

Implementasi Compatibility Layer Pada Jaringan Server Diskless Berbasis Lubuntu 18.04 LTS

Farid Jatri Abiyu¹, Ibnu Ziad², Ade Silvia Handayani³

^{1,2,3}Politeknik Negeri Sriwijaya, Program Studi D-IV Teknik Telekomunikasi

Jl. Srijaya Negara, Bukit Lama, Bukit Besar, Kota Palembang, Sumatera Selatan 30139, 082175140935,

e-mail: faridabiyyu021098@gmail.com

ARTICLE INFO

ABSTRACT

Article history:

Received 30 Oktober 2020

Received in revised form 2 November 2020

Accepted 10 November 2020

Available online 10 Desember 2020

Diskless server is a cluster computer network which uses SSH (Secure Shell) protocol to grant the client an access to the host's directory and modify it's content so that the client don't need a hardisk (Thin Client). One way to design a diskless server is by utilizing "Linux Terminal Server Project", an open source-based script for Linux. However, using Linux has it own drawback, such as it can't cross platform for running an application based on Windows system which are commonly used. This drawback can be overcome by using a compatibility layer that converts a windows-based application's source code. The data which will be monitored is the compatibility layer implementation's result, and quality of service, which includes the throughput, packet loss, delay, and jitter. The result of measurement from those four parameters resulting in "Excellent" for throughput, "Perfect" for packet loss and delay, and "Good" for jitter.

Keywords: *Diskless Server, Compatibility Layer, Linux Terminal Server Project, Quality of Service*

1. Pendahuluan

Diskless server adalah sistem jaringan cluster computer yang memungkinkan client mengakses direktori utama dari host, sehingga data host dapat diakses dan dimodifikasi dan semua proses komputasi memanfaatkan resource dari host[1]. Metode ini digunakan untuk membangun sebuah jaringan komputer yang memiliki kebutuhan dasar yang sama, namun dengan biaya yang lebih murah, seperti sekolah, laboratorium dan kantor.

Salah satu metode pembangunan diskless server adalah menggunakan CCBoot pada sistem operasi Microsoft Windows [2]. Microsoft Windows serta CCBoot adalah perangkat lunak berlisensi, sehingga untuk menggunakan perangkat lunak tersebut, dibutuhkan serial number yang didapatkan dengan cara membeli perangkat lunak tersebut kepada pihak yang bersangkutan. Lisensi tersebut dapat dihindari dengan menggunakan LTSP (Linux Terminal Server Project) yang berbasis open source pada sistem operasi berbasis Linux [3].

Perangkat lunak komputer yang lazim digunakan cenderung menggunakan sistem operasi Windows dan tidak mendukung cross platform untuk dijalankan. Salah satu metode yang diterapkan untuk mengatasi masalah ini adalah dengan menggunakan Virtual Machine [4]. Virtual

Received Oktober 23, 2020; Revised November 29, 2020; Accepted November 12, 2020

Machine adalah sebuah perangkat lunak yang menjalankan sistem operasi diatas sistem operasi lain [5]. Penggunaan Virtual Machine dapat membebani pemakaian RAM pada host karena harus menjalankan dua sistem operasi sekaligus, sehingga untuk mengatasi ini, dibutuhkan sebuah converter, yang berfungsi sebagai lapisan “compatibility” (compatibility layer), salah satu caranya adalah dengan menggunakan Wine (Wine Is Not an Emulator)[6]. Wine merupakan perangkat lunak open source yang bekerja seperti sebuah mesin virtual, namun tidak menjalankan sistem operasi diatas sistem operasi utama.

Penelitian ini akan mengimplementasikan compatibility layer pada sebuah jaringan server diskless sebagai alternatif virtual machine yang lebih mudah diterapkan sehingga memberikan pengalaman pada end-user yang lebih baik. Server diskless akan dibangun menggunakan script LTSP, serta pengukuran Quality of Service pada host.

1.2 Linux Terminal Server Project

LTSP merupakan script open-source pada linux yang mengkonversi sebuah node untuk menjadi terminal server dan memungkinkan banyak user mengakses host secara bersamaan [7]. Client tidak memerlukan penyimpanan karena client menggunakan penyimpanan pada direktori /home di host [8]. Client akan menjalankan image pada kernel host dengan proses booting menggunakan iPXE menggunakan interface Ethernet. Jaringan akan beroperasi dengan merequest data untuk mengakses host melalui protokol SSH sehingga membuat seolah olah client beroperasi sendiri dengan sistem operasi sendiri, namun semua proses pengolahan data terjadi di host [9]. Karena alasan tersebut, client LTSP disebut sebagai thin client.

