# ANALISIS DAN PERBANDINGAN PENGGUNAAN LTSP PADA JARINGAN KOMPUTASI KLIEN SERVER SEBAGAI *PRIVATE CLOUD* DENGAN JARINGAN KOMPUTASI KLIEN SERVER KONVENSIONAL

**Oleh:**

**Rizki Bagaswara Putra – 10513713**

**Mahasiswa Jurusan Sistem Informasi UNIKOM**

**ABSTRACT**

*This research makes client-server system with LTSP. LTSP offers solutions and alternatives for the network infrastructure of computing that are still using conventional methods. LTSP provides a solution in the shape of thin-client as the client so that the price to build this system is a lot more inexpensive. The intent of this research so that slowly the system/agency will change over to using the LTSP network infrastructure as the foundation for their computing infrastructure system.*

*The method of research conducted to establish this system is to do simulations and comparisons. Simulation of LTSP and then compare the results with the network client-server computing. On that point are four faces to get a benchmark comparison, namely: cost, time, care, and protection.*

*From the effects of the research, the authors found that the build system with LTSP method has advantages compared to using conventional methods. LTSP using thin-client as the client so that the price is a lot more inexpensive. LTSP uses only a single server, which means the treatment is done only on a single computer. Suggestion system development is how in the future this system is capable to handle fat-client and client-hybrid, and was able to completely replace the conventional client-server model.*

*Keywords: Linux, LTSP, Edubuntu, thin-client*

## I. PENDAHULUAN

### 1.1. Latar Belakang Penelitian

Teknologi akan selalu berkembang, tak terkecuali teknologi pada sistem informasi. Akan tetapi masyarakat Indonesia hanya mengenal satu metode yang digunakan untuk membangun jaringan komputasi klien server. Padahal banyak sekali metode lain yang bisa digunakan sebagai opsi alternatif selain metode klien server yang umum digunakan saat ini.

Pada tahun 1999[1], sebuah komunitas yang berfokus pada pengembangan Linux mengembangkan sebuah metode baru untuk membangun sistem *client-server* yang diberi nama *Linux Terminal Server Project* (LTSP). Tujuan dari LTSP ini adalah bagaimana membangun sebuah koneksi *client-server* yang dapat dinikmati banyak orang secara simultan dan dengan biaya yang murah atau bahkan secara cuma - cuma. Konsep dasar dari LTSP adalah membangun sebuah terminal server bagi Linux yang gratis dan *open source* yang mengizinkan banyak orang

menggunakan satu computer secara simultan. Konsep yang diberikan oleh LTSP memiliki kemiripan dengan konsep *client-server* konvensional. Akan tetapi, LTSP memberikan sebuah fitur berupa *thin-clien*, dimana *legacy* PC masih dapat digunakan sebagai klien sehingga biaya yang dikeluarkan menjadi lebih murah.

Meskipun sudah berumur 19 tahun, LTSP masih dapat dikatakan sebagai teknologi baru di Indonesia. Karena penerapannya masih sangat jarang di Indonesia. Padahal, LTSP menawarkan satu solusi utama yang tidak dimiliki oleh konsep *client-server* konvensional, yaitu LTSP menawarkan sebuah *private cloud* tanpa harus menggunakan koneksi internet. Pada konsep *client-server* konvensional, tidak terdapat fasilitas *private cloud* karena semua data klien tersimpan pada masing – masing *hard drive* PC klien. Sedangkan pada LTSP data klien tersimpan pada *hard drive* server, bahkan ketika user sedang melakukan sebuah pengetikan dokumen, user melakukan hal tersebut langsung menggunakan aplikasi yang sudah terinstall pada server, oleh karena itu LTSP juga dapat digunakan sebagai sebuah *private cloud*. Dilihat dari empat hal, dibandingkan dengan klien server konvensional LTSP mempunyai keuntungan lebih, yaitu dari sisi *cost*, *time*, *maintenance*, dan *security*.

Oleh karena itu, penulis membuat skripsi yang berjudul "**Analisis dan Perbandingan Penggunaan LTSP Pada Jaringan Komputasi Klien Server Sebagai *Private Cloud* dengan Jaringan Komputasi Klien Server Konvensional**". Penulis berharap dengan adanya skripsi ini, semakin meningkatkan kesadaran tentang adanya LTSP dan keuntungan – keuntungan yang ditawarkan sebagai solusi alternatif client-server konvensional. Sehingga akan semakin banyak instansi/organisasi yang mau menerapkannya.

## 1.2. Identifikasi dan Rumusan Masalah

Berdasarkan pada latar belakang, identifikasi dan rumusan masalah adalah sebagai berikut:

### 1.2.1. Identifikasi Masalah

Berdasarkan latar belakang, hasil identifikasi masalah adalah sebagai berikut:

1. LTSP merupakan suatu teknologi yang masih jarang dan belum familiar.
2. Masih sangat sedikit orang yang memahami soal LTSP.
3. Metode komputasi klien server konvensional memiliki kekurangan dari segi *cost*, *time*, *maintenance*, dan *security* dibandingkan dengan LTSP.

### 1.2.2. Rumusan Masalah

Berdasarkan latar belakang, hasil rumusan masalah adalah sebagai berikut:

1. Bagaimana merancang sebuah sistem *clien-server* menggunakan metode LTSP?
2. Bagaimana membuat sebuah *private cloud* menggunakan LTSP?
3. Bagaimana membuat sebuah *thin-client* sebagai klien di dalam sistem *client-server* LTSP?

## 1.3. Maksud dan Tujuan Penelitian

Maksud dan tujuan dari penelitian ini adalah sebagai berikut:

### 1.3.1. Maksud Penelitian

Penulis bermaksud untuk membuat sebuah koneksi client-server yang berbasis LTSP dan membandingkannya dengan koneksi client-server konvensional untuk memberikan sebuah hasil analisa mendalam mengenai LTSP yang mampu memberikan solusi sebagai alternatif pengganti client-server konvensional.

### 1.3.2. Tujuan Penelitian

Tujuan dari penulis melakukan penelitian ini adalah sebagai berikut:

1. Mampu merancang sebuah sistem *client-server* menggunakan metode LTSP.
2. Mampu membuat sebuah *private cloud* dengan menggunakan LTSP.
3. Mampu membuat sebuah *thin-clien* sebagai klien di dalam sistem *client-server* LTSP.

## 1.4. Kegunaan Penelitian

Kegunaan dari penelitian yang dilakukan penulis dapat dikategorikan kedalam dua kelompok, yaitu kegunaan praktis dan kegunaan akademis.

### 1.4.1. Kegunaan Praktis

Kegunaan praktis penelitian ini adalah sebagai berikut:

1. Bagi institusi/organisasi pendidikan
   Institusi/organisasi tidak perlu lagi mengeluarkan biaya besar untuk membeli atau upgrade PC. Karena dengan menggunakan PC yang memiliki arsitektur dan perangkat keras lebih tua yang sudah tidak mumpuni untuk menjalankan sistem operasi secara penuh dapat digunakan sebagai thin-client. Sehingga akan semakin banyak orang yang bisa memiliki akses computer.
2. Bagi institusi/organisasi komersial
   Institusi/organisasi tidak perlu lagi mengeluarkan biaya besar untuk membeli atau upgrade PC. Karena dengan menggunakan PC yang memiliki arsitektur dan perangkat keras lebih tua yang sudah tidak mumpuni untuk menjalankan sistem operasi secara penuh dapat digunakan sebagai thin-client. Hal ini dapat menambah capital cost dan mengurangi biaya operasional.

