

**IMPLEMENTASI HIGH-AVAILABILITY VPN CLIENT  
PADA JARINGAN KOMPUTER FAKULTAS HUKUM  
UNIVERSITAS UDAYANA**

**Putu Topan Pribadi**

Program Studi Teknik Informatika, Jurusan Ilmu Komputer,  
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana  
Email: putu.topan@cs.unud.ac.id

**ABSTRAK**

*Ketersediaan data/layanan sangat dibutuhkan, terutama bagi perusahaan-perusahaan yang sangat membutuhkan data-data ataupun layanan yang bersifat sangat penting. Apabila terjadi kerusakan data ataupun layanan, sangat dibutuhkan adanya server cadangan.*

*Sesuai dengan dasar teori dari teknologi High-Availability, yaitu ilmu untuk menciptakan redundansi dalam setiap sistem dan subsistem untuk memastikan bahwa layanan tetap up dan tersedia. Sehingga dapat dianalogikan dalam implementasi, bila satu server gagal melayani service tertentu, maka tugas server tersebut otomatis akan dilempar ke server lainnya.*

*HA akan diimplementasikan pada layanan VPN Client yang bertujuan untuk memastikan koneksi VPN tetap hidup, agar administrator dapat lebih mudah me-monitoring jaringan dari luar.*

**Kata Kunci:** Cluster Computing, High-availability, Failover, Heartbeat, VPN, OpenVPN

**ABSTRACT**

*Availability of data/services are needed, especially for companies who desperately need the data or services that are very important. When the data and services are corrupt, it is necessary to have a backup server.*

*According to the basic theory of High-Availability technology, the science of creating redundancy in each system and subsystem to ensure that services remain up and available. So that it can be analogized to the implementation, if the server fails to serve a particular service, then the task server will automatically be taken over by the other server.*

*HA will be implemented on the VPN Client service that aims to make a VPN connection alive, so that the administrator can be easier to monitoring the network from outside.*

**Keywords:** Cluster Computing, High-availability, Failover, Heartbeat, VPN, OpenVPN

## PENDAHULUAN

Perkembangan teknologi memberi pengaruh besar bagi segala aspek kehidupan, begitu banyak manfaat yang dapat kita implementasikan. Teknologi saat ini telah memberikan kemudahan dalam penyampaian suatu informasi. Dalam perkembangan di suatu perusahaan, infrastruktur harus sesuai dengan kebutuhan, baik itu di perusahaan besar maupun kecil dan tentu saja sangat berbeda dalam hal persyaratan dan infrastruktur, khususnya infrastruktur jaringan.

Teknologi *backup* otomatis dan *High Availability Server (HA)* merupakan teknologi yang banyak dibutuhkan dilingkungan perusahaan atau entitas bisnis yang menggunakan sistem komputerisasi. *Linux* sebagai sebuah pilihan terbaik untuk sistem server menyediakan berbagai macam *tools powerful* yang dapat didayagunakan dengan biaya minimal.

Seiring penggunaan teknologi informasi tersebut, maka perusahaan dihadapkan pada berbagai kemungkinan terjadinya gangguan pada sistem. Gangguan dapat berupa kerusakan komputer, *storage*, *data corruption*, jaringan, dan bencana alam. Hal ini akan menyebabkan terhambatnya proses bisnis dan kehilangan data yang mengakibatkan kerugian pada perusahaan.

## TINJAUAN PUSTAKA

### *TCP/IP*

*TCP/IP* adalah singkatan dari *Transmission Control Protocol/Internet Protocol*. *TCP* bertugas menerima pesan elektronik dengan panjang sembarang dan membaginya ke dalam bagian-bagian berukuran 64kb. Dengan membagi pesan menjadi bagian-bagian, perangkat lunak yang mengontrol komunikasi jaringan dapat mengirim tiap bagian dan menyerahkan prosedur pemeriksaan bagian demi bagian. Apabila suatu bagian mengalami kerusakan selama transmisi, maka program pengirim hanya perlu mengulang transmisi bagian itu dan tidak perlu mengulang dari awal (Anjik Sukmaaji, 2008: p22).