1.3 DHCP Server

Client membutuhkan IP Address untuk terhubung dengan host. Pengalamatan IP dapat dilakukan dengan dua cara : static IP dan dynamic IP. Pada penelitian ini akan diterapkan pengalamatan IP secara otomatis dengan menggunakan DHCP Server, sehingga client akan secara otomatis merequest dan mendapatkan IP dari host [10]. DHCP server dapat menggantikan peran router sebagai NAT (Network Address Translation) sehingga untuk menghubungkan node, hanya perlu menggunakan switch/hub.

1.4 Monitoring, Remoting dan Broadcasting Pada Client

Host akan memonitor aktivitas client untuk mengetahui apa yang sedang dilakukan client dengan menggunakan software Epopes yang berbasis open source, apabila terjadi masalah, maka host dapat secara remote melakukan troubleshoot apabila diperlukan. Selain meremote dan memonitor, host juga dapat melakukan pesan secara broadcast kepada semua client [11]

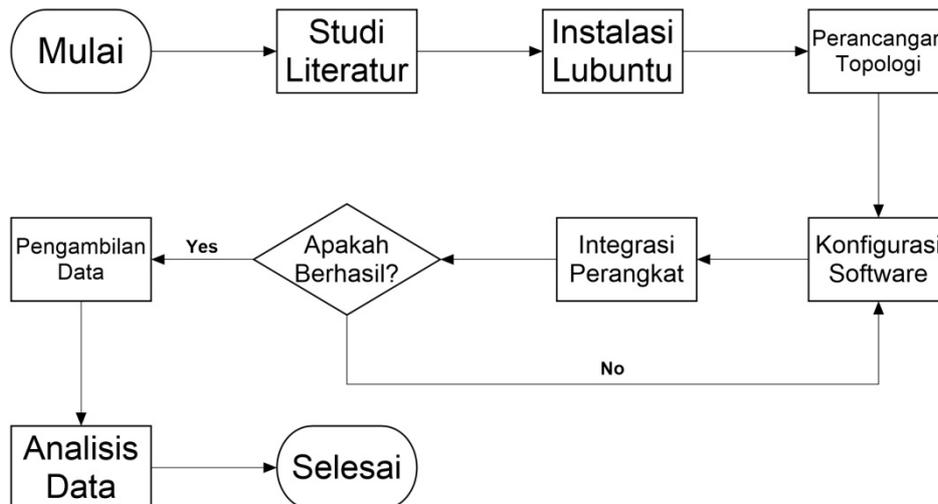
1.5 Wine

Wine merupakan singkatan dari “Wine Is Not an Emulator” adalah sebuah software berbasis open-source yang berfungsi sebagai Compatibility Layer. Wine merubah API (Application Programming Interface) program berbasis Windows menjadi POSIX (Portable Operating System Interface), dan menghemat penggunaan perangkat keras pada PC sehingga memudahkan pengeksekusian program berbasis Windows pada sistem operasi berbasis Linux [5].

1.6 Protokol SSH (Secure Shell)

SSH me-enkripsi komunikasi antara host dan client. SSH merupakan alternatif RLOGIN (Remote Login), RSH (Remote Shell) dan Telnet [9]. Setelah di konfigurasi, client dapat mengakses host sehingga menjadikan host sebagai penyimpanan cloud yang bersifat lokal.

2. Metode Penelitian



Gambar 1. Diagram Alur Penelitian

2.1 Studi Literatur

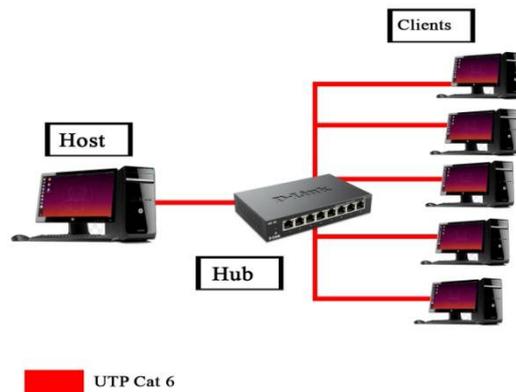
Proses penelitian dimulai dengan melakukan studi literature dengan membaca jurnal-jurnal dan buku-buku yang relevan dengan penelitian ini. Jurnal yang menjadi bahan referensi dijadikan patokan dalam membangun server agar dapat menemukan hasil akhir sebaik mungkin.

