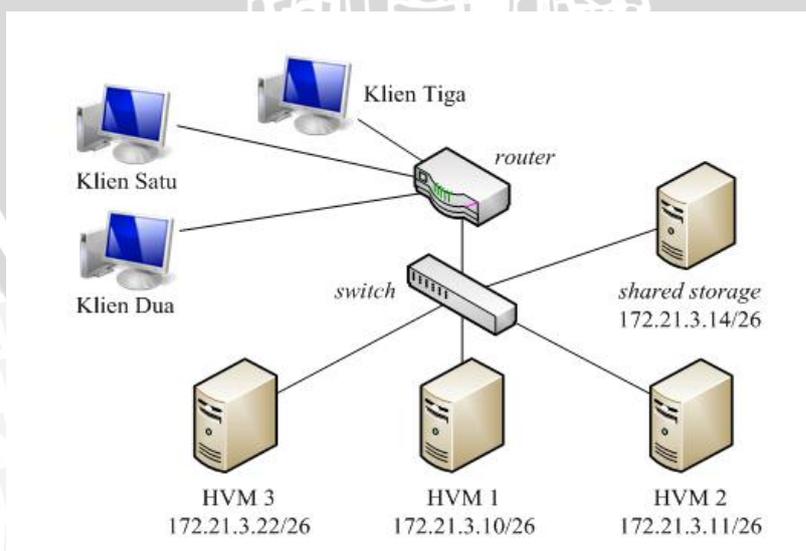


BAB IV PERANCANGAN

Bab ini menjelaskan rancangan sistem virtualisasi *server* yang dibangun dengan menggunakan metode *load balancing* dan *load aggregation*. Dua hal yang akan dibahas dari perancangan sistem virtualisasi *server* ini yaitu perancangan jaringan dan perancangan perangkat lunak.

4.1 Perancangan Jaringan

Setelah memperoleh seluruh kebutuhan perangkat keras dan perangkat lunak melalui tahap analisis kebutuhan, sistem virtualisasi *server* ini kemudian dirancang sesuai dengan tujuan pelaksanaan penelitian. Perancangan jaringan meliputi instalasi komputer HVM, komputer *shared storage*, *switch*, media penghubung, dan perangkat keras lainnya di dalam sebuah segmen jaringan lokal yang sama. Proses perancangan dilakukan dengan menghubungkan seluruh perangkat keras menggunakan kabel UTP melalui *switch* dan kemudian memberikan alamat jaringan untuk setiap HVM dan *shared storage*. Penulis menentukan alamat jaringan 172.21.3.0 dan *subnet mask* 255.255.255.192 untuk setiap komputer HVM dan *shared storage*. Berikut merupakan perancangan jaringan sistem virtualisasi *server*:



Gambar 4.1 Perancangan Jaringan Sistem Virtualisasi *Server*

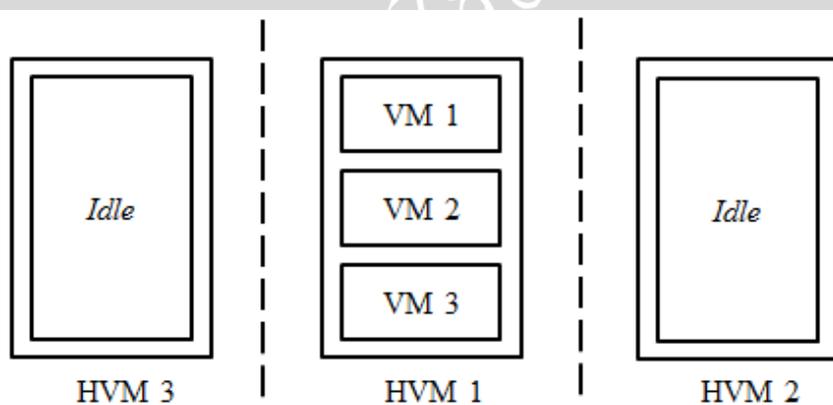
4.2 Perancangan Perangkat Lunak

Pada sistem virtualisasi *server* ini, penggunaan sumber daya komputasi CPU untuk setiap mesin virtual (VM) diatur sebanding dengan sumber daya komputasi CPU *server* fisik. Alasan penggunaan CPU mesin virtual seperti ini adalah untuk mencapai kondisi proporsional ketika sebuah HVM menangani satu mesin virtual, yaitu pada saat terjadi pemindahan mesin virtual. Dalam penelitian ini, penulis hanya memasang tiga buah mesin virtual karena keterbatasan sumber daya komputasi, khususnya pada kapasitas *memory* komputer. Berikut alokasi sumber daya komputasi pada masing-masing *host* dan mesin virtual:

