

PENGEMBANGAN PERANGKAT PRAKTIKUM SISTEM OPERASI UNTUK PEMBUATAN MESIN VIRTUAL SECARA OTOMATIS DI PROXMOX VE SERVER MENGGUNAKAN SOFTWARE ANSIBLE

¹ Rico Cahyo Widodo Putra, ² F. Yudi Limpraptono, ³ Sotyohadi

^{1,2,3} Teknik Elektro S1 ITN Malang, Malang, Indonesia

¹1712017@scholar.itn.ac.id, ²fyudil@lecturer.itn.ac.id, ³sotyohadi@lecturer.itn.ac.id

Abstrak— Teknologi virtualisasi memiliki keuntungan untuk membuat beberapa komputer virtual dalam satu komputer fisik sehingga menghemat sumber daya dan memanfaatkan sumber daya komputer fisik secara maksimal. Saat ini, pada praktikum sistem operasi di Laboratorium Jaringan Komputer ITN Malang masih menggunakan teknologi virtualisasi dengan Virtualbox atau VMWare yang tidak mendukung teknologi container secara langsung dan juga bisa terancam oleh virus dari komputer host yang mengakibatkan kehilangan data dan konfigurasi.

Berdasarkan permasalahan yang muncul, serta adanya metode yang memungkinkan dapat atau mampu menanggulangi permasalahan tersebut, maka muncul suatu gagasan untuk membangun mesin virtual secara otomatis di Proxmox VE menggunakan software Ansible sebagai pendukung praktikum sistem operasi.

Penggunaan mesin virtual pada Proxmox VE mampu menekan penggunaan sumber daya pada tiap komputer praktikan, sehingga kinerja komputer dapat ditingkatkan, serta dengan adanya pembuatan mesin virtual secara otomatis dengan software Ansible dapat membantu dalam pelaksanaan praktikum sistem operasi menjadi lebih efisien.

Kata Kunci: *Virtualisasi, Proxmox VE, Mesin Virtual, Otomatis, Ansible.*

I. PENDAHULUAN

Otomatisasi adalah topik yang banyak dipakai dan diteliti untuk memudahkan suatu pekerjaan. Otomatisasi dapat menjadikan suatu pekerjaan menjadi lebih cepat, efisien dan mampu mengurangi kesalahan sehingga dapat menghemat banyak waktu jika dibandingkan tanpa adanya otomatisasi. Salah satu penerapan otomatisasi adalah dalam bidang teknologi komputer diantaranya pada jaringan komputer, virtualisasi pada perangkat keras komputer, server, aplikasi, dll. [23]. Contoh software yang dapat digunakan untuk

melakukan otomatisasi adalah Ansible. Ansible adalah software otomatisasi open source yang mengotomatisasi penyediaan perangkat lunak, manajemen konfigurasi, dan pemasangan aplikasi [15].

Pelaksanaan praktikum sistem operasi di Laboratorium Jaringan Komputer ITN Malang saat ini, masih menggunakan software pendukung seperti VMWare dan Virtualbox yang diinstal atau dipasangkan didalam sistem operasi pada tiap perangkat komputer, sehingga dengan metode seperti ini sumber daya pada mesin virtual masih tergantung pada sumber daya komputer host. Selain itu, VMWare dan Virtualbox tidak mendukung teknologi kontainer secara langsung, yang mengakibatkan kesalahan proses instalasi atau pembuatan mesin virtual akan berdampak pada pengulangan pembuatan mesin virtual di komputer host. Disamping hal itu, tidak didukungnya teknologi kontainer secara langsung, mengakibatkan penggunaan sumber daya yang besar pada komputer host. Aplikasi VMWare dan Virtualbox juga bisa terancam oleh virus dari komputer host yang mengakibatkan kehilangan data dan konfigurasi [18].

Dengan adanya permasalahan dan kekurangan yang menjadi kendala dalam pelaksanaan praktikum sistem operasi saat ini dan kedepannya, maka terdapat suatu metode yang dapat digunakan untuk menanggulangi permasalahan tersebut. Salah satu metode ini menggunakan virtualisasi dengan Proxmox VE (Proxmox Virtual Environment) yang merupakan distro linux berbasis debian yang bersifat open source-platform virtualisasi dengan web interface untuk pengoperasian dan manajemen peralatan [13]. Selain virtualisasi, untuk memudahkan dalam manajemen konfigurasi disaat pelaksanaan praktikum digunakan juga perangkat lunak otomatisasi menggunakan Ansible [15].

Berdasarkan permasalahan yang muncul, serta adanya metode yang memungkinkan dapat atau mampu menanggulangi permasalahan tersebut, maka muncul suatu gagasan untuk membuat mesin virtual di proxmox VE server menggunakan software Ansible sebagai pendukung praktikum sistem operasi.

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara membuat server proxmox ve pada Laptop untuk teknologi virtualisasi.
2. Bagaimana cara melakukan pembuatan mesin virtual secara otomatis pada Proxmox VE menggunakan software Ansible di Praktikum Sistem Operasi pada Laboratorium Jaringan Komputer dan Cisco, ITN 2 Malang.