2.2 Instalasi Lubuntu

Tahap ini merupakan permulaan dalam membangun server, sistem operasi Lubuntu dipilih karena memiliki desktop environment (DE) yang ringan (lightweight) sehingga dapat menutupi pemakaian RAM secara berlebihan karena penggunaan desktop environment Gnome yang menjadi desktop environment utama dalam sistem operasi Ubuntu. Instalasi dilakukan dengan mounting sebuah live USB dengan program installer Lubuntu. Program ini dapat diunduh di website resmi Lubuntu.

2.3 Perancangan Topologi

Perancangan topologi merupakan tahapan untuk menentukan hubungan antara node dengan node lainnya. Perancangan topologi mempertimbangkan kebutuhan server, dan bagaimana hubungan logis antara node yang akan digunakan. Setelah mempertimbangkan kedua hal tersebut, topologi yang akan digunakan adalah topologi bintang (star), dimana topologi ini memiliki keunggulan dimana apabila salah satu client mengalami gangguan, maka client lain tidak akan terkena dampak gangguan yang sama, dan jika host yang mengalami gangguan, maka tidak perlu melakukan troubleshoot di sisi client, cukup hanya di sisi host.



Gambar 2 Topologi Server

2.4 Konfigurasi Software

Konfigurasi software dibagi menjadi beberapa bagian :

2.4.1 Konfigurasi DHCP Server

Konfigurasi DHCP Server merupakan tahapan untuk membangun DHCP server sebagai pembagi alamat IP otomatis untuk client. DHCP Server di konfigurasi dengan membuat script pada direktori netplan dengan ekstensi ".yaml", script ini akan menentukan IP dan subnet server yang akan dibangun. Gambar 4 menunjukkan script yang berisikan alamat IP yang digunakan oleh host (192.168.67.1) dengan subnet mask (255.255.255.0).

```

GNU nano 2.9.3
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp1s0f1:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.67.1/24]
      gateway4: 192.168.67.1
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]

```

Gambar 3 Script DHCP Server

Setelah membuat script, maka berikutnya yang harus dilakukan adalah menginstall dnsmasq agar dapat menggunakan protokol TFTP (Trivial File Transfer Protocol), protokol tersebut berguna dalam komunikasi antar host dan client dalam merequest IP Address.

2.4.2 Konfigurasi LTSP, Epopetes dan Wine

Langkah ini merupakan proses instalasi LTSP beserta beberapa script lain yang dibutuhkan.

Script yang dikonfigurasi antara lain :

- ltsp : berisi source code dari LTSP
- ltsp-binaries : berisi iPXE dan biner memtest
- nfs-kernel-server : membagikan akses client melalui NFS, yaitu sebuah kumpulan protokol yang digunakan untuk mengakses beberapa sistem berkas melalui jaringan.
- openssh-server : memberikan klien izin akses ke direktori /home pada host dengan menggunakan SSH File System
- Epopetes : adalah software open source yang digunakan host untuk memonitor klien, melakukan broadcast dan meremote klien dari jauh.

- Wine : sebagai compatibility layer. Wine berfungsi sebagai pengganti virtual machine dalam perannya untuk menggunakan aplikasi berbasis windows.

2.4.3 Integrasi Perangkat

Langkah ini merupakan tahap uji coba apakah konfigurasi yang dilakukan sudah benar atau masih salah. Integrasi perangkat dilakukan dengan menghubungkan node client dengan node host menggunakan perantara sebuah hub seperti yang ditunjukkan pada gambar 4 (a) dan (b). Client perlu diubah proses booting default dari hardisk menjadi PXE (LAN) pada menu bios.

Client merequest IP pada host, pada tahap ini, terjadi komunikasi antara host dan client, host akan mengenali mac address client dan memberikan IP pada client didalam subnet yang sama. Tabel 1 adalah alamat IP tiap node yang diberikan oleh host dengan protokol DHCP.