*TCP/IP* adalah standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan *internet*. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*Software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah *TCP/IP stack*.

## **Cluster Computing**

*Cluster*, dalam ilmu komputer dan jaringan komputer adalah sekumpulan komputer (umumnya server jaringan) independen yang beroperasi serta bekerja secara erat dan terlihat oleh klien jaringan seolah-olah komputer-komputer tersebut adalah satu buah *unit* komputer. Proses menghubungkan beberapa komputer agar dapat bekerja seperti itu dinamakan dengan *Clustering*. Komponen *cluster* biasanya saling terhubung dengan cepat melalui sebuah interkoneksi yang sangat cepat, atau bisa juga melalui jaringan lokal (LAN).

Karena menggunakan lebih dari satu buah server, maka manajemen dan perawatan sebuah *cluster* jauh lebih rumit dibandingkan dengan manajemen server *mainframe* tunggal yang memiliki skalabilitas tinggi (semacam IBM AS/400), meski lebih murah.

*Cluster Computing* terbagi ke dalam beberapa kategori, yaitu sebagai berikut:

### **1. High-availability Clusters**

*High-availability* adalah ilmu untuk menciptakan redundansi dalam setiap sistem dan subsistem untuk memastikan bahwa layanan tetap *up* dan tersedia. *HA* pada dasarnya menempatkan satu atau lebih server cadangan dalam modus siaga, yang bisa online dalam beberapa

saat hanya setelah mereka menemukan kegagalan pada sistem utama. (Charles Bookman, 2002).

*High-availability cluster*, yang juga sering disebut sebagai *Failover Cluster* pada umumnya diimplementasikan untuk tujuan meningkatkan ketersediaan layanan yang disediakan oleh kluster tersebut. Elemen kluster akan bekerja dengan memiliki *node-node* redundan, yang kemudian digunakan untuk menyediakan layanan saat salah satu elemen kluster mengalami kegagalan. Ukuran yang paling umum dari kategori ini adalah dua *node*, yang merupakan syarat minimum untuk melakukan redundansi. Implementasi kluster jenis ini akan mencoba untuk menggunakan redundansi komponen kluster untuk menghilangkan kegagalan di satu titik (*Single Point of Failure*).

### **2. Load-balancing Clusters**

Kluster kategori ini beroperasi dengan mendistribusikan beban pekerjaan secara merata melalui beberapa *node* yang bekerja di belakang (*back-end node*). Umumnya kluster ini akan dikonfigurasi sedemikian rupa dengan beberapa *front-end load-balancing* redundan. Karena setiap elemen dalam sebuah kluster *load-balancing* menawarkan layanan penuh,

maka dapat dikatakan bahwa komponen kluster tersebut merupakan sebuah kluster aktif/kluster HA aktif, yang bisa menerima semua permintaan yang diajukan oleh klien.

### 3. *Compute Clusters*

Seringnya, penggunaan utama kluster komputer adalah untuk tujuan komputasi, daripada penanganan operasi yang berorientasi I/O seperti layanan Web atau basis data. Contoh, sebuah kluster mungkin mendukung simulasi komputasional untuk perubahan cuaca atau tabrakan kendaraan.

Desain kluster ini, sering disebut sebagai “*Grid*”. Beberapa *compute cluster* yang dihubungkan secara erat yang didesain sedemikian rupa, umumnya disebut dengan “*Supercomputing*”. Beberapa perangkat lunak *Middleware* seperti MPI atau Paraller Virtual Machine (PVM) mengizinkan program *compute clustering* agar dapat dijalankan di dalam kluster-kluster tersebut.

