

PERANCANGAN *VIRTUAL COMPUTER NETWORK* MENGGUNAKAN COMMAND LINE INTERFACE

I Made Darma Susila
STMIK STIKOM BALI
Jl. Raya Puputan Renon No. 86 Denpasar
Telp. (0361) 244445
darma_s@stikom-bali.ac.id

I Gede Putu Krisna Juliharta
STMIK STIKOM BALI
Jl. Raya Puputan Renon No. 86 Denpasar
Telp. (0361) 244445
krisna@stikom-bali.ac.id

Virtualisasi berbasis command line di linux merupakan sebuah teknologi virtualisasi yang mampu menjalankan sebuah sistem operasi sebagai sebuah aplikasi di atas sebuah sistem operasi linux yang berperan sebagai induk atau host. Hal ini dapat digunakan sebagai solusi dari banyaknya perangkat server yang dibutuhkan untuk menjalankan aplikasi. Dengan bantuan user mode linux dibuat penelitian yang tujuannya membangun virtual server yang ringan dan berbasis open source. Hasil dari penelitian yang dibuat adalah dibuat tiga buah server yaitu domain name server, web server, dan DHCP server didalam satu server induk yang berbasis ubuntu

Kata Kunci : virtualisasi, user mode linux, server

I. PENDAHULUAN

I.1. Latar Belakang

Sekolah, Universitas, Kantor-kantor pemerintahan, perusahaan-perusahaan negara dan swasta menjadikan saat ini komputer sebagai ujung tombak dalam menjalankan segala proses belajar mengajar maupun segala proses bisnis yang ada. Akses layanan yang cepat menjadi suatu tuntutan pokok bagi setiap perusahaan atau perorangan dalam menjalankan setiap proses bisnisnya. Hal ini yang menyebabkan banyaknya kebutuhan akan komputer server. Semakin banyaknya komputer server tentunya menimbulkan kendala seperti, ruangan yang luas, kebutuhan daya yang tinggi, dan mesin pendingin yang memadai. Dilain sisi dengan spesifikasi server yang tinggi, komputer server memberikan layanan yang tidak terlalu berat, sehingga menimbulkan kontradiksi.

Setelah ditemukannya konsep Mesin Virtual (Virtual Machine) munculah teknologi yang mampu menjalankan beberapa Sistem Operasi secara bersamaan di dalam satu buah komputer server yang disebut dengan "Virtual Server". Dengan kehadiran teknologi ini maka permasalahan tempat penyimpanan dan konsumsi energi listrik dapat dihemat secara signifikan.

Ada banyak teknologi Virtualisasi yang tersedia di sistem operasi Linux, bahkan virtualisasi dapat dibangun menggunakan command line. Virtualisasi berbasis command line di linux merupakan sebuah teknologi virtualisasi yang mampu menjalankan sebuah sistem operasi sebagai sebuah aplikasi di atas sebuah

sistem operasi linux yang berperan sebagai induk atau host. Konsep ini sangat unik dan berbeda dengan teknologi virtualisasi lainnya seperti XEN, VirtualBox, KVM, QEMU dan LXC yang mana mesin virtual berjalan sebagai proses biasa pada user space di linux yang menyebabkan mesin virtual terisolasi penuh dari host, sehingga keamanan teknologi virtualisasi ini menjadi sangat tinggi bila dibandingkan dengan teknologi virtualisasi lainnya. Namun terdapat beberapa kekurangan dari teknologi virtualisasi ini antara lain : proses set-up yang sulit serta performa mesin virtual yang rendah.

Dalam sebuah host server virtual yang menggunakan sistem operasi linux biasanya tidak menyediakan antarmuka grafis (GUI). Semua interaksi terhadap sistem menggunakan antarmuka text melalui aplikasi Terminal. Ketidakterdediaan antarmuka grafis ini disebabkan oleh beberapa hal antara lain : Pengoptimalisasian penggunaan sumber daya pada mesin server, dengan menghilangkan GUI maka sumber daya yang digunakan host lebih sedikit.

I.2. Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari perancangan dan Virtual Server Berbasis Command Line adalah untuk membuat perancangan sistem yang melakukan otomatisasi pembuatan, penghapusan dan manajemen mesin virtual

I.3. Rumusan Masalah

Dalam perancangan dan pembangunan Implementasi virtual computer network berbasis command line interface, diperlukan adanya ruang lingkup agar perancangan dan pembangunan bisa terarah dan fokus pada tujuan yang diinginkan. Adapun ruang lingkup dalam perancangan dan pembangunan Virtual Server Berbasis Command Line dengan metode User Mode Linux antara lain:

1. Perancangan virtual server dan guest server dilakukan menggunakan diagram alir atau flow chart
2. Perancangan virtual server yang dibuat adalah web server virtual, domain names server virtual, dan DHCP server virtual.

