eth0.

   Gambaran proses dari langkah satu sampai dengan delapan dapat dilihat pada gambar berikut :



Gambar 3.6 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.

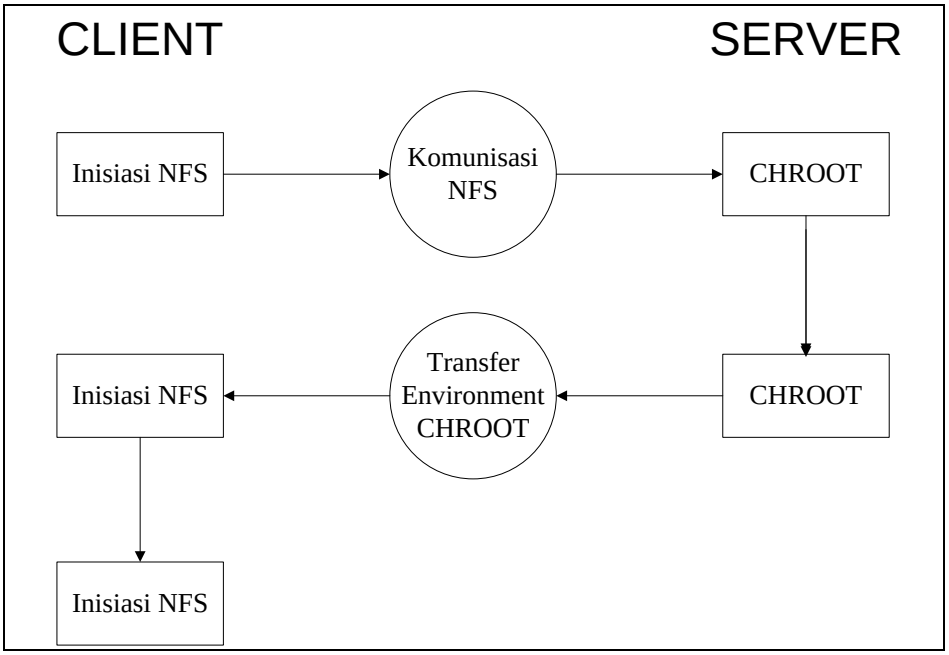
10. Setelah proses penambatan dan *pivot* pada sistem berkas *root* baru selesai, selesailah skrip *shell* /linuxrc kemudian dilanjutkan dengan memanggil program **/sbin/init** yang sebenarnya.
11. Init akan membaca berkas /etc/inittab dan mulai menyiapkan lingkungan komputer kerja.
12. Salah satu dari 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 proses dari langkah sembilan sampai dengan sembilan belas dapat dilihat pada gambar berikut :



Gambar 3.7 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

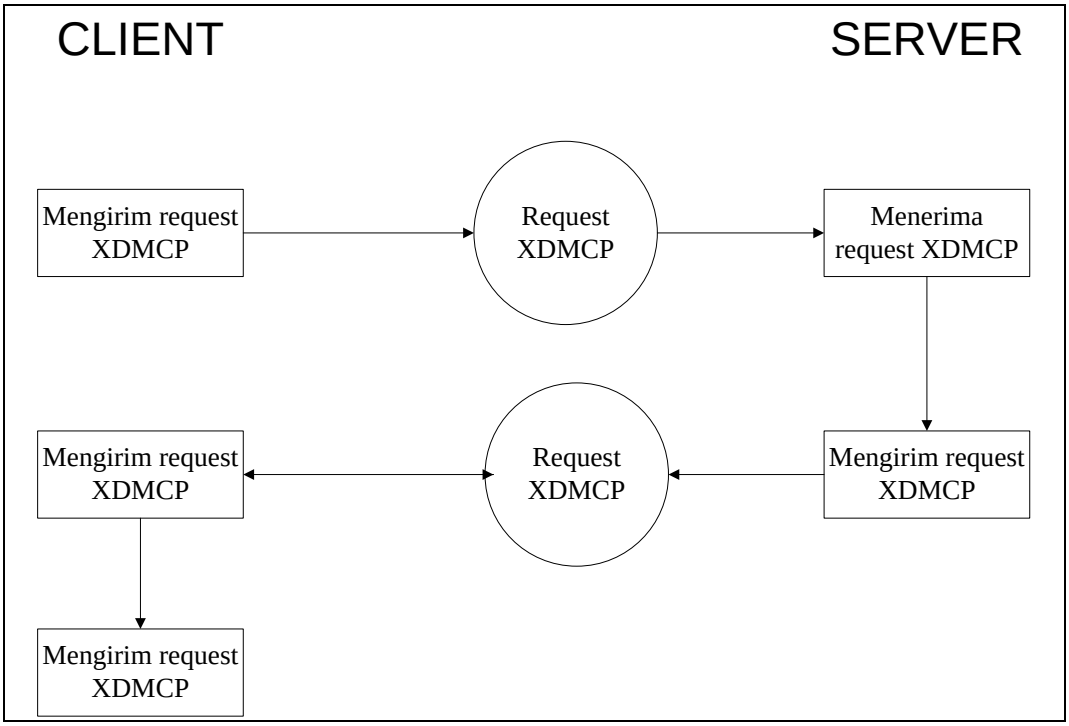
3.3.6 yang lama dimulai dengan 'XF86_', sedangkan modul server X dari Xorg yang lebih baru secara khusus merupakan nama−nama dengan huruf kecil, seperti *ati* atau *trident*.