#### ***Heartbeat***

*Heartbeat* adalah salah satu program yang terpisah atau termasuk dalam fungsi utama dari aplikasi *cluster*. *Heartbeat* bertujuan untuk terus mem-*polling* server dalam konfigurasi *cluster*

untuk memastikan bahwa mereka sudah *up* dan merespon. (Charles Bookman, 2002).

*Heartbeat* adalah Linux *High Availability* yang menggunakan teknik *cluster*, yang bisa digunakan untuk pada beberapa sistem operasi seperti *Linux*, *FreeBSD*, *OpenBSD*, *Solaris*, *MacOS*, yang mengunggulkan kehandalan, ketersediaan, dan *serviceability* (RAS).

*Heartbeat* perlu dikombinasikan dengan *resource* yang diperlukan untuk membangun *failover* yang memiliki kemampuan menghentikan dan memulai *service* yang diinginkan seperti *service IP address*, *webserver*, *mounting* blok *hard disk*, dsb. *Heartbeat* menjalankan *script* inialisasi untuk menjalankan *service* lain saat *heartbeat* dijalankan atau bisa juga mematikan *service* lain saat *heartbeat* dimatikan. *Heartbeat* juga melakukan perpindahan *IP address* dari satu *server* ke *server* yang lain (*IP floating*).

Fungsi dari *Heartbeat* adalah memeriksa layanan dari unit server, jika *service* tersebut mati maka akan langsung menggantikannya dengan *server backup service*. Berikut peran dari *Heartbeat* :

1. Memonitor server online (bekerja atau tidak), bisa menggunakan ethernet, dan juga menggunakan serial port.
2. Melakukan fail over dimana server *master* akan dialihkan ke server *slave*

jika terjadi kegagalan pada server *master*.

### **VPN**

*VPN* adalah singkatan dari *virtual private network*, yaitu sebuah terowongan virtual (*virtual tunnel*) dari jaringan ke jaringan lain yang terenkripsi. *VPN server* dan *VPN client* harus saling ter-autentikasi. *VPN* mengkoneksikan dua jaringan seperti kantor-kantor cabang atau *remote users* tunggal ke kantor. (Carla Schroder, 2008: p265).

*VPN* adalah sebuah koneksi *Virtual* yang bersifat *private* mengapa disebut *virtual* karena pada dasarnya jaringan ini tidak ada secara fisik hanya berupa jaringan *virtual* dan mengapa disebut *private* karena jaringan ini merupakan jaringan yang sifatnya *private* yang tidak semua orang bisa mengaksesnya. *VPN* Menghubungkan *PC* dengan jaringan publik atau internet namun sifatnya *private*, karena bersifat *private* maka tidak semua orang bisa terkoneksi ke jaringan ini dan mengaksesnya. Oleh karena itu diperlukan keamanan data.

### **OpenVPN**

*OpenVPN* adalah aplikasi open source untuk *Virtual Private Networking* (*VPN*), dimana aplikasi tersebut dapat

membuat koneksi *point-to-point tunnel* yang telah terenkripsi. *OpenVPN* menciptakan *VPN* sebenarnya, perluasan terenkripsi dari jaringan yang memerlukan kepercayaan bersama yang akan didirikan antara server dan klien. *OpenVPN* menggunakan *private keys*, *certificate*, atau *username/password* untuk melakukan autentikasi dalam membangun koneksi. Dimana untuk enkripsi menggunakan *OpenSSL* (Carla Schroder, 2008: p266).

### **Linux**

*Linux* adalah nama yang diberikan kepada sistem operasi komputer bertipe *Unix*. *Linux* merupakan salah satu contoh hasil pengembangan perangkat lunak bebas dan sumber terbuka utama. Seperti perangkat lunak bebas dan sumber terbuka lainnya pada umumnya, kode sumber *Linux* dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapa saja.

*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, dan sistem benam seperti pembaca buku elektronik, sistem permainan video (PlayStation 2, PlayStation 3 dan XBox), telepon genggam dan *router*.

