

RANCANG BANGUN INFRASTRUKTUR WEB SERVER BERBASIS DOCKER PADA UBUNTU SERVER

Zaenal Mutaqin Subekti¹, Kikim Mukiman², Subandri³, Muhammad Lutfi Sulthon Auliya
Sulistiyono⁴, Roni Eka Putra⁵

¹Universitas Bani Saleh, zaenalms@ubs.ac.id

²Universitas Bani Saleh, kikim@ubs.ac.id

³Universitas Bani Saleh, andrisubandri@ymail.com

⁴BPPTIK Kementerian Kominfo, lutfisulistiyono@gmail.com

⁵BPPTIK Kementerian Kominfo, roni.ekaputra13@gmail.com

ABSTRAK

Kebutuhan akan infrastruktur teknologi informasi yang efisien, fleksibel, dan skalabel semakin meningkat. Salah satu teknologi yang memberikan solusi terhadap kebutuhan ini adalah docker container. Web server merupakan komponen vital dalam penyediaan layanan berbasis web, seperti hosting situs web, aplikasi web, dan API. Ubuntu Server 22.04 dapat dikolaborasi dengan docker container sehingga dapat menyediakan konterisasi untuk kebutuhan web server seperti nginx. Dari hasil pengujian akses ke web server semua pengguna dapat mengakses ke web server dengan berhasil dan hasil pengujian tes koneksi dari lima pengguna hanya pengguna dari smartphone yang mendapatkan akses lambat dengan maksimal akses mencapai 82,5ms dan akses paling cepat didapatkan kurang dari 1ms, sehingga perlu dilakukan pengujian lebih banyak lagi untuk mendapatkan hasil pengujian yang optimal terhadap akses web server, dengan ada nya docker container pada sistem operasi ubuntu server 22.04 dapat memberikan kemudahan dalam membangun sebuah web server dengan melakukan perintah docker pull sehingga sangat memudahkan pengguna dalam melakukan perancangan web server pada ubuntu server.

Kata Kunci: web server, docker, container, ubuntu, server

ABSTRACT

The need for efficient, flexible, and scalable information technology infrastructure is increasing. One technology that provides a solution to this need is docker container. Web servers are a vital component in providing web-based services, such as website hosting, web applications, and APIs. Ubuntu Server 22.04 can be collaborated with docker container so that it can provide countermeasures for web server needs such as nginx. From the results of testing access to the web server, all users can access the web server successfully and the results of the connection test from five users, only users from smartphones get slow access with a maximum access of 82.5ms and the fastest access is less than 1ms, so more testing is needed to get optimal test results for web server access, with the presence of docker containers on the Ubuntu Server 22.04 operating system, it can provide convenience in building a web server by executing the docker pull command so that it is very easy for users to design a web server on the Ubuntu Server.

Keywords: web server, docker, container, ubuntu, server

PENDAHULUAN

Dalam era digital yang berkembang pesat, kebutuhan akan infrastruktur teknologi informasi yang efisien, fleksibel, dan skalabel semakin meningkat. Salah satu teknologi yang memberikan solusi terhadap kebutuhan ini adalah containerisasi, di mana [1] Docker menjadi salah satu platform yang paling populer. [2] Docker memungkinkan pengembang untuk membuat, mengelola, dan menjalankan aplikasi di dalam wadah (container) yang terisolasi, sehingga mempermudah proses pengembangan, distribusi, dan deployment aplikasi.

[3] Ubuntu Server 22.04, sebagai salah satu sistem operasi berbasis Linux yang andal dan banyak digunakan, menawarkan kestabilan, dukungan jangka panjang, dan kompatibilitas yang tinggi dengan berbagai aplikasi dan teknologi modern, termasuk Docker. Kombinasi antara Docker dan [4] Ubuntu Server 22.04 memberikan landasan yang kuat untuk membangun dan mengelola infrastruktur web server yang efisien.

Web server merupakan komponen vital dalam penyediaan layanan berbasis web, seperti hosting situs web, [5] aplikasi web, dan [6] API. Dalam konteks ini, penerapan Docker pada Ubuntu Server 22.04 memberikan berbagai keuntungan, seperti kemudahan pengelolaan dependensi, isolasi lingkungan aplikasi, dan kemampuan untuk menjalankan beberapa instance web server secara paralel tanpa konflik. Hal ini sangat relevan dalam mendukung kebutuhan bisnis yang memerlukan ketersediaan layanan yang tinggi dan responsif terhadap perubahan.

