

Simulasi Implementasi High Availability Server Menggunakan Ceph Pada Proxmox

Alfido Adha Febriansyah¹, Aditya Prapanca²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹alfido.20068@mhs.unesa.ac.id

²adityaprapanca@unesa.ac.id

Abstrak - Penelitian ini berfokus pada perancangan dan implementasi sistem high availability (HA) menggunakan Ceph pada Proxmox VE, dengan tujuan utama mengurangi downtime dan meminimalisir kebutuhan intervensi manusia dalam pemulihan sistem. Hasil pengujian menunjukkan bahwa sistem yang dikembangkan berhasil mencapai tingkat ketersediaan (HA) yang tinggi, dengan Virtual Machines (VM) mencapai tingkat ketersediaan lebih dari 97% dan container mencapai tingkat ketersediaan di atas 94%. Peningkatan spesifikasi perangkat keras pada Proxmox VE juga memberikan dampak positif pada kinerja HA, terutama pada VM yang mampu memanfaatkan sumber daya tambahan dengan lebih efektif. Skrip otomatis yang dibuat untuk mendeteksi dan memulihkan kegagalan pada Proxmox VE terbukti efektif dalam menjaga stabilitas dan kinerja layanan. Evaluasi dan optimasi kinerja sistem melalui simulasi berbagai skenario menunjukkan kemampuan sistem dalam mempertahankan tingkat ketersediaan yang tinggi dengan waktu pemulihan yang cepat, sehingga mengurangi dampak negatif downtime terhadap layanan. Penelitian ini memberikan kontribusi signifikan dalam pemahaman dan implementasi sistem HA yang efisien dan handal dalam lingkungan virtualisasi berbasis Ceph dan Proxmox VE.

Kata Kunci: High Availability, Ceph, Proxmox VE, Virtual Machine, Container, Downtime, Skrip Otomatis, Stabilitas Sistem

I. PENDAHULUAN

Dalam lanskap teknologi informasi yang terus berkembang, ketersediaan tinggi (high availability/HA) menjadi semakin penting. High availability mengacu pada kemampuan sistem atau layanan untuk tetap beroperasi dan dapat diakses tanpa gangguan yang signifikan, bahkan ketika terjadi kegagalan komponen. Untuk mencapai tujuan ini, berbagai teknologi dan strategi telah dikembangkan, salah satunya adalah penggunaan Ceph, sebuah platform penyimpanan terdistribusi yang dikenal karena skalabilitas dan keandalannya. Ceph menawarkan kemampuan replikasi data yang canggih, self-healing, dan distribusi beban yang merata, menjadikannya pilihan yang menarik untuk membangun infrastruktur high availability.

Salah satu platform virtualisasi yang populer dan banyak digunakan adalah Proxmox Virtual Environment (VE). Proxmox VE menyediakan lingkungan yang fleksibel dan efisien untuk menjalankan mesin virtual (VM) dan container.

Dengan menggabungkan Ceph dan Proxmox VE, dimungkinkan untuk menciptakan infrastruktur server yang

tidak hanya memiliki ketersediaan tinggi tetapi juga skalabilitas dan kinerja yang optimal.

Penelitian ini berfokus pada perancangan dan implementasi sistem high availability server menggunakan Ceph pada Proxmox VE. Tujuan utama dari penelitian ini adalah untuk mengurangi downtime (waktu henti) server seminimal mungkin dan meminimalkan intervensi manusia dalam proses pemulihan sistem. Dengan menggabungkan keunggulan Ceph dalam penyimpanan terdistribusi dan keandalan Proxmox VE dalam virtualisasi, diharapkan dapat dicapai tingkat ketersediaan yang tinggi untuk layanan-layanan penting.

Penelitian ini akan membahas secara rinci langkah-langkah dalam merancang, mengimplementasikan, dan menguji sistem high availability yang diusulkan. Selain itu, akan dilakukan analisis mendalam terhadap hasil pengujian untuk mengevaluasi kinerja sistem dan mengidentifikasi area-area yang memerlukan perbaikan lebih lanjut. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam pengembangan sistem high availability yang lebih efisien dan handal di masa depan.