### Linux Ubuntu

*Ubuntu* merupakan salah satu distribusi *Linux* yang berbasiskan *Debian* dan didistribusikan sebagai *software* bebas. Nama *Ubuntu* berasal dari filosofi dari Afrika Selatan yang berarti “Kemanusiaan kepada sesama”. *Ubuntu* didesain untuk kepentingan penggunaan personal, namun versi server *Ubuntu* juga tersedia, dan telah dipakai secara luas.

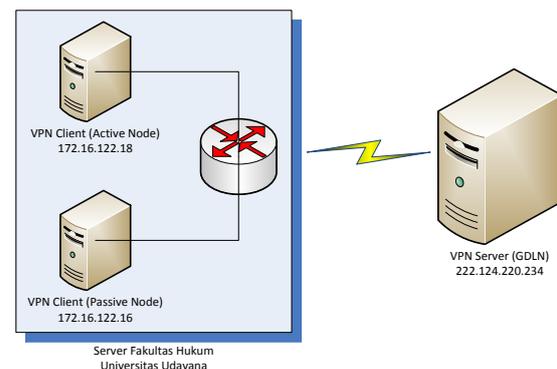
Proyek *Ubuntu* resmi disponsori oleh *Canonical Ltd.* yang merupakan sebuah perusahaan yang dimiliki oleh pengusaha Afrika Selatan Mark Shuttleworth. Tujuan dari distribusi *Linux Ubuntu* adalah membawa semangat yang terkandung di dalam Filosofi *Ubuntu* ke dalam dunia perangkat lunak. *Ubuntu* adalah sistem operasi lengkap berbasis *Linux*, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional.

### Linux CentOS

*CentOS* singkatan dari *Community ENTerprise Operating System* (Sistem Operasi Perusahaan buatan Komunitas/Masyarakat) adalah sistem operasi gratis yang dibuat dari *source code Red Hat Enterprise Linux (RHEL)*. Proyek ini berupaya untuk 100% binari kompatibel dengan produk hulunya (*RHEL*). Dan tentu saja menggunakan paket *RPM*.

Karena *CentOS* dikompilasi dari *SRPM RHEL* maka *CentOS* 100% kompatibel dengan *RHEL*, isi dari *CentOS* hampir sama dengan *RHEL*. Bedanya pada *CentOS* semua atribut *RHEL* dibuang, misalnya *README.TXT RHEL* diganti menjadi *README.TXT CentOS*.

### PERANCANGAN DAN KONFIGURASI SISTEM



*VPN Server* terletak di server ILKOM yang berada di GDLN UNUD, dengan menggunakan distro *Linux Ubuntu* 8.04 (*IP* publik: 222.124.220.234). Sedangkan untuk *VPN Client*, dibuatkan dua buah server yaitu *Primary Server*

(*Active Node*) dengan menggunakan *Linux CentOS 5.5* (172.16.122.18) dan *Secondary Server* (*Passive Node*) dengan menggunakan *Linux CentOS 5.8* (172.16.122.16), dimana nantinya *Primary Server* akan menjadi server utama untuk *service VPN Client* dan *Secondary Server* akan aktif jika *Primary Server* gagal menjalankan *service VPN Client* atau server *hang/mati*.

Untuk dapat mengakses keseluruhan server yang berada di Fakultas Hukum Universitas Udayana, *administrator* jaringan juga harus mempunyai koneksi *VPN Client*. Untuk koneksi *VPN* penulis menggunakan *OpenVPN*, karena *OpenVPN* merupakan salah satu aplikasi *VPN* yang *opensource*. Untuk *High-Availability service*, penulis menggunakan *Heartbeat*. Komunikasi *Heartbeat* antar server dilakukan melalui interface *Ethernet* dengan cara mem-*broadcast* ke jaringan *IP* yang sama.