Penelitian ini bertujuan untuk merancang dan membangun web server menggunakan Docker pada Ubuntu Server 22.04. Fokus penelitian ini meliputi proses instalasi dan konfigurasi Docker, pembuatan container untuk web server. Dengan pendekatan ini, diharapkan dapat memberikan panduan praktis dan solusi efektif bagi pengembang atau administrator sistem dalam mengelola web server berbasis container.

TINJAUAN PUSTAKA

[7] Sebuah jurnal dengan judul pengujian dan implementasi efektivitas sistem virtualisasi berbasis docker platform, Penulisan ilmiah ini bermaksud menganalisis dan merancang sebuah platform virtualisasi berbasis Docker yang dipasang di operasi system Linux Ubuntu 17.10, meliputi proses instalasi, konfigurasi, dan menjalankan aplikasi Docker. Docker adalah sebuah projek terbuka yang berfungsi membantu cara kerja untuk developer atau sistem administrator untuk membangun, mengemas dan menjalankan aplikasi dimanapun dalam sebuah kontainer. arsitektur menggunakan metode klien server, Docker klien, mengirimkan permintaan kedocker daemon untuk membangun, mendistribusikan dan menjalankan kontainer Docker. Keduanya Docker klien dan Docker daemon berkomunikasi via socket menggunakan Restful API. Docker menyediakan alat dan platform untuk mengelola siklus hidup pada pada kontainer yaitu developer dan sistem administrator dapat mengembangkan aplikasi dan komponen pendukungnya menggunakan kontainer. Kontainer adalah sebuah wadah menjadi unit untuk mendistribusikan dan menjalankan uji coba aplikasi yang akan siap dipasang.

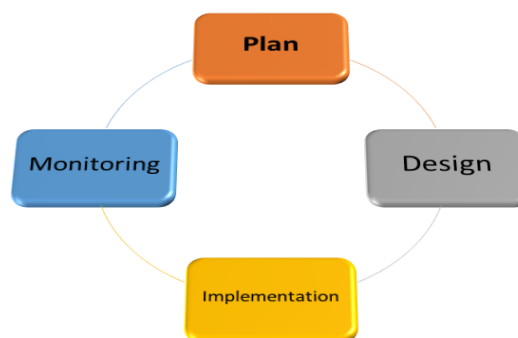
[8] Penerapan docker container sebagai teknologi ramah skalabilitas dibanding teknik virtualisasi untuk membangun website di ubuntu 18.04.4 lts Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan, dalam membangun website, Docker Container bekerja dengan cara mengikat service, library dan dependensinya bersamaan dengan aplikasi tersebut dalam lingkungan yang terisolasi. Sehingga penggunaan Docker Container ini tidak memerlukan pemasangan sistem operasi tambahan pada sistem operasi utama untuk mengisolasi aplikasi. Hal inilah yang menyebabkan proses developing aplikasi web menjadi ramah

skalabilitas, yaitu tidak boros dalam pemakaian sumber daya. Cara membangun website publik menggunakan teknologi Docker ini yaitu dengan memanfaatkan fitur dari Docker yang disebut Docker Compose. Dimana setiap service yang dibutuhkan dikonfigurasi dan dijalankan dengan Docker Compose tersebut didalam Virtual Privat Server (VPS). Penggunaan teknologi Docker Container jauh lebih unggul dibandingkan dengan penggunaan teknologi Virtualisasi berbasis Hypervisor. Dalam pengujian yang dilakukan pada penelitian ini diperoleh data bahwa Docker Container hanya memerlukan sumber daya CPU sekitar 4,0 – 12,7 % saja dari kapasitas penuh CPU, sedangkan Virtualisasi Hypervisor dengan Virtual Box memerlukan sumber daya CPU sekitar 16,2 – 29,4 %. Pada penggunaan sumber daya memori Docker hanya memerlukan 2,6 GiB saja dari total memori 5,7 GiB, sedangkan Virtual Box memerlukan memori sebesar 4,6 GiB. Begitu pula pada penggunaan swap Docker unggul karena hanya memerlukan swap sebesar 0,15 dibanding Virtual Box yang memerlukan swap 0,21