Tujuan dibuatnya penelitian ini yaitu:

1. Membuat server virtualisasi menggunakan Proxmox VE.
2. Menggunakan software Ansible untuk membuat mesin virtual secara otomatis di Proxmox VE.
3. Memudahkan dan meningkatkan perangkat pada pelaksanaan Praktikum Sistem Operasi.

II. KAJIAN PUSTAKA

A. Virtualisasi

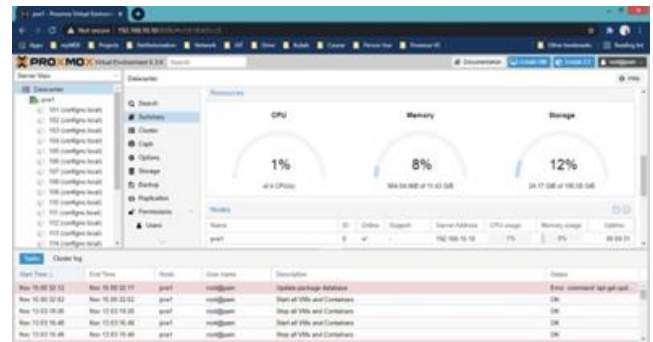
Secara umum virtualisasi adalah suatu teknologi pada sebuah perangkat lunak yang memungkinkan satu perangkat keras untuk menjalankan beberapa sistem operasi dan layanan pada saat yang sama [1]. Pengelola teknologi virtualisasi biasa disebut sebagai hypervisor yang merupakan jenis perangkat lunak yang membangun mesin virtual dan beroperasi secara terpisah dengan sistem operasi host maupun antar mesin virtual. *Container* adalah salah satu teknologi virtualisasi yang saat ini banyak digunakan karena memiliki keunggulan lebih hemat *memory*, *processor*, *storage* dan memiliki waktu startup yang jauh lebih cepat daripada server riil dan virtualisasi berbasis *Virtual Machine* (VM).

B. Proxmox Virtual Environment

Proxmox VE adalah salah satu platform majemen server untuk teknologi virtualisasi yang memiliki keunggulan utama memiliki lesensi gratis (open source) yang memakai distribusi Linux berbasis Debian dengan kernel Ubuntu LTS yang dimodifikasi dan memungkinkan penerapan dan pengelolaan mesin virtual dan container [13]. Proxmox VE memiliki antarmuka web-console dan command-line untuk mealakukan menejemen virtualisasi. Dua jenis virtualisasi yang didukung oleh Proxmox VE, yaitu berbasis container dengan LXC (Linux Container) yang menggantikan OpenVZ mulai dari versi 4.0 dan virtualisasi penuh dengan KVM (Kernel-based Virtual Machine). Proxmox VE dilisensikan di bawah GNU Affero General Public License, versi 3.

Proxmox VE memiliki banyak kelebihan sebagai teknolgi virtualisasi, antara lain:

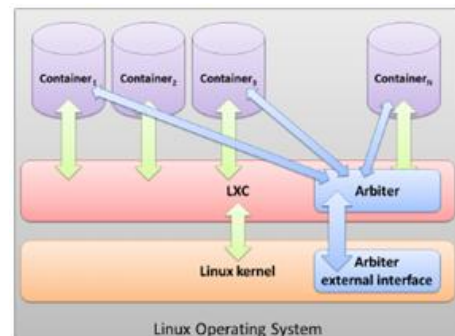
- a. Memiliki lesensi open-source, sehingga bisa digunakan oleh siapapun secara free.
- b. Mendukung platform virtualisasi berbasis LXC dan KVM.
- c. Membutuhkan sumber daya seperti memory dan processor yang sedikit dalam menjalankan mesin virtual.
- d. Mendukung auto-backup sesuai jadwal yang ditentukan baik internal storage maupun eksternal storage.
- e. Mendukung teknologi Cluster dan High Availability sehingga Proxmox VE memiliki peforma dan ketersediaan yang baik sebagai server.
- f. Mendukung banyak model storage: LVM, iSCSI, Local Directory dan NFS.



Gambar 1. Tampilan Dashboard Proxmox VE Pada Web Browser

C. Linux Container (LXC)

Linux Container (LXC) adalah teknologi virtualisasi pada level sistem operasi sehingga mampu menjalankan banyak sistem operasi atau layanan di dalam lingkungan yang saling terisolasi. LXC adalah teknologi virtualisasi berbasis *container*, dan berjalan dengan baik pada sistem operasi Linux, terutama pada distribusi Ubuntu dan Red Hat beserta turunannya. Berbeda dengan *Virtual Machine* (VM) yang membutuhkan banyak *resource* komputer *host* karena VM memerlukan level *kernel* pada sistem operasinya secara lengkap serta semua program dan aplikasi dasar harus diinstal terlebih dahulu, sedangkan teknolgi *container* memerlukan lebih sedikit *resource* dalam penggunaannya karena *container* menjalankan virtualisasi pada level sistem operasi sehingga menggunakan *kernel* yang sama dengan komputer *host* dari *container* tersebut..