### 1.4.2. Kegunaan Akademis

Kegunaan akademis penelitian ini adalah sebagai berikut:

1. Bagi penulis, penelitian ini dapat menambah wawasan dan pengetahuan mengenai LTSP.
2. Bagi penulis lain, jika ingin menerapkan LTSP, penelitian ini dapat menjadi sebuah rujukan atau referensi. Hasil penelitian ini juga dapat menjadi sebuah rujukan atau referensi yang dapat digunakan di masa depan demi pengembangan lebih lanjut dari LTSP.

## 1.5. Batasan Masalah

Batasan masalah dalam skripsi ini adalah:

1. Sistem LTSP *thin-client* yang dibangun tidak meliputi *user session*.
2. Hanya menggunakan jenis *thin-client* sebagai klien.
3. Konfigurasi yang dilakukan hanya berlaku pada Edubuntu 14.04 LTS.

4. Sistem LTSP yang dibangun masih sebuah produk beta.
5. LTSP yang digunakan adalah LTSP 5

## II. LANDASAN TEORI

### 2.1. Linux

Linux adalah sebuah sistem operasi kernel yang dirilis pertama kali pada 5 Oktober 1991 oleh Linus Torvald.[2] Pada awal perkembangannya, Linux merupakan sebuah sistem operasi gratis untuk komputer pribadi yang berbasis pada arsitektur Intel x86, tetapi kemudian dikembangkan lebih lanjut sehingga Linux mampu mendukung lebih banyak arsitektur perangkat keras komputer dibanding dengan sistem operasi lain.[3] Dominasi Linux pada *smartphones*, Android, yang dibangun menggunakan Linux kernel, membuat Linux menjadi sistem operasi terbesar yang memiliki jumlah perangkat terinstalasi untuk kebutuhan umum.[4]

Linux telah lama dikenal untuk penggunaannya di server, dan didukung oleh perusahaan-perusahaan komputer ternama seperti Intel, Dell, Hewlett-Packard, IBM, Novell, Oracle Corporation, Red Hat, dan Sun Microsystems. Linux digunakan sebagai sistem operasi di berbagai macam jenis perangkat keras komputer, termasuk komputer desktop, superkomputer,[5] dan sistem benam seperti pembaca buku elektronik, sistem permainan video (PlayStation 2, PlayStation 3 dan Xbox[6][7]), telepon genggam dan router. Para pengamat teknologi informatika beranggapan kesuksesan Linux dikarenakan Linux tidak bergantung kepada vendor (*vendor independence*), biaya operasional yang rendah, dan kompatibilitas yang tinggi dibandingkan versi UNIX tak bebas, serta faktor keamanan dan kestabilannya yang tinggi dibandingkan dengan sistem operasi lainnya seperti Microsoft Windows. Ciri-ciri ini juga menjadi bukti atas keunggulan model pengembangan perangkat lunak sumber terbuka (*opensource software*).

Perkembangan Linux adalah salah satu contoh yang paling menonjol dari kolaborasi antara perangkat lunak bebas dan open source. Kode sumber yang mendasari dapat digunakan, dimodifikasi, dan didistribusikan-komersial atau non-komersial-oleh siapapun di bawah lisensi GNU General Public License. Biasanya, Linux dikemas dalam bentuk yang dikenal sebagai distribusi Linux, untuk penggunaan desktop dan server. Beberapa distro Linux utama yang populer meliputi Debian, Ubuntu, Linux Mint, Fedora, openSUSE, Arch Linux, dan distibusi komersial seperti Red Hat Enterprise Linux dan SUSE Linux Enterprise Server. Distribusi Linux termasuk kernel Linux, utilitas pendukung, dan *library* serta biasanya sejumlah besar aplikasi perangkat lunak untuk memenuhi tujuan distribusi.

Distribusi yang berorientasi ke arah penggunaan desktop biasanya termasuk X11, imlementasi Wayland atau Mir sebagai sistem *windowing*, dan lingkungan desktop yang menyertainya seperti GNOME atau KDE SC. Beberapa distribusi tersebut mungkin memiliki sumber daya intensif desktop yang kurang seperti LXDE atau Xfce, untuk digunakan pada komputer tua atau yang kurang kuat. Distribusi yang dimaksudkan untuk berjalan di server dapat menghilangkan semua lingkungan grafis dari instalasi standar, dan bukan termasuk perangkat lunak lain untuk mendirikan dan mengoperasikan solusi tumpukan seperti lampu. Karena Linux dapat

di distribusikan kembali secara bebas, siapa pun dapat membuat distribusi untuk setiap penggunaan.

## 2.2. Edubuntu

Edubuntu adalah salah satu distro Linux yang dirancang untuk dipergunakan dalam sekolah/ruang kelas. Sebelumnya, Edubuntu dikenal sebagai Ubuntu Education Edition yang memiliki arti ”pendidikan untuk semua orang”. Edubuntu merupakan varian dari Ubuntu, Edubuntu dibuat dengan menggunakan Ubuntu sebagai pondasi dasar. Salah satu fitur dari Edubuntu adalah *built-in* Linux Server Terminal Project (LTSP). Edubuntu memiliki banyak aplikasi edukasi termasuk GCompris, KDE Education Suite, Sabayon Profile Manager, Pessulus Lockdown Editor, Edubuntu Menueditor, LibreOffice, Gnome Nanny dan iTalc, dan masih banyak lagi. GUI standar dari Edubuntu adalah Unity disamping masih ada Gnome yang menjadi alternatif GUI.[8]

Tujuan utama Edubuntu adalah untuk memungkinkan pendidik dengan pengetahuan teknis yang terbatas dan keterampilan untuk mengatur laboratorium komputer atau lingkungan belajar on-line dalam satu jam atau kurang dan kemudian secara efektif mengelola lingkungan tersebut.

Desain utama Edubuntu adalah manajemen terpusat untuk konfigurasi, pengguna, dan proses, Fasilitas untuk bekerjasama dalam sebuah kelas. Sama pentingnya adalah perkumpulan untuk terbaik tersedia perangkat lunak bebas dan material digital untuk pendidikan. Menurut pernyataan tujuan di Website Edubuntu resmi: "tujuan kami adalah untuk mengumpulkan sebuah sistem yang berisi semua yang terbaik free software tersedia dalam pendidikan dan membuatnya mudah untuk menginstal dan memelihara."[9] Hal ini juga bertujuan untuk memungkinkan berpenghasilan rendah lingkungan untuk memaksimalkan peralatan (tua) mereka tersedia.

**Tabel 2.1 Tabel Sejarah Versi Edubuntu**

## 2.3. Linux Server Terminal Project (LTSP)

Linux Terminal Server proyek (LTSP) adalah sebuah server terminal yang bebas dan open source untuk Linux yang memungkinkan banyak orang untuk secara bersamaan menggunakan komputer yang sama. Aplikasi berjalan pada server dengan sebuah terminal dikenal sebagai *thin client* (juga dikenal sebagai X terminal) yang menangani input dan output. Umumnya, Terminal bertenaga rendah, tidak memiliki hard disk dan lebih tenang serta lebih dapat diandalkan daripada komputer desktop karena mereka tidak memiliki bagian yang bergerak.

Teknologi ini menjadi populer di sekolah karena memungkinkan sekolah untuk memberikan siswa akses ke komputer tanpa membeli atau *upgrade* mesin desktop. Peningkatan akses ke komputer menjadi lebih murah seperti *thin client* dapat berupa komputer lama yang tidak lagi cocok untuk menjalankan OS desktop secara penuh. Bahkan CPU yang relatif lambat dengan RAM sebesar hanya 128 MB dapat memberikan kinerja yang sangat baik sebagai *thin client*. Selain itu, penggunaan sumber daya komputasi yang terpusat berarti bahwa kinerja lebih dapat

diperoleh dengan biaya yang sedikit melalui upgrade ke sebuah server tunggal daripada di seluruh computer yang ada.