Tabel 4.1 Alokasi Sumber Daya Komputasi

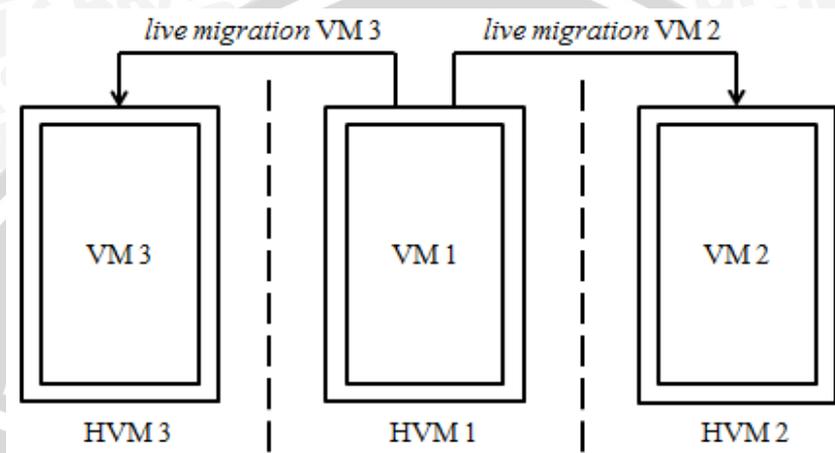
<i>Server</i>	CPU/vCPU	<i>Memory</i>	<i>Hard Disk</i>
HVM 1	4	2 GB	500 GB
HVM 2	4	2 GB	500 GB
HVM 3	4	2 GB	500 GB
VM 1	4	1 GB	50 GB
VM 2	4	1 GB	50 GB
VM 3	4	1 GB	50 GB

Setelah menentukan alokasi sumber daya komputasi sistem, penulis kemudian merancang perangkat lunak yang akan digunakan agar menjadi sebuah kesatuan sistem. Berikut merupakan ilustrasi cara kerja sistem virtualisasi *server* yang akan dibangun dengan metode *load balancing* dan *load aggregation*:



Gambar 4.2 Sistem Virtualisasi *Server* Beban Kerja CPU Rendah

Gambar 4.2 merupakan ilustrasi sistem virtualisasi *server* ketika HVM satu mengalami beban kerja rendah CPU *server* fisik. Ketika akses klien sedikit terhadap VM, HVM satu yang berfungsi sebagai *server* fisik utama tidak mengalami kelebihan beban kerja. HVM dua dan HVM tiga berada dalam kondisi *idle* karena tidak ada VM yang berjalan di dalamnya.



Gambar 4.3 Sistem Virtualisasi *Server* Beban Kerja CPU Tinggi

Gambar 4.3 di atas merupakan ilustrasi pada saat terjadi beban kerja tinggi CPU HVM satu, proses *load balancing* dilakukan dengan mendistribusikan beban kerja CPU tersebut ke HVM lain yang tersedia. Secara otomatis, sistem akan memindahkan VM secara berurutan ke HVM dua atau HVM tiga sesuai dengan besarnya beban kerja CPU VM. Proses pemindahan VM tersebut dilakukan berdasarkan parameter batas beban kerja CPU maksimum HVM satu. Pada saat beban kerja VM kembali menurun, proses *load aggregation* dilakukan secara otomatis pula yaitu dengan memindahkan VM yang telah dimigrasi menuju ke HVM satu. Seperti itulah proses umum yang terjadi pada sistem virtualisasi *server* pada saat menangani beban kerja tinggi CPU *server* fisik.

