

IMPLEMENTATION OF CONTAINER ORCHESTRATOR MANAGEMENT IN LEARNING MANAGEMENT SYSTEM

Muhamad Subhi Basit^{*1}, Arif Pratama², Januponsa Dio Firizqi^{*3}, Richardus Eko Indrajit⁴

^{1,2,3,4}Information Technology, Universitas Pradita, Indonesia

Email: ¹muhamad.subhi@student.pradita.ac.id, ²arif.pratama@student.pradita.ac.id,
³januponsa.dio@sudent.pradita.ac.id@email.ac.id, ⁴eko.indrajit@pradita.ac.id

(Article received: August 04, 2023; Revision: August 11, 2023; published: August 21, 2023)

Abstract

The pandemic that started in 2019 made technology even more important. Accelerating digital adaptation and transformation in the education sector, namely the Learning Management System (LMS) is very helpful in the teaching and learning process. Various LMS platforms are widely used and developed by various agencies. Pradita University is also developing an LMS which is made with the node.js programming language and the next.js framework which can be accessed on the pradita.digital site. The choice of using the wrong technology can cause the problematic LMS that is used to become slow and heavy, to the point where it cannot be accessed. The use of technology with the hypervisor concept which was previously commonly used has various weaknesses. For this reason, the Pradita LMS was developed using container orchestrator technology and Rancher Kubernetes as container orchestrator management. By utilizing Rancher as a container orchestrator management, it can make it easier to use with a user friendly GUI. In this study aims to prove that the use of Rancher in improving the quality and performance of LMS Pradita. Among them by identifying system efficiency, providing better scalability, increasing the availability of Pradita LMS, Rancher is able to manage applications and containers used for Pradita LMS, and saving costs used in implementing physical devices. By using the experimental method applied in this study, the results of system efficiency testing and scalability testing carried out with a load of up to 200 users can run properly and efficiently, then the availability test proves that the LMS is still available even when there are workers who are not operating, and testing rancher management capabilities lead to conclusions Rancher runs optimally and uses resources efficiently in accordance with the objectives in this study.

Keywords: containerization, container orchestrator management, learning management system, Rancher.

PENERAPAN MANAJEMEN ORKESTRATOR KONTAINER PADA LEARNING MANAGEMENT SYSTEM

Abstrak

Pandemi yang berlangsung mulai dari tahun 2019 membuat teknologi menjadi semakin penting. Percepatan adaptasi dan transformasi digital pada sektor pendidikan yaitu dengan *Learning Management System* (LMS) sangat membantu dalam proses belajar mengajar. Berbagai *platform* LMS banyak digunakan dan dikembangkan oleh berbagai instansi. Pradita University turut mengembangkan LMS yang dibuat dengan bahasa pemrograman node.js dan *framework* next.js yang dapat diakses pada situs pradita.digital. Pemilihan penggunaan teknologi yang tidak tepat dapat mengakibatkan LMS bermasalah yang digunakan menjadi lambat dan berat, hingga tidak dapat diakses. Penggunaan teknologi dengan konsep *hypervisor* yang sebelumnya umum digunakan memiliki berbagai kelemahan. Untuk itu pada LMS Pradita dikembangkan menggunakan teknologi orkestrator kontainer dan Rancher Kubernetes sebagai manajemen orkestrator kontainer. Dengan pemanfaatan Rancher sebagai manajemen orkestrator kontainer dapat mempermudah pemakaian dengan GUI yang *user friendly*. Dalam penelitian ini bertujuan untuk membuktikan bahwa penggunaan Rancher dalam meningkatkan kualitas dan performa LMS Pradita. Diantaranya dengan mengidentifikasi efisiensi sistem, menyediakan skalabilitas yang lebih baik, meningkatkan ketersediaan LMS Pradita, Rancher mampu mengelola aplikasi dan kontainer yang digunakan untuk LMS Pradita, dan menghemat biaya yang digunakan dalam penerapan perangkat fisik. Dengan menggunakan metode *experimental* yang diterapkan dalam penelitian ini memberikan hasil pengujian efisiensi sistem dan pengujian skalabilitas yang dilakukan dengan beban hingga 200 *users* dapat berjalan dengan baik dan efisien, lalu pengujian ketersediaan membuktikan bahwa LMS masih tetap tersedia bahkan ketika ada worker yang tidak beroperasi, dan pengujian kemampuan manajemen rancher menghasilkan kesimpulan Rancher berjalan dengan optimal dan menggunakan sumber daya secara efisien sesuai dengan tujuan dalam penelitian ini.

Kata kunci: *kontainerisasi, learning management system, manajemen orkestrator kontainer, Rancher.*

1. PENDAHULUAN

Pandemi yang disebabkan oleh virus corona yang muncul pertama pada tahun 2019 menyebabkan banyak perubahan di antaranya dibidang teknologi. Penggunaan teknologi dan pemakaian produk digital diterapkan diberbagai sektor usaha, kesehatan maupun pendidikan. Penggunaan teknologi membawa perubahan yang signifikan dalam cara belajar, mengajar, berinteraksi maupun dalam hal mengakses informasi [1]. Di dunia pendidikan proses belajar mengajar dipaksa untuk lebih cepat beradaptasi dan bertransformasi yang semula belajar mengajar tatap muka secara luring menjadi secara daring [2]. Penggunaan *electronic learning (e-learning)* sebagai sarana belajar mengajar secara digital menggantikan proses belajar mengajar tatap muka dalam kelas luring [3]. Meskipun beberapa universitas diseluruh dunia telah mengadopsi *e-learning* selama beberapa tahun dari sebelum pandemi, kesiapan untuk transformasi penuh menggunakan *e-learning* masih tidak diperkirakan akan berjalan dengan mudah. Transformasi tiba-tiba ke pembelajaran jarak jauh memaksa institusi pendidikan untuk mengadopsi pembelajaran online dan bergantung pada infrastruktur *e-learning* universitas, kompetensi kognitif mahasiswa, dosen dan staf [4].

Salah satu infrastruktur penting dalam penerapan *e-learning* adalah *Learning Management System (LMS)*. *Learning Management System (LMS)* adalah platform perangkat lunak yang digunakan untuk menyediakan, mengelola, dan menyampaikan berbagai jenis materi pelajaran, pelatihan, dan pembelajaran secara daring. Dengan semakin berkembangnya teknologi dan meningkatnya permintaan akan pembelajaran jarak jauh [5]. LMS telah menjadi kunci dalam mendukung pengalaman pembelajaran jarak jauh [6]. Dengan pengalaman yang diperoleh selama masa adopsi pendidikan jarak jauh yang dipaksa, universitas dapat mengoptimalkan penyediaan pembelajaran daring dan menawarkan model pendidikan yang lebih fleksibel dan beradaptasi dengan kebutuhan mahasiswa di masa depan. Transformasi ini juga menegaskan pentingnya teknologi dan platform digital dalam membentuk masa depan pendidikan tinggi yang lebih dinamis dan inklusif [7]. Berdasarkan pengamatan yang telah dilakukan maka dikembangkan produk LMS yang dibuat sendiri untuk diterapkan di Pradita University pada situs *pradita.digital* dengan menggunakan bahasa pemrograman *node.js* dan *framework next.js*.

