

# Development of Access Mechanism of E-learning Linux Container-Based

Septya Andi Suryanto

Program Studi Teknik Informatika  
Universitas Muhammadiyah Malang

septyaandi24@gmail.com

Syaifuddin

Program Studi Teknik Informatika  
Universitas Muhammadiyah Malang

udinsaif@gmail.com

Diah Rizqiwati

Program Studi Teknik Informatika  
Universitas Muhammadiyah Malang

diah.rizqiwati@gmail.com

## ABSTRACT

The rapid development and utilization of technology penetrated to the world of education. E-learning learning method is used for the improvement of the quality of the students. Utilization of E-learning would require an adequate infrastructure to run optimally. To apply E-learning on each course, especially at the University level of server resources are needed quite a lot. Then used virtualization technology to save and optimize resources. Various virtualization technologies can be utilized in such cases, one of which is Docker. The purpose of this research is to apply E-learning on Docker virtualization and develop Docker access mechanism which initially to access it must use IP to DNS to facilitate access and administration process. The results showed that E-learning is very possible in the application of Docker virtualization technology and access mechanism can be developed into DNS.

## Keywords

Architecture, DNS, Docker, E-Learning, LXC, Virtualization

## 1. PENDAHULUAN

Pemanfaatan *E-learning* menjadikan proses belajar mengajar tidak harus berada di ruang kelas tetapi di mana saja selama tersedia koneksi internet dan alat untuk mengaksesnya. Untuk mengimplementasikan *E-learning* dibutuhkan suatu *server* sebagai penyedia layanan agar dapat diakses oleh pengguna. Banyak universitas dan perguruan tinggi sudah menerapkan *E-learning* sebagai penunjang sistem pembelajarannya. Agar manajemen *E-learning* terstruktur maka tiap program studi memiliki sistem *E-learning* masing-masing. Dengan banyaknya program studi perlu biaya dan sumber daya yang sangat besar apabila setiap program studi menyediakan server masing-masing. Oleh karena itu, pemanfaatan teknologi virtualisasi sangat penting untuk digunakan dalam kasus seperti ini.

Teknologi virtualisasi adalah teknologi yang memungkinkan beberapa sistem operasi dan layanan untuk berjalan pada perangkat keras pada waktu yang bersamaan. Mekanisme tersebut menjadikan sumber daya perangkat keras seperti prosesor, memori, penyimpanan, dan perangkat jaringan digunakan secara bersamaan. Terdapat beberapa jenis teknologi virtualisasi salah satunya adalah Docker. Docker merupakan kerangka kerja virtualisasi berbasis sistem operasi Linux 64-bit. Docker tidak melakukan virtualisasi secara penuh melainkan melakukan virtualisasi di atas kernel yang berjalan pada sistem operasi di bawahnya. Sehingga Docker mampu menunjukkan layanan aplikasi web yang lebih baik dari segi kinerja, ketersediaannya

tinggi, dan hemat sumber daya server dibandingkan dengan virtualisasi lainnya[4][7].

Pada penelitian ini, peneliti melakukan pengembangan mekanisme akses *E-learning* berbasis Docker. Pengaksesan secara langsung oleh user terhadap web server pada container melalui port yang dibuka menjadikan beban kerja pada CPU meningkat sehingga pemrosesan website menjadi berat jika diakses oleh banyak user pada satu waktu. NGINX merupakan server HTTP dan Reverse Proxy dengan kinerja yang tinggi yang dapat mereduksi penggunaan sumber daya CPU dan memori serta operasi yang stabil. NGINX sebagai reverse proxy dapat menjadi solusi port mapping pada Docker dan juga menyembunyikan alamat internal server[2]. Mekanisme kerja NGINX sebagai reverse proxy adalah NGINX melakukan perubahan URL yang dikirim oleh user menjadi URI yang ditujukan ke server dan web server akan merespon dengan mengirimkan konten ke NGINX untuk diteruskan ke user.

## 2. KAJIAN PUSTAKA

### 2.1 E-learning

*E-learning* atau bisa juga disebut sebagai pembelajaran elektronik merupakan bentuk pemanfaatan teknologi informasi di bidang pendidikan dalam bentuk digital yang biasanya dijumpai oleh teknologi internet[6].