[9]Pemanfaatan Virtualisasi Di Ubuntu Untuk Optimalisasi Sumber Daya: Manfaat, Tantangan, Dan Solusi, Virtualisasi di Linux Ubuntu menawarkan berbagai keuntungan dan tantangan yang penting untuk dibahas guna memahami potensi penuh dari teknologi ini. Dalam penelitian ini, tiga solusi virtualisasi utama dievaluasi: KVM (Kernel-based Virtual Machine), VirtualBox, dan Docker. Setiap solusi memiliki karakteristik dan kasus penggunaan yang unik, yang mempengaruhi efektivitasnya dalam berbagai skenario. Virtualisasi di Linux Ubuntu menawarkan beragam solusi yang memungkinkan penggunaan sumber daya fisik secara lebih efisien dan fleksibel. Penelitian ini telah mengevaluasi tiga solusi utama, yaitu KVM (Kernel-based Virtual Machine), VirtualBox, dan Docker, yang masing-masing memiliki keunggulan dan tantangan tersendiri dalam implementasi dan penggunaan. Docker memperkenalkan pendekatan containerization yang ringan dan efisien, memungkinkan isolasi aplikasi yang baik dan deployment cepat, Keunggulan Docker termasuk waktu startup yang cepat dan portabilitas aplikasi yang tinggi. Namun, memerlukan manajemen jaringan dan orkestrasi yang lebih kompleks, terutama dalam lingkungan produksi skala besar.

METODE PENELITIAN

Pada tahap ini penulis menggunakan beberapa tahapan dalam penelitian yaitu, pertama [10]plan, merupakan rencana rancangan bangun infrastruktur web server berbasis docker yang akan di terapkan, kedua yaitu design merupakan langkah kedua dari metode penelitan ini yang bertujuan untuk melakukan design dari topology yang akan digunakan, Langkah ketiga yaitu implementation merupakan tahapan dengan topologi yang sudah ada maka selanjutnya dilakukan konfigurasi sesuai dengan kebutuhan pada rancang bangun, Langkah keempat yaitu monitoring merupakan tahapan setelah dilakukan implementasi konfigurasi kemudian dilakukan testing dan dilihat hasil nya apakah sudah sesuai dengan yang diharapkan, atau masih ada perbaikan supaya sesuai dengan harapan.



Gambar 1. Metode Penelitian

HASIL DAN PEMBAHASAN

Lakukan instalasi ubuntu server 22.04, setelah selesai instalasi ubuntu server 22.04 lakukan update sistem operasi ubuntu server, seperti dibawah ini.

```
root@admin123:~# apt-get update
```

Gambar 2. Update ubuntu server

Lakukan Instalasi paket prasyarat

```
root@admin123:~# sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Gambar 3. Instalasi paket prasyarat

Penambahan PGP

```
root@admin123:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK
```

Gambar 4. Penambahan PGP

Lakukan penambahan Repository docker untuk persiapan instalasi docker

```
root@admin123:~# sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"  
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable'  
Description:  
Archive for codename: focal components: stable  
More info: https://download.docker.com/linux/ubuntu  
Adding repository.  
Press [ENTER] to continue or Ctrl-c to cancel.  
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list  
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list  
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Get:4 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]  
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [53.3 kB]  
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Fetched 111 kB in 6s (19.0 kB/s)  
Reading package lists... Done
```

Gambar 5. Penambahan Repository Docker

Lakukan Update packet repository untuk memastikan bahwa penambahan repository berhasil

```
root@admin123:~# apt-get update  
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 https://download.docker.com/linux/ubuntu focal InRelease  
Hit:4 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done
```

Gambar 6. Update Repository

Lakukan pengecekan versi docker yang akan diinstall

```
root@admin123:~# apt-cache policy docker-ce  
docker-ce:  
  Installed: (none)  
  Candidate: 5:27.4.1-1~ubuntu.20.04~focal  
  Version table:  
 * 5:27.4.1-1~ubuntu.20.04~focal 500  
   500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages  
 5:27.4.0-1~ubuntu.20.04~focal 500  
   500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages  
 5:27.3.1-1~ubuntu.20.04~focal 500  
   500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

Gambar 7. Pengecekan versi docker

Lakukan Installasi docker pada ubuntu server 22.04

```

root@admin123:~# sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libsllp0 pigz
  sllrp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libsllp0
  pigz sllrp4netns
0 upgraded, 10 newly installed, 0 to remove and 124 not upgraded.
Need to get 124 MB of archives.
After this operation, 446 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://id.archive.ubuntu.com/ubuntu/jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.7.24-1 [29.5 MB]
Get:3 http://id.archive.ubuntu.com/ubuntu/jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu/jammy/main amd64 libsllp0 amd64 4.6.1-1build1 [61.5 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu/jammy/universe amd64 sllrp4netns amd64 1.0.1-2 [28.2 kB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd64 0.19.3-1-ubuntu.20.04-focal [30.7 MB]
  