Pengembangan LMS Pradita terkait infrastruktur, perangkat, server, kecepatan data, dan penyimpanan data merupakan instrument penting yang harus dipastikan dapat berjalan baik. [8]. Terdapat beberapa metode *deployment* aplikasi *web*

ke dalam server, salah satu metode yang memanfaatkan teknologi virtualisasi berbasis *hypervisor* dapat digunakan. Dengan metode *hypervisor* setiap aplikasi dan *dependency* yang dibutuhkan di-*deploy* ke dalam mesin virtual yang berbeda. Dengan metode ini dapat meningkatkan skalabilitas, karena setiap aplikasi berjalan pada sumber daya (CPU, memori, penyimpanan data) yang berbeda sehingga dapat dengan mudah ditambahkan sesuai kebutuhan. Sehingga membutuhkan resource besar. Karena setiap *Virtual Machine* menjalankan *guest OS* beserta kernelnya sendiri terpisah dari *host* [9]. Kebutuhan sumber daya yang besar ini dikarenakan adanya virtualisasi penuh dari mulai sistem operasi hingga proses yang berjalan. Hal ini berbeda dengan kontainer yang melakukan virtualisasi pada tingkat aplikasi. Dengan begitu, penggunaan sumber daya seperti pemakaian CPU, Ram dan perangkat fisik pada setiap kontainer lebih sedikit dibandingkan mesin virtual [10]. Virtualisasi penuh berarti membuat seakan terdapat computer atau sistem operasi lain berjalan di dalam komputer. Sepeti contoh menginstal Linux dalam Windows, demikian pula sebaliknya jika menginstall Windows dalam Linux [11].

Tantangan-tantangan terkait efisiensi dan skalabilitas, menggunakan sumber daya yang lebih efektif sehingga memunculkan opsi yang membuat organisasi atau institusi pendidikan beralih ke kontainerisasi dan orkestrasi untuk menyederhanakan dan meningkatkan penerapan LMS Pradita. Pola arsitektur perangkat lunak inovatif seperti *microservices* telah muncul untuk meningkatkan modularitas aplikasi dan menyederhanakan pengembangan, pengujian, penskalaan, dan penggantian komponen. Teknologi berbasis kontainer, seperti Docker Swarm, Kubernetes, Apache Mesos, dan Cattle, telah memainkan peran krusial dengan memungkinkan penerapan dan penskalaan cepat *microservices* dengan biaya overhead yang rendah [12]. *Microservice* merupakan salah satu contoh arsitektur perangkat lunak yang memecah aplikasi menjadi beberapa bagian kecil dan aplikasi kecil tersebut adalah *service* [13]. Sedangkan jika menggunakan arsitektur monolitik yang merupakan sebuah arsitektur pengembangan aplikasi yang kode program, database dan tampilan program menjadi satu, dan semua layanan dijalankan dalam sebuah server sehingga satu server akan menangani semua layanan dalam aplikasi [14].

Rancher Kubernetes, sebagai platform manajemen kontainer yang kuat dan *user-friendly*, telah menjadi pilihan populer untuk mengelola kluster Kubernetes dan aplikasi kontainer. Rancher merupakan platform manajemen Kubernetes yang

berfungsi untuk mengelola, mengamankan, dan mengimplementasikan container dalam skala besar. Dengan Rancher, pengguna dapat dengan mudah menyusun, meluncurkan, dan mengelola aplikasi yang berjalan dalam wadah dengan antarmuka pengguna yang intuitif dan juga menyediakan antarmuka grafis yang memudahkan pengguna dalam mengelola infrastruktur Kubernetes serta mengelola siklus dalam container [15].

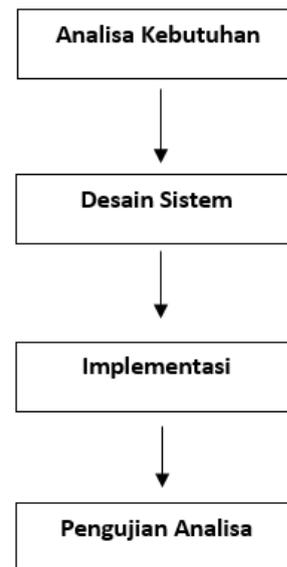
Tujuan dari penelitian implementasi Rancher Kubernetes dalam *Learning Management System* (LMS) adalah untuk mengevaluasi manfaat, keefektifan, dan tantangan penggunaan Rancher Kubernetes dalam meningkatkan kualitas dan performa LMS Pradita. Beberapa tujuan khusus yang dapat diidentifikasi dalam penelitian ini antara lain:

- Meningkatkan Efisiensi Sistem: Menilai sejauh mana implementasi Rancher Kubernetes dapat meningkatkan efisiensi operasional LMS Pradita, termasuk pengelolaan dan penyebaran aplikasi, penjadwalan tugas, dan alokasi sumber daya secara lebih optimal.
- Menyediakan Skalabilitas yang Lebih Baik: Mengidentifikasi apakah Rancher Kubernetes dapat menghadirkan kemampuan skalabilitas yang lebih baik bagi LMS Pradita untuk mengatasi peningkatan jumlah pengguna dan permintaan beban kerja yang fluktuatif.
- Meningkatkan Ketersediaan LMS Pradita: Menilai keandalan dan ketersediaan LMS Pradita setelah mengadopsi Rancher Kubernetes, serta sejauh mana platform tersebut dapat menangani kegagalan dan pemulihan dengan cepat.
- Menilai Kemampuan Manajemen Aplikasi: Mengevaluasi sejauh mana Rancher Kubernetes dapat membantu dalam manajemen aplikasi LMS Pradita, termasuk implementasi, pembaruan, penurunan, dan penskalaan aplikasi dengan mudah.
- Analisis Biaya dan Penghematan: Menganalisis dampak ekonomi dari penggunaan Rancher Kubernetes dalam LMS, termasuk efisiensi biaya, penghematan infrastruktur, dan manfaat jangka panjang.
- Rekomendasi dan Tantangan: Memberikan rekomendasi bagi organisasi dan institusi pendidikan tentang potensi keuntungan dan tantangan yang mungkin dihadapi dalam mengadopsi Rancher Kubernetes dalam LMS.

Melalui penelitian ini, diharapkan dapat mendapatkan wawasan yang mendalam tentang bagaimana Rancher Kubernetes dapat meningkatkan efisiensi, keandalan, dan kualitas infrastruktur dari *Learning Management System*. Hasil penelitian ini dapat menjadi panduan bagi pengambil keputusan dalam organisasi pendidikan untuk memutuskan bahwa Rancher Kubernetes sesuai untuk diimplementasikan dalam infrastruktur pembelajaran mereka.

2. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah metode *experimental design*. Dalam upaya melakukan penerapan manajemen orkestrasi kontainer menggunakan Rancher Kubernetes yang di *deploy* pada lingkungan infrastruktur server berbasis kubernetes klaster dan untuk mengetahui data kinerja dari LMS ketika digunakan. Maka beberapa tahapan dilakukan untuk mencapai tujuan tersebut. Berikut dapat dilihat pada gambar 1 ini merupakan tahapan yang dilakukan dalam penelitian ini:



Gambar 1. Alur penelitian.

2.1. Analisa Kebutuhan

Tahapan awal dari penelitian ini yaitu melakukan analisis terhadap kebutuhan yang diperlukan dalam menunjang proses penelitian. Analisis kebutuhan dibagi menjadi beberapa bagian, yaitu:

2.1.1. Kebutuhan Data

1. Studi Literatur

Penelitian ini mengambil referensi dari jurnal nasional, jurnal internasional, laporan, makalah, serta artikel dari internet yang relevan dengan implementasi model *deployment container* pada dengan Kubernetes dan manajemen orkestrasi kontainer dengan Rancher yang sesuai dengan topik penelitian ini.

2. Observasi

Penelitian ini juga melakukan observasi dengan mengunjungi situs referensi mengenai orkestrasi kubernetes dan manajemen orkestrasi Rancher Kubernetes. Serta observasi kebutuhan-kebutuhan yang mendukung dari *platform* LMS.

2.1.2. Kebutuhan Fungsional Sistem

Kebutuhan fungsional dari penerapan manajemen orkestrasi kontainer menggunakan Rancher Kubernetes yaitu:

1. Rancher dapat mempermudah dalam manajemen orkestrasi dengan GUI yang disediakan
2. Rancher dapat mempermudah dalam memonitor pemakaian CPU, RAM dan Pods
3. Mampu meenjalankan dengan baik LMS Pradita yang dibuat
4. LMS dapat berjalan walaupun salah satu host worker tidak berjalan (*down*)
5. Dapat meningkatkan efisiensi operasional LMS Pradita.

2.1.3. Kebutuhan Perangkat

Spesifikasi perangkat yang digunakan dalam penelitian ini, yaitu:

- 1.Perangkat Virtual Server
- 1 frontend rancher
- 4 virtual machine
- 1 master, 3 worker

spesifikasi :

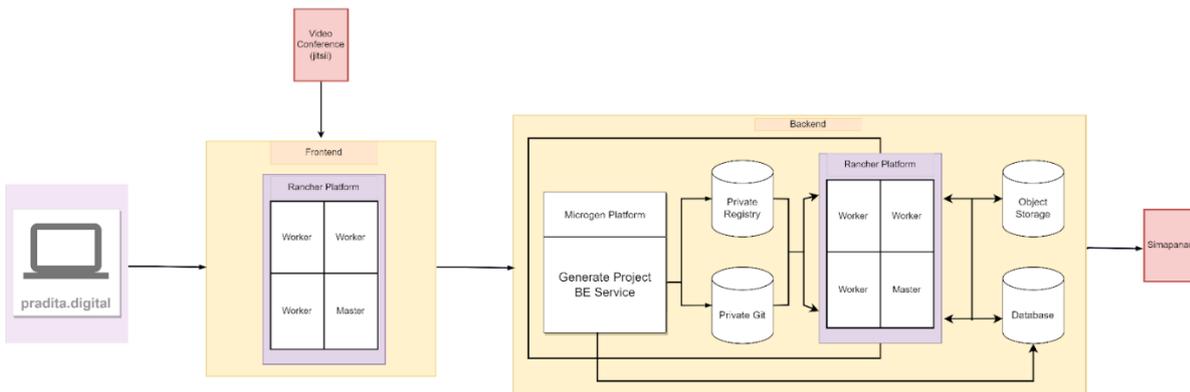
- vm master : cpu 4 core, memory 8 gb, storage 100 gb
- vm worker : cpu 8 core, memory 16 gb, storage 250 gb.

2.2. Desain Sistem

Sistem yang dirancang dalam penelitian ini menggunakan empat node yang berada dalam cluster kubernetes. Kontainer yang berada dalam beberapa node terhubung dalam entitas klaster. Berikut merupakan desain sistem dalam penelitian ini:

2.2.1. Desain Topologi

Pada gambar 2 dapat dilihat terdapat 2 klaster yaitu 1 klaster front end dan 1 klaster back end. Setiap klaster memiliki 4 node dimana salah 1 node bertugas sebagai master dan 3 node lainnya sebagai worker. Node master akan bertugas sebagai pembagi tugas terhadap aplikasi yang akan di *deploy*. Node master juga akan membagi tugas kepada worker yang mempunyai *load* yang rendah.



Gambar 2. Alur Infrastruktur LMS Pradita

2.2.2. Instalasi Infrastruktur

Setelah desain system dirancang. Melanjutkan untuk proses instalasi dalam perangkat. Beberapa instalasi yang dilakukan antara lain:

1. Instalasi 4 Virtual Machine dengan Operating System Ubuntu 20.04

Gambar 3 menunjukkan proses instalasi virtual machine. Gambar 4 menunjukkan proses instalasi docker, sedangkan Gambar 5 menunjukkan proses instalasi manajemen orkestrator Rancher Kubernetes

```

login as ubuntu
ubuntu@10.11.60.33's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-153-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 1 15:56:01 UTC 2023

System load: 0.72           Users logged in: 0
Usage of /:  21.0% of 242.11GB   IPv4 address for docker0: 172.17.0.1
Memory usage: 33%           IPv4 address for ens3: 192.168.1.100.27
Swap usage:  0%             IPv4 address for weave: 10.42.192.0
Processes:   508

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure R8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.
https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

63 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Sat Feb 25 19:15:24 2023 from 10.11.60.254
ubuntu@frontend-worker1-idx-openstack:~$
    
```

Gambar 3. Proses instalasi virtual machine

2. Instalasi docker

```

root@ids:~# apt install docker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  wmdocker
The following NEW packages will be installed:
  docker wmdocker
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 14.3 kB of archives.
After this operation, 58.4 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 wmdocker amd64 1.5-2 [13.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/universe amd64 docker all 1.5-2 [1316 B]
Fetched 14.3 kB in 1s (18.4 kB/s)
Selecting previously unselected package wmdocker.
(Reading database ... 101323 files and directories currently installed.)
Preparing to unpack .../wmdocker_1.5-2_amd64.deb ...
Unpacking wmdocker (1.5-2) ...
Selecting previously unselected package docker.
Preparing to unpack .../archives/docker_1.5-2_all.deb ...
Unpacking docker (1.5-2) ...
Setting up wmdocker (1.5-2) ...
Setting up docker (1.5-2) ...
Processing triggers for man-db (2.9.1-1) ...
root@ids:~#

