

Perancangan dan Penerapan *High Availability* Web Server dengan *Failover Clustering* di Lingkungan Pendidikan Tinggi

Khayreen Najmi^{*1}, Muhammad Zulfariansyah²

^{1,2}Prodi Teknik Informatika, Universitas Nahdlatul Ulama Kalimantan Timur, Samarinda

e-mail: ¹khayreen.najmi@gmail.com, ²zulfariansyah@unukaltim.ac.id

Abstrak

Universitas Nahdlatul Ulama Kalimantan Timur menghadapi tantangan dalam menjaga stabilitas web server untuk layanan informasi dan system akademik. Peningkatan jumlah pengguna dan beban aplikasi menyebabkan penurunan performa dan kegagalan layanan. Solusi yang ditawarkan dalam penelitian ini adalah penerapan failover clustering dan high availability menggunakan proxmox virtual environment (VE) dan ceph. Penelitian ini menggunakan metode Network Development Life Cycle (NDLC), yang terdiri dari enam tahap, yaitu analysis, design, simulation prototyping, implementation, monitoring, dan management. Hasil pengujian menunjukkan bahwa sistem berhasil melakukan failover otomatis dengan rata-rata waktu respon 1,85 detik dan tingkat ketersediaan layanan mencapai lebih dari 99%. Hasil ini menunjukkan bahwa implementasi failover clustering dan high availability efektif dalam meningkatkan keandalan dan kontinuitas layanan web server di lingkungan universitas.

Kata kunci— Failover Clustering, High Availability, Proxmox VE, Ceph, NDLC

1. PENDAHULUAN

Web server merupakan salah satu infrastruktur penting dalam menunjang layanan informasi dan akademik di lingkungan perguruan tinggi. Akan tetapi gangguan system dapat mengakibatkan ketidakterediaan layanan yang berdampak pada proses administrasi dan pembelajaran. Universitas Nahdlatul Ulama Kalimantan Timur (UNU Kaltim) menghadapi permasalahan tersebut akibat peningkatan jumlah pengguna dan beban kerja system.

Untuk mengatasi masalah tersebut, menerapkan failover clustering dan high availability adalah solusi yang tepat. Karena failover clustering memungkinkan system tetap berjalan dengan mengalihkan layanan ke node cadangan saat terjadi kegagalan. Sementara itu, high availability menjamin ketersediaan layanan melalui mekanisme redundansi dan replika data. Dengan menggunakan Proxmox VE dan Ceph, system dapat dikelola secara terpusat dan mampu melakukan migrasi virtual machine secara otomatis.

Metode yang digunakan dalam pengembangan ini adalah Network Development Life Cycle (NDLC). Metode ini terdiri dari enam tahapan, yaitu analysis, design, simulation prototyping, implementation, monitoring, dan management. Tujuan dari penelitian ini adalah untuk merancang system failover clustering dan high availability yang dapat meningkatkan stabilitas, ketersediaan dan performa layanan web server di UNU Kaltim.

2. METODE PENELITIAN

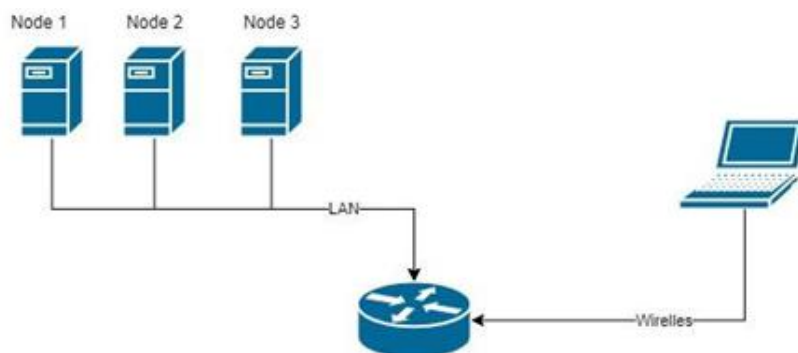
Penelitian ini menggunakan metode pengembangan system NDLC, yang terdiri dari beberapa tahapan-tahapan. Tahapannya adalah analysis, design, simulation prototyping, implementation, monitoring, dan management. Data dikumpulkan melalui observasi pada pembelajaran sebelumnya, serta kajian literature terkait teknik failover clustering, high availability yang ada. Tahapan-tahapan dari metode NDLC adalah sebagai berikut:

2.1 Analysis

Tahap analisis merupakan langkah awal yang sangat krusial karena berfungsi untuk mengidentifikasi secara menyeluruh kebutuhan perangkat keras (hardware) dan perangkat lunak (software) yang akan digunakan dalam proyek implementasi sistem virtualisasi berbasis Proxmox VE. Pada tahap ini dilakukan pemetaan terhadap perangkat yang dibutuhkan, mulai dari laptop untuk administrator, router jaringan untuk konektivitas, kabel Ethernet untuk komunikasi antar perangkat, serta perangkat server yang akan menjadi node dalam sistem. Selain itu, dilakukan juga seleksi terhadap sistem operasi pendukung seperti Debian GNU/Linux yang akan dijalankan di dalam virtual machine. Identifikasi ini penting untuk memastikan seluruh komponen saling kompatibel dan mendukung arsitektur sistem yang akan dibangun.

2.2 Design

Pada tahap desain, dilakukan perancangan arsitektur jaringan dan sistem menggunakan pendekatan topologi jaringan bintang (star topology) yang dikenal karena keandalannya dalam menjaga koneksi antar node tetap stabil meskipun terjadi kegagalan pada salah satu bagian. Desain ini mencakup tiga buah node server yang akan dikonfigurasi menjadi bagian dari satu cluster dan satu laptop administrator yang berfungsi sebagai pusat kendali dan manajemen sistem. Dalam rancangan ini ditentukan pula alokasi IP address, pengaturan interface jaringan, dan alur komunikasi antar node, sehingga nantinya implementasi dapat dilakukan secara sistematis dan minim risiko.



Gambar 1. Design Topologi

2.3 Simulation Prototyping

Tahap simulation prototyping bertujuan untuk melakukan pengujian awal terhadap desain yang telah dirancang sebelumnya menggunakan Proxmox VE sebagai platform virtualisasi. Dalam tahap ini, dibuat simulasi lingkungan virtual dari topologi yang telah

ditentukan, untuk memastikan bahwa konfigurasi dan rancangan jaringan dapat berjalan sebagaimana mestinya. Pengujian ini mencakup simulasi komunikasi antar node, kemampuan sistem untuk membentuk cluster, serta respons terhadap berbagai kondisi operasional. Dengan demikian, simulasi ini menjadi sarana validasi awal sebelum sistem diterapkan secara nyata, sekaligus menjadi bahan evaluasi bila ditemukan kendala teknis atau kekurangan dalam desain.

2.4 Implementation

Tahap implementasi merupakan langkah realisasi dari hasil analisis dan desain yang telah dibuat sebelumnya. Pada tahap ini, dilakukan instalasi Proxmox VE pada setiap node server dan proses pembentukan cluster antar node tersebut. Setelah itu, dilakukan instalasi Ceph sebagai solusi penyimpanan terdistribusi yang memungkinkan sistem bekerja secara redundan dan toleran terhadap kegagalan. Selain itu, dilakukan konfigurasi High Availability (HA) Group untuk memastikan layanan tetap berjalan meskipun salah satu node mengalami kegagalan. Di akhir tahap ini, juga dilakukan instalasi dan konfigurasi virtual machine berbasis Debian sebagai sistem operasi tamu, yang akan menjadi basis bagi layanan-layanan sistem yang dibutuhkan.

2.5 Monitoring

Setelah implementasi selesai, tahap monitoring menjadi kunci untuk memastikan sistem berjalan sesuai ekspektasi. Monitoring dilakukan dengan mengamati performa dan stabilitas sistem melalui pengujian failover, yaitu dengan mematikan salah satu node untuk melihat bagaimana sistem merespons dan berapa lama waktu yang dibutuhkan untuk memindahkan layanan secara otomatis ke node lain. Selain itu, dicatat pula waktu pemulihan (recovery time) dan status dari node-node lainnya. Pengujian ini sangat penting untuk mengevaluasi tingkat ketersediaan (availability) dan reliabilitas sistem dalam skenario dunia nyata.

2.6 Management

Tahap manajemen merupakan fase akhir yang berfokus pada pengelolaan sistem secara berkelanjutan setelah implementasi dan pengujian selesai dilakukan. Pengelolaan ini dilakukan melalui dashboard Proxmox VE yang menyediakan antarmuka visual untuk memantau status node, kinerja VM, penggunaan sumber daya (CPU, RAM, storage), serta kesehatan cluster dan Ceph storage. Selain pemantauan, dilakukan pula tindakan-tindakan manajerial seperti penambahan atau penghapusan VM, migrasi beban kerja antar node, update sistem, serta pencatatan log untuk kebutuhan audit dan troubleshooting. Tahap ini penting untuk menjaga sistem tetap optimal, responsif, dan mampu beradaptasi dengan perubahan kebutuhan operasional di masa depan.