#### 2.1.1 Moodle

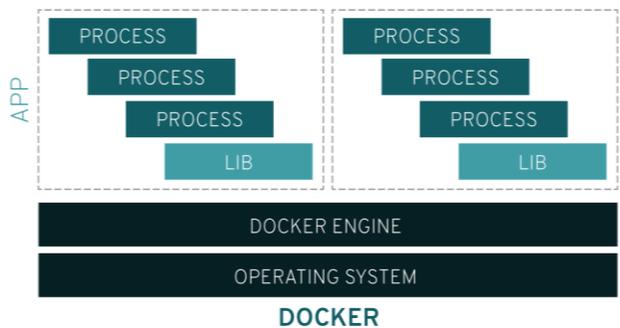
Moodle adalah suatu LMS (Learning Management System) gratis yang memungkinkan untuk membuat pengalaman pembelajaran daring secara kuat, fleksibel, dan menarik[1]. Moodle merupakan akronim dari Modular Object-Oriented Dynamic Learning Environment diperkenalkan pada tahun 1998 dan dirilis pada tahun 2001. Moodle memberlakukan dirinya sebagai solusi terbaik dan menjadi Learning Management System yang paling banyak digunakan. Beberapa fitur yang ditawarkan oleh Moodle bagi penggunaannya adalah pengguna dapat melakukan posting item baru, penugasan, sumber jurnal elektronik, obrolan antar pengguna, forum diskusi, kuis, survey, dan lain-lain.

### 2.2 Docker

Docker merupakan *platform* virtualisasi *container* yang dibangun di atas teknologi sumber terbuka yang tersedia untuk umum[13]. Docker memberikan antarmuka untuk membuat dan mengendalikan kontainer dengan mudah dan aman. Dengan begitu pengembang dapat mengemas aplikasinya ke dalam kontainer Docker dengan mudah dan ringan yang bisa beroperasi di segala tempat tanpa melakukan modifikasi. Docker sendiri terdiri dari dua komponen utama, yaitu Docker *Engine* dan Docker *Hub*[11].

### 2.2.1 Docker Engine

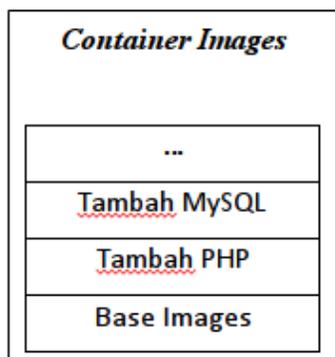
Seperti ilustrasi pada gambar 1, Docker *engine* berjalan tepat di atas sistem operasi. Untuk menjalankan aplikasi, Docker *engine* tidak perlu lagi membuat suatu virtualisasi sistem operasi agar aplikasi dapat berjalan, melainkan langsung berjalan di dalam suatu *container* yang berjalan dan di kelola oleh Docker *engine*.



Gambar 1. Arsitektur Docker

### 2.2.2 Docker Images

Docker *images* merupakan suatu rangkaian lapisan data di atas *images* dasar[11]. Setiap *container* Docker dimulai dari *images* dasar seperti Ubuntu, Debian, dan distro linux lainnya. Ketika pengguna melakukan perubahan pada suatu *container*, bukan melakukan perubahan secara langsung pada *image* dari suatu *container* melainkan menambahkan suatu lapisan yang mengandung perubahan itu sendiri, seperti menambahkan aplikasi MySQL.



Gambar 2. Arsitektur Docker images

### 2.2.3 Docker Container

Docker *container* merupakan suatu lingkungan virtual yang memanfaatkan dua fitur yang dimiliki oleh Linux, yaitu *namespace* dan *cgroups*. *Cgroups* memberikan mekanisme untuk menghitung dan membatasi sumber daya yang dapat di proses dan diakses oleh setiap *container*. *Namespace* membungkus sumber daya sistem operasi menjadi beberapa hal (*container*) yang berbeda[11].