II. LANDASAN TEORI

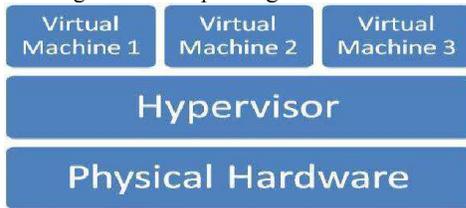
II.1. Virtualisasi

Virtualisasi merupakan sebuah kerangka (framework) atau metodologi pembagian sumber daya

sebuah komputer ke dalam beberapa lingkungan eksekusi dengan cara menerapkan satu atau beberapa konsep dan teknologi seperti pemartisian perangkat lunak, Time-sharing, simulasi mesin secara sebagian atau menyeluruh, emulasi, QOS dan lainnya. Virtualisasi memungkinkan adanya abstraksi dan isolasi dari fungsi-fungsi tingkat rendah dan perangkat keras yang mendasar. Hal ini memberikan portabilitas pada fungsi-fungsi tingkat tinggi dan pembagian serta agregasi sumber daya fisik.[1]

Dalam kegiatan berkomputer setiap hari secara tidak sadar kita sudah menggunakan teknologi virtualisasi, sistem operasi melakukan virtualisasi terhadap sumber daya perangkat keras sehingga komputer bisa menjalankan beberapa aplikasi secara bersamaan, hal ini juga mempermudah para Programmer dalam membuat aplikasi, program tidak perlu mengetahui / menulis kode tentang manajemen memori, disk, berkas dan lainnya. Itu semua adalah tugas sistem operasi. Lain halnya dengan virtualisasi perangkat keras.

Virtualisasi Perangkat Keras memungkinkan untuk menjalankan beberapa sistem operasi secara bersamaan di dalam sebuah perangkat keras, yang mana sistem operasi yang di-virtualisasikan tidak menyadari dirinya sedang berjalan di dalam mesin virtual. Sistem operasi yang berjalan di dalam lingkungan virtualisasi biasa disebut dengan Sistem operasi guest.



Gambar 2.1 :Konsep Dasar Virtualisasi

II.2. Metode Pendekatan Virtualisasi

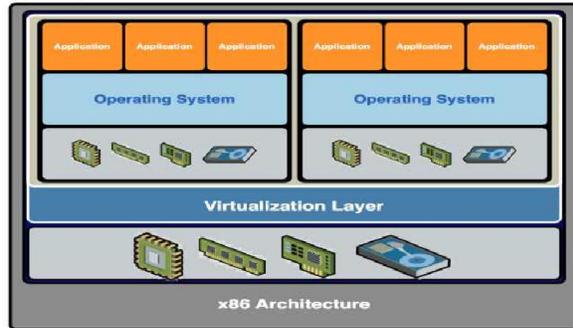
Dalam melakukan proses virtualisasi terdapat beberapa pendekatan yang digunakan, tiap metode mempunyai kelebihan dan kekurangannya. Ada 3 metode pendekatan virtualisasi yaitu : Emulasi Perangkat Keras, Paravirtualisasi dan Virtualisasi sistem operasi atau yang lebih dikenal dengan istilah Container.

1. Emulasi Perangkat Keras

Pada metode virtualisasi Emulasi perangkat keras, perangkat lunak virtualisasi (biasanya disebut dengan hypervisor) membuat dan menyediakan sebuah lingkungan perangkat keras yang di-emulasi. Lingkungan ini biasanya disebut dengan VMM (Virtual Machine Monitor). Sistem operasi yang di-virtualisasikan nantinya akan berkomunikasi dengan VMM bukan dengan perangkat keras yang sesungguhnya. VMM mengoordinasikan akses antara sistem operasi guest dan perangkat keras.[2]