Dengan mengubah komputer yang ada menjadi *thin client*, sebuah lembaga pendidikan juga dapat memperoleh lebih banyak kontrol atas bagaimana siswa menggunakan sumber daya komputasi seperti semua sesi pengguna dapat dimonitor pada server.

Dalam bentuknya saat ini (v5.x), LTSP bergantung pada distribusi untuk mengintegrasikan arsitektur LTSP ke dalam produk mereka masing-masing. Di seri v4.x, LTSP adalah sebuah paket add-on untuk setiap distribusi. Beberapa distro mengintegrasikan LTSP kedalam sistem utama mereka (Ubuntu, Debian) atau sebagai produk terpisah, seperti Edubuntu (Ubuntu), K12LTSP (CentOS) dan Skolelinux (Debian), KIWI-LTSP (SUSE). LTSP adalah merek dagang terdaftar dari DisklessWorkstations.com, LLC.

Proyek pendiri dan Ketua LTSP adalah Jim McQuillan, dan LTSP didistribusikan dibawah ketentuan dari GNU General Public License.[1]

Proses booting pada klien LTSP adalah sebagai berikut:

1. Di LTSP server, lingkungan chroot diatur dengan operasi sistem Linux minimal dan lingkungan X.
2. Kedua komputer akan boot[13] dari perangkat boot lokal (seperti harddisk, disk CD-ROM atau USB), dimana beban kernel Linux kecil dari perangkat yang menginisialisasi sistem dan semua peripheral yang sudah mengenali.
3. Atau, *thin client* akan menggunakan PXE atau *boot* melalui jaringan, Bagian dari on-board Ethernet firmware, untuk meminta IP alamat dan boot server (LTSP server) menggunakan protokol DHCP. PXE boot loader (PXElinux) yang dimuat, yang kemudian mengambil Linux kernel dan initrd dari layanan *Trivial File Transfer Protocol* (TFTP) yang biasanya berjalan di LTSP server. Menggunakan utilitas initrd, kernel akan meminta alamat DHCP IP (baru) dan alamat server dimana itu dapat me-*mount filesystem root* yang (chroot yang disebutkan di atas). Ketika informasi ini diperoleh, klien Mount jalan pada filesystem root yang melalui layanan *Network File System* (NFS) atau *Network Block Device* (NBD) yang berjalan di LTSP server.
4. Klien kemudian memuat Linux dari root filesystem NFS yang telah terpasang (atau citra filesistem NBD) dan memulai sistem Window X. Pada XDMCP login manajer di LTSP server. Dalam kasus MueKow terbaru (LTSP v5.x), klien pertama membangun SSH *tunnel* menuju LTSP server lingkungan X, yang akan memulai LDM (LTSP *Display Manager*) login manajer secara lokal. Dari titik ini ke depan, Semua program yang dimulai pada LTSP server, tapi ditampilkan dan dioperasikan dari klien.

**Tabel 2.2 Perbedaan Antara LTSP 4 dan 5[14]**

## 2.4. Thin Client

*Thin client* (kadang-kadang juga disebut *lean*, *zero* atau klien ramping) adalah sebuah komputer atau sebuah program komputer yang sangat bergantung pada komputer lain (server) untuk memenuhi peran komputasinya. Hal ini berbeda dari

klien tradisional, yang merupakan sebuah komputer yang dirancang untuk mengambil peran ini dengan sendirinya. Peran tertentu diasumsikan oleh server dapat bervariasi, mulai dari menyediakan kestabilan data (misalnya, untuk node *diskless*) untuk informasi aktual pengolahan atas nama klien.

*Thin client* terjadi sebagai komponen infrastruktur komputer yang lebih luas, di mana banyak klien berbagi perhitungan mereka dengan server yang sama. Dengan demikian, infrastruktur *thin-client* dapat dilihat sebagai penyedia beberapa layanan komputasi melalui beberapa user interface. Hal ini diinginkan dalam konteks yang mana *fat-client* memiliki lebih banyak fungsi atau kuasa dibanding yang dibutuhkan oleh infrastruktur. Komputasi *thin-client* juga merupakan cara mudah mempertahankan layanan komputasi dengan mengurangi biaya total kepemilikan.[15]

Jenis yang paling umum dari *thin client* modern adalah low-end terminal komputer yang hanya menyediakan antarmuka pengguna grafis – atau lebih baru-baru ini, dalam beberapa kasus, browser web-untuk pengguna akhir.

Server, dalam mengambil secara keseluruhan pengolahan beban dari beberapa klien, membentuk titik tunggal kegagalan untuk klien mereka. Ini memiliki aspek positif dan negatif. Di satu sisi, model ancaman keamanan perangkat lunak menjadi lebih terfokus pada server. Klien tidak menjalankan perangkat lunak; oleh karena itu, hanya sejumlah kecil komputer (server) harus diamankan di tingkat perangkat lunak, daripada mengamankan perangkat lunak yang diinstal pada setiap komputer klien tunggal (meskipun komputer klien mungkin masih memerlukan keamanan fisik dan otentikasi kuat, untuk mencegah akses yang tidak sah, tergantung pada persyaratan). Di sisi lain, setiap denial of service serangan terhadap server akan membatasi akses banyak klien. Perangkat lunak server biasanya ditulis dengan teknologi mesin virtual sehingga setiap klien terisolasi dan kecelakaan klien mudah ditangani dan reboot. Titik tunggal kegagalan bisa tetap eksis, namun. Jika server crash, data rugi mungkin.

Untuk jaringan kecil, ini titik tunggal kegagalan properti mungkin akan diperluas. Hosting server dapat diintegrasikan dengan server file dan print server relevan untuk kliennya. Ini dapat menyederhanakan jaringan dan pemeliharaannya, tetapi mungkin meningkatkan risiko terhadap server.

Dalam prakteknya, redundansi dapat disediakan baik dalam bentuk tambahan konektivitas dari server ke jaringan maupun di server sendiri, menggunakan fitur seperti RAID, didistribusikan server (beberapa jaringan server muncul sebagai satu server ke pengguna), berkerumun filesistem (yang memungkinkan file yang dapat diakses dari beberapa server), VMWare ketersediaan yang tinggi dan toleransi kesalahan atau Citrix XenApp *load balancing*.

Sementara server harus cukup kuat untuk menangani beberapa sesi klien sekaligus, klien dapat dikumpulkan dari hardware yang jauh lebih murah daripada *fat* klien. Banyak klien memiliki RAM minimal, beberapa bahkan tidak memiliki *hard drive*. Ini mengurangi konsumsi daya klien, dan membuat sistem *marginal scalable*, yaitu harganya relatif murah untuk menghubungkan terminal klien tambahan. *Thin client* biasanya memiliki biaya total kepemilikan sangat rendah ,

tetapi kebutuhan akan infrastruktur server kuat mengimbangi penghematan biaya. *Thin client* juga umumnya menggunakan daya yang sangat rendah dan mungkin bahkan tidak memerlukan kipas pendingin, tapi server mengkonsumsi daya tinggi dan hampir selalu membutuhkan ruangan server denga lingkungan terkendali ber-AC.

Karena klien terbuat dari perangkat keras biaya rendah dengan beberapa bagian yang bergerak, mereka dapat beroperasi dalam lingkungan yang lebih ekstrem daripada komputer konvensional. Namun, mereka pasti memerlukan sambungan jaringan ke server mereka, yang harus diisolasi dari lingkungan yang ekstrem. Karena *thin client* murah, mereka menawarkan risiko rendah terhadap pencurian secara umum, dan mudah untuk mengganti jika dicuri atau rusak. Karena mereka tidak memiliki *boot* rumit, masalah *boot* dikontrol secara terpusat di server.