## 3. PENELITIAN TERDAHULU

Pada penelitian terdahulu, sistem *E-learning* Moodle berhasil diterapkan pada layanan Amazon EC2[9], VMS[10], dan Docker[3]. Berbeda dengan penelitian [3], proses konfigurasi dan *deployment* pada penelitian [9][10] dilakukan secara manual dan satu persatu, seperti melakukan instalasi web server dan mysql *server* disetiap proses *deployment* sistem *E-learning* Moodle. Hal tersebut dikarenakan implementasi sistem *E-learning* Moodle menggunakan teknologi virtualisasi berjenis VMS. Pada penelitian [3] sistem *E-learning* Moodle berhasil diterapkan pada platform Docker beserta program untuk mempermudah proses *deployment* dan manajemen sistem *E-learning* [3]. Program yang digunakan untuk melakukan manajemen *container* pada penelitian tersebut memiliki fungsi untuk melakukan eksekusi perintah untuk melakukan pembuatan Moodle, mendaftar *container* yang ada, menghapus, memulai dan memberhentikan *container* dengan berbasis Bash script. Dalam mekanisme pengaksesan Moodle, user harus menambahkan port yang telah ditentukan disetiap *container* seperti "http://<domain/ IP server>:<port container>". Menghafal angka port untuk masing-masing Moodle menjadikan sistem kurang efektif baik bagi administrator juga user jika terdapat lebih dari dua aplikasi Moodle.

## 4. METODOLOGI PENELITIAN

Pada penelitian ini, peneliti melakukan perancangan dan pengembangan arsitektur jaringan dengan memanfaatkan teknologi virtualisasi Docker. Metode eksperimen digunakan dalam penelitian ini dengan membuat simulasi yang sesuai dengan kasus ini.

### 4.1 Studi Literatur

Dalam tahapan ini, penulis melakukan studi literatur terkait judul yang diambil. Dalam merancang web dan Docker dibutuhkan literatur yang perlu dipelajari dari berbagai sumber seperti jurnal, artikel, informasi dari buku maupun dokumentasi aplikasi atau teknologi yang digunakan. Sumber pustaka antara lain terkait dengan *E-learning*, pengembangan web, Docker dan virtualisasi.

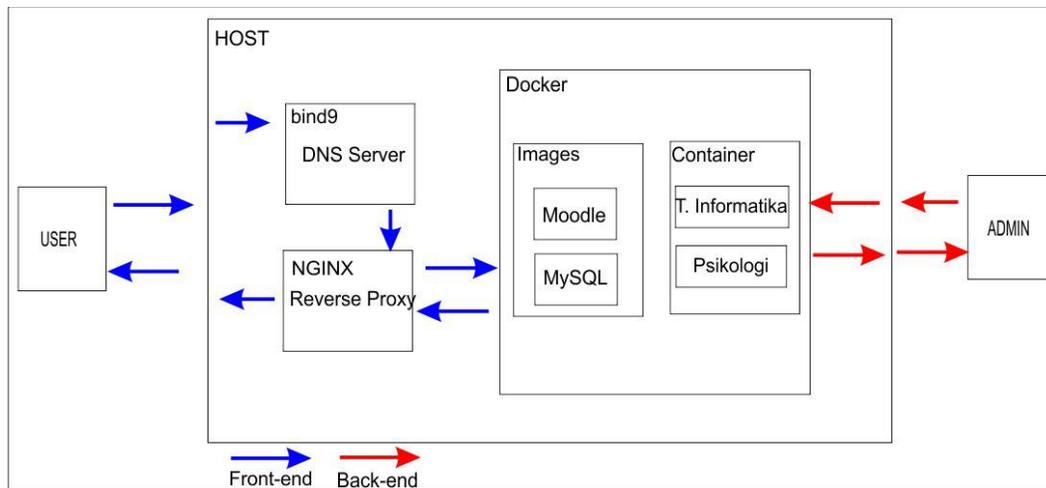
### 4.2 Analisa Kebutuhan

Berdasar studi literatur yang dilakukan, dilakukan analisa kebutuhan dari sistem yang akan dibangun. Analisa yang dilakukan meliputi beberapa hal antara lain arsitektur docker (1), arsitektur Moodle (2), mekanisme DNS (3), dan NGINX sebagai *reverse proxy* (4).

### 4.3 Model Penelitian

#### 4.3.1 Perancangan Arsitektur

Arsitektur sistem yang akan diimplementasikan digambarkan sesuai dengan gambar 3. Sistem yang akan dibangun akan diimplementasikan menggunakan mesin virtual yang mana semua perangkat lunak dijadikan dalam satu host dengan Ubuntu Server 16.04 LTS sebagai sistem operasinya. Dalam host tersebut terdiri dari bind9 sebagai DNS *server* (1), Nginx sebagai *reverse proxy* (2), dan Docker sebagai virtualisasi (3). Docker *images* pada sistem ini terdiri dari *images* Moodle beserta web servernya yaitu Apache Web Server dan MySQL.