II. METODE

Tujuan dari penelitian ini adalah untuk menyelidiki kinerja dan keandalan implementasi High Availability Server menggunakan teknologi Ceph di lingkungan virtualisasi Proxmox melalui pendekatan eksperimental. Desain eksperimental yang digunakan melibatkan variabel independen seperti konfigurasi Ceph, konfigurasi jaringan, dan beban kerja sistem. Grup eksperimen terbagi menjadi grup kontrol yang tidak menggunakan implementasi High Availability dan grup perlakuan yang menerapkan implementasi High Availability menggunakan Ceph.

Penelitian ini mencakup semua server yang menggunakan platform virtualisasi Proxmox dengan kebutuhan High Availability. Sampel penelitian diambil dari konfigurasi server yang ada dalam lingkungan simulasi dengan spesifikasi tertentu

untuk menguji kinerja dan keandalan implementasi High Availability menggunakan Ceph.

Pengumpulan data dilakukan melalui pengukuran waktu pemulihan saat terjadi kegagalan node, ketersediaan layanan selama periode pengujian, dan kinerja sistem seperti throughput dan latency. Instrumen yang digunakan meliputi skrip otomatis untuk deteksi dan pemulihan kegagalan, serta perangkat lunak monitoring untuk mengumpulkan data performa.

Analisis data dilakukan dengan menggunakan metrik yang relevan dan metode statistik untuk membandingkan hasil antara kedua grup (kontrol dan perlakuan). Analisis ini mencakup perbandingan waktu pemulihan, tingkat ketersediaan, dan kinerja sistem. Hasil dari analisis ini diharapkan dapat memberikan wawasan yang berharga dalam implementasi High Availability menggunakan Ceph pada Proxmox serta memberikan landasan untuk pengembangan sistem yang lebih andal di masa depan.

Untuk menunjang seluruh proses pada penelitian, diperlukan beberapa kebutuhan, yaitu:

1. Kebutuhan perangkat keras (hardware), Berikut merupakan spesifikasi dari perangkat keras yang digunakan untuk menunjang penelitian ini:
 - Laptop
 - Prosesor: Intel Core i7-10750H
 - RAM: 24 GB
 - Penyimpanan: SSD 512 GB
 - Sistem Operasi: Windows 11
2. Kebutuhan perangkat lunak (software), Berikut beberapa perangkat lunak yang digunakan untuk menunjang penelitian ini:
 - VM Manager
 - Proxmox VE
 - Ceph

Pengujian dilakukan dengan menguji VM dan container yang memiliki spesifikasi berbeda di Proxmox yang juga memiliki spesifikasi yang berbeda-beda. Untuk mengevaluasi performa High Availability (HA) pada Proxmox menggunakan Ceph, setiap skema memiliki durasi pengujian selama 12 jam dengan pola downtime yang berbeda. Berikut adalah skema pengujiannya:

1. Pengujian Awal (2 jam):
Server dibiarkan menyala tanpa gangguan selama 2 jam pertama.
2. Pengujian Utama (8 jam):
Setiap satu jam, server akan dimatikan otomatis oleh skrip yang dipasang untuk menghitung waktu downtime. Proses ini diulangi sebanyak 8 kali.
3. Pengujian Akhir (2 jam):

Server dibiarkan menyala tanpa gangguan selama 2 jam terakhir untuk memastikan bahwa sistem High Availability (HA) yang diuji dapat mempertahankan stabilitas dan ketersediaan layanan setelah melalui serangkaian pengujian downtime yang disengaja

TABEL 1
 TABEL SKEMA PENGUJIAN

| No. | Durasi | Aktivitas | Keterangan |
|-----|-------------|--|---------------------------------|
| 1 | 0 - 2 jam | Server menyala | Pengujian awal, tanpa downtime |
| 2 | 2 - 3 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 1 |
| 3 | 3 - 4 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 2 |
| 4 | 4 - 5 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 3 |
| 5 | 5 - 6 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 4 |
| 6 | 6 - 7 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 5 |
| 7 | 7 - 8 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 6 |
| 8 | 8 - 9 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 7 |
| 9 | 9 - 10 jam | Server dimatikan otomatis, downtime dihitung | Pengujian utama, downtime 8 |
| 10 | 10 - 12 jam | Server menyala | Pengujian akhir, tanpa downtime |

Dengan skema pengujian yang melibatkan downtime yang disengaja setiap jam selama 8 jam, penelitian ini dapat melakukan analisis mendalam terhadap kemampuan server dalam mempertahankan ketersediaan tinggi (High Availability/HA) dalam menghadapi gangguan berulang.

