

**IMPLEMENTASI HORIZONTAL POD AUTOSCALER PADA
KUBERNETES DALAM MENINGKATKAN SKALABILITAS
APLIKASI WEB BERBASIS PHP**

TUGAS AKHIR



HIKMAH NURSIDI

NIM. 21106050001

**STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA**

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2025

HALAMAN PENGESAHAN



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-912/Un.02/DST/PP.00.9/06/2025

Tugas Akhir dengan judul : Implementasi Horizontal Pod Autoscaler pada Kubernetes dalam Meningkatkan Skalabilitas Aplikasi Web berbasis PHP

yang dipersiapkan dan disusun oleh:

Nama : HIKMAH NURSIDIK
Nomor Induk Mahasiswa : 21106050001
Telah diujikan pada : Jumat, 23 Mei 2025
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Valid ID: 683683538b15f

Ketua Sidang

Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D.
SIGNED



Valid ID: 68367b56d7978

Penguji I

Dr. Ir. Bambang Sugiantoro, M.T., IPU.,
ASEAN Eng.
SIGNED



Valid ID: 683661291f046

Penguji II

Dwi Otik Kurniawati, M.Eng.
SIGNED



Valid ID: 683700db8138b

Yogyakarta, 23 Mei 2025
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

LEMBAR PERNYATAAN

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Hikmah Nursidik

NIM : 21106050001

Dosen Pembimbing : Ir. Muhammad Taufiq Nuruzzaman, S.T., M.Eng., Ph.D.

Saya menyatakan bahwa skripsi saya yang berjudul **"Implementasi Horizontal Pod Autoscaler pada Kubernetes dalam Meningkatkan Skalabilitas Aplikasi Web berbasis PHP"** merupakan hasil penelitian saya sendiri. Sejauh pengetahuan saya, penelitian ini belum pernah dilakukan atau dipublikasikan di tempat lain, kecuali sebagai acuan atau kutipan yang tercantum dalam daftar pustaka pada naskah ini.

Skripsi ini juga diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Yogyakarta, 19 Mei 2025

Yang menyatakan,



HIKMAH NURSIDIK

NIM. 21106050001

SURAT PERSETUJUAN TUGAS AKHIR



Universitas Islam Negeri Sunan Kalijaga



FM-UINSK-BM-05-03/R0

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Di Yogyakarta

Assalammu 'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka saya selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Hikmah Nursidik

NIM : 21106050001

Judul Skripsi : Implementasi Horizontal Pod Autoscaler pada Kubernetes dalam Meningkatkan Skalabilitas Aplikasi Web berbasis PHP

sudah dapat diajukan kembali kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara dapat segera dimunaqasyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalammu 'alaikum wr. wb.

Yogyakarta, 16 Mei 2025

Pembimbing

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D.

MOTTO

خَيْرُ النَّاسِ أَنْفَعُهُمُ لِلنَّاسِ

“Sebaik-baiknya manusia ialah yang bermanfaat untuk manusia yang lainnya”



HALAMAN PERSEMBAHAN

Dalam mengerjakan Tugas Akhir ini, hal utama bagi penulis adalah meniatkannya *lillahi ta'ala*. Dengan niat itu pula, penulis dapat menyelesaikan tugas akhir ini dan mempersembahkannya khusus untuk:

Kedua Orang Tua Tercinta, ***Bapak Arif Samijan dan Ibu Ragil Mumtathiah,***

Yu Novi Ngumrina dan Keluarga,

Yu Meli Sofingatun dan Keluarga,

Mas Bagus Firmansyah dan Keluarga

Semua Orang yang Pernah Membuat Saya dari Tidak Tahu Menjadi Tahu

Almamater Tercinta

Program Studi Informatika

Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN Sunan Kalijaga, yang diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen Tugas Akhir ini merupakan hak milik UIN Sunan Kalijaga Yogyakarta.