Gambar 3. Arsitektur Sistem

### 4.3.2 Implementasi

Pada tahapan ini, peneliti melakukan pembuatan beberapa container yang terdiri dari container Moodle dan MySQL yang dibangun berdasarkan *images* Moodle dan MySQL. Tujuan dari pemisahan antara *container* Moodle dan MySQL adalah agar pencadangan data baik dari Moodle maupun MySQL menjadi lebih mudah.

```

ta@ta:~$ docker run -d --name db-$jurusan --net moodlenet --ip $ip2 \
-e MYSQL_DATABASE=moodle -e \ MYSQL_ROOT_PASSWORD=moodle -e \ MYSQL_USER=moodle -e MYSQL_PASSWORD=moodle \ septyaandi/mysqlmaster
  
```

Gambar 4. Perintah membuat container E-learning

```

ta@ta:~$ docker run -d --name www-$jurusan --net moodlenet --ip \ $ip1 -p $port1:80 --link db-$jurusan \
-e VIRTUAL_HOST=$jurusan.moodlenet.com \ jhardison/moodle
  
```

Gambar 5. Perintah membuat container basis data E-learning

Pada jaringan internal docker dibuat suatu cluster jaringan yang dikhususkan untuk moodle dan mysql yang mana tiap container akan diberikan IP statis. Pada penelitian ini cluster jaringan akan dinamai moodlenet dengan subnet 172.19.0.0/16.

```

ta@ta:~$ docker network create moodlenet --subnet 172.18.0.0/16
  
```

Gambar 6. Perintah membuat cluster jaringan Docker

Untuk mempermudah pembuatan *E-learning*, peneliti membuat *script* berbasis bash. Dalam membuat membuat *E-learning* melalui *script* bash ini inputan yang harus dimasukkan antara lain nama kelompok belajar sebagai identitas container dan ip moodle. Selain hal tersebut parameter yang harus diisi akan diisi otomatis oleh *script* bash tersebut.

```

#####
###   E-LEARNING   ###
#####
Your container is : psikolgi
Your IP's site (www-psikolgi) : 172.18.0.5:8081
Congrat's your site has been created!
ta@ta:~$
  
```

Gambar 7. Hasil program script bash pembuatan E-learning

## 5. HASIL

```

#####
###   E-LEARNING   ###
#####
Let's start:
Jurusan:
tarbiyah
ip moodle: (ex: 172.18.0.2)
172.18.0.20
[sudo] password for ta:
5c7c58a816a83d2446ecb17b9fb1930246245bc0a9224498b82a638a8667794d
c7e120722dd5e58e62d4a63f43d8d149bd08d86fceb0b7f17032801bc47ba8f
##### (100%)
Build finished.
  
```

Gambar 8. Proses program script bash dalam membuat E-learning

Mekanisme akses *E-learning* berbasis linux *container* Docker dapat dikembangkan dengan menambahkan *reverse proxy* agar *E-learning* dapat diakses melalui DNS. Mekanisme awal untuk mengakses *E-learning* Moodle adalah melalui IP dari *host* dengan menambahkan *port* yang dispesifikasikan pada masing-masing *container*. Dengan menambahkan Nginx sebagai *reverse proxy*, *container* Docker dapat diakses dengan menggunakan nama subdomain.

Gambar 8 merepresentasikan proses pembuatan container suatu kelompok belajar. Masukkan yang diminta oleh program adalah nama jurusan atau kelompok belajar dan IP untuk container. Untuk port yang di publish secara otomatis diisikan oleh program dengan mencari port yang tidak terpakai oleh komputer host yang dimulai dari angka 8081. Selain itu, program jugamelakukan konfigurasi secara otomatis pada Nginx dengan menambahkan *proxy\_pass* pada virtual host nginx. Berikut merupakan pengaturan nginx yang dihasilkan oleh program.

```
#psikolgi#
server {
    listen 80;
    server_name psikolgi.moodlenet.com;
    location / {
        proxy_pass http://172.18.0.5:80;
        proxy_set_header Host $host:$server_port;
    }
}
```