III. HASIL DAN PEMBAHASAN

Implementasi dan evaluasi sistem High Availability (HA) menggunakan CEPH pada Proxmox VE menjadi fokus dalam Penelitian ini.

A. Proses Instalasi

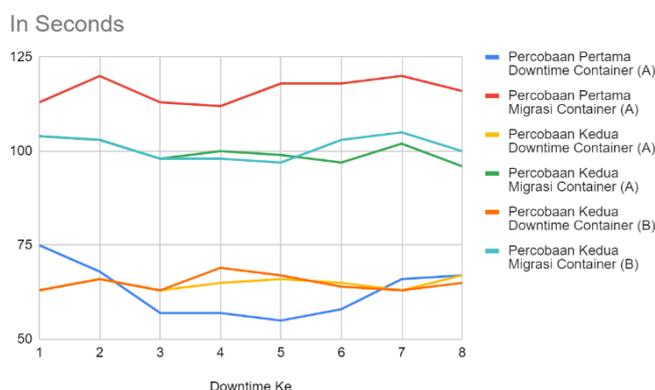
Proses instalasi Proxmox VE dilakukan pada server dengan dua disk (50GB untuk sistem dan 25GB untuk Ceph). Tahapan instalasi meliputi persiapan dan pembuatan media instalasi, pemilihan disk dan partisi, konfigurasi jaringan, serta pengaturan lokasi dan pengguna. Instalasi berjalan lancar tanpa kendala berarti.

1. Persiapan dan Instalasi Media:
 - File ISO Proxmox VE diunduh dan media bootable dibuat menggunakan Rufus.
 - Sistem berhasil di-boot dari media instalasi dan proses instalasi dimulai dengan memilih opsi "Install Proxmox VE".
2. Pemilihan Disk dan Partisi:
Disk 50GB dipilih untuk instalasi sistem Proxmox VE dan konfigurasi partisi otomatis berjalan dengan baik.
3. Konfigurasi Jaringan:
IP statis, gateway, dan DNS server "1.1.1.1" diatur selama instalasi, dan jaringan terkonfigurasi dengan baik.
4. Konfigurasi Lokasi dan Pengguna:
Zona waktu dipilih sesuai dengan lokasi (WIB), password untuk user root dibuat, dan email untuk notifikasi diatur.
5. Proses Instalasi dan Reboot:
Instalasi sistem berjalan lancar tanpa error dan sistem dapat diakses melalui antarmuka web setelah reboot.

Server kembali dinyalakan dan pengukuran dilakukan untuk memastikan stabilitas.

Berikut spesifikasi skema yang diujikan:

1. Proxmox Percobaan Pertama: 1GB RAM, 1 CPU
2. Proxmox Percobaan Kedua: 4GB RAM, 2 CPU
3. Proxmox Percobaan Ketiga: 7GB RAM, 3 CPU
4. Container (A): 2GB RAM, 1 CPU
5. Container (B): 4GB RAM, 1 CPU
6. VM (A): 2GB RAM, 1 CPU
7. VM (B): 4GB RAM, 2 CPU