INTISARI

Aplikasi web berbasis PHP rentan mengalami penurunan performa saat lonjakan trafik tak terduga, berpotensi menyebabkan kegagalan sistem. Kubernetes, sebagai platform orkestrasi container yang populer, menawarkan fitur Horizontal Pod Autoscaler (HPA) untuk menyesuaikan jumlah Pod secara otomatis berdasarkan metrik seperti penggunaan CPU atau memori. Penelitian ini bertujuan menguji implementasi HPA pada Kubernetes dalam meningkatkan skalabilitas aplikasi web berbasis PHP yang di-deploy di server lokal.

Metode yang digunakan adalah pengujian eksperimental dengan dua skenario, yaitu *tanpa HPA* (satu Pod statis) dan *dengan HPA* (*autoscaling* hingga delapan Pod). Simulasi beban dilakukan menggunakan Locust dengan skala 100 hingga 1.500 *user* virtual. Hasil pengujian menunjukkan bahwa pada skenario tanpa HPA, *success rate* turun hingga 68%, *average response time* melonjak di atas 13.000 ms, dan *throughput* stagnan sekitar 1.080 *bytes per request*. Sebaliknya, dengan HPA, *success rate* mencapai 100%, *average response time* turun di bawah 4.100 ms, dan *throughput* meningkat menjadi 2.500 *bytes per request*.

Simpulan penelitian membuktikan bahwa HPA secara signifikan meningkatkan skalabilitas aplikasi dalam menghadapi lonjakan trafik. Untuk pengembangan selanjutnya, disarankan optimalisasi konfigurasi HPA berbasis *multi-metric* dan pengujian pada skala pengguna yang lebih besar.

Kata Kunci : *Autoscaling*, Kubernetes, *Horizontal Pod Autoscaler*, Skalabilitas, *Load Testing*

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

PHP-based web applications are prone to performance degradation during unexpected traffic spikes, risking system failure. Kubernetes, a popular container orchestration platform, offers the Horizontal Pod Autoscaler (HPA) feature to automatically adjust the number of Pods based on metrics such as CPU or memory usage. This study aims to test the implementation of HPA in Kubernetes to enhance the scalability of a PHP-based web application deployed on a local server.

The experimental method compared two scenarios: without HPA (a single static Pod) and with HPA (autoscaling up to eight Pods). Load testing was conducted using the Locust tool, simulating 100 to 1,500 virtual users. Results revealed that without HPA, the success rate dropped to 68%, average response time surged above 13,000 ms, and throughput stagnated at approximately 1,080 bytes per request. In contrast, with HPA enabled, the success rate reached 100%, average response time decreased below 4,100 ms, and throughput improved to 2,500 bytes per request.

These findings demonstrate that HPA significantly enhances application scalability during traffic surges. For future work, optimizing HPA configuration with multi-metric scaling and testing at larger user scales are recommended.

Keyword : *Autoscaling, Kubernetes, Horizontal Pod Autoscaler, Skalabilitas, Load Testing*

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala limpahan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul *“Implementasi Horizontal Pod Autoscaler pada Kubernetes dalam Meningkatkan Skalabilitas Aplikasi Web berbasis PHP”* dengan lancar.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Dalam proses penyusunan tugas akhir ini, penulis banyak menerima bantuan, dukungan, serta arahan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Keluarga tercinta, Bapak Samijan dan Ibu Ragil Mumtathiah, Yu Novi Ngumrina beserta keluarga, Yu Meli Sofingatun beserta keluarga, serta Mas Bagus Firmansyah beserta keluarga yang selalu memberikan dukungan moril maupun materil, serta doa yang tiada henti.
2. Bapak Prof. Noorhaidi, M.A., M.Phil., Ph.D., selaku Rektor Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
3. Ibu Prof. Dr. Dra. Hj. Khurul Wardati, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Ir. Muhammad Taufiq Nuruzzaman, S.T., M.Eng., Ph.D., selaku dosen pembimbing yang telah dengan sabar membimbing, memberikan arahan, kritik, dan motivasi selama proses penyusunan tugas akhir ini.
5. Bapak Dr. Ir. M. Mustakim, S.T., M.T., selaku Ketua Program Studi Teknik Informatika.
6. Bapak Ir. Muhammad Didik Rohmad Wahyudi, S.T., M.T., selaku dosen penasihat akademik yang senantiasa memberikan arahan dan pendampingan selama masa studi.