Gambar 2. Arsitektur Umum Linux Container (LXC)

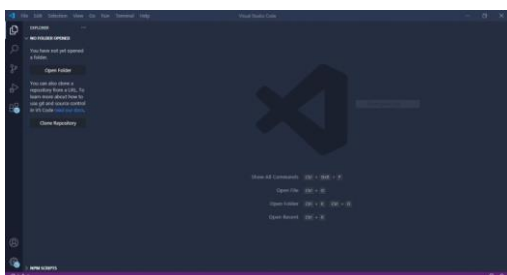
D. Ansible

Ansible adalah software otomatisasi open source yang mengotomatisasi penyediaan perangkat lunak, manajemen konfigurasi, dan pemasangan aplikasi [15]. Ansible berjalan pada banyak sistem mirip Unix, dan dapat mengkonfigurasi sistem mirip Unix serta Microsoft Windows. Ansible dibuat oleh Michael DeHaan dan diakuisisi oleh Red Hat pada tahun 2015. Ansible-Playbook adalah ansible yang menggunakan Playbook untuk melakukan otomatisasi. Playbook adalah file berekstensi YAML yang berisi ketentuan konfigurasi dan penerapan aturan otomatisasi di Ansible dan memungkinkan Ansible untuk melakukan konfigurasi pada *node* yang dikelola. Setiap Playbook memetakan sekelompok host ke ansible-role dan setiap role direpresentasikan dalam ansible-task.

Ansible memiliki dua istilah dalam arsitektur otomatisasinya, yaitu mesin kontroler (*controller*) dan *node/host*. Kontroler pada Ansible memiliki fungsi untuk melakukan manajemen atau mengatur setiap *node* yang terhubung dengannya melalui koneksi SSH (*Secure Shell*). Informasi alamat untuk memetakan setiap *node* disimpan pada file *ansible-host* atau pada file *ansible-inventory*. Karena Ansible berkomunikasi hanya dengan SSH antara kontroler dan *node*, maka Ansible memiliki arsitektur *agentless* atau tanpa agen yang artinya *node* yang terhubung ke Ansible tidak perlu memasang atau menjalankan agen dari Ansible sehingga lebih mudah dalam pengoperasiannya. Tidak seperti perangkat lunak otomatisasi lain seperti *Chef* dan *Puppet* yang membutuhkan agen untuk berkomunikasi antara kontroler dan *node* yang terhubung dengannya untuk melakukan otomatisasi atau manajemen konfigurasi

E. Visual Studio Code

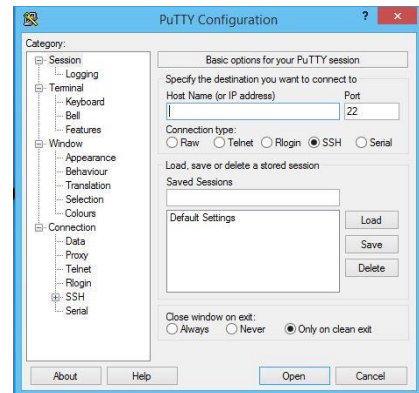
Visual Studio Code (VSCode) adalah suatu aplikasi kode editor berbasis *open-source* yang dikembangkan oleh Microsoft yang mendukung berbagai sistem operasi, seperti: Linux, Windows, MacOS. VSCode memiliki kelebihan mendukung berbagai bahasa pemrograman, seperti: Python, Java Script, PHP, C, C++, C#, dll. Selain itu VSCode juga mendukung bahasa pemrograman konfigurasi seperti YAML. VSCode juga memiliki fitur layanan terintegrasi dengan Github sehingga kode yang kita tulis bisa langsung kita unggah ke Github sebagai lumbung kode, selain itu VSCode memiliki kelebihan, yaitu pengguna bisa memasang atau menambahkan ekstensi secara mudah sesuai kebutuhan untuk menunjang performa dari pengguna dalam mengoperasikannya.



Gambar 3. Tampilan Utama Visual Studio Code

F. PuTTY

PuTTY adalah aplikasi sumber terbuka (open source) yang mendukung koneksi dengan banyak protokol jaringan, seperti SSH, Telnet, SCP, rlogin, serial port atau console [20]. PuTTY awalnya dibuat untuk Microsoft Windows, tetapi telah dikembangkan ke berbagai sistem operasi lain seperti Linux dan MacOS. PuTTY dibuat dan dikelola terutama oleh Simon Tatham, seorang programmer Inggris.



Gambar 4. Tampilan Utama PuTTY

G. Switch

Switch adalah perangkat jaringan komputer yang menghubungkan beberapa perangkat jaringan, seperti; komputer, router, access point, dll. Agar dapat saling berkomunikasi atau bertukar data informasi dari satu perangkat ke perangkat lainnya. Switch mengenali perangkat yang terhubung dengannya menggunakan protokol MAC (Media Access Control) Address pada tiap perangkat, sehingga pada saat ada suatu data atau informasi yang dikirim dari perangkat pengirim ke penerima, maka switch akan meneruskan data tersebut hanya ke MAC Address tujuan data tersebut, atau tidak disebar (broadcast) ke seluruh perangkat yang terhubung dengannya [7].