3. HASIL DAN PEMBAHASAN

3.1 Instalasi Proxmox

Pada penelitian ini, penerapan *failover clustering* dan *high availability* menggunakan proxmox VE, dan tiga buah CPU (node) yang difungsikan sebagai server dalam sistem. setiap node diberikan konfigurasi *IP address* dan *hostname* yang berbeda untuk menghindari konflik jaringan. Rincian konfigurasi yang digunakan adalah sebagai berikut:

- a. Node 1 : hostname unukaltim.local *IP Address* 192.168.100.2

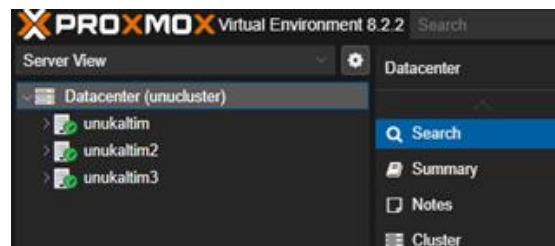
b. Node 2 : hostname unukaltim2.local *IP Address* 192.168.100.3

c. Node 3 : hostname unukaltim3.local *IP Address* 192.168.100.4

Selain itu, router dengan *IP Address* 192.168.100.1 digunakan sebagai *gateway* utama dalam jaringan lokal. Router ini berfungsi untuk mengelola koneksi antar node serta menyediakan akses administratif melalui laptop yang terhubung secara nirkabel.

3.2 Membuat Cluster

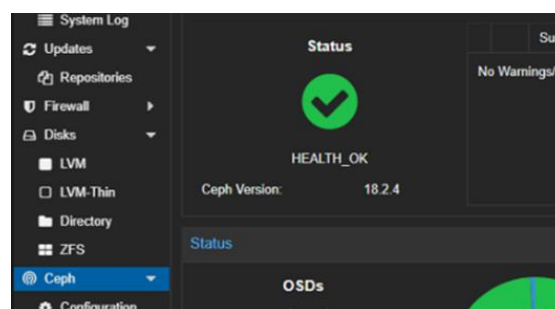
Tujuan utama dari pembuatan cluster adalah untuk menyatukan ketiga node ke dalam satu kelompok atau kluster yang terintegrasi. Dengan demikian, sistem dapat memastikan ketersediaan layanan yang tinggi melalui mekanisme *failover*. Jika salah satu node mengalami kegagalan, node lainnya dapat mengambil alih tugas secara otomatis, sehingga operasional tetap berjalan tanpa gangguan.



Gambar 2. Cluster dari seluruh Node

3.3 Instalasi Ceph

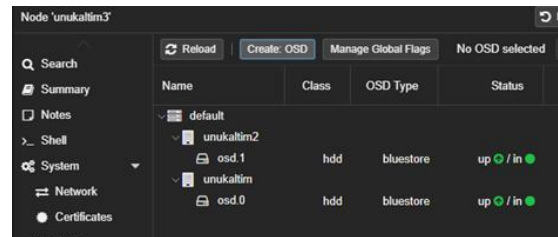
Ceph digunakan sebagai *storage* untuk *virtual machine* dengan beberapa keunggulan, seperti mengurangi risiko kehilangan data melalui replikasi otomatis dan memberikan performa yang baik, sehingga ideal untuk menangani beban kerja berat. Namun, instalasi Ceph memerlukan koneksi jaringan internet dan harus dilakukan pada setiap node dalam cluster agar sistem dapat berjalan secara terintegrasi.



Gambar 3. Status Ceph Aktif

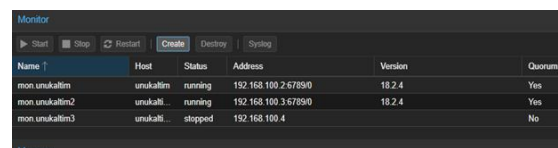
3.4 OSD, Monitor, Pools

Object Storage Daemon (OSD) berperan sebagai komponen penyimpanan data utama dalam cluster ceph, menangani penyimpanan, replikasi, dan pemulihan data.



Gambar 4. Membuat OSD

Monitor berfungsi untuk menjaga integritas dan konsistensi status cluster. Dengan monitor, administrator dapat segera mendeteksi masalah atau kegagalan pada node, memungkinkan perbaikan atau pemindahan VM secara cepat untuk mencegah gangguan.