7. Bapak Mandahadi Kusuma, M.Eng. dan Bapak Eko Hadi Gunawan, M.Eng. yang telah banyak membantu dalam pelaksanaan penelitian di Laboratorium Terpadu Informatika.
8. Seluruh Bapak/Ibu dosen Informatika UIN Sunan Kalijaga yang telah memberikan ilmu, inspirasi, dan semangat belajar selama masa perkuliahan.
9. Abah K.H. Munir Syafa'at dan Ibu Nyai Hj. Barokah Nawawi, serta seluruh keluarga besar Pondok Pesantren Hidayatul Mubtadi-ien, Kotagede, atas doa dan dorongan spiritual yang tiada henti.
10. Bank Indonesia yang telah memberikan beasiswa dan dukungan finansial, sehingga penulis dapat lebih fokus dalam menyelesaikan studi.
11. Rekan-rekan organisasi, khususnya keluarga besar FKIST dan EXACT, yang telah menjadi wadah berproses, berkembang, dan saling menguatkan.
12. Keluarga Besar Pusat Teknologi Informasi dan Pangkalan Data UIN Sunan Kalijaga, khususnya Mas Yoga, Pak Isna, dan Pak Gatra yang selalu memberi motivasi.
13. Teman-teman seperjuangan Informatika angkatan 2021 UIN Sunan Kalijaga Yogyakarta, atas kebersamaan, semangat, dan saling bantu selama masa studi.
14. Seluruh pihak yang tidak dapat disebutkan satu per satu, namun telah memberikan kontribusi dalam penyusunan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan. Oleh karena itu, segala bentuk kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan di masa yang akan datang. Semoga karya ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi pembaca sekalian.

Yogyakarta, 19 Mei 2025

Penulis,

Hikmah Nursidik

NIM. 21106050001

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
LEMBAR PERNYATAAN	iii
SURAT PERSETUJUAN TUGAS AKHIR.....	iv
MOTTO.....	v
HALAMAN PERSEMBAHAN.....	vi
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR.....	vii
INTISARI.....	viii
ABSTRACT.....	ix
KATA PENGANTAR.....	x
DAFTAR ISI	xii
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
DAFTAR KODE SUMBER	xvi
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	4
2.1 Tinjauan Pustaka	4
2.2 Landasan Teori	8
2.2.1 Virtualisasi.....	8
2.2.2 Proxmox Virtual Environment (PVE).....	9
2.2.3 Terraform.....	9
2.2.4 Ansible.....	10
2.2.5 Docker	10
2.2.6 Kubernetes.....	11

2.2.7 <i>Horizontal Pod Autoscaler</i>	14
2.2.8 <i>Load Testing</i>	15
2.2.9 <i>Locust</i>	16
BAB 3 METODE PENGEMBANGAN SISTEM	17
3.1 Lokasi dan Waktu.....	17
3.2 Alat dan Bahan	17
3.3 Langkah-langkah Pengembangan Sistem	18
BAB 4 PERANCANGAN DAN EVALUASI SISTEM	21
4.1 Rancangan Sistem	21
4.1.1 Analisis Kebutuhan	21
4.1.2 Implementasi	21
4.1.3 Pengujian.....	38
4.1.4 Evaluasi	42
4.2 Evaluasi Sistem	43
4.2.1 Sebelum Implementasi HPA	43
4.2.2 Setelah Implementasi HPA.....	45
BAB 5 PENUTUP.....	48
5.1 Kesimpulan	48
5.2 Saran.....	48
DAFTAR PUSTAKA	50
LAMPIRAN.....	53

DAFTAR TABEL

Tabel 2. 1 Ringkasan Tinjauan Pustaka.....	6
Tabel 3. 1 Perangkat Lunak (<i>Software</i>)	18
Tabel 4. 1 Skenario Pengujian.....	40
Tabel 4. 2 Perbandingan Performa Sebelum dan Sesudah Implementasi HPA.....	47