```

Gambar 4. Proses instalasi docker

3. Instalasi Rancher

```

root@ids:~# sudo docker run -d \
> --v $HOST_VOLUME:/var/lib/mysql \
> --restart=unless-stopped \
> -p 8080:8080 \
> rancher/server
Unable to find image 'rancher/server:latest' locally
latest: Pulling from rancher/server
bae38266908: Pull complete
29ede3c02ff2: Pull complete
da4e69f33106: Pull complete
8d49e3f5d27f: Pull complete
b09e1abb17d6: Pull complete
42247db4917: Pull complete
78d37de643ce: Pull complete
69d13e08a4fe: Pull complete
24dfdc36a2b7: Pull complete
bc433fed3823: Pull complete
b92e18df556: Pull complete
d8e3024894f: Pull complete
07bf18e8ec0: Pull complete
339e2408f91: Pull complete
9372455de0b8: Pull complete
5a33b348bf45: Pull complete
328697d8874: Pull complete
b379bfb954de: Pull complete
ba7e1999a31: Pull complete
0c19aca4f8a1: Pull complete
e03fc76c9997: Pull complete
Digest: sha256:95b55603122c28baea4e8d94663aa34ad770bbc624a9ed6ef986fb3ea5224d91
Status: Downloaded newer image for rancher/server:latest
ec2a11ce9569d535c3cc9f232f2fcdc6f4a6d1bbd71ddb814fd166a38ac5c5
root@ids:~#

```

Gambar 5. Proses instalasi Rancher.

2.3. Implementasi

Setelah menganalisa kebutuhan, perancangan sistem, dan instalasi infrastruktur tahap selanjutnya adalah yaitu mengkonfigurasi dan menerapkan rancangan sistem. Sudah dibuat 2 kluster yaitu kluster front end dan back end. Dimana di dalam setiap kluster terdapat 4 worker yang sudah berjalan untuk memenuhi kebutuhan LMS Pradita dari sisi infrastruktur.

2.4. Pengujian dan Analisa

Pada tahap terakhir yaitu tahapan pengujian dan analisa dilakukan pengujian untuk menentukan efisiensi sistem, skalabilitas, ketersediaan, pengujian kemampuan manajemen Rancher terhadap aplikasi LMS Pradita. Berikut merupakan penjelasan pada tiap pengujian, yaitu:

A. Pengujian Efisiensi sistem

Pengujian efisiensi sistem mengacu pada kemampuan sistem untuk melakukan tugas dan fungsi dengan menggunakan sumber daya pada CPU seminimal mungkin. Pengujian efisiensi sistem ini dilakukan untuk mengetahui dan menilai efisiensi sistem dengan memberikan beban dari *request user* dengan jumlah 50 *user*, 100 *user*, dan 200 *user* pada saat melakukan aktivitas login, akses mata kuliah, unggah file, dan melakukan *video conference*.

B. Pengujian Skalabilitas

Pengujian skalabilitas sistem mengacu pada kemampuan sistem untuk beradaptasi dan mengatasi pertumbuhan beban atau ukuran dengan tetap mempertahankan kinerja dan respons yang baik. Pengujian skalabilitas sistem ini dilakukan untuk mengetahui dan menilai skalabilitas ketika diberikan beban dari request user dengan jumlah 50 user, 100 user, dan 200 user pada saat melakukan aktivitas login, akses mata kuliah, unggah file, dan melakukan *video conference*.

C. Pengujian Ketersediaan

Pengujian ketersediaan system mengacu pada kemampuan sistem dan memverifikasi sejauh mana aplikasi LMS Pradita dapat tetap beroperasi dengan kinerja yang baik dan responsif selama periode waktu tertentu bahkan jika ada 1 vm atau worker yang mati dan tidak beroperasi.

D. Pengujian Kemampuan Manajemen Rancher

Pengujian kemampuan manajemen Rancher untuk memverifikasi kemampuan dari platform manajemen kontainer Rancher untuk memonitor dan menambah atau mengurangi worker sesuai dengan kebutuhan pada LMS Pradita.

Setelah pengujian dilakukan, akan diperoleh data hasil pengujian yang kemudian dilakukan Analisa menggunakan parameter yang ditentukan untuk mengetahui kualitas hasil pengujian.

3. HASIL DAN PEMBAHASAN

Setelah ditentukan kebutuhan sistem, instalasi, konfigurasi dan penentuan pengujian. Untuk mendapat hasil penelitian sesuai kebutuhan maka dilakukan tahapan pengujian baik dari sisi *host* maupun *client*. Berikut pembahasan dari pengujian penerapan manajemen okestrasi kontainer yang menggunakan Rancher pada LMS Pradita di situs pradita.digital.

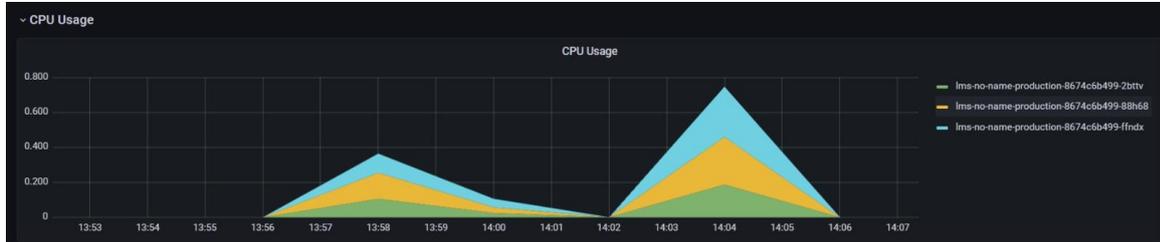
3.1. Pengujian Efisiensi Sistem

Pengujian efisiensi LMS Pradita yang menggunakan Rancher ini merupakan proses mengukur dan memastikan bahwa aplikasi yang dijalankan di dalam kluster front end dan back end melalui Rancher berjalan dengan optimal dan menggunakan sumber daya secara efisien. Tujuan dari pengujian ini adalah untuk mengidentifikasi dan mengatasi masalah kinerja, optimalkan penggunaan sumber daya, serta memastikan bahwa aplikasi berfungsi dengan baik di lingkungan yang dikelola oleh Rancher. Sehingga dapat menilai sejauh mana implementasi Rancher Kubernetes dapat meningkatkan efisiensi operasional LMS Pradita, termasuk pengelolaan dan penyebaran aplikasi, penjadwalan tugas, dan alokasi sumber daya secara lebih optimal.

Pengujian efisiensi sistem ni dilakukan dengan memberikan beban dari *request user* dengan jumlah 50 user, 100 user, dan 200 user secara

bertahap. *Request user* yang diberikan pada saat melakukan aktivitas login, akses mata kuliah, unggah file, dan melakukan video conference yang berupa grafik menggunakan Grafana. Grafana merupakan salah satu penyedia layanan monitoring yang dapat

tersinkronisasi dengan berbagai *resource* dan menampilkannya dengan bentuk grafik [16]. Berikut grafik yang ditampilkan merupakan grafik pemakaian CPU dan Bandwidth sebagai hasil penilaian efisiensi sistem dari infrastruktur yang digunakan.