Metode emulasi ini bekerja dengan cara sebagai berikut : saat sistem operasi guest memberikan instruksi, hypervisor mengubah instruksi yang diberikan oleh sistem operasi guest kepada VMM menjadi sebuah instruksi native kemudian menjalankannya di perangkat keras yang sesungguhnya, sehingga mampu

menjalankan sebuah sistem operasi guest tanpa memodifikasi sistem operasi guest itu sendiri. Karena setiap instruksi yang berjalan harus diterjemahkan terlebih dahulu, metode Emulasi Perangkat Keras ini menjadi agak lambat dibandingkan dengan metode virtualisasi lainnya. Secara garis besar metode virtualisasi Emulasi Perangkat Keras digambarkan seperti gambar di bawah ini :



Gambar 2.2 : Emulasi Perangkat Keras

Metode virtualisasi ini memanfaatkan arsitektur teknis prosesor x86 yang sudah terkenal untuk membuat perangkat lunak emulasi dari chip x86 tersebut, karena komoditas chip x86 digunakan secara luas, metode virtualisasi ini menjadi sebuah pilihan yang ekonomis bila dilihat dari sudut pandang ekonomi.

Oleh karena perangkat lunak virtualisasi ini (hypervisor) menyediakan sebuah antarmuka yang konsisten terhadap sistem operasi guest dan memegang tanggung jawab untuk memperhitungkan bagaimana sistem operasi guest berkomunikasi dengan perangkat keras yang sebenarnya, metode virtualisasi ini menjadi sangat fleksibel. Hal ini mengizinkan sistem operasi guest yang dipasang pada satu hypervisor dipindahkan dan dijalankan pada hypervisor lainnya.

Penerapan yang sering menggunakan metode Emulasi perangkat keras adalah dalam pembangunan perangkat lunak dan jaminan kualitas sebab metode ini mengizinkan beberapa sistem operasi yang berbeda berjalan secara bersamaan sehingga mampu untuk memfasilitasi pengujian perangkat lunak pada berbagai sistem operasi. Emulasi perangkat keras juga digunakan dalam konsolidasi Server, dimana sejumlah lingkungan sistem operasi dan aplikasinya dipindahkan dari Server fisik yang terpisah ke dalam sebuah Server fisik yang menjalankan perangkat lunak virtualisasi.

2. Paravirtualisasi

Paravirtualisasi merupakan sebuah metode virtualisasi yang memiliki sedikit perbedaan dengan Emulasi perangkat keras, paravirtualisasi memiliki konsep : daripada mengemulasikan seluruh perangkat keras secara lengkap lebih baik membuat sebuah lapisan tipis yang berfungsi untuk mengoordinasikan akses oleh sistem operasi guest terhadap sumber daya perangkat keras. Dengan kata lain paravirtualisasi tidak membuat sebuah mesin virtual secara menyeluruh untuk menampung sistem operasi guest, melainkan mengizinkan sistem operasi guest berinteraksi secara langsung dengan hypervisor [2].

Pendekatan ini memiliki dua keunggulan pertama, metode ini memiliki performa yang lebih baik dibandingkan dengan emulasi perangkat keras karena metode ini mempunyai jumlah kode yang relatif sedikit. Emulasi perangkat keras memasukan sebuah lapisan emulasi perangkat keras secara keseluruhan antara sistem operasi guest dan perangkat keras, lain halnya dengan paravirtualisasi dimana lapisan tipis kode yang berada di antara sistem operasi guest dan perangkat keras bertindak sebagai “polisi lalu lintas” yang memberikan akses sumber daya fisik kepada satu sistem operasi guest dan menghentikan semua sistem operasi guest untuk mengakses sumber daya yang sama pada waktu yang bersamaan.

Keunggulan kedua dari metode ini adalah, paravirtualisasi tidak membatasi driver perangkat keras dimana pada virtualisasi emulasi perangkat keras, penggunaan driver dibatasi hanya dapat menggunakan driver yang tersedia pada perangkat lunak virtualisasi. Bahkan dalam pendekatan paravirtualisasi ini tidak menyediakan driver sama sekali, melainkan metode ini menggunakan driver yang ada pada sistem operasi guest. Hal ini mengizinkan sistem operasi guest untuk menggunakan seluruh kemampuan perangkat keras secara efisien jika dibandingkan dengan metode emulasi perangkat keras yang membatasi driver perangkat keras. Gambaran mendasar paravirtualisasi dapat dilihat pada gambar di bawah ini :



Gambar 2.3 : Konsep dasar Paravirtualisasi

Pendekatan virtualisasi ini memanfaatkan kemampuan sistem operasi yang disebut dengan shared memory untuk mencapai performa yang tinggi dalam pengoperasiannya, shared memory adalah memori yang bisa diakses oleh dua program yang berbeda. Paravirtualisasi menggunakan shared memory untuk mengirim data dari dan ke sistem operasi guest antar sistem operasi guest dan juga pada hypervisor yang memungkinkan tercapainya performa tinggi.