Di sisi lain, untuk mencapai kesederhanaan ini, *thin client* kadang-kadang tertinggal di belakang *thick client* (PC desktop) dalam hal ekstensi. Sebagai contoh, jika sebuah utilitas perangkat lunak lokal atau set *device driver* yang diperlukan untuk mendukung perangkat periperal lokal terlampir (misalnya printer, scanner, perangkat keamanan biometrik), sistem operasi *thin client* mungkin kekurangan sumber daya yang dibutuhkan untuk sepenuhnya mengintegrasikan dependensi yang diperlukan. *Thin client* modern berusaha untuk mengatasi keterbatasan ini, melalui pemetaan port atau USB perangkat lunak pengalihan. Namun, metode ini tidak dapat mengatasi semua skenario kasus penggunaan untuk sejumlah besar jenis perifer yang dimasukkan menggunakan hari ini.

**Gambar 2.1 Bentuk Fisik Thin Client**

## 2.5. Jaringan klien server

Model komputasi klien-server adalah sebuah struktur aplikasi terdistibusi yang membagi – bagi tugas atau beban kerja antara penyedia sumber daya atau layanan, disebut server, dan pemohon layanan, disebut klien.[17] Klien dan server berkomunikasi melalui jaringan komputer pada hardware yang terpisah, tetapi baik server maupun klien bisa berada dalam sistem yang sama. Sebuah *host* server menjalankan satu atau lebih program server yang berbagi sumber daya dengan klien. Klien tidak berbagi sumber daya, tetapi meminta konten atau fungsi layanan dari server. Oleh karena itu, klien memulai sesi komunikasi dengan server yang menunggu permintaan masuk. Contoh aplikasi komputer yang digunakan klien-server model adalah Email, Jaringan pencetakan, dan World Wide Web.

Sementara merumuskan model klien-server di tahun 1960-an dan 1970-an, ilmuwan komputer di Xerox dan Xerox PARC menggunakan istilah *server-host* (atau *serving host*) dan *user-host* (atau *using-host*).[18][19] Satu konteks di mana para peneliti menggunakan istilah-istilah ini pada desain bahasa perograman jaringan komputer yang disebut *Decode-Encode Language* (DEL).[18] tujuan bahasa ini adalah untuk menerima perintah dari satu komputer (*user-host*), yang akan mengembalikan laporan status kepada pengguna dengan mengkodekan perintah di paket jaringan. Komputer berkemampuan DEL lain, *server-host*, menerima paket-paket, menterjemahkan mereka, dan mengembalikan data terformat kepada *user-host*. Program DEL pada user-host menerima hasil untuk menunjukkan kepada

pengguna. Ini adalah transaksi klien-server. Pengembangan DEL baru mulai pada tahun 1969, dimana Departemen Pertahanan Amerika Serikat mendirikan ARPANET (pendahulu internet).

*Client-host* dan *server-host* memiliki arti yang berbeda dari klien dan server. Sebuah host adalah setiap komputer yang terhubung ke jaringan. Sedangkan kata server dan klien dapat merujuk ke komputer atau program komputer, *server-host* dan *user-host* selalu mengacu pada komputer. Host adalah komputer serbaguna, multifungsi; klien dan server adalah program yang hanya berjalan pada *host*. Dalam model klien-server, server ini lebih mungkin untuk lebih fokus kepada tugas melayani.

Penggunaan awal kata klien terjadi pada "*Separating Data from Function in a Distributed File System*", sebuah makalah tahun 1978 oleh ilmuwan komputer Xerox PARC Howard Sturgis, James Mitchell dan Jay Israel. Penulis berhati-hati dalam mendefinisikan istilah untuk pembaca, dan menjelaskan bahwa mereka menggunakannya untuk membedakan antara pengguna dan pengguna jaringan node (klien).[20] (pada tahun 1992, kata server telah masuk ke dalam istilah umum.)[21][22]

## 2.6. VirtualBox

VM Oracle VirtualBox (dahulu Sun VirtualBox, Sun xVM VirtualBox dan Innotek VirtualBox) adalah sebuah hypervisor untuk komputer x86 dari Oracle Corporation. Innotek GmbH pertama mengembangkan produk sebelum di akuisisi oleh Sun Microsystems dalam 2008. Oracle telah melanjutkan pengembangan sejak 2010.

VirtualBox dapat diinstal pada sistem operasi *host* yang sudah ada; VirtualBox bisa membuat dan mengelola mesin virtual *guest*, masing-masing dengan sistem operasi *guest* dan lingkungan virtual sendiri. Sistem operasi *host* yang didukung termasuk Linux, OS X, Windows XP, Solaris, dan OpenSolaris; Ada juga *port* untuk FreeBSD[23] dan Genode.[24] Sistem operasi *guest* yang didukung termasuk versi dan turunan dari Windows, Linux, BSD, OS/2, Solaris, Haiku dan lain-lain.[25] Sejak rilis 3.2.0, VirtualBox juga memungkinkan virtualisasi terbatas dari *guest* OS X pada perangkat keras Apple, meskipun OSx86 juga dapat diinstal menggunakan VirtualBox.[26][27]

*Guest Additions* harus diinstal pada sistem operasi guest untuk mendapatkan pengalaman terbaik.[28] *Guest Additions* terdiri dari *device driver* dan aplikasi sistem yang mengoptimalkan sistem operasi *guest* untuk kinerja dan kegunaan yang lebih baik.[29] Sejak versi 4.3 (dirilis pada Oktober 2013[30]), Windows *guest* pada perangkat keras yang didukung dapat mengambil keuntungan dari *driver* WDDM yang termasuk dalam paket *Guest Additions*; Hal ini memungkinkan untuk mengaktifkan Windows Aero bersama dukungan Direct3D.

VirtualBox awalnya ditawarkan oleh Innotek GmbH dari Weinstadt, Jerman dibawah lisensi perangkat lunak berpemilik, membuat satu versi produk tersedia tanpa biaya, untuk penggunaan pribadi atau evaluasi, VirtualBox *Personal Use and Evaluation License* (PUEL).[31] Pada bulan Januari 2007, berdasarkan nasihat oleh LiSoG, Innotek GmbH merilis VirtualBox *Open Source Edition* (OSE) sebagai

perangkat lunak bebas dan open source, persyaratan dari GNU General Public License (GPL), versi 2.[32]

Innotek GmbH juga memberikan kontribusi untuk pengembangan dukungan OS/2 dan Linux dalam virtualisasi[33] dan OS/2 Port[34] produk dari Connectix yang kemudian diakuisisi oleh Microsoft. Secara khusus, Innotek mengembangkan kode "upload" dalam Microsoft Virtual PC dan Microsoft Virtual Server, yang memungkinkan berbagai *host-guest* OS berinteraksi seperti clipboard berbagi atau ukuran viewport dinamis. Sun Microsystems mengakuisisi Innotek pada Februari 2008.[35][36][37] Oracle Corporation mengakuisisi Sun pada bulan Januari 2010 dan mengubah nama produk mejadi "Oracle VM VirtualBox".[38][39][40]

Dengan versi 4 VirtualBox, dirilis pada bulan Desember 2010, paket inti adalah perangkat lunak gratis yang dirilis di bawah GNU *General Public License* versi 2 (GPLv2). Ini adalah paket fitur penuh, termasuk beberapa komponen proprietary yang tidak tersedia di bawah GPLv2. Komponen ini menyediakan dukungan untuk perangkat USB 2.0, *Remote Desktop Protocol* (RDP) dan *Preboot Execution Environment* (PXE) dan untuk kartu Intel dirilis terpisah "VirtualBox Oracle VM VirtualBox *extension pack*" dibawah kepmilikan *Personal Use and Evaluation License* (PUEL), yang memungkinkan penggunaan perangkat lunak untuk penggunaan pribadi, penggunaan pendidikan atau evaluasi, secara gratis.[31]

Oracle mendefinisikan penggunaan pribadi sebagai situasi di mana satu orang menginstal perangkat lunak, dan hanya individu, dan teman-teman serta keluarga, yang menggunakan perangkat lunak. Oracle tidak peduli apakah menggunakannya untuk tujuan komersial atau non-komersial.[41] Oracle akan mempertimbangkan penggunaan non-pribadi, misalnya, jika administrator jaringan melakukan istalalasi dengan banyak salinan dari perangkat lunak dan diinstal pada banyak mesin yang berbeda, atas banyak pengguna yang berbeda. Jenis situasi akan memerlukan pembelian lisensi khusus.