Gambar 5. Tampilan Monitor

Setelah berhasil menginstall ceph *storage*, OSD, dan monitor. Langkah selanjutnya adalah mengkonfigurasi pool. Dengan menggunakan pool, setiap disk virtual pada VM tidak hanya tersimpan di satu node, tetapi didistribusikan ke seluruh node dalam cluster melalui mekanisme replikasi Ceph. Keuntungan utama dari penggunaan pool adalah kemampuan untuk menangani kegagalan pada salah satu node tanpa mengganggu ketersediaan data. Jika terjadi kegagalan hardware pada salah satu node, data tetap tersedia karena telah direplikasi di node lainnya, sehingga meningkatkan resiliensi dan kontinuitas layanan.

3.4 Mengkonfigurasi High Availability

Fitur *high availability* (HA) berfungsi untuk memastikan bahwa *virtual machine* (VM) dapat secara otomatis dipindahkan ke node lain jika terjadi kegagalan pada salah satu node dalam sistem. Terdapat lima menu pada pengaturan *High Availability* (HA), yaitu *Default*, *Freeze*, *Failover*, *Migrate*, dan *Conditional*, di mana menu yang paling sering digunakan adalah *Failover* dan *Migrate*. *Failover* berfungsi untuk memindahkan VM secara otomatis ke node lain jika node yang sedang berjalan mengalami kegagalan, sedangkan *Migrate* digunakan untuk memindahkan VM secara manual dari satu node ke node lainnya untuk keperluan pemeliharaan atau manajemen beban, namun *migrate* juga bisa digunakan secara otomatis (*live migration*). *Freeze* jarang digunakan karena berfungsi untuk menghentikan perpindahan otomatis meskipun terjadi masalah pada node, sementara *Default* dan *Conditional* memungkinkan pemindahan VM berdasarkan kondisi tertentu, seperti status node atau beban kerja yang telah ditentukan.

3.5 Membuat virtual machine

Membuat virtual machine (VM) menggunakan Linux Debian 10.9.0 sebagai sistem operasi berbasis linux yang digunakan dalam penelitian ini sebagai platform utama untuk menguji mekanisme failover dan high availability dengan memastikan sistem dapat berpindah secara otomatis antar node saat terjadi kegagalan pada salah satu server.

3.6 Tahap Monitoring

Pada tahap monitoring dilakukan pengujian *failover* pada *virtual machine* (VM) dengan melakukan percobaan pada server node dengan mematikan salah satu node, dan juga *stopwatch* untuk mengetahui berapa lama waktu *respond time* yang dibutuhkan untuk *virtual machine* (VM) dapat diakses kembali. Pengujian ini bertujuan untuk mengetahui berapa lama waktu yang dibutuhkan pada perpindahan dari node 1 ke node yang lainnya melalui beberapa pengujian. Adapun skenario yang diberikan adalah sebagai berikut:

- a. Skenario 1: ketiga node dalam posisi hidup kemudian node 1 dimatikan.
- b. Skenario 2: node 2 dan node 3 dalam posisi hidup kemudian node 2 dimatikan (Node 1 dalam posisi mati).
- c. Skenario 3: node 1 dan node 2 dalam posisi hidup kemudian node 3 dimatikan.

Tabel 1 Hasil Pengujian *Respond Time* (second)

Skenario	Node yang dimatikan	respond
1	Node 1	1.83
2	Node 2	1.86
3	Node 3	1.87

Hasil pengujian menunjukkan bahwa implementasi metode *failover clustering* pada virtualisasi server berhasil dilakukan dengan waktu perpindahan rata-rata dibawah 2 menit, memastikan kontinuitas layanan tanpa gangguan yang signifikan. Pengujian pada tiga skenario perpindahan ketiga node menunjukkan hasil yang konsisten dan stabil, dengan rata-rata waktu perpindahan 1.83 detik (node 1 ke node 2), 1,86 detik (node2 ke node 3), 1.87 (node 3 ke node 1). Hasil ini membuktikan bahwa metode *failover clustering* efektif dalam menjaga ketersediaan layanan *virtual machine*.

4. KESIMPULAN

Dari penelitian yang telah dilakukan, penerapan *failover clustering* dan *high availability* pada web server UNU KALTIM terbukti efektif dalam meningkatkan keandalan sistem. Dengan teknologi *failover clustering*, jika salah satu server mengalami gangguan, maka server cadangan secara otomatis mengambil alih tugas tanpa mengganggu layanan. Penerapan *high availability* juga berhasil membuat layanan tetap tersedia hingga 99% dengan jeda waktu yang singkat saat terjadi gangguan. Penggunaan sistem operasi *debian* mendukung stabilitas sistem, sehingga pengujian menunjukkan *virtual machine* (VM) dapat berpindah antar server dengan lancar.