Selain kelebihan yang dimiliki, metode pendekatan virtualisasi ini juga memiliki kekurangan yang cukup signifikan, karena paravirtualisasi begitu ringan dan merupakan sebuah lapisan tipis, metode ini memerlukan bantuan pada beban kerja yang berat, dengan kata lain, paravirtualisasi mengharuskan sistem operasi guest untuk dimodifikasi sebelum dapat dijalankan dan berinteraksi dengan antarmuka paravirtualisasi. Hal ini bisa hanya dilakukan dengan memiliki akses terhadap sumber kode dari sistem operasi guest. Akses terhadap sumber kode sistem operasi hanya dimungkinkan pada sistem operasi Open source seperti Linux dan BSD, tetapi hal ini tidak mungkin dilakukan pada sistem operasi komersial seperti Microsoft Windows. Ketidak-

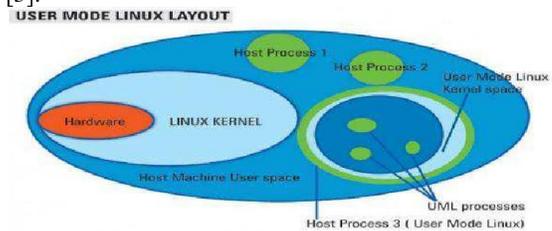
mampuan paravirtualisasi menjalankan sistem operasi dari Microsoft merupakan kekurangan yang sangat signifikan bagi beberapa organisasi yang tergantung dengan sistem operasi Microsoft ini. Tetapi dengan kehadiran chip baru dari Intel dan AMD yang menyematkan fitur virtualisasi yang disebut dengan VT pada prosesor intel dan AMD-V pada prosesor AMD hal ini dapat diatasi. Fitur virtualisasi pada prosesor ini mengizinkan sistem operasi yang tidak dimodifikasi dapat dijalankan di atas sebuah hypervisor paravirtualisasi.

Chip baru dari Intel dan AMD sering disebut dengan istilah virtualization-enabled yang berarti chip-chip ini memiliki fitur tambahan yang memungkinkan beberapa fungsi dari sebuah hypervisor dipindahkan ke dalam chip prosesor. Dengan dipindahkannya beberapa fungsi hypervisor ke chip, menyebabkan virtualisasi dengan menggunakan chip ini menjadi lebih cepat.

II.3. User Mode Linux (UML)

User Mode Linux merupakan salah satu metode virtualisasi yang merupakan penerapan dari pendekatan virtualisasi sistem operasi. UML secara eksklusif hanya mendukung sistem operasi Linux. UML adalah sebuah mesin Linux virtual yang berjalan di atas sistem operasi Linux, secara teknis UML merupakan port dari Linux ke Linux yang memiliki arti yang sama dengan Linux di-port ke berbagai macam arsitektur CPU seperti : ubiquitous x86, SUN SPARC, IBM PowerPC, DEC Alpha dan masih banyak lagi yang lainnya. Perbedaananya adalah UML merupakan sebuah Port dari Linux yang antarmukanya didefinisikan oleh Linux tidak seperti Port Linux ke arsitektur CPU lain yang antarmukanya didefinisikan oleh perangkat keras CPU itu sendiri.

UML memiliki banyak kegunaan bagi administrator sistem, pengguna maupun pengembang perangkat lunak. Mesin virtual UML sangat berguna untuk dijadikan lingkungan penelitian yang bisa dibangun dengan cepat dan dibuang ketika tidak dibutuhkan lagi dan lingkungan produksi yang menggunakan perangkat keras secara efisien serta serta lingkungan pengembangan perangkat lunak yang memberikan kemudahan untuk melakukan testing atau percobaan [3].



Gambar 2.4 : Konsep dasar User Mode Linux (UML)

II.4. Perbandingan UML dengan teknologi Virtualisasi lainnya

UML berbeda dengan metode virtualisasi lainnya dimana UML lebih mengarah ke sistem operasi virtual daripada sebuah mesin virtual. Tidak seperti teknologi lain misalnya VMWare yang merupakan

benar-benar sebuah mesin virtual dimana perangkat keras di emulasikan secara lengkap dan bisa menjalankan berbagai sistem operasi di dalamnya, UML hanya bisa berjalan di atas sistem operasi Linux, atau hanya bisa menjadi sistem operasi guest di sistem operasi Linux. Sebagai sistem operasi virtual UML memiliki keunggulan mampu berinteraksi secara penuh dengan sistem operasi induk (Host OS).