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 proses dari langkah sembilan sampai dengan sembilan belas dapat dilihat pada gambar berikut :



Gambar 3.8 Proses XDMCP dan session

3.6.3 Perancangan optimasi sistem Linux Terminal Server Project (LTSP)

Untuk melakukan optimasi ataupun pengembangan pada system *linux terminal server project* (LTSP), dapat dilakukan dengan cara :

1. Modifikasi chroot

   Pada umumnya *Chroot* yang digunakan pada *client* Linux Terminal Server Project (LTSP) memiliki arsitektur dan distribusi linux 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 10.04, maka *chroot* yang digunakan menggunakan system dengan tipe arsitektur 32 bit (i386) dan dengan distribusi ubuntu 10.04. 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***. Dalam hal ini, dilakukan pemodifikasian pada system *chroot* yang digunakan. Hal tersebut berarti kita dapat menggunakan sistem arsitektur yang berbeda pada chroot. Selanjutnya akan dilakukan pembandingan kinerja sistem LTSP yang lebih efektif dan optimal dengan menggunakan *chroot* dengan arsitektur 32 bit (i386) .

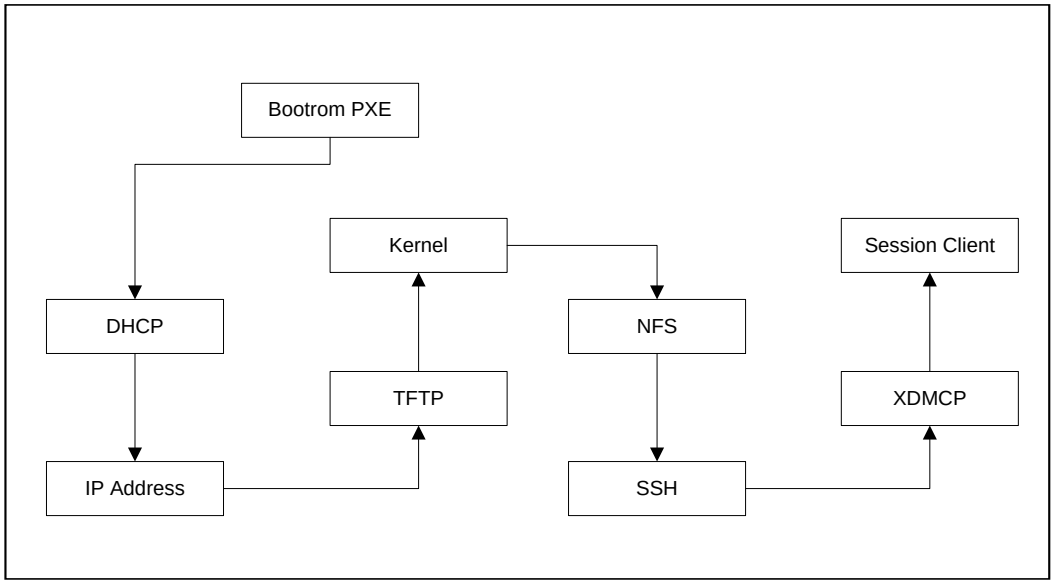
2. Pemasangan Local Applications

   Pada lingkungan LTSP, aplikasi dapat ditentukan agar berjalan pada local *workstation* dengan keuntungan sebagai berikut :