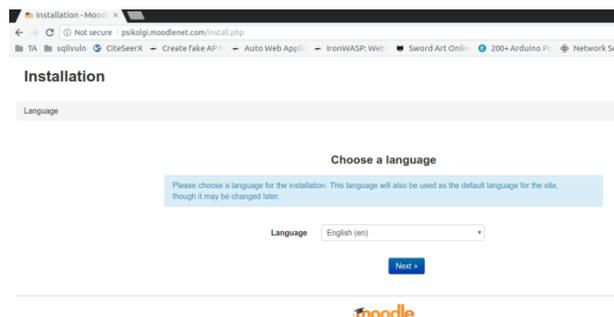
Gambar 9. Konfigurasi reverse proxy pada nginx

Karena DNS server berada pada komputer host yang sama, maka program juga dapat menambahkan pengaturan pada bind9 agar DNS yang ditujukan pada container dapat dikenali. Pengaturan yang ditambahkan pada bind9 adalah pengaturan host bind9.

```
$ttl 38400
moodlenet.com. IN      SOA     ta. admin.gmail.com. (
                    1518635237
                    10800
                    3600
                    604800
                    38400 )
moodlenet.com. IN      NS      ta.
moodlenet.com. IN      A       192.168.100.1
www.moodlenet.com. IN  A       192.168.100.1
elektro.moodlenet.com. IN A       192.168.100.1
informatika.moodlenet.com. IN A       192.168.100.1
sipil.moodlenet.com. IN A       192.168.100.1
psikolgi.moodlenet.com. IN A       192.168.100.1
tarbiyah.moodlenet.com. IN A       192.168.100.1
```

Gambar 10. Konfigurasi DNS server

Pada gambar 9, moodle yang dikhususkan bagi kelompok belajar psikologi dapat diakses melalui DNS dengan URL <http://psikologi.moodlenet.com>. Domain tersebut merupakan hasil kerja dari reverse proxy yang mana dibalik Nginx, domain tersebut di routing kan pada jaringan internal Docker dengan alamat IP container 172.18.0.5 dan port yang publikasikan untuk jaringan eksternal 8081. Jadi cara kerja Nginx sebagai reverse proxy adalah mengalamatkan psikologi.moodlenet.com ke 172.18.0.5:8081.



Gambar 11. Hasil pengaksesan E-learning menggunakan DNS

## 6. PENGUJIAN

Pengujian dilakukan dengan menjalankan beberapa skenario dengan membandingkan sistem awal dengan sistem baru. Sistem awal merupakan pemanfaatan docker sebagai Platform as a service dalam pengembangan aplikasi *E-learning*. Perbedaan dengan sistem yang ada pada penelitian ini adalah sistem awal tidak menggunakan reverse proxy, sedangkan penelitian ini menggunakannya. Skenario yang akan dilakukan antara lain melakukan pengujian dengan parameter user yang mengakses pada satu waktu secara bersamaan dengan jumlah user 100, 250, dan 500. Pada penelitian ini pengujian dibagi menjadi dua, yaitu pengujian untuk mengetahui performa dari sistem dan pengujian untuk mengetahui waktu respon dari *E-learning* yang diimplementasikan pada sistem yang ada pada penelitian ini.

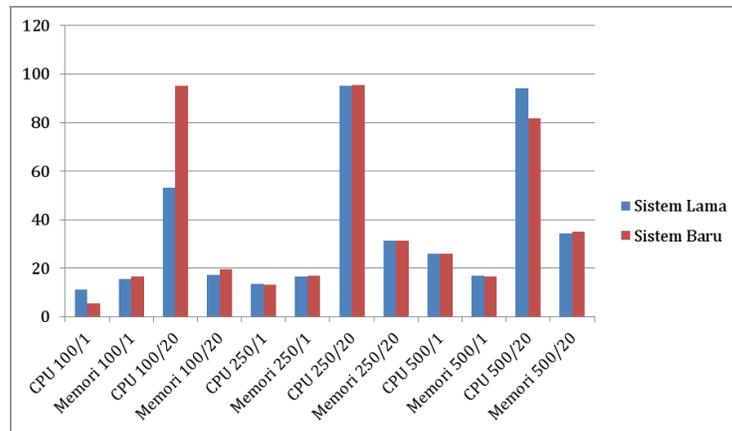
### 6.1 Pengujian Performa

Pada bagian ini peneliti melakukan pengujian performa terhadap container docker. Tujuan dari pengujian ini adalah untuk mengetahui performa CPU dan Memori ketika menjalankan sistem *E-learning* yang diterapkan pada container docker. Aplikasi yang digunakan untuk melakukan pengujian adalah apache Jmeter 4. Dalam pengujian dilakukan beberapa scenario, antara lain adalah penggunaan thread dengan jumlah user 100 dengan 1 kali akses dalam 1 menit, 100 user dengan 20 kali akses dalam 1 menit, 250 user dengan 1 kali akses dalam 1 menit, 250 user dengan 20 kali akses dalam 1 menit, 500 user dengan 1 kali akses dalam 1 menit, dan 500 user dengan 20 kali akses dalam 1 menit. Pengujian dilakukan dengan membandingkan antara sistem lama dengan sistem yang diteliti.