Bila dibandingkan dengan metode virtualisasi sistem operasi lainnya seperti LXC, XEN, BSD Jails dan Solaris Zones yang menggunakan chroot sebagai metode untuk menjalankan sistem operasi yang reintegrasi ke dalam sistem, UML berjalan sebagai sebuah proses tersendiri pada user-space di Linux serta memiliki Kernel sendiri. Ini memberikan keuntungan bagi UML karena berjalan secara independen dari sistem operasi induk dengan dampak menurunnya performa sistem operasi virtual [3].

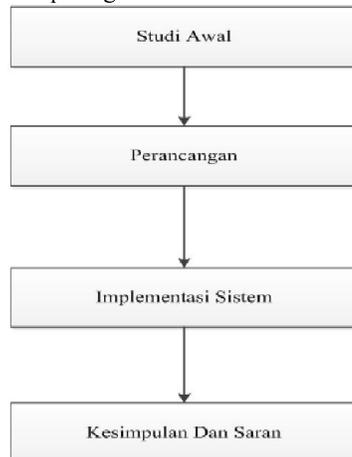
III. METODE PENELITIAN

III.1. Tempat dan Waktu Penelitian.

Tempat dan waktu penelitian dilakukan di STMIK STIKOM Bali, dimulai dari bulan februari 2014 hingga agustus Agustus 2014.

III.2. Perancangan Penelitian

Dalam melakukan penelitian ini, penulis melakukan langkah-langkah penelitian yang dapat digambarkan pada gambar 3.1 :



Gambar 3.1. Langkah Penelitian

- a. Studi awal
Dalam melakukan studi awal, penulis melakukan : pencarian materi, perancangan teknik-teknik tentang virtualisasi server yang telah ada, mempelajari karakteristik, kelebihan, dan kekurangan dari virtualisasi tersebut.
- b. Perancangan
Pada tahapan ini, penulis melakukan perancangan flow chart sistem dari virtualisasi yang akan dibuat sehingga terlihat bagaimana

proses pembuatan dari awal sampai menjadi DNS virtual dan web server virtual

- c. Implementasi
Pada tahap ini dilakukan proses pembuatan sistem dengan melakukan instalasi user mode linux dan melakukan konfigurasi sesuai dengan permasalahan yang telah di deskripsikan.
- d. Kesimpulan dan Saran.

Pada tahap akhir penulis membuat kesimpulan dan saran dari semua proses penelitian yang dilakukan.

III.3. Instrument Penelitian

Alat penelitian yang akan digunakan dalam penelitian dibagi dalam dua bagian utama yaitu:

a. Perangkat Lunak

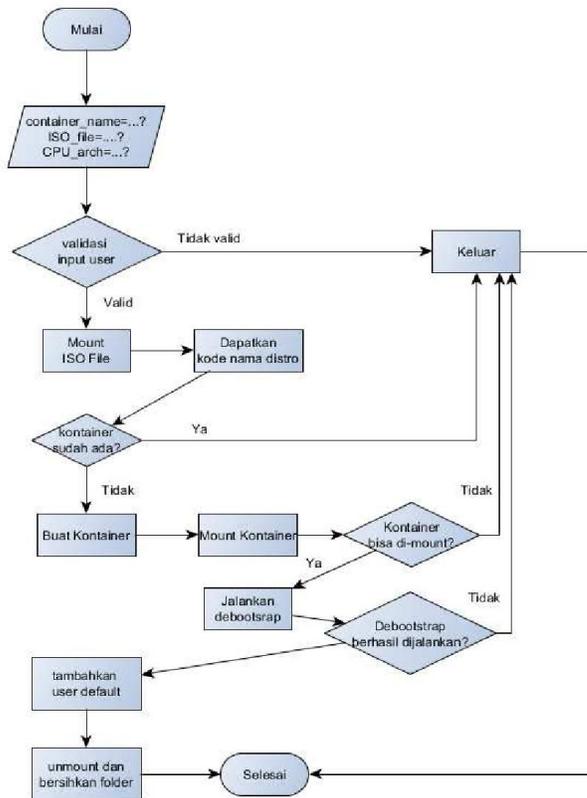
Perangkat lunak yang digunakan dalam perancangan sistem terdiri dari dua bagian yaitu :