Sebelum versi 4, ada dua paket perangkat lunak VirtualBox. Paket lengkap ditawarkan gratis di bawah PUEL, dengan lisensi untuk penggunaan komersial lainnya dibeli dari Oracle. Sebuah paket kedua yang disebut VirtualBox *Open Source Edition* (OSE) dirilis di bawah GPLv2. Ini menghilangkan komponen proprietary yang sama yang tidak tersedia di bawah GPLv2.[41][42] VirtualBox memerlukan penggunaan kompilator Open Watcom untuk membangun BIOS sejak versi 4.2.[43] Meskipun VirtualBox memiliki dukungan eksperimental untuk *guest* Mac OS X, perjanjian lisensi pengguna akhir dari Mac OS X tidak mengizinkan sistem operasi untuk dijalankan pada perangkat keras non-Apple, ditegakkan dalam sistem operasi dengan panggilan Apple *System Management Controller* (SMC) di semua mesin Apple untuk memverifikasi keaslian perangkat keras.[44]

## III. METODE PENELITIAN

### 3.1. Kerangka Penelitian

Penulis membagi langkah kerja ke dalam 3 tahap besar dan 5 tahap kecil. 3 tahap besar tersebut adalah tahap pra-analisis, tahap proses pembuatan, dan tahap

simulasi. Sedangkan 5 tahap kecil adalah tahap analisa kebutuhan sistem, tahap memilih distro yang tepat, tahap instalasi distro, tahap konfigurasi server, dan tahap *booting thin-client*. Berikut adalah gambaran flowchart ranah kerja yang penulis gunakan dalam menyusun skripsi ini.

**Gambar 3.1 Langkah Kerja Pembuatan LTSP *Thin-client***

### 3.1.1. Pra-analisis

Pada tahap ini dilakukan analisis proses bisnis yang terjadi, kebutuhan yang harus mampu dipenuhi oleh server. Diharapkan server mampu menangani lebih dari 10 *thin-client*, mampu melakukan *monitoring* terhadap klien, satu username untuk satu klien, dan terdapat insfrastruktur yang mampu menunjang kinerja server tersebut serta sistem operasi yang menyediakan fasilitas LTSP *thin-client*.

#### 3.1.1.1. Analisa Kebutuhan Sistem

Sistem harus mampu menangani lebih dari 10 klien, sistem juga harus mampu melakukan *monitoring* pada setiap klien yang terkoneksi pada sistem. Selain itu sistem juga harus mampu membuat satu username untuk satu klien agar keamanan data klien terjaga dengan baik. Kebutuhan perangkat keras server sangat tergantung oleh jumlah komputer klien yang akan terkoneksi dengan server. Semakin banyak komputer klien yang terkoneksi, maka akan semakin tinggi spesifikasi perangkat keras server. Selain itu kebutuhan akan sistem operasi yang akan digunakan pada sistem juga harus memiliki fasilitas LTSP dan mampu menunjang kebutuhan sistem.

#### 3.1.1.2. Memilih Distro yang sesuai

Terdapat banyak sekali macam distro Linux (http://en.wikipedia.org/wiki/List_of_Linux_distributions) beberapa distro populer Linux, yaitu: Ubuntu, Edubuntu, Fedora, Debian. Pada skripsi ini, penulis memilih distro Edubuntu karena pada Edubuntu LTSP telah tersedia secara *built-in*. Dengan adanya fasilitas LTSP secara built-in membuat konfigurasi server menjadi lebih cepat karena sudah tidak perlu melakukan konfigurasi jaringan, DNS server, jumlah klien yang mampu ditangani secara manual yang tersisa hanyalah bagaimana membuat konfigurasi sesuai dengan kebutuhan sistem yang diperlukan, keamanan data klien yang tersimpan pada server, serta keamanan jaringan yang digunakan untuk koneksi server dan klien.

### 3.1.2. Proses Pembuatan

Tahap ini merupakan tahap inti terdiri dari tiga proses yaitu proses instalasi, proses konfigurasi, dan proses *booting* thin-client. Pada tahap ini mulai dilakukan perancangan sistem sesuai dengan hasil analisa kebutuhan sistem. Dimulai dari perakitan perangkat lunak pada server yang akan digunakan, instalasi distro linux yang mampu memenuhi kebutuhan sistem, proses konfigurasi pada server agar mampu menangani klien sampai pada *booting thin-client*.

#### 3.1.2.1. Instalasi Distro

Pada tahap ini dilakukan proses instalasi LTSP distro Edubuntu. Edubuntu memiliki fasilitas LTSP yang telah built-in sehingga lebih mudah di

implementasikan. Selain itu, edubuntu sudah mampu memenuhi hasil analisa kebutuhan sistem yang diperlukan. Edubuntu yang digunakan juga telah memiliki fasilitas LTS (*Long-Term Support*) yang artinya edubuntu memiliki dukungan jangka panjang yang terdiri dari s*oftware update*, *kernel update*, *security update*, dan *bug removal*.

#### 3.1.2.2. Konfigurasi Server

Pada tahap ini dilakukan konfigurasi LTSP sesuai dengan yang dibutuhkan oleh sistem dan juga proses manajemen thin-client. Server harus mampu menangani lebih dari 10 klien dan juga mampu melakukan *monitoring* pada setiap klien yang terkoneksi pada server. Selain itu, server juga harus mampu menjamin keamanan data klien yang tersimpan pada server serta konfigurasi juga dilakukan untuk mengamankan server itu sendiri.

#### 3.1.2.3. *Booting* thin-client

Pada tahap ini dilakukan proses uji coba booting pada thin-client apakah klien sudah mampu booting melalui network atau belum. Klien harus mampu booting melalui jaringan dan terkoneksi pada server. Klien juga harus mampu login pada server dengan menggunakan username yang telah disediakan pada server. Ketika telah terkoneksi pada server, klien harus mampu menjalankan pekerjaan-pekerjaan yang biasa dilakukan secara normal seperti pengetikan dokumen, *browsing* internet, mengirim e-mail, aktivitas multimedia, dll.

### 3.1.3. Simulasi

Tahap ini merupakan proses penerapan langkah-langkah membangun LTSP. Setelah malakukan analisa secara keseluruhan mengenai gambaran sistem seperti apa yang ingin dibangun, maka pada tahap ini dilakukan proses pembangunan sistem tersebut. Dimulai dari tahap instalasi distro edubuntu, konfigurasi server, dan *booting thin-client*. Hasil akhir dari simulasi adalah sebuah kesimpulan mengenai apakah sudah sesuai sistem yang telah dibuat dengan sistem yang diinginkan sesuai dengan analisa kebutuhan sistem.

## IV. HASIL DAN PEMBAHASAN

### 4.1. Hasil Analisa dan Hasil Perbandingan

Berdasarkan hasil simulasi penulis membuat sebuah hasil analisa dan perbandingan dari segi cost, time, maintenance, security.