DAFTAR GAMBAR

Gambar 2. 1 Mekanisme <i>Horizontal Pod Autoscaler</i> pada Kubernetes [17].....	15
Gambar 4. 1 Generate SSH Key	22
Gambar 4. 2 Konfigurasi Cloud-Init	23
Gambar 4. 3 Membuat VM Template	24
Gambar 4. 4 Diagram Arsitektur Sistem	25
Gambar 4. 5 Daftar Node	28
Gambar 4. 6 Daftar Pod	29
Gambar 4. 7 Verifikasi Hasil <i>Deploy</i>	37
Gambar 4. 8 Tampilan Todo List.....	38
Gambar 4. 9 Diagram Arsitektur Pengujian	38
Gambar 4. 10 <i>Flowchart Load Testing</i>	39
Gambar 4. 11 Hasil <i>Load Testing</i>	42
Gambar 4. 12 Grafik <i>success rate</i> tanpa HPA.....	44
Gambar 4. 13 Grafik <i>response time</i> tanpa HPA	44
Gambar 4. 14 Grafik <i>throughput</i> tanpa HPA.....	45
Gambar 4. 15 Grafik <i>success rate</i> HPA	46
Gambar 4. 16 Grafik <i>response time</i> HPA.....	46
Gambar 4. 17 Grafik <i>throughput</i> HPA	47

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR KODE SUMBER

Kode Sumber 4. 1 Inisialisasi Database.....	29
Kode Sumber 4. 2 Secret.....	30
Kode Sumber 4. 3 ConfigMap	30
Kode Sumber 4. 4 Persistent Volume	31
Kode Sumber 4. 5 Persistent Volume Claim	31
Kode Sumber 4. 6 Deployment Database	32
Kode Sumber 4. 7 Deployment Aplikasi.....	33
Kode Sumber 4. 8 Autoscaling.....	35
Kode Sumber 4. 9 Perintah untuk Deployment	37
Kode Sumber 4. 10 Pengujian Locust.....	40
Kode Sumber 4. 11 <i>Run Locust File</i>	41

DAFTAR LAMPIRAN

Lampiran A File Konfigurasi Provider.....	53
Lampiran B File Manajemen Variabel	53
Lampiran C File Konfigurasi Main.tf	54
Lampiran D Konfigurasi Ansible Playbook.....	58



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Aplikasi web berbasis PHP tetap menjadi pilihan utama dalam pengembangan sistem informasi, layanan publik, dan situs web dinamis [1]. Namun, aplikasi ini sering menghadapi tantangan performa saat terjadi lonjakan trafik yang tidak terduga, seperti pada periode pendaftaran massal atau promosi besar-besaran [2]. Lonjakan trafik semacam ini dapat menyebabkan peningkatan waktu respons, penurunan kualitas layanan, dan bahkan kegagalan sistem secara keseluruhan.

Pada arsitektur server tradisional, kapasitas sumber daya biasanya bersifat statis dan tidak mudah beradaptasi dengan perubahan beban kerja. Ketika trafik rendah, sumber daya yang ada mungkin tidak dimanfaatkan secara optimal, sementara saat trafik tinggi dan kapasitas server tidak mencukupi, maka menyebabkan *bottleneck* yang mempengaruhi performa aplikasi. Situasi ini menekankan kebutuhan akan infrastruktur yang mampu menyesuaikan kapasitas layanan secara dinamis sesuai dengan fluktuasi beban kerja.

Kubernetes, sebagai platform orkestrasi container yang populer, menawarkan fitur *Horizontal Pod Autoscaler* (HPA) yang memungkinkan penyesuaian jumlah pod secara otomatis berdasarkan metrik tertentu, seperti penggunaan CPU atau memori [3]. Dengan HPA, aplikasi dapat menyesuaikan jumlah pod secara *real-time* sesuai kebutuhan beban kerja, menjaga performa tetap optimal, dan menghindari penggunaan sumber daya yang tidak perlu.