Switch dibagi menjadi dua jenis, yaitu switch manage dan switch unmanage. Switch manage adalah switch yang bisa dikonfigurasi lebih lanjut dengan memanfaatkan fitur yang ada pada switch tersebut sesuai keinginan pengguna, sedangkan switch unmanage adalah switch yang hanya bisa digunakan dan tidak bisa dikonfigurasi lebih lanjut..



Gambar 5. Switch TP-Link

H. Router

Router adalah perangkat jaringan yang memecah broadcast domain menjadi beberapa bagian, router juga berfungsi untuk menghubungkan dua segmen jaringan atau lebih [7]. Dengan protokol routing, router dapat menghubungkan jaringan komputer yang berbeda agar dapat saling berkomunikasi, router biasanya digunakan untuk menghubungkan Local Area Network (LAN) dengan Wide Area Network (WAN) atau jaringan internet atau biasa disebut dengan istilah router gateway.



Gambar 6. Router MikroTik

I. Personal Computer dan Laptop

Menurut American Standard Institute dan sudah didiskusikan serta sudah disetujui dalam suatu pertemuan Internasional Organization For Standardization Technical Commite, komputer adalah suatu pemroses data (data processor) yang dapat melakukan perhitungan yang besar dan cepat, termasuk perhitungan aritmatika yang besar atau operasi logika, tanpa campur tangan dari manusia untuk mengoperasikannya selama pemrosesan. Tujuan pokok dari sistem komputer adalah mengolah data untuk menghasilkan informasi. Supaya tujuan pokok tersebut terlaksana, maka harus ada elemen-elemen yang mendukungnya. Elemen-elemen tersebut brainware (manusia), hardware (perangkat keras) dan software (perangkat lunak). Ketiga elemen sistem komputer tersebut harus saling berhubungan dan membentuk satu kesatuan [22].

Dalam sebuah PC atau Laptop terdapat sebuah perangkat lunak sistem operasi yang bertugas mengatur dan menjalankan perangkat keras pada sistem komputer dan untuk menjembatani antara pengguna dan perangkat keras komputer [22].

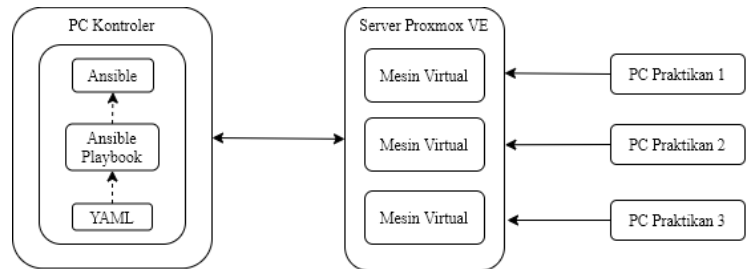


Gambar 7. Laptop Lenovo ThinkPad

III. METODOLOGI PENELITIAN

Pada bagian ini akan membahas mengenai perancangan alat mulai dari perancangan *software* dan perancangan *hardware*.

A. Diagram Blok

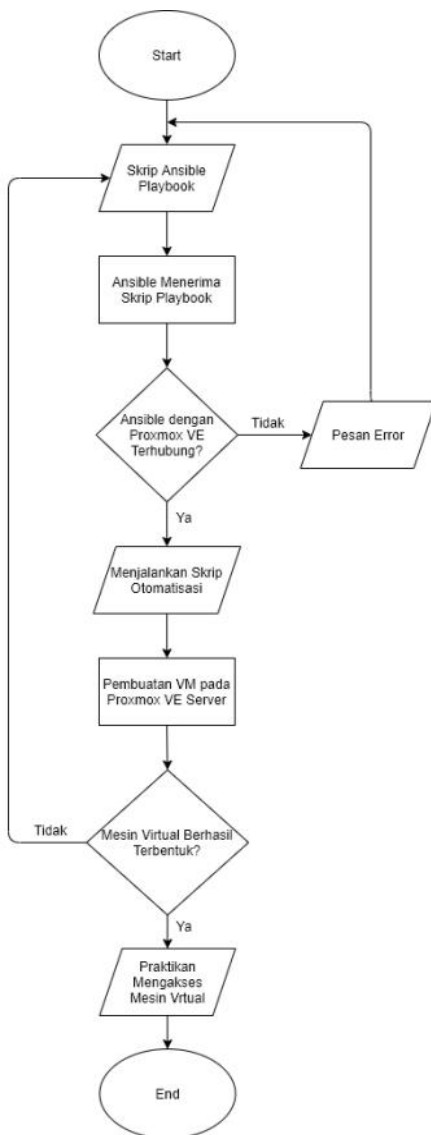


Gambar 8. Blok Diagram Sistem Perancangan Pembuatan Mesin Virtual dengan Ansible DI Proxmox VE Server

Prinsip kerja dari sistem ini, Ansible akan melakukan verifikasi koneksi antara kontroler dan server Proxmox VE kemudian Ansible akan menjalankan dan melakukan pengecekan skrip otomatisasi pembuatan mesin virtual di kontroler dan server Proxmox VE akan menerima skrip tersebut. Tampilan interface web browser Proxmox VE akan melakukan verifikasi mesin virtual sudah berhasil dibangun secara otomatis atau belum, jika mesin virtual sudah berhasil dibangun praktikan akan mengakses mesin virtual menggunakan software dari masing-masing komputer praktikan.