### 4.1.1. Cost

Dilihat dari sisi biaya, maka LTSP menjadi lebih murah dibanding dengan klien server konvensional. Karena *thin client* merupakan sebuah PC *diskless* yang artinya tidak membutuhkan *hard drive* sendiri sebagai tempat penyimpanan data, akan tetapi menggunakan hard drive pada server sebagai tempat penyimpanan data. Hal ini membuat LTSP menjadi jauh lebih berbeda dibanding koneksi klien server konvensional, sebab pada LTSP server juga berperan sebagai private cloud yang juga menyediakan tempat penyimpanan data bagi klien.

Berikut adalah tabel perbandingan total biaya yang harus dikeluarkan jika ingin membangun sebuah sistem klien server dengan menggunakan metode konvensional dan dengan menggunakan metode LTSP.

**Tabel 4.1 Perbandingan Total Biaya**

Spesifikasi dari perangkat keras yang digunakan adalah:

1. IBM System X3500M5-15A: Xeon E5-2620v3, 1x16GB PC4-17000 2133Mhz, 300GB Hot Swap SAS 2.5" G3HS HDD
2. ASUS Desktop K31AD-ID002D: Intel Core i3-4160, 2GB DDR3, 500GB HDD
3. NCOMPUTING L130: Virtual desktop, NIC, Audio, PS/2

Dari Tabel 4.1 dapat disimpulkan bahwa dengan menggunakan metode LTSP jumlah biaya yang dibutuhkan untuk membangun sebuah jaringan klien server lebih rendah daripada menggunakan metode klien server konvensional.

### 4.1.2. Time

**Tabel 4.2 Perbandingan Waktu Instalasi**

Dari Tabel 4.2 dapat disimpulkan bahwa proses instalasi sistem klien server menggunakan LTSP jauh lebih cepat. Karena segala proses baik itu proses instalasi sistem operasi, proses konfigurasi jaringan, proses instalasi aplikasi, proses *update*, dan proses *setup* firewall hanya dilakukan pada server, sehingga proses yang dilakukan hanya satu kali. Sedangakan untuk instalasi sistem klien server menggunakan metode konvensional, proses instalasi sistem operasi, proses konfigurasi jaringan, proses instalasi aplikasi, proses *update*, dan proses *setup* firewall dilakukan baik pada server maupun klien, yang artinya melakukan dua kali proses kerja yang membutuhkan waktu lebih lama.

### 4.1.3. Maintenance

**Tabel 4.3 Perbandingan Proses *Maintenance***

Dari Tabel 4.3 dapat disimpulkan bahwa jaringan klien server menggunakan LTSP membutuhkan sedikit perawatan dibandingkan dengan jaringan klien server menggunakan metode konvensional. Hal ini berdampak langsung pada biaya yang dikeluarkan pada proses perawatan sistem. Dengan sedikitnya proses perawatan yang dibutuhkan oleh LTSP, maka biaya yang dikeluarkan pun akan lebih rendah dibanding dengan jaringan klien server menggunakan metode konvensional, sebab metode konvensional membutuhkan lebih banyak perawatan dimana perawatan tersebut dibutuhkan baik oleh server maupun klien.

### 4.1.4. Security

**Gambar 4.24 *Thin clients* mampu mencegah pencurian data fisik dari klien; akan tetapi mereka tidak mencegah pencurian data melalui jaringan luar**

Jika dilihat dari segi keamanan data, LTSP menawarkan keamanan data yang lebih baik dibanding dengan klien server konvensional. Karena pada klien server konvensional klien diharuskan memiliki sistem operasi sendiri sehingga

kemungkinan terjadinya infeksi malware sangat tinggi. Sedangkan pada LTSP klien menggunakan sistem operasi yang terpusat pada server sehingga kemungkinan terjadinya infeksi malware sangat rendah. Selain itu, pada LTSP data tersimpan langsung pada server, sehingga hanya orang-orang tertentu yang memiliki hak akses terhadap data itu sendiri yang mampu mengaksesnya, sehingga kerahasiaan dan keamanan data pada sistem yang menggunakan LTSP dapat lebih terjaga dan lebih aman.

Akan tetapi, pada kasus tertentu LTSP masih belum mampu menangani pengamanan data dari pencuri data yang beroperasi melalui jaringan. Karena pencuri data mencuri langsung dari jaringan yang menghubungkan dirinya dengan server. Oleh karena itu, sistem keamanan yang digunakan pada server harus memiliki kredibilitas yang tinggi untuk meminimalisir kemudahan akses data dari luar.

**Tabel 4.4 Perbandingan *Security Issues***

# V. KESIMPULAN DAN SARAN

## 5.1. Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian skripsi ini, adalah sebagai berikut:

1. LTSP mampu menjadi solusi alternatif sebagai koneksi klien server.
2. LTSP mampu mengurangi beban biaya bagi instansi/organisasi yang ingin membangun koneksi klien server.
3. LTSP merupakan sebuah inovasi baru dalam dunia komputasi modern.
4. Thin client terbukti mampu menggantikan PC klien biasa meskipun memiliki teknologi yang jauh lebih terbelakang.
5. Thin client memberikan solusi ekonomis bagi instansi/organisasi yang ingin menerapkan LTSP.

## 5.2. Saran

Saran yang penulis dapat berikan bagi pembaca atau penulis lain yang ingin mengembangkan sistem ini adalah sebagai berikut:

1. Klien dapat melakukan auto logoff ketika sudah tidak terpakai tanpa harus melalui server.
2. Klien dapat melakukan resume activity ketika pindah PC, tanpa harus kehilanggan aktivitas yang sedang dikerjakan sebelum pindah PC.
3. Sistem mampu melayani bukan hanya thin client, tetapi juga fat client dan hybrid client.
4. Sistem mampu memberikan fitur dan layanan yang lebih banyak.
5. Sistem mampu menggantikan koneksi klien server konvensional secara menyeluruh.

## VI. DAFTAR PUSTAKA


1. "Introduction to LTSP". http://www.ltsp.org/. Diambil pada 14 Juni 2015.
2. Linus Benedict Torvalds (5 Oktober 1991). "Free minix-like kernel sources for 386-AT". Newsgroup:comp.os.minix. Diambil pada 15 Juni 2015.
3. Barry Levine (26 August 2013). "Linux' 22th Birthday Is Commemorated - Subtly - by Creator". Simpler Media Group, Inc. Diambil pada 15 Juni 2015. "Originally developed for Intel x86-based PCs, Torvalds' "hobby" has now been released for more hardware platforms than any other OS in history."
4. "Mobile/Tablet Operating System Market Share". http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qptimeframe=M . Diambil pada 15 Juni 2015.
5. Lyons, Daniel. "Linux menguasai superkomputer". Forbes.com. Diambil pada 15 Juni 2015.
6. "Sony Open Source Code Distribution Service". Sony Electronics. Diambil pada 15 Juni 2015.
7. "Sharp Liquid Crystal Television Instruction Manual" . Sharp Electronics. p.24. Diambil pada 15 Juni 2015.
8. "Edubuntu 11.10 Release Announcement". http://www.edubuntu.org/news/11.10-release . Diambil pada 15 Juni 2015.
9. "About Edubuntu". Edubuntu.org. Diambil pada 15 Juni 2015.
10. "Ubuntu 12.04 to feature extended support period for desktop users". Canonical.com. Diambil pada 15 Juni 2015.
11. Sneddon, Joey. "Ubuntu 15.04 Gets Tentative Release Date of April 23, 2015". OMG Ubuntu. Diambil pada 15 Juni 2015.
12. "WilyWerewolf/ReleaseSchedule - Ubuntu Wiki". ubuntu.com. Diambil pada 15 Juni 2015.
13. James A. McQuillan. Chapter 1.1 from the LTSP 4.1 manual: The steps that the workstation will go through. http://ltsp.mirrors.tds.net/pub/ltsp/docs/ltsp-4.1-en.html#AEN67. Diambil pada 15 Juni 2015.
14. Jonathan Carter, et al. Ubuntu LTSP Tour: Technical differences (between LTSP 4 and 5). Ubuntu.com. Diambil pada 15 Juni 2015.
15. Nieh, Jason; Novik, Naomi &., Yang, S. Jae (December 2005). "A Comparison of Thin-Client Computing Architectures". *Technical Report CUCS-022-00* (New York: Network Computing Laboratory, Columbia University). Diambil pada 15 Juni 2015.
16. http://www.hp.com/hpinfo/newsroom/feature_stories/2008/images/08thinclient-2.jpg. Diambil pada 15 Juni 2015.
17. "Distributed Application Architecture". Sun Microsystem. Diambil pada 15 Juni 2015.
18. Rulifson, Jeff (June 1969). *DEL*. IETF. RFC 5. Diambil pada 15 Juni 2015.
19. Shapiro, Elmer B. (March 1969). *Network Timetable*. IETF. RFC 4. Diambil pada 15 Juni 2015.
20. Sturgis, Howard E.; Mitchell, James George; Israel, Jay E. (1978). Xerox PARC. Diambil pada 15 Juni 2015.
21. Harper, Douglas. "server". *Online Etymology Dictionary*. Diambil pada 15 Juni 2015.
22. "Separating data from function in a distributed file system". *GetInfo*. German National Library of Science and Technology. Iambil pada 15 uni 2015.
23. "VirtualBox – FreeBSD Wiki". Wiki.freebsd.org. 16 Juni 2009. Diambil pada 15 Juni 2015.
24. "Release notes for the Genode OS Framework 14.02". Genode.org. 28