- Sistem Operasi Ubuntu 12.04 LTS
- User Mode Linux

IV. hasil dan Pembahasan

IV.1. Diagram Alir Pembuatan Countainer

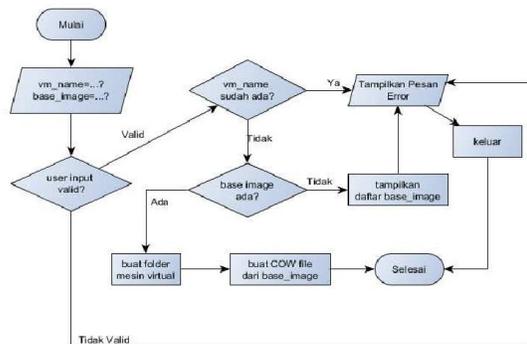
Gambar 4.1 merupakan flowchart fungsi create_container, Diagram alir (Flowchart) ini secara garis besar proses pembuatan container adalah sebagai berikut : pertama-tama file kosong dibuat pada direktori yang ditentukan pada variabel BASE_IMAGE_DIR, lalu diformat dengan sistem berkas ext3 dan dikaitkan (mount) pada direktori yang ditentukan oleh variabel CONT_MOUNT kemudian program debootstrap dijalankan dengan folder tujuan diarahkan ke folder folder pada variabel CONT_MOUNT.



Gambar 4.1 : Flowchart fungsi create_container

IV.2. Diagram alir Virtual Machine

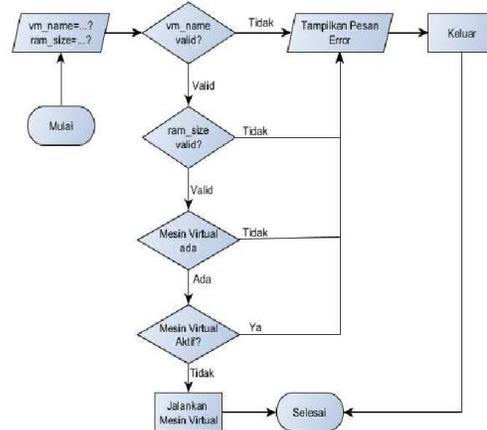
Masukan dari pengguna dari skrip ini berupa nama container, lokasi file ISO dari sistem operasi dan arsitektur CPU, untuk saat ini arsitektur CPU yang didukung masih terbatas pada arsitektur CPU amd64 atau 64bit. Setelah menerima masukan dari pengguna skrip akan mengecek kode nama distro dari file ISO sistem operasi yang dimasukan user, setiap distribusi sistem operasi Linux mempunyai kode nama distro, misalnya Ubuntu 10.04 memiliki kode nama Lucyd Lynx dan Debian 7 mempunyai kode nama wheezy. Jika terjadi eror atau kesalahan pada eksekusi program, skrip akan melakukan fungsi pembersihan / clean-up dan menghentikan eksekusi (gambar 4.2)



Gambar 4.2 Membuat Virtual Machine

IV.3. Diagram Alir start_vm()

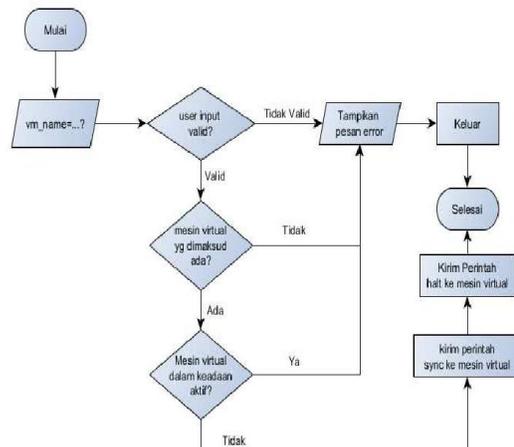
Gambar 4.3 merupakan flowchart (diagram alir) fungsi start_vm, fungsi ini menjalankan mesin virtual yang dimaksud, masukan user berupa nama mesin virtual yang akan dijalankan dan jumlah RAM yang diberikan pada mesin virtual yang bersangkutan, jika mesin virtual tidak ditemukan atau mesin virtual sudah dalam keadaan aktif, pesan eror ditampilkan, tetapi jika mesin virtual yang dimaksudkan ada, jumlah ram yang diberikan tepat serta mesin virtual sedang tidak dalam keadaan aktif maka skrip akan menjalankan mesin virtual yang dimaksud.