Gambar 6. Grafik CPU yang digunakan saat diuji dengan beban 50 dan 100 user

Pada gambar 6 merupakan hasil pengujian dengan beban 50 user CPU yang terpakai hanya 0.37 % sedangkan untuk pengujian dengan beban 100 user

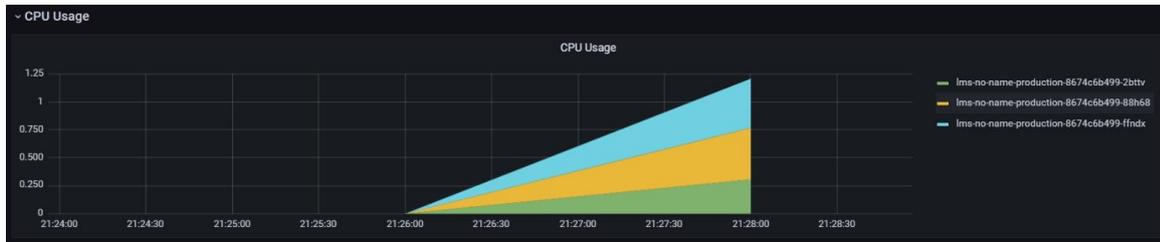
CPU yang digunakan tentu bertambah namun tidak signifikan diangka 0.77 %.



Gambar 7. Grafik Bandwidth yang digunakan saat diuji dengan beban 50 dan 100 user

Pada gambar 7 merupakan hasil pengujian dengan beban 50 user Receive Bandwidth yang terpakai hanya 25 kb/s dan Transmit Bandwidth 80

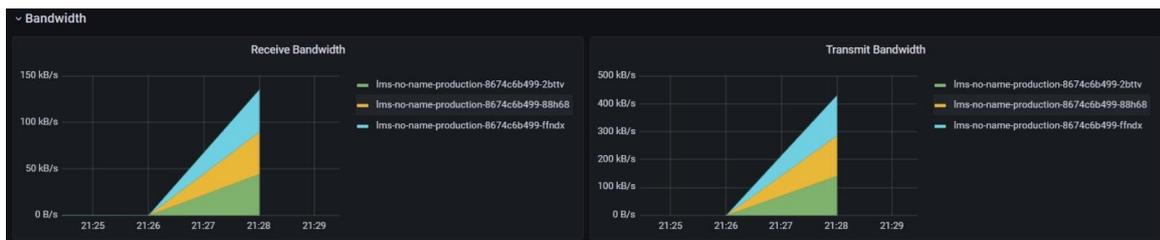
kb/s sedangkan untuk pengujian dengan beban 100 user Receive Bandwidth yang terpakai hanya 67 kb/s dan Transmit Bandwidth 220 kb/s



Gambar 8. Grafik CPU yang digunakan saat diuji dengan beban 200 user

Pada gambar 8 merupakan hasil pengujian dengan beban hingga 200 user pemakaian kapasitas CPU yang terpakai di 1.20 %

Pada gambar 9 merupakan hasil pengujian dengan beban hingga 200 user Receive Bandwidth yang terpakai 132 kb/s dan Transmit Bandwidth 424 kb/s.



Gambar 9. Grafik Bandwidth yang digunakan saat diuji dengan beban 200 user

Dari hasil pengujian sistem yang telah diberikan beban kerja hingga 200 user dapat dikatakan bahwa LMS Pradita yang berjalan dalam kluster front end

dan back end melalui Rancher berjalan dengan optimal dan menggunakan sumber daya secara efisien..

3.2. Pengujian Skalabilitas

Pengujian skalabilitas LMS Pradita yang menggunakan Rancher ini merupakan proses mengidentifikasi apakah Rancher Kubernetes dapat menghadirkan kemampuan skalabilitas yang lebih baik bagi LMS Pradita untuk mengatasi peningkatan jumlah pengguna dan permintaan beban kerja yang fluktuatif. Pengujian melibatkan kemampuan Rancher dalam mengelola dan mengatasi pertumbuhan beban serta kompleksitas sistem yang terkait dengan LMS Pradita. LMS Pradita yang digunakan untuk mengelola proses pembelajaran dan

interaksi pengguna, oleh karena itu, pengujian skalabilitas sangat penting untuk memastikan ketersediaan dan kinerja sistem yang andal.

Pengujian skalabilitas sistem ini dilakukan untuk mengetahui dan menilai skalabilitas ketika diberikan beban dari request user dengan jumlah 50 user, 100 user, dan 200 user pada saat melakukan aktivitas login, akses mata kuliah, unggah file, dan melakukan *video conference* menggunakan tools Vegeta. Vegeta perangkat lunak *open-source* yang digunakan untuk melakukan pengujian kinerja *load testing* [17].

```

root@ids:~# echo "GET https://pradita.digital" | vegeata attack -duration=120s | tee results.bin | vegeata report
Requests      [total, rate, throughput]    6000, 50.01, 50.00
Duration      [total, attack, wait]        2m0s, 2m0s, 16.643ms
Latencies     [min, mean, 50, 90, 95, 99, max] 5.646ms, 16.03ms, 14.469ms, 21.436ms, 27.313ms, 55.042ms, 104.759ms
Bytes In      [total, mean]                36480000, 6080.00
Bytes Out     [total, mean]                0, 0.00
Success       [ratio]                      100.00%
Status Codes  [code:count]                 200:6000
Error Set:
root@ids:~# echo "GET https://pradita.digital" | vegeata attack -duration=2m -rate=100 | tee results.bin | vegeata report
Requests      [total, rate, throughput]    12000, 100.01, 100.00
Duration      [total, attack, wait]        2m0s, 2m0s, 10.241ms
Latencies     [min, mean, 50, 90, 95, 99, max] 4.869ms, 14.739ms, 12.67ms, 21.92ms, 28.64ms, 54.453ms, 198.744ms
Bytes In      [total, mean]                72960000, 6080.00
Bytes Out     [total, mean]                0, 0.00
Success       [ratio]                      100.00%
Status Codes  [code:count]                 200:12000
Error Set:
root@ids:~# █

```

Gambar 10. Hasil pengujian yang digunakan saat diuji dengan beban 50 dan 100 user

Dalam pengujian yang ditunjukkan pada gambar 10, pengujian menggunakan vegeta yang dengan beban 100 user berhasil dilakukan dengan *throughput* atau jumlah pengiriman 100.00, *latencies* atau durasi data terkirim dengan minimal 5.644 ms dan maksimal

104.759ms. Sedangkan pengujian dengan beban 100 user berhasil dilakukan dengan *throughput* atau jumlah pengiriman 100.00, *latencies* atau durasi data terkirim dengan minimal 4.869ms dan maksimal 198.744ms.

```

root@ids:~# echo "GET https://pradita.digital" | vegeata attack -duration=2m -rate=200 | tee results.bin | vegeata report
Requests      [total, rate, throughput]    24000, 200.01, 199.98
Duration      [total, attack, wait]        2m0s, 2m0s, 16.959ms
Latencies     [min, mean, 50, 90, 95, 99, max] 5.151ms, 16.162ms, 13.952ms, 24.332ms, 30.528ms, 53.196ms, 180.237ms
Bytes In      [total, mean]                145920000, 6080.00
Bytes Out     [total, mean]                0, 0.00
Success       [ratio]                      100.00%
Status Codes  [code:count]                 200:24000
Error Set:
root@ids:~# █