25. Februari 2014. Diambil pada 15 Juni 2015.
26. "Guest_OSes". VirtualBox. 12 Juni 2009. Diambil pada 15 Juni 2015.
27. "How to Install Mac OS X Snow Leopard in VirtualBox on Windows 7" . Redmond Pie. 10 Juli 2010. Diambil pada 15 Juni 2015.
28. Kevin Purdy (4 May 2010). "VirtualBox 3.2 Beta Virtualizes Mac OS X (On Macs)". Lifehacker. Diambil pada 15 Juni 2015.
29. Gary Newell. "Run Ubuntu Linux Within Windows Using VirtualBox". linux.about.com. Diambil pada 15 Juni 2015.
30. "Chapter 4 Guest Additions". VirtualBox. Diambil pada 15 Juni 2015.
31. "Oracle VM VirtualBox 4.3 Now Available" (Press release). Oracle Corporation. 15 Oktober 2013. Diambil pada 15 Juni 2015. Generally available today, Oracle VM VirtualBox 4.3 delivers the latest enhancements to the world's most popular, free and open source, cross-platform virtualization software."
32. "VirtualBox_PUEL – VirtualBox". VirtualBox. 10 September 2008. Diambil pada 15 juni 2015.
33. "GPL". VirtualBox. Diambil pada 15 Juni 2015.
34. Ronny Ong View profile More options. "Additions Version History – microsoft.public.virtualpc | Google Groups". Groups.google.com. Diambil pada 15 Juni 2015.
35. "Connectix Announces First Virtual Computing Solution for OS/2 Users; Virtual PC Lets Enterprises Run OS/2 and Windows Concurrently on a Single PC | Business Wire | Find Articles at BNET". Findarticles.com. 1 Juni 2002. Diambil pada 15 Juni 2015.
36. "Sun Microsystems Announces Agreement to Acquire Innotek, Expanding Sun xVM Reach to the Developer Desktop" (Press release). Sun Microsystems. 12 Februari 2008. Diambil pada 15 Juni 2015.
37. "E-Commerce News: Business: Sun Gets Desktop Virtualization Chops With Innotek Buy". Ecommercetimes.com. Diambil pada 15 Juni 2015.
38. "Sun Welcomes Innotek". Sun Microsystems, Inc. Diambil pada 15 Juni 2015. Pada 20 February 2008 Sun selesai mengakuisisi Innotek.
39. "Oracle and Virtualization". Oracle Corporation.Diambil pada 15 Juni 2015.
40. "VirtualBox Joins Oracle's Enterprise Virtualization Portfolio". systemnews. February 25, 2010. Diambil pada 15 Juni 2015.
41. Hawley, Adam (February 26, 2010). "The Oracle VM Product Line Welcomes Sun!". *Oracle Virtualization Blog*. Oracle Corporation. Diarsipkan dari aslinya pada 7 April 2010. Diambil pada 15 Juni 2015.
42. "Licensing: Frequently Asked Questions". VirtualBox. Diambil pada 15 Juni 2015.
43. "Editions". VirtualBox. Diambil pada 15 Juni 2015.
44. "Copyright file of Virtualbox 4.2 in Wheezy-backports". Debian. Diambil pada 15 Juni 2015.

**LAMPIRAN**

**Tabel 2.1 Tabel Sejarah Versi Edubuntu**

| Versi | Tanggal rilis | Nama kode | Sub Versi | Didukung sampai |
|---|---|---|---|---|
| 4.10 | 20 Oktober 2004 | Warty Warthog | Sounder | 30 April 2006 |

| | | | | |
|---|---|---|---|---|
| 5.04 | 8 April 2005 | Hoary Hedgehog | Array | 31 Oktober 2006 |
| 5.10 | 13 Oktober 2005 | Breezy Badger | Colony | 13 April 2007 |
| 6.06 LTS | 1 Juni 2006 | Dapper Drake | Flight | Juni 2009 (desktop)<br>Juni 2011 (server) |
| 6.10 | 26 Oktober 2006 | Edgy Eft | Knot | April 2008 |
| 7.04 | 19 April 2007 | Feisty Fawn | Herd | Oktober 2008 |
| 7.10 | 18 Oktober 2007 | Gusty Gibbon | Tribe | April 2009 |
| 8.04 LTS | 21 April 2008 | Hardy Heron | Siege | April 2011 (desktop)<br>April 2013 (server) |
| 8.10 | 30 Oktober 2008 | Intrepid Ibex | | 30 April 2010 |
| 9.04 | 23 Maret 2009 | Jaunty Jackalope | | 23 Oktober 2010 |
| 9.10 | 29 Oktober 2009 | Karmic Koala | | 30 April 2011 |
| 10.04 LTS | 29 April 2010 | Lucid Lynx | | 9 Mei 2013 (desktop)<br>30 April 2015 (server) |
| 10.10 | 10 Oktober 2010 | Maverick Meerkat | | 10 April 2012 |
| 11.04 | 28 April 2011 | Natty Narwhal | | 28 Oktober 2012 |
| 11.10 | 13 Oktober 2011 | Oneiric Ocelot | | 9 Mei 2013 |
| 12.04 LTS | 26 April 2012 | Precise Pangolin | | 26 April 2017[10] |
| 12.10 | 18 Oktober 2012 | Quantal Quetzal | | 16 Mei 2014 |
| 13.04 | 25 April 2013 | Raring Ringtail | | 27 Januari 2014 |
| 13.10 | 17 Oktober 2013 | Saucy Salamander | | 17 Juli 2014 |
| 14.04 LTS | 17 April 2014 | Trusty Taht | | April 2019 |
| 14.10 | 23 Oktober 2014 | Utopic Unicorn | | Juli 2015 |
| 15.04 | 23 April 2015[11] | Vivid Vervet | | Januari 2016 |
| 15.10 | 22 Oktober 2015[12] | Wily Werewolf | | Juli 2016 |