A. Flowchart Sistem

Diagram alir atau *flowchart* sistem ini menjelaskan alur mulai dari menjalankan dan pengecekan skrip ansible oleh sistem sampai mesin virtual berhasil dibangun di server Proxmox VE. Ketika skrip *Ansible-playbook* dengan ekstensi *YAML* berhasil dibuat, maka Ansible terlebih dahulu akan melakukan uji konektifitas antara kontroler dengan server Proxmox VE, jika konektifitas berhasil terbentuk maka Ansible akan menjalankan dan melakukan pengecekan skrip *Playbook* yang dikirim ke server Proxmox VE. Pada saat skrip *Playbook* ini dijalankan, terjadi proses pembuatan mesin virtual pada server Proxmox VE. Setelah selesai menjalankan skrip *Playbook*, Ansible akan menampilkan pesan bahwa skrip tersebut sukses dijalankan dan kontroler dapat melakukan verifikasi apakah mesin virtual sudah berhasil dibangun di Proxmox VE dengan cara mengakses *interface web browser* Proxmox VE dan praktikan akan mengakses mesin virtual tersebut.

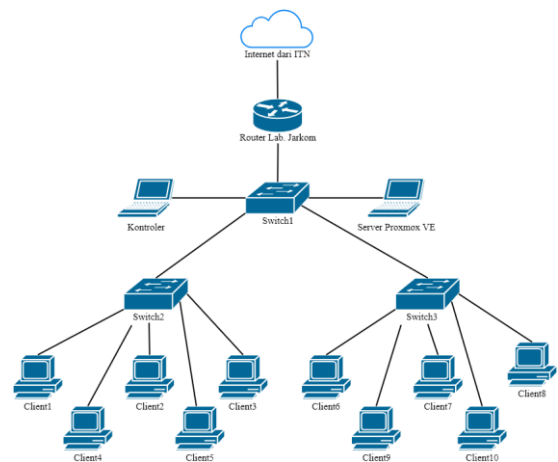


Gambar 9. Flowchart Sistem

B. Perancangan Perangkat Keras

Tabel 1. Kebutuhan Perangkat

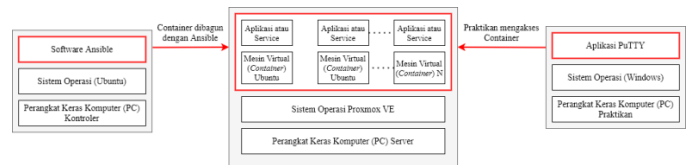
No	Perangkat	Kuantitas
1	Laptop Asus A455L	1
2	Laptop Lenovo ThinkPad X230	1
3	Switch	3
4	Router MikroTik RB941-2nD	1
5	Personal Computer (PC)	10



Gambar 10. Rancangan Topologi Jaringan

C. Blok Diagram Hubungan Antara Perangkat Keras dan Perangkat Lunak Dalam Sistem Yang Dirancang

Blok diagram hubungan antara perangkat keras dan perangkat lunak dalam sistem yang dirancang menggunakan sistem *client-server* dimana mesin virtual (*container*) dengan sistem operasi Ubuntu dibangun menggunakan *software* Ansible pada komputer kontroler di dalam server Proxmox VE. Mesin virtual yang berhasil terbentuk, akan diakses oleh masing-masing komputer praktikan dengan sistem operasi Windows menggunakan aplikasi PuTTY.



Gambar 11. Blok Diagram Hubungan Antara Perangkat Keras

D. Perancangan Perangkat Lunak

Perancangan perangkat lunak sistem ini terdiri dari tiga bagian utama yaitu program pembuatan dan manajemen mesin virtual pada VSCode, perancangan software Ansible pada kontroler dan Perancangan Server Proxmox VE. Pemrograman pada VSCode terdiri dari tujuh program pembuatan dan manajemen mesin virtual berektensi YAML yang dilanjutkan dengan perancangan software Ansible yang digunakan untuk membuat mesin virtual secara otomatis. Perancangan pada server dilakukan manajemen konfigurasi Proxmox VE sebagai sistem operasi utama yang digunakan untuk *virtualisasi* tempat pembuatan mesin virtual. Software Ansible akan menjalankan program YAML dan akan dikirimkan ke server Proxmox VE yang akan membangun mesin virtual secara otomatis. Mesin virtual yang berhasil dibangun akan diakses oleh praktikan melalui masing-masing komputer praktikan menggunakan aplikasi PuTTY dengan protokol SSH

IV. SIMULASI DAN ANALISA

A. Hasil Penggunaan Sumber Daya Pada Masing-Masing Mesin Virtual

Pada saat dilakukan pengujian untuk menjalankan mesin virtual dan membuat web server dengan WordPress pada masing-masing mesin virtual, diperoleh hasil penggunaan sumber daya pada masing-masing mesin virtual pada tabel 2