```

Gambar 11. Hasil pengujian yang digunakan saat diuji dengan beban 200 user

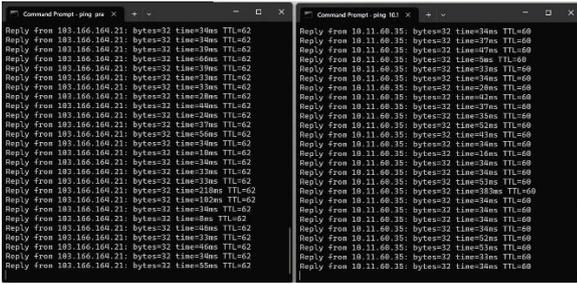
Pada gambar 11 dapat dilihat dalam pengujian menggunakan vegeta yang dengan beban 200 user berhasil dilakukan dengan *throughput* atau jumlah pengiriman 199.98, *latencies* atau durasi data terkirim dengan minimal 5.151ms dan maksimal 180.237ms. Dengan hasil uji pengiriman dengan beban yang bertambah dari 50, 100, dan 200 user semua dapat terkirim dan stabil saat menjalankan LMS Pradita.

3.3. Pengujian Ketersediaan

Pengujian ini menilai keandalan dan ketersediaan LMS Pradita setelah mengadopsi Rancher Kubernetes, serta sejauh mana platform tersebut dapat menangani kegagalan dan pemulihan dengan cepat. Pengujian ini merupakan salah satu bagian penting dalam memastikan platform manajemen kontainer ini dapat berfungsi dengan baik

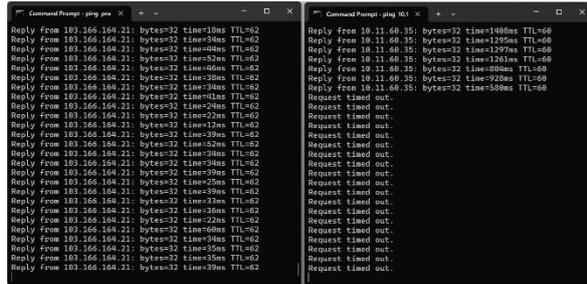
dalam situasi kegagalan atau pemutusan sambungan pada salah satu node worker. Ketersediaan Rancher jika suatu saat ada salah satu worker mati akan menentukan apakah kluster Kubernetes tetap beroperasi secara stabil dan apakah aplikasi yang dijalankan di dalamnya masih dapat diakses oleh pengguna.

Pengujian ketersediaan sistem mengacu pada kemampuan sistem dan memverifikasi sejauh mana aplikasi LMS Pradita dapat tetap beroperasi dengan kinerja yang baik dan responsif selama periode waktu tertentu. Jadi akan diuji dengan shutdown 1 virtual machine atau membuat worker mati atau tidak beroperasi.



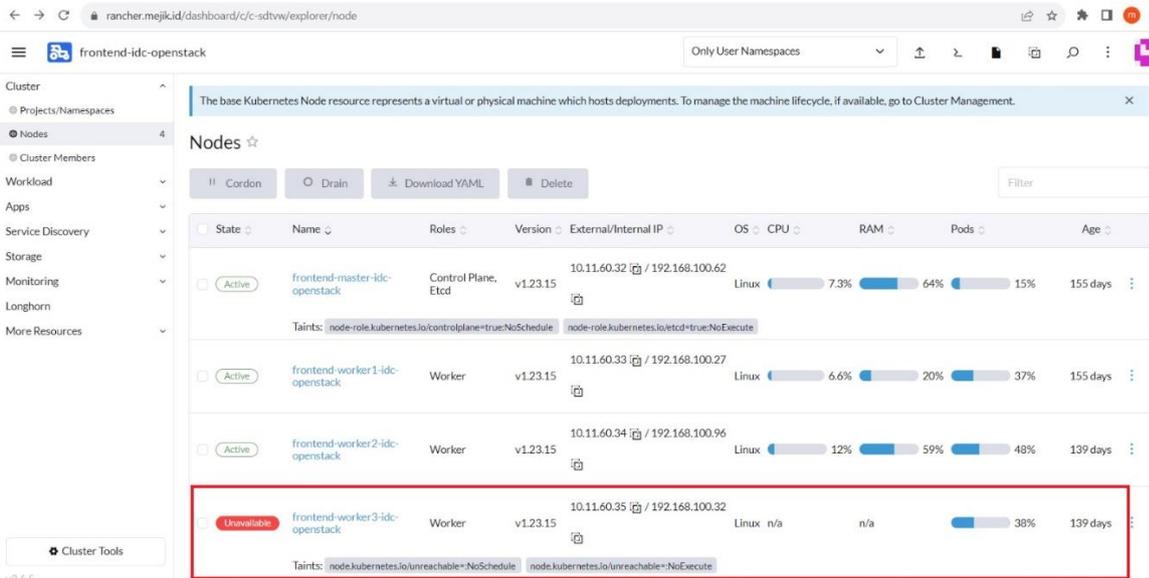
Gambar 12. Prompt ping LMS Pradita dan ping worker aktif

Dapat dilihat pada gambar 12 di bagian sisi kiri merupakan proses ping atau memanggil untuk LMS Pradita, sementara di bagian sisi kanan merupakan proses ping atau memanggil worker atau vm.

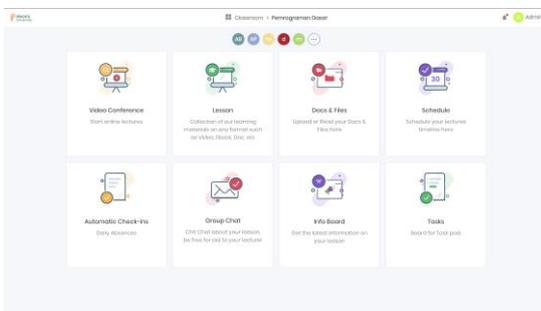


Gambar 13. Prompt ping LMS Pradita aktif dan ping worker mati

Dapat dilihat pada gambar 13 tersebut di bagian sisi kiri merupakan proses ping atau memanggil untuk LMS Pradita aktif, sementara di bagian sisi kanan merupakan proses ping atau memanggil worker atau vm sudah mati dengan status Request timed out. Pada gambar 14 terlihat salah satu worker dengan ip address 10.11.60.35 termonitor dengan status *unavailable* menunjukkan bahwa worker atau vm tersebut berstatus *down*.