Dari hasil pengujian dapat dilihat bahwa penggunaan CPU di beberapa scenario sistem lama lebih baik dari pada sistem baru. Hal tersebut dapat dipengaruhi karena pada sistem baru terdapat beberapa aplikasi tambahan seperti nginx dan bind9. Namun dari segi penggunaan memori kedua sistem tidak terdapat perbedaan yang signifikan.

Tabel 1. Hasil Pengujian Performa CPU Dan Memori

Jumlah User		100/1	100/20	250/1	250/20	500/1
CPU (%)	Sistem Lama	11.12	53.05	13.37	95.27	25.94
	Sistem Baru	5.48	95.02	13.03	95.45	25.96
MEMORI (%)	Sistem Lama	15.42	17.29	16.49	31.25	16.84
	Sistem Baru	16.71	19.58	16.75	31.39	16.66
Total User		100	2000	250	5000	500



Gambar 12. Grafika Pengujian Performa CPU Dan Memori

## 6.2 Pengujian Waktu Respon

Pada bagian ini peneliti melakukan pengujian waktu respon E-learning . Tujuan dari pengujian ini adalah untuk mengetahui waktu respon ketika mengakses E-learning yang diterapkan pada container docker. Aplikasi yang digunakan untuk melakukan pengujian adalah apache Jmeter 4. Aspek yang diuji dalam pengujian ini adalah latency dan throughput.

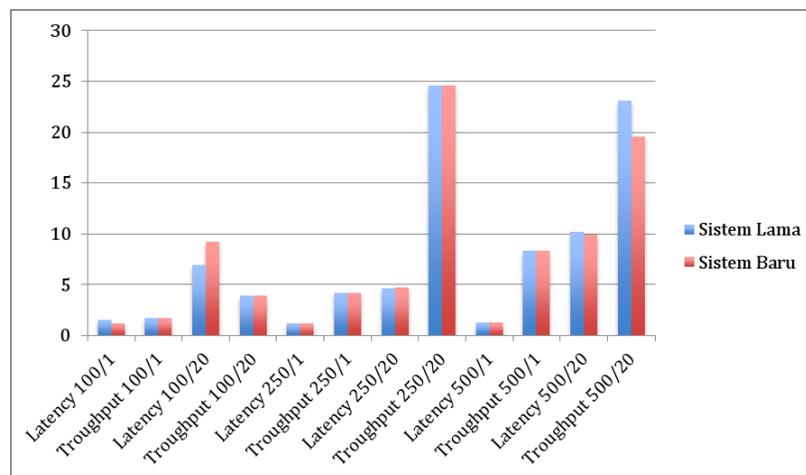
Dalam pengujian dilakukan beberapa scenario, antara lain adalah penggunaan thread dengan jumlah user 100 dengan 1 kali akses dalam 1 menit, 100 user dengan 20 kali akses dalam 1 menit, 250 user dengan 1 kali akses dalam 1 menit, 250 user dengan 20 kali

akses dalam 1 menit, 500 user dengan 1 kali akses dalam 1 menit, dan 500 user dengan 20 kali akses dalam 1 menit. Pengujian dilakukan dengan membandingkan antara sistem lama dengan sistem yang diteliti.

Dari hasil pengujian dapat dilihat bahwa waktu respon, baik latency maupun troughput di beberapa scenario tidak memiliki perbedaan yang signifikan. Namun pada ketika terdapat user yang mengakses berjumlah banyak (dalam scenario terakhir) troughput yang dihasilkan pada sistem baru lebih baik dari pada sistem lama.