Tabel 2. Hasil Pengujian Sumber Daya Rata-Rata Pada Mesin Virtual

ID Mesin Virtual	Penggunaan Rata-Rata CPU	Penggunaan Rata-Rata Memory	Penggunaan Disk
101	36.58% dari 1 CPU	42.14% (215.73 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
102	36.50% dari 1 CPU	48.78% (249.76 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
103	36.52% dari 1 CPU	50.83% (260.24 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
104	36.49% dari 1 CPU	49.19% (251.84 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
105	36.42% dari 1 CPU	46.56% (238.40 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
106	36.42% dari 1 CPU	51.86% (265.52 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
107	36.56% dari 1 CPU	53.21% (272.43 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
108	36.51% dari 1 CPU	50.25% (257.27 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
109	36.49% dari 1 CPU	52.71% (269.86 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
110	36.50% dari 1 CPU	52.34% (267.98 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
111	36.56% dari 1 CPU	51.85% (265.46 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
112	36.58% dari 1 CPU	49.88% (255.39 MB dari 512 MB)	26.85% (1.30 GB dari 4.86 GB)
113	36.56% dari 1 CPU	50.54% (258.75 MB dari 512 MB)	26.90% (1.31 GB dari 4.86 GB)

Dari hasil penggunaan sumber daya pada masing-masing mesin virtual pada Tabel 4.1, penggunaan sumber daya pada *container* rata-rata 36% dari 1 CPU, 49% dari 512MB memori dan 26% dari 4,86GB *disk*, yang digunakan untuk membangun web server dengan web WordPress.

B. Hasil Pengujian Sumber Daya Pada Server Proxmox VE

Pada saat dilakukan pengujian dengan menjalankan semua mesin virtual yang berjumlah tiga belas mesin virtual, diperoleh hasil penggunaan sumber daya pada server Proxmox VE seperti pada tabel 3

Tabel 3. Hasil Pengujian Penggunaan Sumber Daya Pada Server Proxmox VE

Penggunaan Rata-Rata CPU	Penggunaan Rata-Rata Memory	Penggunaan Disk
4.15% dari 4 CPU	37.86% (4.33 GB dari 11.43 GB)	15.88% (22.30 GB dari 140.45 GB)

Dari hasil penggunaan sumber daya berdasarkan Tabel 3, server Proxmox VE dengan menggunakan Laptop Lenovo ThinkPad X230 dengan spesifikasi RAM: 12GB, SSD:

240GB, Prosesor: Intel Core i5-3320M, dapat digunakan sebagai server virtualisasi dengan Proxmox VE yang menjalankan 13 mesin virtual *container* dengan spesifikasi masing-masing *container* RAM: 512MB, Disk: 5GB, CPU: 1, dengan penggunaan sumber daya rata-rata 4,15% dari 4 CPU, 37,86% dari 11,43GB memori dan 15,88% dari 140,45GB *disk*.

C. Hasil Pengujian Penggunaan Maksimal Disk Pada Server Proxmox VE

Pada saat melakukan uji penggunaan maksimal *disk* pada server Proxmox VE, didapatkan nilai maksimal *container* yang bisa dibangun adalah 150 *container*. Dilakukan uji penggunaan *disk* dengan jumlah *container* yang telah ditentukan sehingga didapat hasil pada tabel 4.

Tabel 4. Penggunaan *Disk* Pada Server Proxmox VE Dengan Jumlah *Container* Yang Sudah Ditentukan.

Jumlah Container	Disk Local-LVM yang Digunakan	Status Container
30	19,88% (27,92 GB dari 140,45 GB)	Berjalan
60	39,77% (55,86 GB dari 140,45 GB)	Berjalan
90	59,65% (83,78 GB dari 140,45 GB)	Berjalan
120	79,52% (111,69 GB dari 140,45 GB)	Berjalan
150	99,39% (139,60 GB dari 140,45 GB)	Berjalan

Dari hasil yang diperlihatkan pada Tabel 4, penggunaan disk pada server Proxmox VE yang dijalankan pada perangkat dengan spesifikasi RAM: 12GB, SSD: 240GB, Prosesor: Intel Core i5-3320M. Server Proxmox VE pada sistem ini dapat dibangun 150 mesin virtual *container* dengan penggunaan sumber daya disk sebesar 99,39% dari 140,45GB. Kapasitas disk 140,45GB merupakan kapasitas yang dialokasikan oleh Proxmox VE untuk menyimpan *container* yang dibangun.

D. Hasil Pengujian Perbandingan Penggunaan Sumber Daya Pada Sistem Saat Ini Dan Sistem Yang Dikembangkan

Pada percobaan ini menggunakan dua komputer, dengan spesifikasi CPU AMD A4-500 APU, RAM 2 GB, sistem operasi Windows 10 Pro dan komputer dengan spesifikasi CPU Core i3-7100, RAM 4 GB, sistem operasi Windows 10 Pro.

Pengujian sistem saat ini menggunakan aplikasi VirtualBox yang di instal pada komputer praktikan dan dibuat mesin virtual Ubuntu Server 18.04 LTS. Pengujian pada sistem yang dikembangkan menggunakan mesin virtual pada Proxmox VE server menggunakan prinsip *client-server* dan menggunakan mesin virtual *container* dengan sistem operasi Ubuntu Server 18.04 LTS sehingga didapatkan hasil perbandingan seperti pada Tabel 5.