Gbr 1. Gambar semua percobaan Container

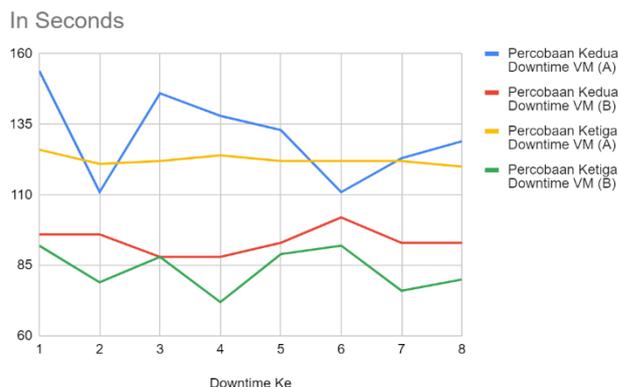
B. Konfigurasi Repositori

Setelah instalasi, konfigurasi repositori non-subscription dilakukan untuk memastikan stabilitas dan akses ke pembaruan terbaru. Ini dilakukan dengan menambahkan URL repositori pada file konfigurasi apt di Proxmox VE.

C. Pengujian Sistem

Pengujian dilakukan dalam beberapa tahap, melibatkan simulasi downtime dan analisis kemampuan sistem dalam mempertahankan HA. Pengujian ini bertujuan untuk mengukur ketersediaan dan performa sistem dalam kondisi yang berbeda:

1. Pengujian Awal:
Server dimatikan otomatis setiap jam selama delapan kali untuk menghitung downtime.
2. Pengujian Utama:
Pengujian berlanjut dengan server menyala selama dua jam setelah pengujian awal.
3. Pengujian Akhir:



Gbr 2. Gambar semua percobaan VM

Dalam penelitian ini, pengujian dilakukan dengan memanfaatkan dua skrip Python yang dirancang khusus. Skrip pertama berfungsi untuk memantau downtime server secara otomatis dengan mengirimkan perintah ping ke alamat IP server setiap detik dan mencatat waktu terjadinya downtime. Sementara itu, skrip kedua bertanggung jawab untuk mengotomatiskan proses mematikan dan menghidupkan server sesuai dengan skema pengujian yang telah ditentukan.

sebelumnya. Skrip ini sangat penting dalam menjalankan simulasi kegagalan server dan mengukur waktu pemulihan sistem secara akurat. Hasil dari pengujian ini kemudian dianalisis dan disajikan dalam bentuk tabel untuk memberikan gambaran yang jelas mengenai kinerja sistem high availability (HA) yang diuji.. Dari Gambar 1 dan Gambar 2 didapat tabel sebagai berikut:

TABEL 2
 TABEL HA SEMUA PERCOBAAN

| Percobaan | Container (A) HA (%) | Container (B) HA (%) | VM (A) HA (%) | VM (B) HA (%) |
|-------------------|----------------------|----------------------|---------------|---------------|
| Percobaan Pertama | 96.682 | - | - | - |
| Percobaan Kedua | 96.951 | 96.925 | 97.581 | 98.266 |
| Percobaan Ketiga | 94.513 | 94.631 | 97.733 | 98.453 |

Dari tabel di atas, dapat dilihat bahwa VM (Virtual Machines) memiliki tingkat ketersediaan yang lebih tinggi dibandingkan Container pada setiap percobaan. Hal ini menunjukkan bahwa VM lebih andal dalam menjaga ketersediaan layanan, yang sangat penting dalam implementasi High Availability (HA).

Hasil pengujian ini menunjukkan bahwa VM pada Proxmox VE lebih unggul dalam menjaga ketersediaan dibandingkan dengan Container. Oleh karena itu, untuk lingkungan yang sangat mengutamakan High Availability, penggunaan VM dengan alokasi sumber daya yang memadai pada hypervisor menjadi pilihan yang lebih tepat. Penelitian ini memberikan panduan praktis bagi organisasi dalam memilih teknologi yang tepat untuk mencapai tingkat ketersediaan layanan yang optimal.

IV. SIMPULAN

Berdasarkan dari percobaan yang dilakukan dalam penelitian ini, maka dapat disimpulkan sebagai berikut:

1. Penelitian ini berhasil merancang dan mengimplementasikan sistem high availability (HA) menggunakan Ceph, sebuah platform penyimpanan terdistribusi yang dikenal karena skalabilitas dan keandalannya, pada Proxmox Virtual Environment (VE), sebuah platform virtualisasi open-source yang populer.

