

PENERAPAN DOCKER UNTUK MEMBANGUN INFRASTRUKTUR PRIVATE CLOUD STORAGE BERBASIS IAAS

Muhammad Rizki Alamsyah¹, Henni Indah Wahanani², Mohammad Idhom³

E-mail : ¹rizky.vertigo@gmail.com , ²henniindah.if@upnjatim.ac.id , ³idhom@upnjatim.ac.id

^{1,2,3}Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur

Abstrak

Internet telah mengalami perluasan dan perkembangan pada bidangnya masing-masing. Tidak hanya sebagai media pengiriman data dan informasi, internet dapat memberikan layanan yang memudahkan masyarakat dalam melakukan aktivitasnya sehari-hari, dan salah satu layanan tersebut adalah cloud computing. Cloud Computing adalah sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan informasi. Cloud Storage termasuk layanan yang paling penting dalam cloud computing karena berguna untuk menyimpan data digital dengan memanfaatkan adanya virtual server sebagai media penyimpanannya. Penelitian ini menerapkan Docker pada infrastruktur private cloud storage yang dibangun. Docker adalah sebuah platform yang dibangun berdasarkan teknologi container yang bersifat open source untuk membangun, mengemas dan menjalankan aplikasi dimanapun dalam sebuah container dan memiliki kelebihan hanya membutuhkan lebih sedikit ruang daripada virtualisasi mesin. Container sendiri adalah sebuah teknologi pada layer aplikasi yang mengemas kode beserta dependensinya menjadi satu. Dari penelitian yang dilakukan, didapatkan hasil penggunaan CPU sebesar 9,4% dan RAM mencapai 1.25GB pada percobaan pertama, CPU sebesar 10,8% dan RAM mencapai 1.32GB pada percobaan kedua, CPU sebesar 13,7% dan RAM mencapai 1.18GB pada percobaan ketiga.

Kata kunci: *Docker, Cloud Computing, Cloud Storage, IaaS, Containerization*

1. PENDAHULUAN

Internet telah mengalami perluasan dan menjadi semakin berkembang sesuai dengan bidangnya masing-masing. Tidak hanya sebagai media pengiriman data dan informasi, kini internet juga dapat memberikan layanan yang dapat memberikan kemudahan masyarakat dalam melakukan aktivitasnya sehari-hari. Salah satu dari layanan tersebut yaitu cloud computing atau komputasi berbasis awan[1].

Cloud computing atau yang biasa disebut komputasi awan sendiri adalah sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan informasi. Data dan informasi tersebut dapat diakses secara bersamaan oleh komputer yang telah terkoneksi satu sama lain[2]. Cloud computing telah menarik banyak peneliti, karena memiliki karakteristik berbagi sumber daya dan memiliki konsumsi energi yang rendah. Tampaknya, cloud computing tak hanya dapat menyediakan sumber daya komputasi yang tak terbatas bagi pengguna, namun juga dapat menyediakan sumber daya penyimpanan yang tak terbatas bagi pengguna. Cloud storage adalah salah satu layanan yang paling penting dalam cloud computing karena memungkinkan interkoneksi semua perangkat elektronik[3].

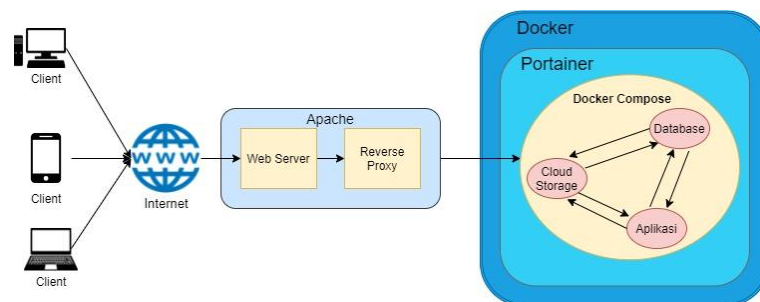
Cloud storage adalah teknologi yang berguna untuk menyimpan data digital dengan memanfaatkan adanya virtual server sebagai media penyimpanan berbasis cloud computing. Cloud storage termasuk kedalam model IaaS (Infrastructure as a Service). Cloud storage adalah layanan penyimpanan dan menggunakan database secara online, dimana data dikelola dan disimpan secara virtual yang dapat diakses melalui jaringan[4].