Meskipun banyak penelitian telah mengeksplorasi implementasi HPA pada Kubernetes di lingkungan *cloud* publik, studi mengenai penerapan HPA untuk aplikasi PHP pada infrastruktur lokal (*on-premise*) masih terbatas. Padahal, banyak organisasi yang masih mengandalkan server lokal untuk menjalankan aplikasi mereka karena pertimbangan keamanan, biaya, atau regulasi [4]. Oleh karena itu, diperlukan penelitian yang mengevaluasi efektivitas HPA dalam meningkatkan skalabilitas aplikasi web berbasis PHP pada lingkungan *on-premise*.

1.2 Rumusan Masalah

Aplikasi web berbasis PHP yang dioperasikan pada server lokal masih rentan terhadap penurunan performa saat mengalami lonjakan trafik. Meskipun Kubernetes dengan fitur *Horizontal Pod Autoscaler* (HPA) telah banyak dimanfaatkan di lingkungan *cloud* publik untuk mengatasi masalah skalabilitas, penerapannya dalam konteks *on-premise* masih jarang diteliti. Oleh karena itu, permasalahan yang akan diselesaikan pada penelitian ini adalah: Bagaimana efektivitas implementasi *Horizontal Pod Autoscaler* (HPA) dalam meningkatkan skalabilitas aplikasi web berbasis PHP di lingkungan *on-premise*?

1.3 Batasan Masalah

Dalam penelitian ini, batasan masalah ditetapkan dengan mempertimbangkan kemampuan dari peneliti seperti waktu, biaya, tenaga, dan keahlian yang tersedia, serta berfungsi untuk menggambarkan ruang lingkup penelitian secara jelas. Oleh karena itu, peneliti menetapkan batasan-batasan demi kelancaran serta keberhasilan penelitian ini sebagai berikut:

1. Penelitian dilakukan pada infrastruktur server lokal (*on-premise*) yang berada di *mini data center* Laboratorium Informatika, UIN Sunan Kalijaga Yogyakarta.
2. Objek penelitian terbatas pada aplikasi web berbasis PHP dengan arsitektur sederhana, tanpa melibatkan aplikasi multikomponen kompleks atau *microservices* berskala besar.
3. Penelitian hanya membahas implementasi dan pengujian *Horizontal Pod Autoscaler* (HPA), tanpa mencakup mekanisme *autoscaling* lain seperti *Vertical Pod Autoscaler* (VPA), *Cluster Autoscaler*, ataupun *custom scaling* berbasis *event-driven*.
4. Evaluasi kinerja dilakukan dengan simulasi beban trafik menggunakan *tools* pengujian *load test*, dengan skenario trafik terkontrol, bukan trafik nyata dari pengguna aktif secara langsung.
5. Analisis performa difokuskan pada tiga metrik utama terdiri dari *response time*, *throughput*, dan *request success rate*, tanpa melakukan optimasi atau audit kode sumber aplikasi PHP itu sendiri.

6. Pengujian dilakukan sepenuhnya di server lokal tanpa model pembiayaan berbasis penggunaan seperti pada layanan cloud komersial.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengevaluasi efektivitas implementasi *Horizontal Pod Autoscaler* (HPA) dalam meningkatkan skalabilitas dan performa aplikasi web berbasis PHP yang dijalankan pada infrastruktur lokal (*on-premise*). Melalui studi ini, diharapkan dapat diperoleh pemahaman mengenai bagaimana HPA dapat membantu aplikasi beradaptasi terhadap fluktuasi beban kerja.

1.5 Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan kontribusi nyata dalam beberapa aspek berikut:

1. Teoritis

Memberikan wawasan mengenai penerapan *Horizontal Pod Autoscaler* (HPA) pada Kubernetes untuk meningkatkan skalabilitas dan kestabilan aplikasi web berbasis PHP, serta menjadi referensi dalam pengelolaan beban aplikasi berbasis container.