Arsitektur sistem high availability yang dihasilkan dari penelitian ini dirancang dengan cermat, melibatkan konfigurasi cluster Ceph yang terdiri dari beberapa node penyimpanan, serta instalasi dan pengaturan komponen-komponen penting seperti monitor daemon (untuk pemantauan kesehatan cluster), OSD (Object Storage Daemon) untuk pengelolaan penyimpanan objek, dan manager daemon untuk orkestrasi tugas-tugas administratif. Untuk menguji ketahanan sistem terhadap gangguan, simulasi kegagalan dilakukan dengan sengaja mematikan salah satu node server. Waktu pemulihan, yaitu waktu yang dibutuhkan sistem untuk kembali beroperasi normal setelah kegagalan, diukur sebagai metrik utama untuk mengevaluasi efektivitas mekanisme high availability yang diterapkan. Selain itu, performa sistem HA juga dievaluasi secara menyeluruh melalui serangkaian percobaan yang berlangsung selama 12 jam. Hasil penelitian ini memberikan bukti kuat bahwa Virtual Machines (VM) menunjukkan tingkat stabilitas dan performa HA yang lebih unggul dibandingkan dengan container, terutama dalam hal menjaga konsistensi ketersediaan layanan, bahkan ketika dihadapkan pada gangguan atau kegagalan. Keunggulan VM ini terlihat dari kemampuannya untuk mempertahankan tingkat ketersediaan layanan yang tinggi secara konsisten selama periode pengujian. Lebih lanjut, penelitian ini juga mengungkapkan bahwa peningkatan spesifikasi perangkat keras pada Proxmox VE, seperti penambahan kapasitas RAM, jumlah core CPU, atau penggunaan perangkat penyimpanan yang lebih cepat, memberikan dampak positif yang signifikan terhadap kinerja HA, terutama pada VM. Hal ini menunjukkan bahwa VM mampu memanfaatkan peningkatan sumber daya dengan lebih efektif untuk meningkatkan ketahanan dan ketersediaan layanan, sehingga menjadi pilihan yang lebih optimal dalam lingkungan yang membutuhkan tingkat HA yang tinggi.

2. Penelitian ini berhasil merancang dan mengimplementasikan sistem high availability (HA) menggunakan Ceph pada Proxmox VE, dengan fokus pada minimalisasi intervensi manusia dalam pemulihan sistem melalui skrip otomatis. Hasil pengujian menunjukkan bahwa virtual machine (VM) memiliki stabilitas dan performa HA yang lebih unggul dibandingkan container, mencapai tingkat ketersediaan lebih dari 97% dalam 70 berbagai skenario kegagalan yang disimulasikan. Peningkatan spesifikasi pada Proxmox, seperti penambahan sumber daya komputasi dan memori, juga memberikan dampak positif yang signifikan pada kinerja

HA, terutama pada VM, yang mampu memanfaatkan sumber daya tambahan dengan lebih efektif untuk menjaga keberlangsungan layanan. Secara keseluruhan, penelitian ini berhasil mengembangkan dan mengimplementasikan skrip otomatis yang efektif untuk deteksi dan pemulihan kegagalan pada lingkungan virtualisasi berbasis Ceph dan Proxmox VE. Skrip ini mampu mengidentifikasi berbagai jenis kegagalan, seperti kegagalan node, kegagalan jaringan, dan kegagalan penyimpanan, serta melakukan tindakan pemulihan yang sesuai secara otomatis, seperti migrasi VM ke node yang sehat atau aktivasi replika data pada Ceph. Dengan demikian, penelitian ini memberikan kontribusi signifikan dalam meningkatkan ketersediaan dan stabilitas layanan pada infrastruktur virtualisasi, mengurangi downtime, dan meminimalkan kerugian akibat gangguan layanan.