Gambar 14. Tampilan salah satu worker down di cluster rancher



Gambar 15. Classroom LMS Pradita

Pada gambar 15 menampilkan halaman kelas dalam LMS Pradita masih dapat diakses. Hal ini dapat membuktikan hasil uji ketersediaan berhasil. Meskipun ada 1 worker atau vm lain yang tidak beroperasi, akan di *back up* oleh worker lainnya

sehingga LMS Pradita tetap dapat diakses oleh pengguna.

3.4. Pengujian Kemampuan Manajemen Rancher

Pengujian kemampuan manajemen Rancher untuk memverifikasi kemampuan dari platform manajemen kontainer Rancher untuk memenuhi kebutuhan dan mendukung LMS Pradita, diantaranya sebagai berikut:

- Rancher sebagai manajemen orkestrator dapat diimplementasikan dan menjalankan LMS Pradita dengan efektif dan efisien. Hal ini dapat dibuktikan dengan membuat klustering dan worker node sehingga dapat menjalankan LMS Pradita.
- Dapat memonitor penggunaan infrastruktur yang digunakan. Dengan Rancher admin atau pengelola dapat memonitor penggunaan

- infrastruktur seperti CPU, RAM, VM, Pods, Kluster dan Node yang aktif.
- Dapat mendukung pembaruan atau pengurangan sesuai kebutuhan LMS Pradita. Dengan Rancher jika terdapat fitur atau kebutuhan baru pada LMS Pradita yang perlu di dukung dapat menambahkan worker baru agar LMS Pradita dapat berjalan baik. Hal ini karena pada orkestrator mempunyai kemampuan *horizontal scale* yaitu kemampuan dari sebuah orkestrator atau platform manajemen kontainer untuk menambahkan lebih banyak *instance* atau layanan dengan menambahkan lebih banyak node ke dalam kluster [18]. Sehingga Rancher dapat menyediakan lingkungan yang mendukung skalabilitas dan stabilitas dalam penggunaan LMS Pradita. Ilustrasi penambahan worker dapat dilihat pada gambar 16
 - Dapat menghemat biaya dalam penerapan infrastruktur untuk LMS Pradita. Dengan menggunakan rancher sebagai implemtasi infrasturktur akan memudahkan dari sisi *horizontal scale*, ketika terdapat penggunaan yang tinggi maka dapat menambahkan worker, juga jika penggunaan tidak terlalu besar dapat dikurangi worker tersebut. Sehingga dapat mengefisiensikan dari segi biaya. Hal ini berbanding terbalik jika menggunakan virtual machine dan tidak menggunakan orkestrator yang akan sulit untuk mengkontrol dan menambah biaya pada segi infrastruktur.



Gambar 16. Penambahan worker Rancher pada klaster front end.

4. DISKUSI

Hasil dari penelitian ini membuktikan bahwa Rancher yang digunakan sebagai manajemen orkestrator dapat berjalan dan mendukung kebutuhan infrastruktur LMS Pradita. Rancher terbukti efektif dan efisien, dapat menyediakan skalabilitas yang baik, meningkatkan ketersediaan, mampu memanagerial aplikasi juga lebih hemat biaya dalam membangun infrastruktur.

Jika menganalisa dari penelitian terdahulu dari Tegar Sukma Hendrana, dan I Made Suartana dengan judul PENERAPAN CONTAINER LOAD BALANCING UNTUK MANAJEMEN TRAFIK PADA LEARNING MANAGEMENT SYSTEM yang di publikasi pada tahun 2022 dapat disimpulkan jika dalam penelitian ini berfokus pada penerapan container load balancing pada LMS Moodle menggunakan Docker Swarm. Dengan tujuan menganalisis bagaimana server Moodle di lingkungan Docker Swarm cluster dapat mengatasi beban traffic dengan metode container load balancing. Dan menghasilkan bahwa penggunaan Docker Swarm dengan metode container load balancing berhasil mengatasi beban traffic pada aktivitas login, view course, assignments, dan quiz dalam LMS Moodle. Lalu dari penelitian terdahulu dari Saleh Dwiyatno, Edy Rakhmat, Oki Gustiawan dengan judul IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER yang di publikasi pada tahun 2020 dapat disimpulkan dalam penelitian ini berfokus pada implementasi virtualisasi berbasis Docker pada server yang sebelumnya menggunakan virtualisasi berbasis hypervisor. Dengan tujuan eksplorasi penggunaan Docker sebagai solusi pemaketan dan peluncuran aplikasi dalam container yang terisolasi. Dan menghasilkan bahwa Docker dapat digunakan untuk virtualisasi server dengan efektif, menggantikan virtualisasi berbasis hypervisor. Docker container mampu memaketkan dan meluncurkan aplikasi dengan isolasi yang diperlukan.

Lalu pada penelitian PENERAPAN MANAJEMEN ORKESTRATOR KONTAINER PADA LEARNING MANAGEMENT SYSTEM yang peneliti tulis ini berfokus pada penggunaan manajemen orkestrator Rancher Kubernetes pada LMS Pradita. Dengan tujuan mengevaluasi efisiensi, skalabilitas, ketersediaan, dan kemampuan manajemen Rancher dalam konteks LMS. Dengan hasil dalam penelitian ini menunjukkan bahwa Rancher Kubernetes dapat mengelola LMS Pradita dengan baik dan efisien. Pengujian efisiensi, skalabilitas, dan ketersediaan menunjukkan hasil positif, dan Rancher dapat mengelola sumber daya dengan optimal.

Sebagai masukan dan saran dalam penelitian Penerapan Manajemen Orkestrator Kontainer Pada Learning Management System ini, untuk menghindari *single point of failure* yang menjadi elemen dalam sistem jika mengalami kegagalan, akan menyebabkan seluruh sistem atau aplikasi menjadi tidak berfungsi [19]. Penulis menyarankan agar di lakukan membuat kluster baru yg di letakan di lokasi lain atau *data center* lain dengan spesifikasi kluster yang sama dengan kluster yang *existing* sekarang.

Secara singkat dapat disimpulkan ketiga penelitian berfokus pada penggunaan teknologi kontainer, seperti Docker dan Kubernetes, untuk meningkatkan efisiensi, skalabilitas, ketersediaan,

dan manajemen trafik pada sistem-sistem seperti Learning Management System. Penelitian ini lebih berfokus pada penggunaan Rancher Kubernetes pada LMS dan mengujinya dalam hal efisiensi dan skalabilitas. Penelitian kedua lebih berfokus pada Docker Swarm dan container load balancing untuk mengatasi beban trafik pada LMS Moodle. Sementara itu, penelitian ketiga lebih bersifat eksploratif dalam menggantikan virtualisasi berbasis hypervisor dengan Docker pada server-server yang ada. Meskipun fokus dan pendekatan penelitian berbeda, semuanya menunjukkan hasil positif terkait penerapan teknologi kontainer dalam lingkungan yang berbeda.