```

Gambar 8. Instalasi docker

Lakukan pengecekan status docker

```

root@admin123:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-12-25 09:32:13 WIB; 3min 10s ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 10876 (dockerd)
     Tasks: 9
    Memory: 22.5M
       CPU: 702ms
    CGroup: /system.slice/docker.service
            └─10876 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 25 09:32:11 admin123 systemd[1]: Starting Docker Application Container Engine...
Dec 25 09:32:12 admin123 dockerd[10876]: time="2024-12-25T09:32:12.033882800+07:00" level=info msg="Starting up"
Dec 25 09:32:12 admin123 dockerd[10876]: time="2024-12-25T09:32:12.036328104+07:00" level=info msg="OTEL tracing is not configured,
Dec 25 09:32:12 admin123 dockerd[10876]: time="2024-12-25T09:32:12.036597376+07:00" level=info msg="detected 127.0.0.53 nameserver,
Dec 25 09:32:12 admin123 dockerd[10876]: time="2024-12-25T09:32:12.255378973+07:00" level=info msg="Loading containers: start."
Dec 25 09:32:13 admin123 dockerd[10876]: time="2024-12-25T09:32:13.206984769+07:00" level=info msg="Loading containers: done."
Dec 25 09:32:13 admin123 dockerd[10876]: time="2024-12-25T09:32:13.265762219+07:00" level=info msg="Docker daemon" commit=c710b88 co
Dec 25 09:32:13 admin123 dockerd[10876]: time="2024-12-25T09:32:13.266302030+07:00" level=info msg="Daemon has completed initializat
Dec 25 09:32:13 admin123 dockerd[10876]: time="2024-12-25T09:32:13.335921248+07:00" level=info msg="API listen on /run/docker.sock"
Dec 25 09:32:13 admin123 systemd[1]: Started Docker Application Container Engine.
lines 1-22/22 (END)
root@admin123:~#
  
```

Gambar 9. Pengecekan status docker

Lakukan pengecekan docker version pada ubuntu server yang sudah terinstall

```

root@admin123:~# docker version
Client: Docker Engine - Community
 Version: 27.4.1
  API version: 1.47
  Go version: go1.22.10
  Git commit: b9d17ea
   Built: Tue Dec 17 15:45:52 2024
  OS/Arch: linux/amd64
  Context: default

Server: Docker Engine - Community
 Engine:
  Version: 27.4.1
  API version: 1.47 (minimum version 1.24)
  Go version: go1.22.10
  Git commit: c710b88
   Built: Tue Dec 17 15:45:52 2024
  OS/Arch: linux/amd64
  Experimental: false
 containerd:
  Version: 1.7.24
  GitCommit: 88bf19b2105c8b17560993bee28a01ddc2f97182
  runc:
  Version: 1.2.2
  GitCommit: v1.2.2-0-g7cb3632
 docker-init:
  Version: 0.19.0
  GitCommit: de40ad0
root@admin123:~#
  
```

Gambar 10. Versi docker yang sudah terinstall

Lakukan download image docker nginx pada hub docker

```
root@admin123:~# docker pull nginx
```

Gambar 11. Pull nginx

Setelah download selesai lakukan pengecekan image yang sudah ready pada ubuntu server

```
root@admin123:~# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest f876bfc1cc63 4 weeks ago 192MB
```

Gambar 12. Cek docker image

Lakukan run docker nginx dengan nama webserver2

```
root@admin123:~# docker run -d -p 82:80 --name webserver2 nginx
2152a20591396fc6abf6837e53354a68242c597590be7614d3c298d7b029cf86
```