V. SARAN

Berikut adalah saran yang dapat diberikan untuk penelitian selanjutnya:

1. Penelitian selanjutnya dapat mempertimbangkan untuk menggunakan perangkat keras yang lebih kuat untuk menjalankan Proxmox VE dan Ceph, serta menguji skenario yang lebih kompleks dan mendapatkan hasil yang lebih akurat tentang kinerja dan skalabilitas sistem. Misalnya, penelitian dapat menguji bagaimana sistem merespons 71 ketika beberapa node server gagal secara bersamaan atau ketika beban kerja sistem sangat tinggi. Penelitian ini juga dapat diperluas dengan menguji berbagai konfigurasi Ceph dan Proxmox VE, seperti jumlah OSD yang berbeda, ukuran cluster yang berbeda, dan jenis beban kerja yang berbeda. Hal ini akan memberikan pemahaman yang lebih komprehensif tentang bagaimana Ceph dan Proxmox VE dapat digunakan bersama untuk mencapai ketersediaan tinggi dalam berbagai situasi. Selain itu, penelitian selanjutnya dapat fokus pada pengembangan skrip otomatis yang lebih canggih untuk mendeteksi dan memulihkan kegagalan pada Proxmox VE. Skrip ini dapat menggunakan algoritma machine learning untuk memprediksi potensi kegagalan dan mengambil tindakan pencegahan sebelum kegagalan terjadi. Dengan demikian, penelitian ini dapat memberikan kontribusi yang lebih signifikan dalam meningkatkan ketersediaan dan keandalan sistem high availability berbasis Ceph pada Proxmox VE.
2. Penelitian selanjutnya dapat mengembangkan sistem yang lebih komprehensif dengan mengintegrasikan mekanisme

pemantauan yang lebih canggih. Pemantauan secara realtime terhadap kesehatan dan kinerja komponen sistem dapat membantu mendeteksi potensi masalah sebelum berkembang menjadi kegagalan. Selain itu, otomatisasi pemulihan dapat ditingkatkan dengan mengembangkan algoritma yang lebih cerdas untuk mengidentifikasi akar penyebab kegagalan dan melakukan tindakan pemulihan yang sesuai tanpa intervensi manusia. Penelitian lebih lanjut juga dapat mengeksplorasi penggunaan teknologi kecerdasan buatan (AI) dan machine learning (ML) untuk memprediksi dan mencegah downtime. Dengan menganalisis pola-pola data historis, AI dan ML dapat mengidentifikasi anomali dan potensi masalah, sehingga memungkinkan tindakan pencegahan yang proaktif untuk menghindari downtime yang tidak terduga.

REFERENSI

- [1] A. Yudhi, "Implementasi Computer Cluster Berbasis Open Source untuk Penyeimbang Beban Sistem Jaringan Komputer," *Jurnal Teknik Informatika dan Sistem Informasi Volume 2 Nomor 1 April 2016*, vol. 2, no. 1, 2016.
- [2] Mulyanto and A. Ahmad, "Implementasi Highly Available Website Dengan Distributed Replicated Block Device," *IJCCS, Vol.10, No.2, July 2016, pp. 149-160*, vol. Vol 10, no. 2016, Jul. 2016.
- [3] N. Kong, K. Kit, and M. Aibin, "Study on High Availability and Fault Tolerance," 2023.
- [4] J. Mantik, A. T. Irmawan, R. Maulany, and A. Thymoti, "Implementation of proxmox server monitoring system with laravel and vue.js," Online, 2024.
- [5] M. R. Mesbahi, A. M. Rahmani, and M. Hosseinzadeh, "Reliability and high availability in cloud computing environments: a reference roadmap," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, Dec. 2018, doi: 10.1186/s13673-018-0143-8.
- [6] S. R. Siregar and S. Ramadan Siregar, "(media cetak) Efisiensi Fisik Komputer Server dengan Menerapkan Proxmox Virtual Environment," *Journal of Computer System and Informatics (JoSYC)*, vol. 1, no. 2, pp. 83-87, 2020.
- [7] K. Marzuki, M. I. Kholid, I. P. Hariyadi, and L. Z. A. Mardedi, "Automation of Open VSwitch-Based Virtual Network Configuration Using Ansible on Proxmox Virtual Environment," *International Journal of Electronics and Communications Systems*, vol. 3, no. 1, p. 11, Jun. 2023, doi: 10.24042/ijecs.v3i1.16524.
- [8] Y. Ariyanto, B. Harijanto, V. A. H. Firdaus, and S. N. Arief, "Performance analysis of Proxmox VE firewall for network security in cloud computing server implementation," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1757-899X/732/1/012081.
- [9] B. Arifwidodo, V. Metayasha, and S. Ikhwan, "Analisis Kinerja Load Balancing pada Server Web Menggunakan Algoritma Weighted Round Robin pada Proxmox VE," *Jurnal Telekomunikasi dan Komputer*, vol. 11, no. 3, p. 210, Dec. 2021, doi: 10.22441/incomtech.v11i3.11775.
- [10] S. Farizy Emi Sita Eriana Ji Surya Kencana No, P. A. Gd, and U. Pamulang Tangerang Selatan -Banten, *Universitas Pamulang Sistem Informasi Cloud Computing = Komputasi Awan i CLOUD COMPUTING = KOMPUTASI AWAN*. [Online]. Available: www.unpam.ac.id
- [11] I. Ikbal, "TATA KELOLA TEKNOLOGI INFORMASI DI UNIVERSITAS PQR BERBASIS FRAMEWORK COBIT 4.1."
- [12] I. Ar-Razy, R. Kridalukmana, and E. D. Widiyanto, "Implementasi Cloud Storage Menggunakan OwnCloud yang High-Availability,"