5. KESIMPULAN

Hasil pengujian menunjukkan telah membawa hasil yang sangat positif dalam menangani beban *traffic* dari permintaan pengguna. Dalam pengujian efisiensi sistem, terlihat bahwa penggunaan sumber daya CPU dan Bandwidth berjalan dengan sangat efisien bahkan saat diberi beban hingga 200 pengguna. Uji skalabilitas juga memberikan hasil yang mengesankan, menunjukkan bahwa LMS Pradita mampu dengan stabil menangani beban hingga 50, 100, dan 200 pengguna.

Tak hanya itu, pengujian ketersediaan juga menunjukkan tingkat keandalan yang tinggi. Meskipun salah satu worker mengalami kegagalan, LMS Pradita tetap dapat diakses dan beroperasi dengan lancar berkat bantuan dari worker lainnya. Hal ini membuktikan bahwa infrastruktur Kubernetes yang diterapkan telah mencapai tingkat ketersediaan yang tinggi.

Selain itu, dalam pengujian manajemen dengan Rancher, platform ini menunjukkan kemampuannya dalam menjalankan dan mengelola LMS Pradita dengan sangat efektif. Dengan menggunakan Rancher, tim pengelola dapat dengan mudah memantau penggunaan infrastruktur, melakukan pembaruan atau pengurangan sesuai kebutuhan, dan bahkan mengoptimalkan biaya dalam pengelolaan infrastruktur.

Kesimpulannya, implementasi Kubernetes dengan Rancher telah membuktikan diri sebagai solusi yang efisien, skalabel, dan memiliki ketersediaan tinggi untuk mengelola platform pembelajaran daring LMS Pradita. Keberhasilan pengujian ini menunjukkan bahwa kombinasi teknologi ini memiliki potensi untuk diadopsi di berbagai lembaga pendidikan di Indonesia, memberikan pengalaman pembelajaran yang lebih baik dan dapat diandalkan bagi pengguna.

DAFTAR PUSTAKA

- [1] Verawati, U. J., Alifa, Y. D. N., Millah, Z., & Nissa, Z. K. (2023). Implementasi Pembelajaran E-Learning Sebagai Transformasi Pendidikan di Era Digital. *Social Science Academic*, 1(2), 221-228.
- [2] Indrawati, B. (2020). Tantangan dan peluang pendidikan tinggi dalam masa dan pasca Pandemi Covid-19. *Jurnal Kajian Ilmiah*, 1(1).
- [3] Fahmi, N., Dzakiah, D., Alhabsyi, F., & Mudaimin, M. (2022). Dinamika E-Learning pada Model Pembelajaran di Perguruan Tinggi (Tinjauan Kondisi Teknologi di Masa Pandemi Covid-19). *Iqra: Jurnal Ilmu Kependidikan Dan Keislaman*, 17(2), 44-51.
- [4] A. Yauma, I. Fitri, and S. Ningsih, "Learning Management System (LMS) pada E-Learning Menggunakan Metode Agile dan Waterfall berbasis Website," *Jurnal JTIC (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 5, no. 3, pp. 323–328, 2021.
- [5] A. Garad, A. M. Al-Ansi, and I. N. Qamari, "The role of e-learning infrastructure and cognitive competence in distance learning effectiveness during the covid-19 pandemic," *Jurnal Cakrawala Pendidikan*, vol. 40, no. 1, pp. 81–91, 2021.
- [6] Y. Ghilay, "Effectiveness of learning management systems in higher education: Views of lecturers with different levels of activity in LMSs," *Ghilay, Y.(2019). Effectiveness of Learning Management Systems in Higher Education: Views of Lecturers with Different Levels of Activity in LMSs. Journal of Online Higher Education*, vol. 3, no. 2, pp. 29–50, 2019.
- [7] A. Yauma, I. Fitri, and S. Ningsih, "Learning Management System (LMS) pada E-Learning Menggunakan Metode Agile dan Waterfall berbasis Website," *Jurnal JTIC (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 5, no. 3, pp. 323–328, 2021.
- [8] S. Ashour, G. A. El-Refae, and E. A. Zaitoun, "Post-pandemic higher education: Perspectives from university leaders and educational experts in the United Arab Emirates," *Higher Education for the Future*, vol. 8, no. 2, pp. 219–238, 2021.
- [9] A. Al-Hunaiyyan, S. Al-Sharhan, and R. AlHajri, "Prospects and challenges of learning management systems in higher education," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, 2020.
- [10] S. Dwiyatno, E. Rachmat, A. P. Sari, and O. Gustiawan, "Implementasi virtualisasi server berbasis docker container," *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, vol. 7, no. 2, pp. 165–175, 2020.
- [11] M. Zulfikar, A. Bhawiyuga, and A. Basuki,

- “Implementasi Lab Virtual berbasis Teknologi Kontainer Multi Klaster dengan Orkestrator Kubernetes,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 6, pp. 2649–2654, 2022.
- [12] A. Widarma and Y. H. Siregar, “Rancangan Teknologi Virtualisasi Untuk Optimalisasi Server Di Universitas Asahan,” *CESS (Journal of Computer Engineering, System and Science)*, vol. 4, no. 2, pp. 313–319, 2019.
- [13] I. M. Al Jawarneh *et al.*, “Container orchestration engines: A thorough functional and performance comparison,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019, pp. 1–6.
- [14] D. Al Fansha, M. Y. H. Setyawan, and M. N. Fauzan, “Load Test pada Microservice yang menerapkan CQRS dan Event Sourcing,” *Jurnal Buana Informatika*, vol. 12, no. 2, pp. 126–134, 2021.
- [15] N. Siagian, T. E. Tamba, H. H. O. Situmorang, and H. Samosir, “Aplikasi Apotek Berbasis Web Menggunakan Arsitektur Microservices (Studi Kasus Apotek Glen, Kab. Toba),” *Journal of Applied Technology and Informatics Indonesia*, vol. 1, no. 2, pp. 22–28, 2021.
- [16] M. Moravcik, M. Kontsek, P. Segec, and D. Cymbalak, “Kubernetes-evolution of virtualization,” in *2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, IEEE, 2022, pp. 454–459.
- [17] K. T. Widagdo, T. I. Bayu, and Y. A. Susetyo, “Pemodelan Sistem Monitoring Sensor Curah Hujan Menggunakan Grafana,” *Indonesian Journal of Computing and Modeling*, vol. 2, no. 2, pp. 1–8, 2019.
- [18] I. Tzanettis, C.-M. Androna, A. Zafeiropoulos, E. Fotopoulou, and S. Papavassiliou, “Data Fusion of Observability Signals for Assisting Orchestration of Distributed Applications,” *Sensors*, vol. 22, no. 5, p. 2061, 2022.
- [18] D. Balla, C. Simon, and M. Maliosz, “Adaptive scaling of Kubernetes pods,” in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, IEEE, 2020, pp. 1–5.
- [19] E. Nugroho, “ANALISIS PENCEGAHAN SINGLE POINT OF FAILURE PADA JARINGAN SERVICE LAYER 2 VPN DENGAN METODE FAST REROUTE MPLS TRAFFIC ENGINEERING.” Institut Teknologi Telkom Jakarta, 2022.