Gambar 4.3 : Flowchart start_vm

IV.4. Diagram alir stop_vm()

Gambar 4.4 merupakan flowchart fungsi stop_vm, Flowchart ini menjelaskan tentang fungsi untuk menghentikan mesin virtual yang sedang berjalan, masukan user berupa nama mesin virtual yang akan dihentikan. Jika masukan user tidak valid, mesin virtual tidak ditemukan atau mesin virtual tidak sedang dalam keadaan aktif, pesan eror ditampilkan. Dan jika masukan user valid, mesin virtual yang dimaksud ada dan mesin virtual sedang dalam keadaan aktif skrip akan menjalankan proses mematikan mesin virtual dengan pertama-tama mengirimkan perintah sync ke mesin virtual dan kemudian perintah halt untuk mematikan mesin virtual.

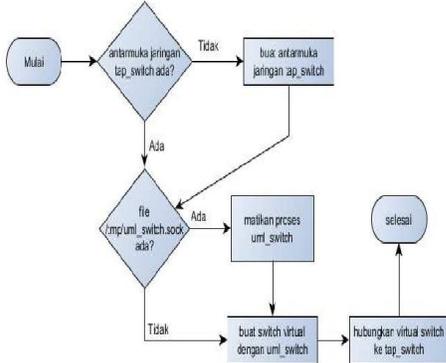


Gambar 4.4 : Flowchart stop_vm

IV.5. Flowchart fungsi setup_switch()

Gambar 4.5 merupakan flowchart fungsi setup_switch, Flowchart ini menjelaskan tentang fungsi untuk penambahan switch virtual pada skrip BASH yang telah dibuat, fungsi ini menambahkan sebuah switch virtual di dalam sistem operasi host yang nantinya akan dihubungkan dengan mesin virtual. Switch ini memanfaatkan *unix socket* sebagai media berkomunikasi pada skrip ini *unix socket* yang digunakan diset ke /tmp/uml_switch.sock.

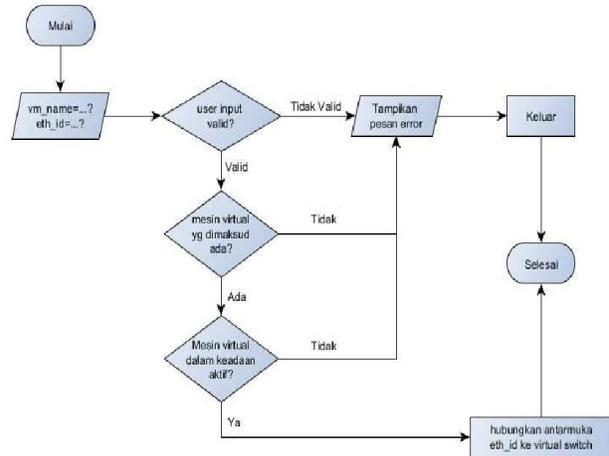
Jika antarmuka jaringan *tap_switch* tidak ada maka skrip akan membuat sebuah antarmuka jaringan berbasis TUN/TAP dengan nama *tap_switch* dan setelah antarmuka jaringan *tap_switch* dibuat ataupun antarmuka jaringan ini sudah ada maka selanjutnya skrip melakukan pengecekan terhadap keberadaan berkas /tmp/uml_switch.sock jika berkas ada maka skrip akan menghentikan program *uml_switch* dan membuat ulang *switch* virtual dan kemudian menghubungkan *switch* virtual ini dengan antarmuka jaringan *tap_switch*.



Gambar 4.5 : Flowchart setup_switch

IV.6. Flowchart fungsi attach_to_switch()

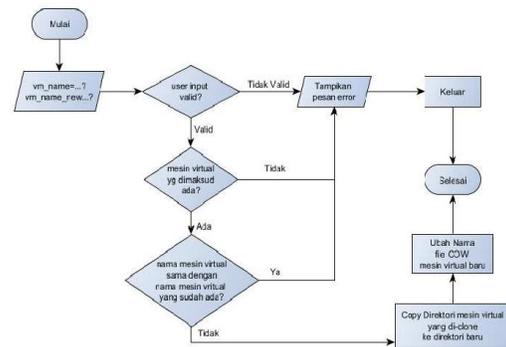
Gambar 4.6 merupakan flowchart fungsi attach_to_switch, Flowchart ini menjelaskan tentang fungsi untuk penghubungan mesin virtual ke switch virtual pada skrip BASH yang telah dibuat, fungsi ini menghubungkan mesin virtual ke switch virtual dengan terlebih dahulu membuat antarmuka jaringan pada mesin virtual, membuat antarmuka jaringan pada sistem operasi host dengan TUN/TAP, lalu menghubungkan antarmuka jaringan pada host dan mesin virtual dengan switch yang telah dipersiapkan melalui fungsi setup_switch() dan yang terakhir menghubungkan antarmuka.



