

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam era transformasi digital yang pesat, kebutuhan akan sistem yang memiliki tingkat ketersediaan tinggi (*High Availability*) dan toleransi terhadap kegagalan (*Fault Tolerance*) menjadi sangat penting bagi organisasi maupun lembaga yang bergantung pada layanan berbasis server. Gangguan kecil pada sistem dapat menyebabkan *downtime* yang akan berakibat fatal pada kegiatan operasional, menurunkan produktivitas hingga kerugian finansial. Menurut (Saxena & Singh, 2023) sistem dengan tingkat ketersediaan tinggi berperan penting dalam menjaga keberlangsungan layanan digital, terutama dalam lingkungan komputasi awan dan infrastruktur jaringan berskala besar. Oleh karena itu, diperlukan infrastruktur yang mampu menjamin layanan tetap berjalan meskipun terjadi kegagalan pada salah satu komponen sistem.

Salah satu solusi yang banyak digunakan untuk memenuhi kebutuhan tersebut adalah platform virtualisasi Proxmox Virtual Environment (Proxmox VE). Proxmox VE tersedia dalam dua versi, yaitu *enterprise edition* (berbayar) dan *community edition* (gratis). Versi *enterprise* menyediakan repository stabil, dukungan resmi dan pembaharuan terjamin sedangkan versi *community* menggunakan *repository non-enterprise* tanpa dukungan komersial, sehingga potensi error dan ketidakstabilan lebih tinggi. Namun, secara fitur inti seperti clustering, *High Availability*, *storage management* dan integrasi Ceph, kedua versi tetap menyediakan kemampuan yang sama. Dalam konteks penelitian akademik dan lingkungan laboratorium, *community edition* menjadi pilihan ideal karena gratis, terbuka, legal digunakan, serta mampu menampilkan performa real dari teknologi HA tanpa adanya optimasi premium dari *enterprise repository*. Namun, meskipun versi gratis Proxmox VE secara luas digunakan dalam praktik, efektivitas fitur *High Availability* dan *Fault Tolerance* pada versi ini masih jarang diuji secara ilmiah. Hal ini menjadi isu penting karena mayoritas

pengguna di lingkungan akademik atau organisasi kecil mengandalkan versi gratis, tetapi belum memiliki data yang jelas mengenai kemampuannya dalam menghadapi kegagalan fisik yang nyata.

Studi lain oleh (Saskara et al., 2025) memperlihatkan bahwa Proxmox VE mampu meningkatkan ketersediaan layanan hingga 99,89% pada implementasi sistem informasi akademik, tetapi penelitian tersebut hanya terbatas pada pengujian di tingkat aplikasi tanpa mensimulasikan kerusakan perangkat fisik.

Kondisi tersebut menimbulkan masalah kritis yang menjadi dasar penelitian ini, yaitu belum ada kajian mendalam mengenai performa High Availability dan Fault Tolerance pada Proxmox VE sumber terbuka yang menggunakan repositori komunitas (gratis) ketika dihadapkan pada kegagalan fisik seperti pemutusan daya listrik dan kerusakan hard disk. Selain itu belum terdapat data empiris yang mengevaluasi bagaimana perilaku sistem, seperti rata rata waktu *failover*, rata rata waktu *downtime* dan rata rata waktu *recovery data*.

Berdasarkan hal tersebut, penelitian ini dilakukan untuk menguji ketahanan sistem cluster Proxmox VE dengan tiga server fisik terhadap dua jenis kegagalan nyata (*real hardware failure simulation*). Pengujian dilakukan dengan mensimulasikan pemutusan daya listrik (*power failure*) dan pelepasan *hard disk* (*disk failure*) secara langsung pada salah satu *node cluster* yang sedang beroperasi. Dalam penelitian ini digunakan Ceph Storage sebagai penyimpanan terdistribusi untuk mendukung mekanisme *High Availability* pada cluster Proxmox VE. Ceph dipilih karena mampu mempertahankan ketersediaan data meskipun terjadi kegagalan fisik pada salah satu disk, sehingga pengujian *disk failure* dapat dilakukan secara realistis tanpa menyebabkan hilangnya data. Pengukuran rata-rata waktu *failover* akan dilihat dari log yang menyimpan timestamp kejadian seperti node mati, VM berpindah dan VM hidup kembali, sedangkan untuk pengukuran rata-rata waktu *downtime* layanan digunakan analisis timestamp saat layanan FTP dan Mail Server berhenti merespon hingga kembali normal. Untuk menghindari human error dan memastikan konsistensi data, proses perhitungan waktu akan di automatisasi oleh *script Bash* (Shell Script). Melalui