   1. Mengurangi beban load pada server. Pada jaringan yang besar dengan aplikasi yang secara *intensive* menggunakan banyak memori seperti Firefox, dengan menjalankan aplikasi yang sepenuhnya dijalankan pada *client* dapat memberikan hasil yang lebih baik bagi server LTSP.
   2. Sering kali dijumpai suatu aplikasi tetap berjalan padahal *client* yang bersangkutan sudah tidak dapat mengakses aplikasi tersebut (*runaway application),* pemasangan aplikasi pada *client* dilakukan agar tidak terlalu mengganggu kinerja server

3.6.4 Perancangan *Client* Linux Terminal Server Project (LTSP)

Client pada jaringan diskless melakukan proses booting melalui jaringan, menggunakan program yang disebut network boot loader. Dalam penelitian ini metode booting yang digunakan terbatas menggunakan PXE (Pre-boot Execution Environment). Tahapan proses *booting* pada *client* dapat dilihat pada gambar berikut :



Gambar 3.9 Proses *booting* pada *client* LTSP.

Berikut penjelasan dari proses *booting* pada *client* :

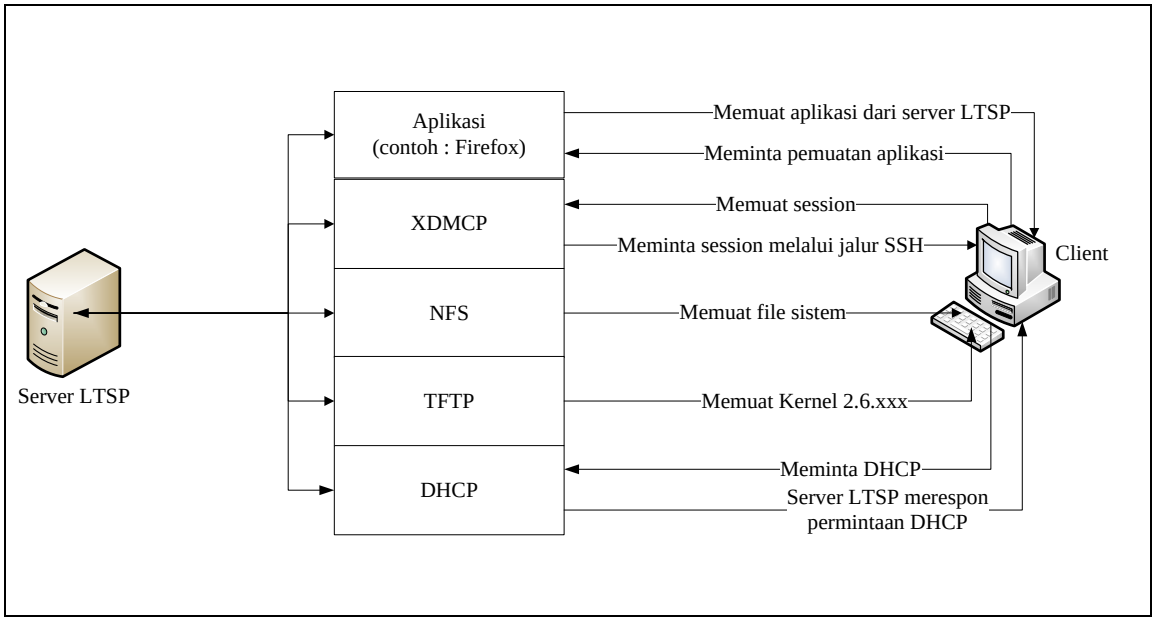
1. Pada saat komputer *client* melakukan booting melalui PXE, *client* akan merequest IP Address dari LTSP Server melalui protokol DHCP (server), DHCP akan membaca file konfigurasi dhcp.conf yang berada pada /etc/ltsp/dhcp.conf.
2. Setelah *client* mendapatkan IP address dari protocol DHCP, Server kemudian memuat Linux kernel dari *preconfigured Linux Image*

(pxelinux.0 atau nbi.img) kedalam BOOT Rom *client* dengan menggunakan protokol TFTP *(Trivial File Transfer Protokol*) yang telah berjalan pada *services* LTSP Server.