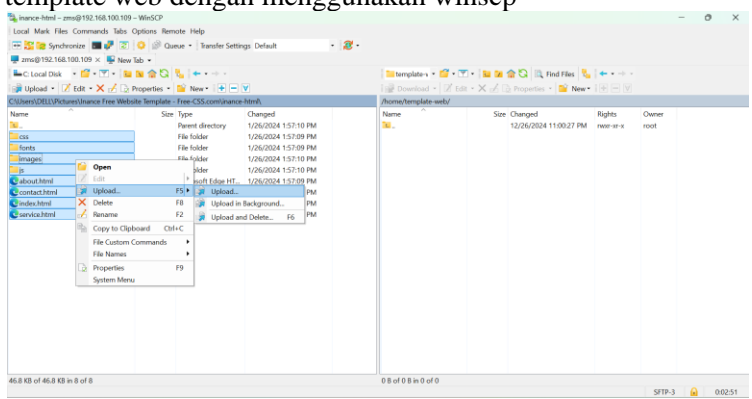
Gambar 13. Run docker

Pengecekan docker yang sedang aktif

```
root@admin123:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2152a2059139 nginx "/docker-entrypoint..." 4 seconds ago Up 4 seconds 0.0.0.0:82->80/tcp, [::]:82->80/tcp webserver2
```

Gambar 14. Docker yang sedang berjalan

Siapkan template web yang akan diupload ke web server dalam sebuah folder, kemudian lakukan Upload template web dengan menggunakan winscp



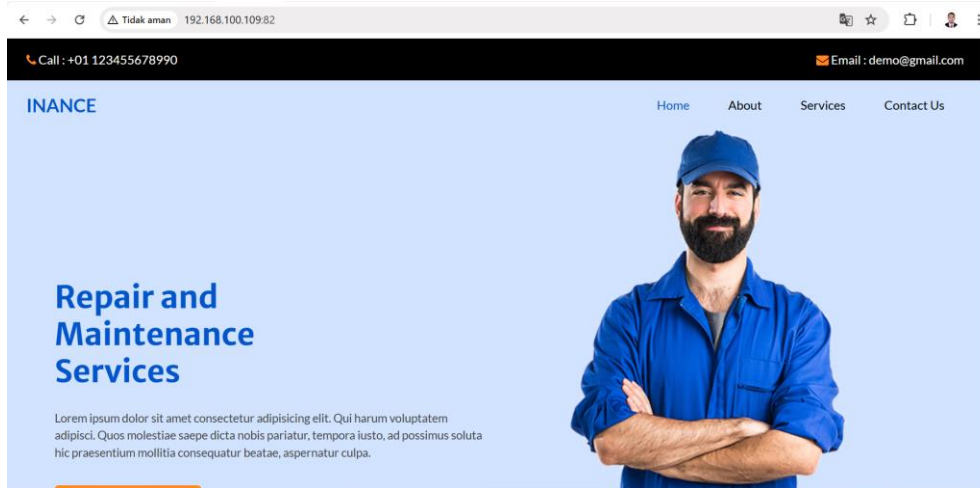
Gambar 16. Upload template web

Lakukan Copy file template web dari ubuntu server ke docker nginx, disini dengan web server nginx menggunakan nama webserver2

```
root@admin123:~# docker cp /home/template-web/css/ webserver2:usr/share/nginx/html
Successfully copied 274kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/fonts/ webserver2:usr/share/nginx/html
Successfully copied 345kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/images/ webserver2:usr/share/nginx/html
Successfully copied 895kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/js/ webserver2:usr/share/nginx/html
Successfully copied 225kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/index.html webserver2:usr/share/nginx/html
Successfully copied 27.6kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/about.html webserver2:usr/share/nginx/html
Successfully copied 8.7kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/contact.html webserver2:usr/share/nginx/html
Successfully copied 8.7kB to webserver2:usr/share/nginx/html
root@admin123:~# docker cp /home/template-web/service.html webserver2:usr/share/nginx/html
Successfully copied 9.73kB to webserver2:usr/share/nginx/html
root@admin123:~#
```

Gambar 17. Copy file template web ke docker

Pengujian akses web server dari client



Berikut hasil pengujian untuk akses web server dari client

Tabel 1. Hasil pengujian

No	Nama Client	Ip Address	Hasil Pengujian
1	laptop A	192.168.100.92	Berhasil
2	smartphone	192.168.100.130	Berhasil
3	laptop B	192.168.100.107	Berhasil
4	laptop C	192.168.100.135	Berhasil
5	laptop D	192.168.100.110	Berhasil

Dari hasil pengujian terlihat bahwa semua client dapat mengakses ke web server nginx