4.2.1 Penggunaan Perangkat Lunak

Berdasarkan pada kebutuhan perangkat lunak yang telah ditentukan pada tahap analisis kebutuhan, berikut merupakan penjelasan penggunaan seluruh perangkat lunak pada sistem virtualisasi *server*:

- **Sistem Operasi** – Sistem operasi yang digunakan pada sistem virtualisasi *server* ini adalah Fedora-14-x86_64. Sistem operasi ini dipasang pada

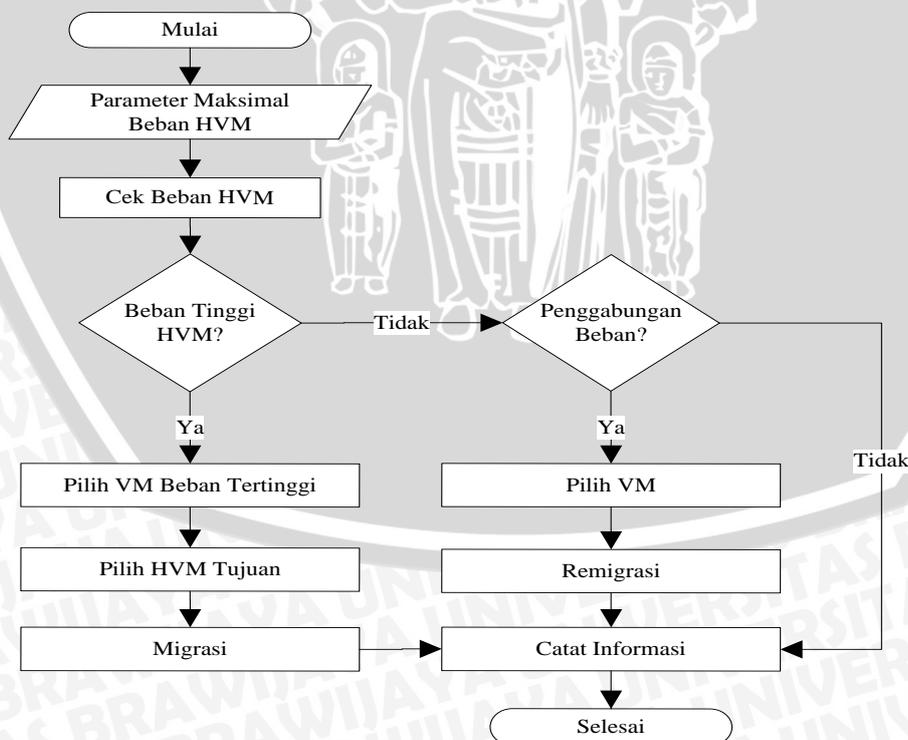
seluruh komputer HVM dan *shared storage*. Pada dasarnya, seluruh jenis distribusi Linux dapat digunakan pada sistem ini namun penulis memilih Fedora karena telah teruji dapat beroperasi baik dengan *KVM Hypervisor*.

- **VM Hypervisor** – Perangkat lunak ini berfungsi untuk membuat dan menjalankan mesin virtual di dalam *server* fisik. Dalam penelitian ini penulis memilih *KVM* sebagai *VM Hypervisor*. Perangkat lunak ini dipasang di setiap komputer HVM dan seluruhnya dikonfigurasi identik agar tidak terjadi kegagalan ketika mesin virtual dipindahkan.
- **Web Server** – *Web server* yang digunakan yaitu *Apache Web server*. Perangkat lunak ini berfungsi sebagai media untuk melakukan pengujian sistem. *Apache Web server* dipasang pada masing-masing mesin virtual yang berfungsi sebagai *server* virtual.
- **Load Tester** – Untuk menguji sistem virtualisasi *server* yang dibangun, maka diperlukan alat yang dapat memicu terjadinya beban kerja tinggi CPU pada sistem. *Load tester* yang digunakan penulis yaitu *Apache Jmeter* karena perangkat lunak ini memiliki fitur yang lengkap dan handal untuk melakukan *stress load*. Perangkat lunak ini dipasang di beberapa komputer klien yang berada di luar sistem virtualisasi *server* yang dibangun.
- **File Sharing** – *File sharing* berfungsi untuk melakukan pengaksesan *virtual disk* yang disimpan di dalam komputer *shared storage*. *File sharing* yang digunakan dalam penelitian ini yaitu *Network File System*. Penggunaan *File sharing* memiliki fungsi untuk mengurangi besarnya data *VM* yang harus dipindahkan, sehingga proses migrasi dapat berjalan dengan waktu yang lebih cepat. Selain itu, *NFS* digunakan untuk menyimpan data beban kerja CPU mesin virtual dari HVM dua dan HVM tiga.
- **System Monitoring** – Untuk mendapatkan informasi persentase beban kerja yang sedang ditanggung oleh *server* fisik maupun mesin virtual, sistem ini memerlukan perangkat lunak tambahan yaitu *sysstat* & *virt-top*. *Sysstat* digunakan untuk melakukan monitoring beban kerja HVM, sedangkan *virt-top* digunakan untuk monitoring beban kerja seluruh mesin virtual. Kedua perangkat lunak tersebut digunakan pada masing-masing HVM.