2. Praktis

Menjadi panduan bagi pengembang dan pengelola infrastruktur dalam menerapkan *autoscaling* pada server lokal, agar aplikasi tetap responsif saat terjadi lonjakan trafik dan sumber daya dapat dioptimalkan dengan lebih efisien.

BAB 5

PENUTUP

5.1 Kesimpulan

Setelah mengimplementasikan *Kubernetes Cluster* menggunakan Terraform, sistem secara keseluruhan memenuhi spesifikasi teknis yang direncanakan. Dengan master node yang memiliki 4 core CPU dan 8GB RAM, serta worker node dengan 2 core CPU dan 8GB RAM, aplikasi PHP dan database MySQL dapat berjalan dengan baik. Penggunaan Ansible playbook juga sangat membantu dalam mengotomatisasi proses konfigurasi Kubernetes, sehingga mengurangi potensi kesalahan manual dan memastikan pengelolaan yang lebih efisien.

Selanjutnya, hasil pengujian menunjukkan bahwa penerapan *Horizontal Pod Autoscaler* (HPA) membawa perubahan yang signifikan terhadap performa aplikasi. Tanpa HPA, sistem mengalami penurunan drastis dalam performa, dengan *success rate* yang turun hingga 68%, *response time* yang melonjak lebih dari 13.000 ms, dan *throughput* yang stagnan. Namun, setelah HPA diterapkan, sistem dapat menyesuaikan jumlah pod sesuai dengan beban yang ada, menjaga *success rate* tetap 100%, menurunkan *response time* menjadi di bawah 4.100 ms, dan meningkatkan *throughput* lebih dari dua kali lipat. HPA terbukti sangat efektif dalam menjaga kestabilan dan skalabilitas aplikasi web berbasis PHP, bahkan di bawah beban tinggi.

5.2 Saran

Dalam rangka pengembangan sistem dan penelitian selanjutnya, berikut rekomendasi yang dapat dipertimbangkan:

1. Optimalisasi konfigurasi *Horizontal Pod Autoscaler* (HPA) dengan menggunakan kombinasi metrik seperti CPU, *memory*, dan *network I/O* guna meningkatkan responsivitas sistem terhadap segala jenis beban.
2. Mengintegrasikan *Vertical Pod Autoscaler* (VPA) sehingga dapat membantu meningkatkan efisiensi penggunaan sumber daya pada tiap pod tanpa harus selalu menambah jumlah pod baru.

3. Untuk mengetahui batas skalabilitas sistem, disarankan melakukan pengujian dengan jumlah *virtual user* lebih dari 2.000 dan konfigurasi jumlah pod maksimal yang lebih tinggi.



DAFTAR PUSTAKA

- [1] Koding Akademi, “Bahasa Pemrograman PHP: Apakah Masih Menarik di 2023 Ini?,” Koding Akademi. Accessed: Mar. 07, 2025. [Online]. Available: <https://www.kodingakademi.id/bahasa-pemrograman-php-apakah-masih-menarik-di-2023-ini/>
- [2] Yusuf Muharam and Taufik Hidayat, “Pengembangan Aplikasi Back-End E-Commerce Menggunakan Rest Api Golang Untuk Optimalisasi Kinerja Server,” *Comput. J. Inform.*, vol. 11, no. 01, pp. 7–13, Jun. 2024, doi: 10.55222/computing.v11i01.1479.
- [3] T.-T. Nguyen, Y.-J. Yeom, T. Kim, D.-H. Park, and S. Kim, “Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration,” *Sensors*, vol. 20, no. 16, p. 4621, Aug. 2020, doi: 10.3390/s20164621.
- [4] Rita Puspita Sari, “Cloud vs On-Premise: Mana yang Lebih Cocok untuk Bisnis Anda?,” Cloud vs On-Premise: Mana yang Lebih Cocok untuk Bisnis Anda? Accessed: Mar. 07, 2025. [Online]. Available: <https://www.cloudcomputing.id/pengetahuan-dasar/cloud-vs-on-premise>
- [5] D. T. Gurohman *et al.*, “Penerapan Horizontal Pod Autoscaler dan Redis Cluster Berbasis Kubernetes untuk Meningkatkan Performa Website E-Learning,” *SKANIKA Sist. Komput. Dan Tek. Inform.*, vol. 7, no. 2, pp. 224–235, Jul. 2024, doi: 10.36080/skanika.v7i2.3211.
- [6] A. B. B. Penggalih, A. Widjajarto, and A. Budiyo, “Implementasi dan Profiling Layanan CMS Wordpress Menggunakan Fitur Horizontal Pod Autoscaler pada Google Kubernetes Engine Dengan Metrik Cpu, Pod, dan Transaction,” *EProceedings Eng.*, vol. 11, no. 4, 2024.
- [7] A. B. Pratama, “Analisis Performansi Web Server pada Kubernetes Cluster untuk Mendukung Scalability dengan Metode Horizontal Pod Autoscaler,” Politeknik Negeri Jakarta, 2023. Accessed: Nov. 03, 2025. [Online]. Available: <https://repository.pnj.ac.id/id/eprint/13164/1/Agus%20Budi%20Pratama%20Bab%201%20-%20Bab%205.pdf>