Berikut hasil pengujian koneksi dari client ke web server

Tabel 2. Tabel Hasil Pengujian Koneksi

No	Nama Client	Koneksi Pertama	Koneksi Kedua	Koneksi Ketiga	Koneksi Keempat
1	laptop A	2ms	1ms	1ms	<1ms
2	smartphone	16.2ms	32.8ms	54.4ms	82.5ms
3	laptop B	37ms	22ms	6ms	6ms
4	laptop C	1ms	<1ms	1ms	<1ms
5	laptop D	1ms	<1ms	5ms	9ms

Dari hasil pengujian tes koneksi dari lima pengguna hanya pengguna dari smartphone yang mendapatkan akses lambat dengan maksimal akses 82,5ms dan akses paling cepat didapatkan kurang dari 1ms, sehingga perlu dilakukan pengujian lebih banyak lagi untuk mendapatkan hasil pengujian yang optimal.

PENUTUP

Simpulan

Dari hasil pengujian akses web server semua pengguna dapat mengakses ke web server dan dari hasil pengujian tes koneksi dari lima pengguna hanya pengguna dari hp samsung yang mendapatkan akses lambat dengan maksimal akses 82,5ms dan akses paling cepat didapatkan kurang dari 1ms. Dengan ada nya docker container didalam satu sistem operasi dapat memudahkan dalam melakukan virtualisasi dalam sistem operasi ubuntu server.

Saran

Perlu dilakukan pengujian lebih banyak lagi untuk mendapatkan hasil pengujian yang optimal, dan perlu dibangun lebih banyak web server untuk penerapan load balancing sehingga dapat mengoptimalkan kinerja dari pengguna ke akses web server.

REFERENSI

- [1] S. Dwiyatno, E. Rakhmat, and O. Gustiawan, "IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER," vol. 7, no. 2, 2020.
- [2] A. H. Dame, A. U. Zailani, K. Kunci, : Implementasi, and W. Server, "Implementasi Webserver Berbasis Docker Dan Linux," *J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 5, pp. 1084–1090, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [3] M. Ubuntu, S. Pada, and P. T. Xyz, "PERANCANGAN INFRASTRUKTUR DOMAIN NAME SERVER LOKAL," no. 2, pp. 23–29, 2020.
- [4] I. Ihsan, Z. Zulkarnain, and A. Y. Amsal, "DESAIN DAN PERANCANGAN VPS BERBASIS UBUNTU 22.04 UNTUK IMPLEMENTASI WEBSITE SEKOLAH DI SMK ADZKIYA BALIKPAPAN," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, 2023, doi: 10.30656/prosisko.v10i2.7103.
- [5] Z. M. Subekti, "RANCANG BANGUN SISTEM UJIAN ONLINE MENGGUNAKAN," no. 2, pp. 163–168, 2021.
- [6] A. Sinambela, E. Ernawati, and F. F. Coastera, "Implementasi Arsitektur Microservices Pada Rancang Bangun Aplikasi Marketplace Berbasis Web (Studi Kasus : Pasar Tradisional Modern Kota Bengkulu)," *Rekursif J. Inform.*, 2021, doi: 10.33369/rekursif.v9i1.14929.
- [7] H. Mubarak and H. Saptono, "Jurnal Informatika Terpadu BERBASIS DOCKER PLATFORM," *J. Inform. Terpadu*, vol. 5, no. 1, pp. 18–23, 2019, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JIT>
- [8] F. Ariadi, C. Iswahyudi, E. Nurnawati, J. Informatika, and I. Akprind, "PENERAPAN DOCKER CONTAINER SEBAGAI TEKNOLOGI RAMAH SKALABILITAS DIBANDING TEKNIK VIRTUALISASI UNTUK MEMBANGUN WEBSITE DI UBUNTU 18.04.4 LTS," vol. 8, no. 2, 2020.
- [9] M. H. Saputra, Ferdiayansah, and F. Dristyan, "Pemanfaatan Virtualisasi Di Ubuntu Untuk Optimalisasi Sumber Daya: Manfaat, Tantangan, Dan Solusi," *J. Comput. Sci. Technol.*, vol. 3, no. 1, pp. 49–54, Jan. 2025, doi: 10.59435/jocstec.v3i1.431.
- [10] Z. Mutaqin Subekti, K. Mukiman, A. Fikri Adluwal Fadhil, and M. Asyrofi, "Penerapan Limit Akses Browsing Internet pada saat Jam Kerja di PT XYZ," *J. Teknol. Terpadu*, vol. 7, no. 1, pp. 31–38, 2021, doi: 10.54914/jtt.v7i1.342.