- **Program *Bash Shell*** – Beberapa program tambahan berupa *Bash Shell* diperlukan untuk menjalankan proses *load balancing* dan *load aggregation*. Fungsi utama dari *Bash* atau jenis *shell* lainnya adalah sebagai alat untuk berinteraksi dengan sistem operasi komputer. Ketika program dijalankan, *shell* mengambil perintah yang dimasukkan, menentukan program apa saja yang perlu dijalankan, dan kemudian menampilkannya [ALB-07]. Program *Bash Shell* ini dijalankan pada seluruh HVM dan dieksekusi secara berkala dengan menggunakan Linux Crontab. Program *Bash Shell* ini berisi perintah-perintah seperti monitoring, seleksi kondisi, perpindahan mesin virtual, dan juga pencatatan *log* sistem virtualisasi *server*.

4.2.2 Algoritma Program

Untuk dapat melakukan proses *load balancing* dan *load aggregation*, diperlukan sebuah program tambahan berupa *Bash Shell* yang dijalankan terpusat dan berkala pada HVM satu. Pendekatan yang menjadi dasar dari pembuatan program ini adalah beban kerja CPU dari HVM dan VM. Berikut diagram alir algoritma program:



Gambar 4.4 Diagram Alir Algoritma Program

Penjelasan algoritma program:

- Proses awal yang dilakukan adalah menentukan parameter beban kerja maksimum yang dapat ditangani oleh HVM satu.
- Setelah itu, program mendapatkan informasi beban kerja HVM 1, VM 1, VM 2, dan VM 3. Seluruh informasi yang diambil bersumber dari beberapa berkas yang disimpan di HVM satu dan *shared storage*.
- Setelah informasi beban kerja didapat, program melakukan pengecekan apakah HVM satu dalam kondisi kelebihan beban atau tidak.
- Jika HVM satu kelebihan beban
 - Program memeriksa mesin virtual yang berjalan pada HVM satu.
 - Jika VM berjumlah lebih dari satu
 - Program mencari VM yang memiliki beban kerja CPU tertinggi dan kemudian menunjuk salah satu HVM sebagai HVM target.
 - Program melakukan migrasi VM. Migrasi adalah proses perpindahan mesin virtual dari HVM satu menuju HVM dua atau HVM tiga.
 - Jika VM berjumlah satu
 - Program berakhir
- Jika HVM satu tidak kelebihan beban
 - Program memeriksa jumlah mesin virtual yang berjalan pada HVM satu.
 - Jika terdapat VM tidak berada pada HVM satu
 - Program melakukan kalkulasi penggabungan beban, apakah VM dapat dipindahkan kembali ke HVM satu.
 - Jika iya, program melakukan remigrasi VM. Remigrasi adalah proses perpindahan mesin virtual dari HVM dua atau HVM tiga menuju ke HVM satu
 - Jika tidak, maka program berakhir
 - Jika seluruh VM berada pada HVM satu
 - Program berakhir
- Program mencatat seluruh informasi yang terjadi dalam setiap eksekusi.