5. SARAN

Adapun saran untuk penelitian selanjutnya dapat dilakukan dengan cara berikut :

1. Menggunakan perangkat dengan spesifikasi lebih tinggi.
2. Mengukur performa dengan berbagai perangkat penyimpanan network.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh dosen Program Studi Teknik Informatika Universitas Nahdlatul Ulama Kalimantan Timur atas dukungan dalam pelaksanaan penelitian ini. Terima kasih juga kepada dosen pembimbing dan semua pihak yang telah memberikan masukan dan bantuan selama proses penelitian berlangsung.

DAFTAR PUSTAKA

- [1] Agus, Supendar, H., Tutupoly, T. A., Vmware, I. H. A. V. S. M., & Potensia, E. P. P. G. R. (2020). Implementasi High Availability Virtualisasi Server Menggunakan Vmware. *Jurnal Inovasi Dan Sains Teknik Elektro*, 1(2), 101–107.
- [2] Chandra, R. (2024). Virtualisasi Server Menggunakan Proxmox Untuk Mengoptimalkan Resource Server Pada SMK Bhakti Persada. 1(2), 69–80.
- [3] Hartati, Shinta Septiantina, Amelia Luthfi Kamil, & Farkhatus Solikhah. (2021). Teknik Failover Clustering Sebagai Solusi High Availability. *Tematik*, 8(1), 104–109. <https://doi.org/10.38204/tematik.v8i1.579>
- [4] Prasandy, T., & Adhiwibowo, W. (2015). Virtualisasi Server Sederhana Menggunakan Proxmox. *Jurnal Transformatika*, 12(2), 37. <https://doi.org/10.26623/transformatika.v12i2.80>
- [5] Pribadi, Y., PN, A. B., & Irwansyah, M. A. (2020). Analysis of the Use of the Failover Clustering Method to Achieve High Availability on a Web Server (Case Study: Informatics Department Building). *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 8(2), 218. <https://doi.org/10.26418/justin.v8i2.31965>
- [6] Sanjaya, T., & Setiyadi, D. (2019a). Network Development Life Cycle (NDLC) Dalam Perancangan Jaringan Komputer Pada Rumah Shalom Mahanaim. *Jurnal Mahasiswa Bina Insani*, 4(1), 1–10.
- [7] Wahyuddin, M. I., & Jejen, M. (2016). Perancangan dan Implementasi High Availability Cluster Web Server Berbasis DBRD dan Heartbeat. 205–211. <https://doi.org/10.5614/sniko.2015.30>
- [8] Sumarna, S., Nurdin, H., & Handono, F. W. (2019). Laporan Akhir Penelitian - Perancangan NClustering High Availability Web Server Dengan Load Balancing Dan Failover. Jakarta.
- [9] Subekti, Zaenal Mutaqin Kurniawan, Rizky. 2019. Perancangan Jaringan VoIP Berbasis Open Source dengan DNS Pada Mikrotik. *Jurnal Cendekia AMIK Dian Cipta Cendekia*.
- [10] W. Adhiwibowo, “High Availability Web and Database Service Model with Redundacy Servers,” AITI, vol. 17, no. 1, pp. 33–41, Aug. 2020.

- [11] M. A. Putra, I. Fitri, dan A. Iskandar, "Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker," *J. Media Informatika Budidarma*, vol. 4, no. 1, pp. –, Jan. 2020.
- [12] S. Supriyadi dan D. Setiyadi, "Virtualisasi Server Failover Clustering Menggunakan Network Development Life Cycle Di SMK Negeri 1 Kota Bekasi," *J. Mahasiswa Bina Insani*, vol. 4, no. 2, pp. 115–124, Feb. 2020.
- [13] M. Rosalia, R. Munadi, dan R. Mayasari, "Implementasi High Availability Server Menggunakan Metode Load Balancing dan Failover pada Virtual Web Server Cluster," *e-Proceeding Eng.*, vol. 3, no. 3, –, 2019.
- [14] M. Mulyanto dan A. Ashari, "Implementasi Highly Available Website Dengan Distributed Replicated Block Device," *IJCCS*, –, 2021.
- [15] B. P. A. Pratama, B. G. Hardianto, dan D. Dharmayanti, "Implementation of High Availability Based on Load Balancing and Failover Method in E-commerce Using MySQL Database," *Asian J. Eng., Soc. and Health*, vol. 3, no. 10, 2024