Sehingga file yang telah disimpan pada Cloud storage dapat diakses bersamaan secara real time.

Docker adalah salah satu platform bersifat open source yang dibangun berdasarkan teknologi container dan ditujukan untuk developer atau sysadmin untuk membangun, mengemas, dan menjalankan aplikasi dimana pun dalam sebuah container. Container adalah sebuah teknologi pada layer aplikasi yang mengemas kode beserta dependensinya menjadi satu. Beberapa container dapat berjalan di mesin yang sama dan berbagi Kernel OS dengan container lain, masing-masing container berjalan sebagai proses yang terisolasi pada user space. Container sendiri memiliki kelebihan hanya membutuhkan lebih sedikit ruang daripada Virtualisasi Mesin, karena container menggunakan citra yang besarnya hanya berukuran puluhan MB. Container juga dapat menangani lebih banyak aplikasi dan memerlukan lebih sedikit kinerja dari Virtualisasi Mesin dan Sistem Operasi[5].

Berdasarkan penjelasan di atas, maka penulis mengambil judul skripsi “Rancang Bangun Infrastruktur Private cloud storage Berbasis IaaS (Infrastructure as a Service) dengan maksud membangun rancangan infrastruktur cloud storage dengan menggunakan docker container untuk ` 3 mengurangi resource yang dibutuhkan karena dapat dijalankan pada beberapa platform cloud sehingga lebih fleksibel dan memungkinkan user untuk memporting aplikasi antar lingkungan secara mudah[6].

2. METODOLOGI



Gambar 1. Topologi

Pada gambar 1 menunjukkan topologi dari sistem yang digunakan dalam penelitian ini. Dari topologi diatas dapat diartikan bahwa cloud storage, database, dan aplikasi akan dijalankan dalam sebuah container dimana container tersebut dibuat dengan menggunakan Docker Compose dalam sebuah file berformat .yaml. Docker Compose akan membuat container tersebut beserta image didalamnya yang kemudian juga memberikan network untuk container tersebut. Kemudian web server beserta konfigurasi reverse proxy dari container itu sendiri dapat dikonfigurasi melalui superuser admin dari sistem webserver telah di install pada sistem operasi ubuntu. Portainer berguna sebagai User Interface dalam manajemen container docker. Alur dari topologi tersebut adalah jika client ingin mengakses cloud storage yang ada dalam container tersebut, client harus tersambung ke internet terlebih dahulu, kemudian reverse proxy akan mengarahkan ke alamat IP dari container yang dituju.

2.1 Studi Literatur

Tahapan ini dilakukan untuk mengumpulkan beberapa literatur maupun dokumentasi terkait dengan perancangan infrastruktur private cloud storage berbasis IaaS dengan menggunakan docker container sebagai basis pengembangannya, mulai dari jurnal, situs penyedia dokumentasi, serta buku pegangan.

2.2 Analisis Kebutuhan

Kebutuhan sistem sangat penting untuk menunjang kelancaran penelitian, kebutuhan sistem ini sama halnya dengan kebutuhan sumber daya, baik sumber daya perangkat keras maupun sumber daya perangkat lunak. Berikut ini beberapa kebutuhan sistem yang digunakan dalam uji coba:

1. Server

Kebutuhan server fisik dan web server lainnya pada penelitian ini dalam hal implementasi dan pengujian dijalankan dengan menggunakan VPS (Virtual Private Server). Berikut ini adalah spesifikasi dari VPS (Virtual Private Server yang digunakan dalam penelitian:

Tabel 1. Tabel Spesifikasi Virtual Private Server

| CPU | Memory | Storage | Bandwith | OS |
|-----------------------|--------|-----------|----------|------------------|
| 1 vCore Intel Skylake | 2 GB | 55 GB SSD | 2000 GB | Ubuntu 18.04 x64 |