Tabel 5. Hasil Pengujian Perbandingan Penggunaan Sumber Daya Pada Sistem Saat Ini dan Sistem Yang Dikembangkan

Pengujian	Kapasitas RAM	Penggunaan CPU Pada Komputer	Penggunaan RAM Komputer
Sistem Saat Ini	2 GB	Rata-rata 23% - 30% dari 4 core	Rata-rata 1,4 GB (82%) dari 2 GB
	4 GB	Rata-rata 14% - 22% dari 4 core	Rata-rata 2,9 GB (73%) dari 4 GB
Sistem Yang Dikembangkan	2 GB	Rata-rata 5% - 10% dari 4 core	Rata-rata 0,8 GB (47%) dari 2 GB
	4 GB	Rata-rata 2% - 5% dari 4 core	Rata-rata 2,2 GB (55%) dari 4 GB

Dari hasil pengujian pada Tabel 5, penggunaan rata-rata sumber daya pada komputer praktikan menggunakan sistem saat ini dengan software VirtualBox dan sistem yang dikembangkan dengan Proxmox VE server memiliki perbedaan. Penggunaan rata-rata sumber daya pada sistem yang dikembangkan lebih rendah jika dibandingkan dengan sistem saat ini, sehingga komputer praktikan dengan spesifikasi rendah, seperti memiliki RAM 2 GB bisa digunakan untuk praktikum sistem operasi dengan penggunaan sumber daya yang tidak besar dan bisa dikembangkan lebih lanjut.

V. KESIMPULAN

Setelah dilakukan perancangan, perakitan dan pengujian, serta analisa data, maka dapat disimpulkan “Pengembangan Perangkat Praktikum Sistem Operasi Menggunakan Software Ansible Untuk Membangun Mesin Virtual Secara Otomatis Di Proxmox VE Server” yang diantaranya adalah:

1. Mesin virtual *container* yang telah dibangun, rata-rata membutuhkan sumber daya 36% dari 1 CPU, 49% untuk kapasitas memori dari nilai maksimum 512MB dan 26% dari kapasitas media penyimpanan (*disk*) dari maksimal 4,86GB. Untuk membangun web server dan web WordPress.
2. Penggunaan sumber daya pada server Proxmox VE rata-rata 4,15% dari 4 CPU, 37,86% dari memori 11,43GB dan 15,88% dari kapasitas penyimpanan (*disk*) 140,45GB. Yang menjalankan 13 *container* dengan web server dan web WordPress yang dibangun di dalam masing-masing *container*.
3. Penggunaan sumber daya pada komputer praktikan dengan menggunakan sistem yang dikembangkan dengan Proxmox VE server memiliki nilai yang lebih rendah jika dibandingkan dengan sistem yang dipakai saat ini dengan VirtualBox, sehingga komputer praktikan dengan spesifikasi RAM 2 GB bisa digunakan untuk praktikum dengan penggunaan sumber daya yang lebih rendah.
4. Server Proxmox VE yang digunakan dalam sistem dapat membangun dan menjalankan 150 *container* dengan penggunaan *disk* maksimal 99,39% .
5. Mesin virtual *container* yang dibangun dapat diakses secara bersamaan/simultan oleh komputer praktikan menggunakan protokol SSH.
6. Praktikan dapat membangun web server dan web WordPress di mesin virtual *container* yang telah

dibangun dengan perintah-perintah pada *command line interface* (CLI) Linux Ubuntu.

Saran

Penulis menyadari bahwa dalam pembuatan penelitian ini tidak terlepas dari kesalahan dan kekurangan. Baik dalam penulisannya, penjelasan laporan, maupun perancangan dan implementasi sistem. Maka dari itu, penulis memberi beberapa saran agar sistem ini dapat dikembangkan menjadi lebih baik, diantaranya yaitu:

1. Dalam sistem pengembangan perangkat praktikum sistem operasi menggunakan software Ansible dan server Proxmox VE ini sebaiknya menggunakan perangkat khusus server sebagai Proxmox VE servernya.
2. Untuk pengujian penggunaan mesin virtual *container*, bisa ditambahkan lagi untuk menguji kehandalan mesin virtual yang dibangun.