- Jurnal Teknologi dan Sistem Komputer*, vol. 4, no. 2, p. 209, Apr. 2016, doi: 10.14710/jtsiskom.4.2.2016.209-214.
- [13] Y. Pribadi, A. B. Putra Negara, and M. A. Irwansyah, "Analisis Penggunaan Metode Failover Clustering untuk Mencapai High Availability pada Web Server (Studi Kasus: Gedung Jurusan Informatika)," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 2, p. 218, Apr. 2020, doi: 10.26418/justin.v8i2.31965.
- [14] L. Apriliana, U. Darusala, and N. D. Nathasia, "Clustering Server Pada Cloud Computing Berbasis Proxmox VE Menggunakan Metode High Availability," *JOINTECS) Journal of Information Technology and Computer Science*, vol. 3, no. 1, 2018, doi: 10.31328/jo.
- [15] C. Umam, L. B. Handoko, and G. M. Rizqi, "Implementation And Analysis High Availability Network File System Based Server Cluster ARTICLE INFO," *TRANSFORMATIKA*, vol. 16, no. 1, pp. 31–39, 2018, [Online]. Available: www.snia.org/dictionary
- [16] D. Irwan, H. Sukoco, and S. Wahjuni, "Service High Availability Pada Native Server dan Virtual Server Menggunakan Proxmox VE," 2020. [Online]. Available: <http://ejurnal.ubharajaya.ac.id/index.php/JKI>
- [17] V. Oleksiuk and O. Oleksiuk, "The practice of developing the academic cloud using the Proxmox VE platform," *Educational Technology Quarterly*, vol. 2021, no. 4, pp. 605–616, Dec. 2021, doi: 10.55056/etq.36.
- [18] G. Kambourakis, G. D. Gil, and I. Sanchez, "What Email Servers Can Tell to Johnny: An Empirical Study of Provider-to-Provider Email Security," *IEEE Access*, vol. 8, pp. 130066–130081, 2020, doi: 10.1109/ACCESS.2020.3009122.
- [19] A. Arfriandi, "PERANCANGAN, IMPLEMENTASI, DAN ANALISIS KINERJA VIRTUALISASI SERVER MENGGUNAKAN PROXMOX, VMWARE ESX, DAN OPENSTACK," 2012.
- [20] T. Setia Hati Politeknik Negeri Jakarta and T. Setia Hati, "High Availability Server As A Reverse Proxy Using SSL Termination and Caching in PT. Mitra Akses Globalindo," 2019. [Online]. Available: <https://www.incapsula.com>
- [21] M. J. Curtis *et al.*, "Experimental design and analysis and their reporting II: updated and simplified guidance for authors and peer reviewers," *British Journal of Pharmacology*, vol. 175, no. 7. John Wiley and Sons Inc., pp. 987–993, Apr. 01, 2018. doi: 10.1111/bph.14153.