3. Pada saat bersamaan Server akan melakukan mount *filesystem* baru yang dilakukan oleh protokol NFS (Network File System) yang berada di /opt/ltsp/i386. *Client* tidak dapat menambatkan sistem berkas baru sebagai / (*root)*. Pertama−tama harus dimount sebagai /mnt. Kemudian, ia akan melakukan *pivot_root*, yang kemudian akan menukar sistem berkas *root* dengan sistem berkas baru. Ketika selesai, sistem berkas NFS akan di*mount* pada / *(root)*, dan sistem berkas *root* lama akan di*moun*t pada */oldroot*. dan me*load image* dari /opt/ltsp/images/i386.img.
4. *Client* akan membangun saluran SSH (Secure Shell tunnel) untuk berhubungan ke Server LTSP yang kemudian memulai XDMCP (*X Display Manager Control Protocol*) dan mendapatkan *session* dari LTSP.

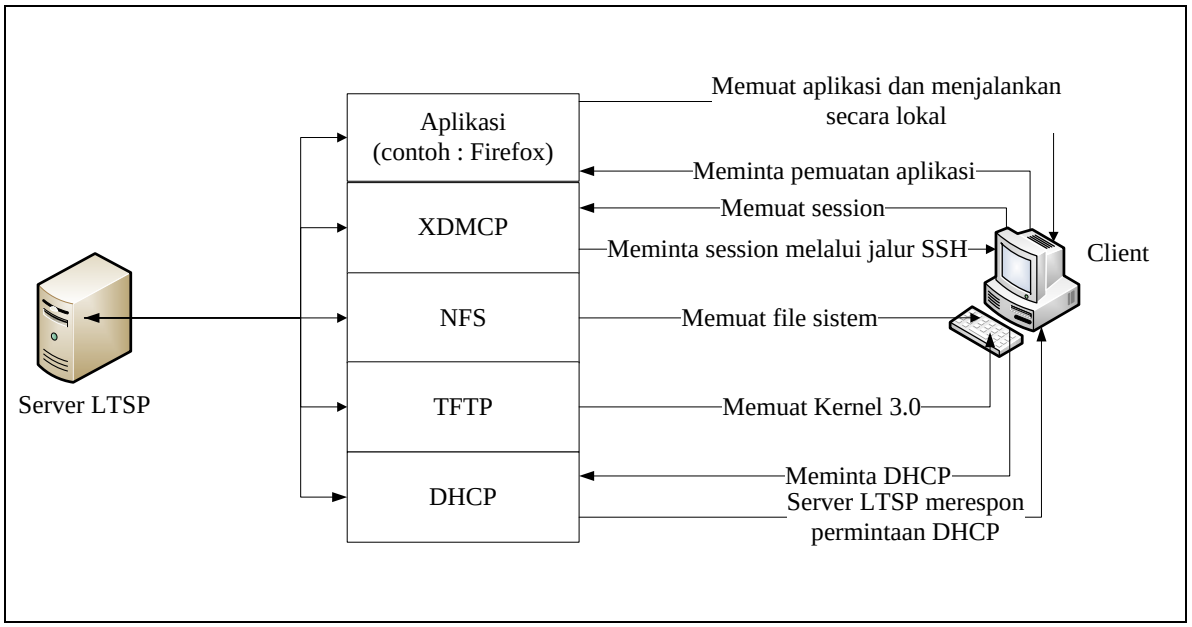
Pada sistem LTSP ini akan dilakukan pemodifikasian berupa pengintegrasian kernel, dan juga pemasangan *localapps*, berikut adalah proses kerja pada LTSP dalam keadaan *default* dan keadaan yang telah dioptimalisasi :

1. LTSP keadaan *default*, pada kondisi ini kernel masih menggunakan versi 2.6.xxx dan pemuatan aplikasi dibebankan pada sever.



Gambar 3.10 Proses kerja LTSP *default.*

2. LTSP yang telah dioptimalisasi, pada kondisi optimasi kernel yang digunakan adalah versi 3.0 dan pemuatan aplikasi dibebankan pada sever.



Gambar 3.11 Proses kerja LTSP optimasi.

Client dalam penelitian ini akan memiliki :

1. Prosesor : Minimal 533 MHz
2. RAM : 128 MB
3. Ethernet : Ethernet card dengan dukungan PXE boot
4. Internal subnet : 192.168.1.0
5. Ip range : 192.168.1.10 s/d 192.168.1.20
6. Subnet : 255.255.255.0