pendekatan ini, penelitian diharapkan mampu Untuk pengukuran rata-rata waktu failover akan dilihat dari log yang menyimpan timestamp kejadian seperti node mati, VM berpindah dan VM hidup kembali, sedangkan untuk pengukuran rata-rata waktu downtime layanan digunakan analisis *timestamp* saat layanan FTP dan Mail Server berhenti merespon hingga kembali normal. memberikan hasil yang lebih realistis mengenai kemampuan Proxmox VE dalam mempertahankan ketersediaan layanan (*service availability*) serta kestabilan sistem pada kondisi ekstrem tanpa menggunakan lisensi komersial.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana kemampuan fitur High Availability dan Fault Tolerance pada Proxmox VE dalam menjaga keberlangsungan layanan saat terjadi kegagalan di salah satu node, seperti simulasi pemutusan daya (*power failure*) atau kerusakan media penyimpanan (*disk failure*)?
2. Berapa rata-rata waktu *downtime* dan *failover* yang terjadi pada sistem *cluster* Proxmox VE ketika salah satu node mengalami gangguan secara tiba tiba (*power failure, disk failure*), serta bagaimana hasil pengujian tersebut dikategorikan sebagai baik atau buruk berdasarkan indikator performa seperti:
 - a) Durasi *downtime* yang tercatat?
 - b) Kecepatan eksekusi *failover* ?
3. Bagaimana rata rata waktu recovery data pada Ceph ketika terjadi *disk failure* pada salah satu node?

1.3 Batasan Masalah

Agar ruang lingkup penelitian tidak terlalu luas, batasan masalah yang diterapkan adalah sebagai berikut:

1. Penelitian hanya menggunakan Proxmox VE versi *Community Edition* tanpa lisensi berbayar (*Enterprise Subscription*).

2. Fokus penelitian pada pengujian fitur *High Availability* bawaan *Proxmox VE* tanpa perangkat tambahan komersial.
3. Penelitian juga mencakup *Fault Tolerance* pada *cluster*, seperti replikasi data dan kemampuan sistem dalam mempertahankan data.
4. *Cluster* dibangun menggunakan tiga node fisik, yang berfungsi sebagai node utama, node cadangan dan node quorum.
5. Pengujian dilakukan dengan simulasi kegagalan fisik secara langsung (*real hardware failure simulation*) yang mencakup :
 - a. Pemutusan daya listrik mendadak (*power failure*) dengan mencabut kabel power node untuk mensimulasikan server yang mati akibat *power loss, overload* listrik.
 - b. Pemutusan atau pelepasan hard disk (*disk failure*) saat node aktif untuk mensimulasikan kerusakan media penyimpanan akibat *bad sector*, konektor longgar, error pada SATA/NVMe atau drive failure lainnya.
6. Parameter evaluasi yang diamati meliputi :
 - a. Rata-rata waktu *downtime* layanan (detik) dan *failover* dari node gagal ke node sehat.
 - b. Rata-rata waktu *recovery* data
7. Penelitian ini tidak membahas aspek keamanan jaringan (*network security*), efisiensi energi maupun skalabilitas cluster diatas tiga node.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Adapun tujuan dari penelitian ini adalah :

1. Mengevaluasi kemampuan *High Availability* dan *Fault Tolerance* pada *Proxmox VE* dalam mempertahankan kontinuitas layanan ketika terjadi kegagalan fisik pada salah satu *node cluster*, seperti *power failure* atau *disk failure*.
2. Mengukur parameter performa sistem dengan membedakan indikator *High Availability* dan *Fault Tolerance*, yaitu :

- a. High Availability :
 - Rata-rata waktu *downtime*
 - Rata-rata waktu *failover*
- b. Fault Tolerance :
 - Rata-rata waktu *recovery* data pada Ceph

1.4.2 Manfaat

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Menyediakan data empiris yang dapat dimanfaatkan oleh praktisi teknologi informasi dalam menilai kelayakan Proxmox VE sebagai solusi HA tanpa lisensi komersial.
2. Menambah referensi ilmiah mengenai performa *High Availability* pada platform virtualisasi Proxmox VE.
3. Memberikan rekomendasi teknis bagi institusi atau organisasi dalam merancang infrastruktur server yang tangguh, stabil, dan efisien dengan memanfaatkan teknologi *open-source*.
4. Menyajikan tolak ukur (*benchmark*) performa seperti durasi *downtime*, waktu *failover*, dan stabilitas *cluster* untuk kebutuhan evaluasi maupun pengembangan lebih lanjut.