VI. DAFTAR PUSTAKA

- [1] Abdurrahman., Soni., & Hafid, Afdhil. 2019. Optimalisasi Sumber Daya Komputer Dengan Virtualisasi Server Menggunakan Proxmox Ve. Jurnal Fasilkom Volume 9 No.2. ISSN: 2089-3353.
- [2] Adiputra, Firmansyah. 2015. Container Dan Docker: Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi Web. Jurnal SimanteC Vol.4, No. 3 Juni 2015. ISSN: 2088-2130.
- [3] Agustian, Bobi., dan Susanto, Dodi. 2018. Virtualisasi Server Dengan Proxmox Untuk Mengoptimalkan Penggunaan Server. Prosiding Seminar Nasional Informatika dan Sistem Informasi Volume 3, Nomor 2, Jul - Okt 2018. ISSN: 2549-4805.
- [4] Ben-Kiki, Oren, dkk. 2009. “YAML Ain’t Markup Language (YAML) Version 1.2 ”. 3rd Edition, Diperbaharui: 01-10-2009.
- [5] Chandra, Albert Y. 2019. Analisis Performansi Antara Apache & Nginx Web Server dalam Menangani Client Request. Jurnal Sistem Dan Informatika (JSI). p-ISSN: 1858-473X, e-ISSN: 2460-3732, DOI: 10.30864/jsi.v14i1.248.
- [6] J., Barrett, Daniel and Silverman, Richard. 2001. “SSH, The Secure Shell: The Definitive Guide”. Publisher: O’Reilly, ISBN: 0-596-00011-1.
- [7] Lammle, Todd. 2016. “CCNA Routing and Switching Complete Study Guide Second Edition”. Copyright: 2016, John Wiley & Sons, Inc., Indianapolis, Indiana. ISBN: 978-1-119-28828-2.
- [8] Maksum, A., dan Listiyoko, L. 2017. Implementasi Proxmox High Availability Server untuk Meningkatkan kinerja Website kampus., eISSN: 2527-6050.
- [9] Masek, P., Stusek, M., Krejci J., Zeman K., Pokorný J., & Kudlacek M. 2018. Unleashing Full Potential of Ansible Framework: University Labs Administration. Conference Paper: May 2018. DOI: 10.23919/FRUCT.2018.8468270.
- [10] Maulana, Halim. 2016. Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source. Jurnal Nasional Informatika dan Teknologi Jaringan, Vol. 1, No. 1. e-ISSN: 2540-7600, p-ISSN: 2540-7597.
- [11] Novianto, Dian. 2015. Pemfilteran Hypertext Transfer Protocol Secure Untuk Penggunaan Internet Yang Aman. Jurnal Informatika, Vol. 15, No.1, Bulan Juni 2015.

- [12] Oualline, Steve. 2001. "Vi iMproved (VIM)". Publisher: Sams 2001, ISBN/ASIN: 0735710015, ISBN-13: 9780735710016.
- [13] Prakoso, Ruli Dimas & Asmunin. 2018. Implementasi Dan Perbandingan Performa Proxmox Dalam Virtualisasi Dengan Tiga Virtual Server. Jurnal Manajemen Informatika Volume 8 Nomor 01 Tahun 2018, 79-85.
- [14] Pramudya, D. B., Ijtihadie, R. M., dan Pratomo, B. A. 2016. Rancang Bangun Aplikasi Simulasi Jaringan untuk Praktikum Jaringan Komputer Memanfaatkan Scapy dalam Lingkungan LXC. Jurnal Teknik ITS Vol. 5, No. 1, (2016) ISSN: 2337-3539 (2301-9271 Print)
- [15] Putra, R. A. dan Sujana, A. P. 2019. Implementasi Cluster Server Pada Raspberry Pi dengan Menggunakan Metode Load Balancing. Komputika: Jurnal Sistem Komputer. Vol. 8, No. 1, April 2019, hlm. 37 - 43. DOI: 10.34010/komputika.v8i1.1623. p-ISSN: 2252-9039, e-ISSN: 2655-3198.
- [16] Rasian, Rio., dan Mursanto, Petrus. 2012. Perbandingan Kinerja Pendekatan Virtualisasi. Jurnal Sistem Informasi: July 2012. DOI: 10.21609/jsi.v5i2.267.
- [17] Risdanto, Batara. 2014. Pengembangan E-Learning Berbasis Web Menggunakan CMS (Content Management System) WordPress Di SMA Negeri 1 Kota Magelang. Program Studi Pendidikan Teknik Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta.
- [18] Suhatman, Rahmat. 2016. Analisa Performansi Server Cloud Berbasis Proxmox Ve untuk Multi Server dan Multi Platform pada Praktikum Administasi Jaringan Komputer. Jurnal Komputer Terapan,. Vol. 2, No. 1, Mei 2016, 17-26.
- [19] Suryono, Tito., dan Afif, M.F. 2012. Pembuatan Prototype Virtual Server Menggunakan Proxmox Ve Untuk Optimalisasi Resource Hardware Di NOC FKIP UNS. IJNS – Volume 1 Nomor 1 – November 2012. ISSN: 2302-5700.
- [20] Tatham, Simon. 2017. "PuTTY User Manual". Copyright 1997-2017, Dokumentasi dibawah lisensi MIT.
- [21] Usada, Elisa, dkk. 2012. Rancang Bangun Sistem Informasi Jadwal Perkuliahan Berbasis JQuery Mobile Dengan Menggunakan Php Dan MySQL. Jurnal Infotel Volume 4, Nomor 2.
- [22] Wimatra, Ayub, dkk. 2008. "Dasar – Dasar Komputer". Akademi Teknik dan Keselamatan Penerbangan Medan, 2008.
- [23] Wiryawan, Rheza. A. dan Rosyid, Nur. R. 2019. Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python. Jurnal SIMETRIS, Vol. 10 No. 2 November 2019. P-ISSN: 2252-4983, E-ISSN: 2549-3108