Tabel 1 IP Address tiap Node

Port Hub	Fungsi Node	IP Address
1	Host	192.168.67.1
2	Client 1	192.168.67.83
3	Client 2	192.168.67.150
4	Client 3	192.168.67.116
5	Client 4	192.168.67.128
6	Client 5	192.168.67.28



(a)



(b)

Gambar 4 (a) Hub yang telah di integrasi dan (b) tampilan penuh hubungan antar host dan client

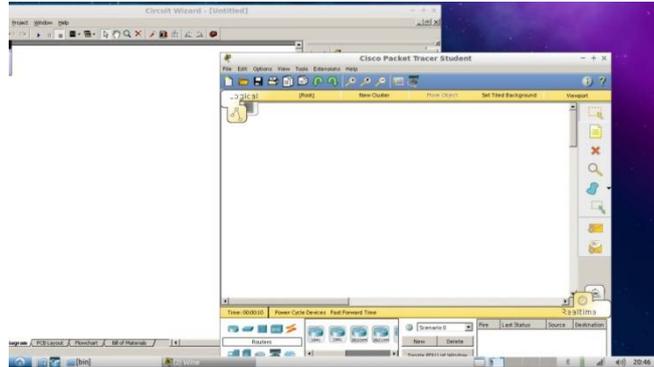
2.4.4 Pengambilan Data

Setelah berhasil menintegrasikan klien, langkah berikutnya adalah pengambilan data. Data yang diperlukan adalah hasil implementasi compatibility layer dan Quality of Service yang mencakupi 4 parameter : Throughput, Packet Loss, Delay dan Jitter pada sisi host saat melayani lima client sekaligus. Pengambilan data Quality Of Service dilakukan dengan menggunakan aplikasi Wireshark.

3. Hasil Penelitian

3.1 Hasil Implementasi Compatibility Layer

Compatibility Layer mengonversi source code pada sebuah program berbasis windows agar dapat berjalan pada sistem Linux. Compatibility Layer yang digunakan adalah WINE (Wine Is Not An Emulator).



Gambar 5 Menjalankan Circuit Wizard dan Cisco Packet Tracer menggunakan WINE

Dengan menggunakan compatibility layer, host tidak perlu menggunakan virtual machine untuk menjalankan program berbasis windows, sehingga dapat menghemat penyimpanan dan memori. Gambar 7 menunjukkan WINE sedang menjalankan dua program sekaligus, yaitu : Cisco Packet Tracer dan Circuit Wizard pada host.

3.2 Quality of Service

Pengukuran QoS dilakukan dengan cara mengambil data pada host dalam keadaan client tidak idle, yaitu keadaan dimana client melakukan aktifitas sehingga akan memenuhi trafik dan data yang didapat akan lebih riil .Pengambilan data dilakukan dengan keadaan lima client aktif menggunakan aplikasi Wireshark selama kurang lebih 180 detik (3 menit). Quality of service memiliki nilai standar TIPHON untuk menentukan indeks kualitas jaringan.

- Throughput

$$= \frac{\text{Byte yang diterima}}{\text{Time Span}}$$

$$= \frac{538413553 \text{ Byte}}{181.568 \text{ Second}}$$

$$= 2965355 \text{ Bps}$$

Dikonversikan ke bit, maka :

$$\begin{aligned} \text{Throughput} &= 2965355 \text{ Bps} \times 8 \text{ bit} \\ &= 23722840 \text{ bps} \\ &= 23 \text{ mbps} \end{aligned}$$

- Packet Loss

$$\frac{\text{Paket yang dikirim} - \text{Paket diterima}}{\text{Paket dikirim}} \times 100\%$$

$$= \frac{587013 - 586556}{587013} \times 100\%$$

$$= 0,077\%$$

- Delay (Latency)

$$\text{Delay Rata-Rata} = \frac{\text{Total Delay}}{\text{Packets}}$$

$$= \frac{181,3951896}{587013}$$

$$= 0,000309139 \text{ Second} = 0,3 \text{ ms}$$

- Jitter Rata-Rata

$$= \frac{\text{Total Jitter}}{\text{Packet} - 1}$$

$$= \frac{0.069415927}{587012}$$

$$= 0,00000011 \text{ Second} = 0,00011 \text{ ms}$$

3.3 Analisa Quality of Service

- Analisa Throughput

Throughput yang didapat dari hasil pengukuran memiliki nilai **23 Mbps** dengan indeks **4 (Sempurna)**. Hal ini terjadi karena saat tidak idle/digunakan, traffic yang melalui jaringan menjadi sibuk, sehingga keadaan ini dapat menentukan nilai throughput client yang sebenarnya. Jika dibandingkan dengan standar TIPHON untuk Throughput, maka server dapat dikatakan memiliki throughput yang sangat bagus (untuk 5 client).

- Analisa Packet Loss

Packet Loss yang didapat dari hasil pengukuran adalah **0,007%** dari paket yang dikirim, dengan indeks **4 (Sempurna)**, hal ini terjadi dikarenakan server hanya menggunakan kabel UTP sebagai media transmisi dengan jarak yang dekat, sehingga dapat mengurangi noise dan redaman ketika pentransmisi data, dimana hal ini sering dialami jaringan wireless.