Gambar 4.6 : Flowchart fungsi attach-to-switch

IV.7. Flowchart fungsi clone_vm()

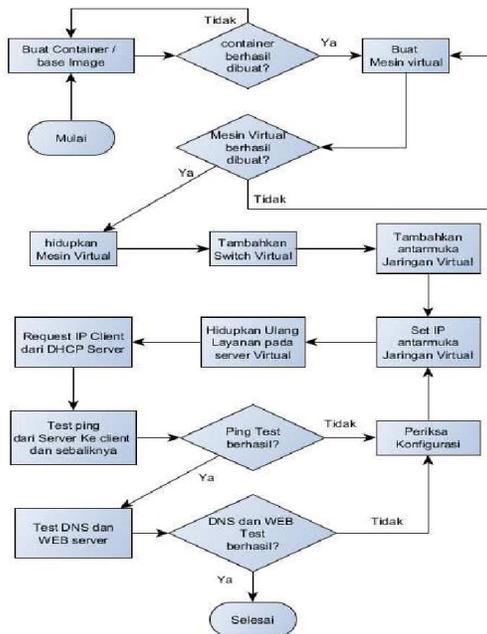
Gambar 4.7 merupakan flowchart fungsi clone_vm, Flowchart ini menjelaskan tentang fungsi untuk penggandaan mesin virtual pada skrip BASH yang akan dibuat, fungsi ini menggandakan mesin virtual yang ada. Isi dan *container* yang diacu oleh mesin hasil penggandaan sama persis tetapi dengan nama yang berbeda. Masukan user berupa nama mesin virtual yang akan digandakan dan nama mesin virtual baru.



Gambar 4.7 Flowchart fungsi clone_vm

IV.8. Flowchart Penggunaan Sistem

Gambar 4.8 merupakan flowchart penggunaan sistem, Flowchart ini menjelaskan tentang penggunaan sistem, dimulai dari proses pembuatan container, pembuatan mesin virtual, konfigurasi switch dan antarmuka jaringan virtual, konfigurasi alamat IP server virtual sampai dengan uji coba konektivitas jaringan, DHCP, DNS dan Web server.



Gambar 4.8 : Flowchart penggunaan sistem

[5] Schaller, Robert R. "Moore's law: past, present and future." Spectrum, IEEE 34.6 (1997): 52.

v. Kesimpulan dan Saran

V.1. Kesimpulan

Setelah melakukan penelitian dapat diambil beberapa kesimpulan antara lain :

1. Telah berhasil dibuat perancangan tiga buah server virtual menggunakan berbasis *command line*
2. Ketiga server virtual yang dalam pembuatannya nanti dibantu easyuml, dirancang untuk dapat berkomunikasi antara satu sama lain.

V.2. Saran

Dalam penelitian ini tak lepas dari ketidaksempurnaan dan kekurangan yang perlu diperbaiki oleh karena itu penulis ingin memberikan saran antara lain

1. Kedepannya dapat diimplementasikan menjadi aplikasi virtual server dengan skenario jaringan yang menghubungkan semua server virtual
2. Penambahan penggunaan mode user mode linux dapat dilakukan untuk mempermudah proses implementasi virtual server berbasis command line.

DAFTAR PUSTAKA

- [1] Mirzaei, Nariman. "Cloud Computing." Pervasive Technology Institute Report, Community Grids Lab, Indiana University (2008): 1-12
- [2] Golden, Bernard. Virtualization for dummies. John Wiley & Sons, 2013.
- [3] Yu, Yang. Os-level virtualization and its applications. ProQuest, 2007.
- [4] Nidhra, Srinivas, and Jagruthi Dondeti. "BLACK BOX AND WHITE BOX TESTING TECHNIQUES--A LITERATURE REVIEW." International Journal of Embedded Systems & Applications 2.2 (2012).