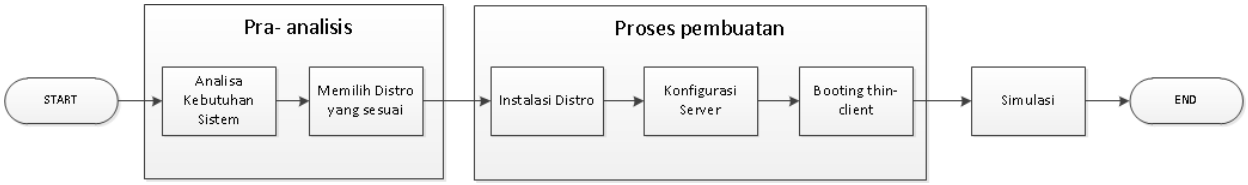
**Tabel 2.2 Perbedaan Antara LTSP 4 dan 5[14]**

| Tujuan | LTSP 4 | LTSP 5 |
|---|---|---|
| **GUI *Export*** | XDMCP | Ssh-X |
| ***Remote login* (X *display manager*)** | KDM/GDM | LTSP *Display Manager* (LDM) |
| **Metode integrasi** | LTSP tarball | Asli sebagai bagian dari distribusi |
| **Root filesystem** | NFS | NBD atau NFS |
| **Server otentikasi** | XDMCP server | SSH server |



**Gambar 2.1 Bentuk Fisik Thin Client**
**(Sumber: hp.com)**

**Gambar 3.1 Langkah Kerja Pembuatan LTSP *Thin-client***




**Tabel 4.1 Perbandingan Total Biaya**

**(Sumber: Bhinneka.com)**

| | **Klien server konvensional** | **Harga** | **LTSP** | **Harga** |
|---|---|---|---|---|
| **Server** | IBM System X3500M5-15A | 48,500,000 | IBM System X3500M5-15A | 48,500,000 |
| **Klien** | ASUS Desktop K31AD-ID002D | 5,549,000 | NCOMPUTING L130 | 1,621,400 |
| **Total** | | 54,049,000 | | 50,121,400 |

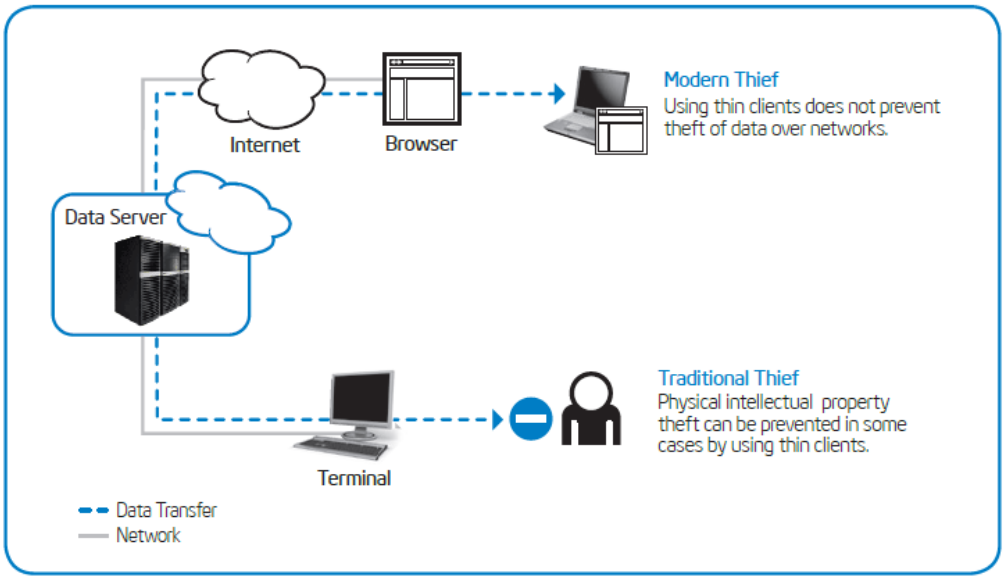
**Tabel 4.2 Perbandingan Waktu Instalasi**

| | **Klien server konvensional** | | **LTSP** | |
|---|---|---|---|---|
| | Server | Klien | Server | *Thin client* |
| Instalasi sistem operasi | 40 menit | 30 menit | 40 menit | - |

| | | | | |
|---|---|---|---|---|
| Konfigurasi jaringan | 30 menit | 15 menit | 30 menit | - |
| Instalasi aplikasi | 5 menit | 5 menit | 5 menit | - |
| Proses update | 60 menit | 30 menit | 60 menit | - |
| Setup firewall | 30 menit | 15 menit | 30 menit | - |
| total | 165 menit | 95 menit | 165 menit | - |
| | 260 menit | | 165 menit | |

**Tabel 4.3 Perbandingan Proses *Maintenance***

| **Prosedur *maintenance*** | **Klien server konvensional** | | **LTSP** | |
|---|---|---|---|---|
| | Server | Klien | Server | Thin client |
| ***Backup* sistem** | √ | √ | √ | - |
| **Pengecekan kapasitas HDD** | √ | √ | √ | - |
| **Cek alarm RAID** | √ | - | √ | - |
| ***Update* OS** | √ | √ | √ | - |
| ***Update control panel*** | √ | √ | √ | - |
| **Cek *update* aplikasi** | √ | √ | √ | - |
| **Cek *remote management tools*** | √ | - | √ | - |
| **Cek *hardware error*** | √ | √ | √ | - |
| **Cek *beban server*** | √ | - | √ | - |
| **Memeriksa akun *user*** | √ | √ | √ | - |
| **Mengganti *password*** | √ | √ | √ | - |
| **Cek sistem keamanan** | √ | √ | √ | - |



**r 4.24 *Thin clients* mampu mencegah pencurian data fisik dari klien; akan tetapi mereka tidak mencegah pencurian data melalui jaringan luar**

**(Sumber: IT@Intel White Paper)**

**Tabel 4.4 Perbandingan *Security Issues***

| Klien server konvensional | LTSP |
|---|---|
| Kehilangan data fisik dapat terjadi. Karena data tersimpan pada setiap HDD klien sehingga risiko terjadinya kebocoran data fisik cukup tinggi. | Pencegahan kehilangan data fisik. Dengan *thin client*, penyimpanan dilakakukan terbatas pada data *center* sehingga dapat mengurangi risiko kebocoran data fisik. |
| *Privileged users*.<br>Pengguna memiliki *administrative privileges* sehingga pengguna mampu melakukan eksploitasi untuk mengubah file sistem dan pengaturan sistem. | *Non-privileged users*.<br>*Administrative privileges* pengguna ditiadakan, menggurangi kemampuan eksploitasi untuk mengubah files sistem dan pengaturan sistem. |
| Tidak ada larangan instalasi aplikasi oleh pengguna.<br>Pengguna dapat melakukan instalasi perangkat lunak tambahan sehingga kelemahan sistem dapat terekspos atau sistem dapat terinfeksi oleh malware | Larangan instalasi aplikasi oleh pengguna.<br>Pengguna tidak dapat melakukan instalasi perangkat lunak tambahan yang mampu mengekspos kelemahan sistem atau menginfeksi sistem dengan malware. |
| Integritas klien.<br>Proses *maintenance* yang diberikan pada setiap klien berbeda sesuai dengan masalah keamanan yang terjadi. *Update* terbaru harus dilakukan secara manual dari satu klien ke klien yang lain. | Integritas klien.<br>Proses *maintenance* seluruh klien dilakukan secara konstan, berdasarkan pada panduan konfigurasi standard. *Update* terbaru dapat dilakukan secara cepat dan konsisten dengan menggunakan *image* yang berbasis server. |
| *Ability to roll back to a known good state*.<br>Hal ini susah dilakukan karena tidak diketahui dimana posisi terbaik untuk dilakukan *roll back* karena klien memiliki *state* yang berbeda. | *Ability to roll back to a known good state*.<br>Dengan *thin client*, hal ini sering dicapai dengan me-*reboot* atau me-*reload* versi sebelumnya dari file container virtual. |