*Primary Server* (*Active Node*) akan menjadi server utama yang akan menjalankan *service VPN Client* jika server berjalan normal dan tidak mengalami kerusakan atau *crash*. Jika *Active Node* mengalami *downtime* atau mati, maka *service VPN Client* akan dipindahkan secara otomatis oleh *Heartbeat* ke *Passive Node*. *Heartbeat* akan terus mem-*polling*

kedua *node* tersebut untuk memastikan *node* mana yang *up* dan merespon. *Service VPN Client* akan berpindah kembali ke *Active Node* jika server sudah berjalan normal.

## PENGUJIAN DAN ANALISIS

Untuk melakukan pengujian, pertama-tama stop *VPN Client* pada kedua *node* lalu jalankan *heartbeat* pada kedua *node*. Pada *Active Node*, cek dengan perintah *ifconfig* maka akan muncul *interface eth0:0* dengan *IP virtual* yang sudah ditentukan sebelumnya, dan juga secara otomatis akan menjalankan *VPN Client* terlihat dari *interface tun0* yang muncul dengan *address*-nya.

Setelah itu matikan *Active Node*, dalam hal ini dianggap server mengalami kegagalan atau *crash* sehingga tidak dapat menjalankan *service VPN Client*. Tunggu sampai beberapa waktu sampai *Active Node* dinyatakan benar-benar mati. Setelah itu, pada *Passive Node*, cek dengan perintah *ifconfig* maka akan terlihat *interface eth0:0* dengan *IP virtual*-nya dan juga terlihat *interface tun0* dengan *address*-nya. Ini menandakan bahwa *Passive Node* berhasil menjadi *backup* dari *service VPN*, sehingga *service* menjadi *High-Availability*.

Jika *Active Node* aktif kembali dan menjalankan *service VPN* sebagaimana

mestinya, maka *Passive Node* akan kembali seperti semula, mematikan *service VPN*-nya sendiri sehingga tidak terjadi redundansi *VPN service*.

Dalam *High-availability cluster*, hal yang sangat penting adalah penanganan *failover*. Ketersediaan data/*service* harus terpenuhi pada saat user menggunakan *service* tersebut.

## KESIMPULAN

Kesimpulan yang dapat penulis tarik dari penelitian ini adalah:

1. Dengan adanya sistem *High-Availability VPN Client*, layanan *VPN* tidak akan terganggu yang diakibatkan oleh kerusakan pada *Primary Server* karena *Secondary Server* akan mengambil alih tugas *Primary Server* dengan baik jika terjadi kegagalan.
2. *High-Availability VPN Client* sangat cocok diterapkan pada sistem yang sangat memerlukan koneksi *VPN* yang stabil.
3. Konsep *High-Availability* yang diimplementasikan pada *VPN Client* di Fakultas Hukum Universitas Udayana merupakan salah satu solusi yang handal untuk membangun koneksi jaringan *VPN*.

## DAFTAR PUSTAKA

- Archlinux.org. 2012. *Simple IP Failover with Heartbeat* [Online] Tersedia: [https://wiki.archlinux.org/index.php/Simple\\_IP\\_Failover\\_with\\_Heartbeat](https://wiki.archlinux.org/index.php/Simple_IP_Failover_with_Heartbeat).
- Bookman, Charles. 2002. *Linux Clustering: Building and Maintaining Linux Clusters*: United States of America. New Riders Publishing.
- Linux-ha.org. 2007. *Ha.cf* [Online] Tersedia: <http://wiki.linux-ha.org/ha.cf>.
- Schroder, Carla. 2008. *Linux Networking Cookbook*: United States of America. O'Really Media, Inc.
- Smartnetsolution.com. 2011. *VPN* [Online] Tersedia: <http://smartnetsolution.com/tutorial/VPN.pdf>.
- Sukmaaji, Anjik. 2008. *Jaringan Komputer*: Yogyakarta. Andi.
- Telkomspeedy.com. 2010. *Instalasi OpenVPN* [Online] Tersedia: [http://opensource.telkomspeedy.com/wiki/index.php/Instalasi\\_OpenVPN](http://opensource.telkomspeedy.com/wiki/index.php/Instalasi_OpenVPN).