- [8] I. Rosyadi, "Implementasi Autoscaling Container Web Server Menggunakan Kubernetes berbasis Proxmox pada Server Universitas Darussalam Gontor," Universitas Darussalam Gontor, 2019.
- [9] R. A. Chandra, M. Murhaban, S. Suryadi, and M. Mukhlizar, "Analisis dan Perbandingan Kinerja Proxmox Virtual Envorment dalam Virtualisasi pada OS Debian dan Ubuntu," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3687–3692, Jun. 2024, doi: 10.36040/jati.v8i3.9795.
- [10] Proxmox Team, "Proxmox VE." Accessed: Mar. 12, 2025. [Online]. Available: https://pve.proxmox.com/wiki/Main_Page
- [11] Avinash Pathak, "Automating Infrastructure Management: Benefits and Challenges of Ansible and Terraform Implementation Across Sectors," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 10, no. 5, pp. 381–394, Oct. 2024, doi: 10.32628/CSEIT241051032.
- [12] Blake Douglas Hatcher, "Automating Server Deployments with Ansible: Utilizing Automation in DevOps," *J. Comput. Sci. Coll.*, vol. 40, no. 3, pp. 42–43, Oktober 2024.
- [13] Dirk Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment | Linux Journal," Linux Journal. Accessed: Mar. 12, 2025. [Online]. Available: <https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment>
- [14] Docker Documentation, "What is Docker?" Accessed: Mar. 12, 2025. [Online]. Available: <https://docs.docker.com/get-started/docker-overview/>
- [15] Kubernetes Documentation, "Kubernetes Concepts," Kubernetes. Accessed: Mar. 12, 2025. [Online]. Available: <https://kubernetes.io/docs/concepts/>
- [16] Kubernetes Documentation, "Horizontal Pod Autoscaling," Kubernetes. Accessed: Mar. 12, 2025. [Online]. Available: <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>
- [17] Kubecost, "The Guide To Kubernetes HPA by Example." Accessed: Mar. 12, 2025. [Online]. Available: <https://www.kubecost.com/kubernetes-autoscaling/kubernetes-hpa/>

- [18] GeeksforGeeks, “Load Testing - Software Testing.” Accessed: Mar. 12, 2025. [Online]. Available: <https://www.geeksforgeeks.org/software-testing-load-testing/>
- [19] Jonatan Heyman, Lars Holmberg, and Andrew Baldwin, “What is Locust? — Locust 2.33.1 documentation.” Accessed: Mar. 12, 2025. [Online]. Available: <https://docs.locust.io/en/stable/what-is-locust.html#>
- [20] Adam Rao, *Aplikasi Todo List*. (Jan. 12, 2024). PHP. Accessed: Apr. 14, 2025. [Online]. Available: <https://github.com/AdamRa0/ToDo-App>