- Analisa Delay

Delay yang terjadi selama pengambilan sampel adalah **0,3 ms** dengan indeks **4 (Sempurna)** dalam standar TIPHON. Delay dapat terjadi akibat jarak fisik antara titik sumber (host) dan titik

tujuan (client), pengaruh media transmisi, serta pengaruh desktop environment yang ringan sehingga software dan hardware tidak terbebani dalam pemrosesan data.

- Analisa Jitter

Jitter merupakan variasi delay yang terjadi pada jaringan, semakin kecil durasi jitter, maka jaringan tersebut akan semakin baik. Sebuah jaringan dapat dikatakan memiliki jitter sempurna apabila nilainya = 0 ms , untuk mencapai ini, jaringan tersebut harus memiliki durasi delay yang sama sepanjang server itu bekerja. Hasil pengukuran yang didapat menggunakan wireshark adalah **0,00011 ms**, sehingga dikategorikan “baik” sesuai standar TIPHON.

4. Kesimpulan

Berdasarkan hasil pembahasan yang dibahas pada bagian 3, dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan hasil pengujian implementasi compatibility layer, dapat disimpulkan bahwa WINE sebagai compatibility layer dapat menjadi alternatif virtual machine yang lebih mudah diimplementasikan karena server tidak memerlukan sistem operasi kedua yang digunakan pada virtual machine untuk menjalankan aplikasi yang tidak mendukung *cross-platform*.
2. DHCP server yang dibangun pada host dapat menggantikan peran router pada server konvensional dalam pembagian alamat IP sehingga dapat menghemat biaya pembangunan server.
3. Hasil pengujian Quality of Service selama tiga menit pada 5 client dengan kondisi tidak idle menghasilkan nilai yang baik pada tiap parameter, hal ini dikarenakan server menggunakan *desktop environment* yang ringan dan transmisi data dilakukan menggunakan kabel UTP sehingga traffic yang berjalan tidak mengalami redaman dan noise yang besar.

Referensi

- [1] Kusuma, D. A. Perancangan Jaringan Diskless Menggunakan Program CCBOOT (Studi Kasus Pada Game Center PHDNET Semarang)
- [2] Nauri YI. Analisis Dan Perancangan Jaringan Komputer Tanpa Harddisk (Diskless) Menggunakan Linux Ubuntu 12.10 (Doctoral dissertation, Universitas Muhammadiyah Surakarta),2013.
- [3] Sonawane, B. S., et al. "Data Redundancy on Diskless Client using Linux Platform." International Journal of Computer Science and Information Security (IJCSIS) 16.4 ,2018.
- [4] Shabaitah AR. Server-Based Desktop Virtualization. Rochester Institute of Technology RIT Scholar Works, 2014
- [5] Springfield RS, Cromer D, Locker H, Waltermann RD, inventors; Lenovo Singapore Pte Ltd, assignee. Diskless client using a hypervisor. United States patent US 8,898,355. 2014.
- [6] Wine HQ Official Website, [https:// www.winehq .org/](https://www.winehq.org/) , Diakses 20 Agustus 2020

- [7] Rakhmat H, Purwanto Y, Dyah I. Perancangan Jaringan Komputer Diskless Berbasis Linux Terminal Server Project Pada Sistem Operasi Ubuntu 8.04. Jurnal Elektro dan Telekomunikasi Terapan. 2014;1(1):15-23.
- [8] Punthawanunt S, Sappajak S, Fujii Y. Fully automated diskless deployment for university's lab. Kasem Bundit Engineering Journal. 2018 Sep 8;8(2):104-16.
- [9] Al-Khazraji SH, Al-Sa'ati MA, Abdullah NM. Building High Performance Computing Using Beowulf Linux Cluster. International Journal of Computer Science and Information Security. 2014 Apr 1;12(4):1.
- [10] Imansyah I K. Analisis Perbandingan Performa Session Based Desktop Virtualization Antara NComputing Dengan Linux Terminal Server Project (LTSP) (Doctoral dissertation, University of Muhammadiyah Malang).
- [11] Gufron G. Perancangan Laboratorium Komputer Menggunakan Jaringan Diskless Berbasis Linux Terminal Server Project dan Pemanfaatan Eoptes Sebagai Aplikasi Monitoring. Jurnal Ipteks Terapan. 2016 May 24;8(1):18-27.