Tabel 2. Hasil Pengujian Waktu Respon E-learning

Jumlah User		100/1	100/20	250/1	250/20	500/1	500/20
Latency (s)	Sistem Lama	1.568	6.920	1.186	4.662	1.229	10.215
	Sistem Baru	1.176	9.184	1.211	4.693	1.234	9.968
Throughput (s)	Sistem Lama	1.7	3.9	4.2	24.6	8.3	23.1
	Sistem Baru	1.7	24.9	4.2	24.6	8.3	19.6
Total User		100	2000	250	5000	500	10000



Gambar 13. Garifka pengujian waktu respon E-learning

## 7. KESIMPULAN

Berdasarkan hasil analisa, implementasi dan pengujian yang telah dilakukan dalam penelitian ini, Peneliti dapat membuat kesimpulan bahwa:

1. Sistem *E-learning* dapat diimplementasikan menggunakan virtualisasi container Docker.
2. Mekanisme untuk mengakses *E-learning* dapat dikembangkan menggunakan DNS sehingga mempermudah pengaksesan.

Antara mekanisme sistem yang lama dan mekanisme sistem yang baru tidak terdapat perbedaan yang signifikan pada waktu respon. Namun pada CPU load sistem mekanisme yang baru menunjukkan bahwa memerlukan CPU load yang lebih banyak daripada yang sebelumnya.

## DAFTAR PUSTAKA

- [1] Rice, William. Moodle 2.0 E-Learning Course Development. Packt Publishing Ltd, 2011.
- [2] "Difference Between a DMZ Server & a Reverse Proxy." [Daring]. Tersedia pada: <http://smallbusiness.chron.com/difference-between-dmz-server-reverse-proxy-71583.html>. [Diakses: 30-Jan-2018].
- [3] A. E. Paksi, Syaifuddin, dan Z. Sari, "Implementasi Container Docker Sebagai Platform As A Service Dalam Pengembangan Aplikasi Berbasis Web Pada *E-learning*," *Jurnal Kinetik*, vol. 1, no. 1, pp. 101-109, 2017.
- [4] F.Adiputra, "Container Dan Docker: Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi Web," *Jurnal Simantec*, vol. 4, no. 3, pp.167-176, 2015.
- [5] L. Darliah, "Pengaruh Kualitas Informasi Dan Penggunaan *E-learning* Terhadap Prestasi Belajar Mahasiswa Pendidikan Ekonomi Fe Uny Dengan Motivasi Belajar Sebagai Variabel Intervening," *Jurnal Pendidikan dan Ekonomi*, vol. 5, no. 2, pp. 150-157, 2016.
- [6] M. Islamiyah, dan L. Widayanti, "Efektifitas Pemanfaatan *E-learning* Berbasis Website Terhadap Hasil Belajar Mahasiswa STMIK Asia Malang Pada Mata Kuliah Fisika Dasar," *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 10, no. 1, pp. 41-46, 2016.
- [7] C. Arango, R. Dernas, J. Sanabria, "Performance Evaluation of Container-based Virtualization for High Performance Computing Environments," arXiv preprint arXiv:1709.10140, 2017.
- [8] S. Tomo dan B. Widada, "Pengaruh Pemanfaatan *E-learning* Terhadap Prestasi Belajar Mahasiswa (Studi Kasus STMIK Sinar Nusantara Surakarta)," *Jurnal Ilmiah Sinus*, vol. 12, no. 1, pp. 35-44, 2014.
- [9] V. Siládi, dan V. Mižurová, "LMS Moodle on Computing Cloud," *4th Interantional Scientific Conference in V4 Countries, Applied Natural Sciences*, pp. 298-302, 2013.
- [10] X. Guo, Q. Shi, D. Zhang, "A Study on Moodle Virtual Cluster in Cloud Computing," *2013 Seventh International Conference on Internet Computing for Engineering and Science*, pp. 15-20, 2013.
- [11] T. Bui, "Analysis of docker security," arXiv preprint arXiv:1501.02967, 2015.
- [12] "What is DOCKER?" [Daring]. Tersedia: <https://www.redhat.com/en/topics/containers/what-is-docker>. [Diakses: 11-Maret-2018]
- [13] Špaček, František, Radomír Sohlich, and Tomáš Dulík. "Docker as platform for assignments evaluation." *Procedia Engineering* 100 (2015): 1665-1671. Harjito, B., and Han, S., 2010, November.

Wireless multimedia sensor networks applications and security challenges. In *Broadband, Wireless Computing, Communication, and Applications (BWCCA), 2010 International Conference on* (pp. 842-846). IEEE.