2. Docker

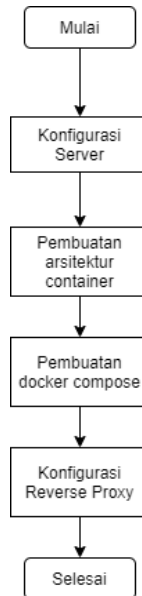
Docker digunakan sebagai platform dalam implementasi pembuatan infrastruktur pada penelitian ini. Docker dipilih karena menggunakan teknologi container yang dapat menghemat penggunaan sumber daya dan konfigurasinya yang lebih fleksibel. Adapun spesifikasi dari sistem container yang digunakan sebagai berikut:

Tabel 2. Tabel Daftar Container

| Image | Port |
|-------------------------|------|
| mariadb:10.5.4 | 3306 |
| nextcloud:19.0.1-apache | 8080 |
| phpmyadmin/phpmyadmin | 8000 |
| collabora/code:4.2.6.2 | 9980 |
| portainer/portainer | 9000 |

Pada tabel 2 menunjukkan image yang digunakan untuk pembuatan sistem, image tersebut akan digunakan sebagai monitoring sumber daya, monitoring dari container yang sedang berjalan, dan sebagai cloud storage beserta dengan office yang berfungsi untuk dapat mengakses file yang ada secara bersamaan. Image tersebut dipasang dan dijalankan pada port yang telah disebutkan.

2.3 Pembuatan Sistem



Gambar 2. Alur pembuatan sistem

Alur pada gambar 2 memperlihatkan bahwa pembuatan sistem memiliki beberapa proses yaitu:

a. Konfigurasi Server

Proses ini dilakukan untuk mempersiapkan server agar dapat digunakan sesuai dengan spesifikasi yang dibutuhkan. Pada server menggunakan sistem operasi Ubuntu 18.04, kemudian melakukan koneksi secured shell connection atau SSH agar dapat melakukan koneksi aman untuk masuk kedalam server. Setelah berhasil melakukan koneksi SSH pada server kemudian mengkonfigurasi domain agar dapat diakses secara global. Setelah domain berhasil dikonfigurasi kemudian adalah memberikan sertifikat atau SSL (Secured Sockets Layer) pada domain yang telah dikonfigurasi agar transfer data yang terjadi dalam website dapat menjadi lebih aman dan terenkripsi.

b. Pembuatan Arsitektur Container

Proses ini dilakukan untuk merancang dan membuat container beserta resourcenya (image, network, volume, port, dan environment) yang dibutuhkan dalam penelitian ini.

c. Pembuatan Docker Compose

Proses ini dilakukan setelah pembuatan arsitektur container yang berguna untuk membuat dan menjalankan lebih dari satu container atau multicontainer secara bersamaan. Pada docker compose dilakukan pembuatan beberapa container yaitu web server, cloud storage, database, office, beserta dengan networknya agar semua container dalam satu file docker compose ini berada dalam satu jaringan dan dapat terhubung satu sama lain.

d. Konfigurasi Reverse Proxy

Proses ini dilakukan untuk menkonfigurasi web server, SSL, dan reverse proxy dari arsitektur docker yang telah dibuat agar client dapat mengakses domain secara langsung dan diteruskan ke port yang dituju tanpa harus menulis ip beserta port secara manual. Proses ini juga berfungsi untuk menyediakan kontrol dan keamanan untuk memastikan kelancaran arus lalu lintas jaringan diantara klien dan server.

2.4 Pengujian

Pengujian ini dilakukan untuk menguji kecepatan laju throughput dari private cloud storage yang telah dibangun dan menguji penggunaan sumber daya yang digunakan saat proses pengujian berlangsung. Pengujian dilakukan sebanyak tiga skenario dengan parameter pengujian pada tabel 3 sebagai berikut:

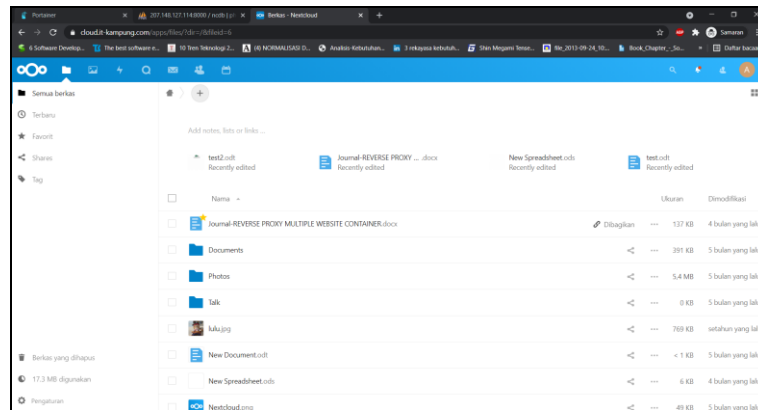
Tabel 3. Tabel Parameter Pengujian

| Pengujian | Parameter | Jumlah Thread | Jenis File | Ukuran |
|-----------|-----------|---------------|------------|--------|
| 1 | HTTPS | 300 | JPG Image | 1 MB |
| 2 | HTTPS | 300 | JPG Image | 2 MB |
| 3 | HTTPS | 300 | JPG Image | 3 MB |

3. HASIL DAN PEMBAHASAN

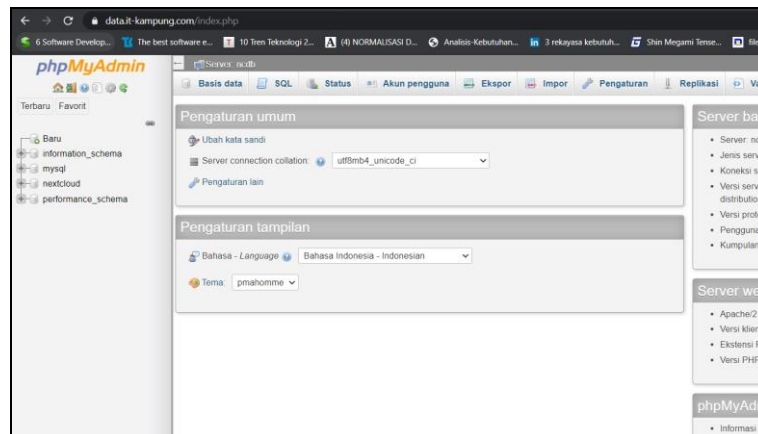
Hasil dari penerapan docker dalam membangun infrastruktur private cloud storage berbasis IaaS dapat dilihat pada ulasan berikut

3.1 Hasil Antarmuka



Gambar 3. Tampilan container private cloud storage

Pada gambar 3 berikut dapat dilihat tampilan dari menu private cloud storage setelah berhasil login. Container dari private cloud storage tersebut telah diberikan domain dan reverse proxy sehingga container dari private cloud storage tersebut dapat diakses secara langsung menggunakan domain, dimana akan langsung diarahkan ke alamat IP beserta port yang telah didefinisikan saat membangun containernya saat domain diakses. Container ini berfungsi sebagai private cloud storage.

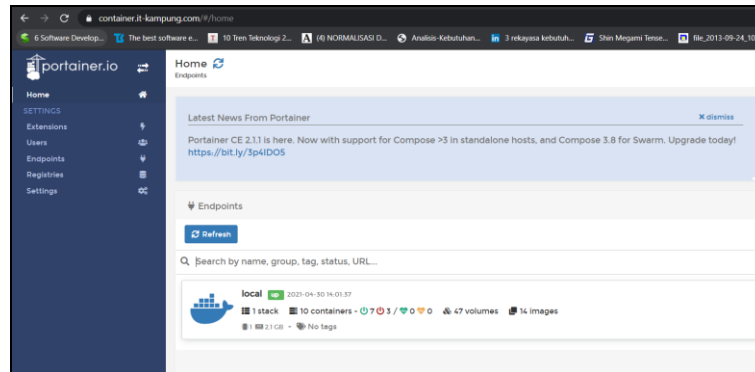


Gambar 4. Tampilan container phpmyadmin

Pada gambar 4 berikut dapat dilihat tampilan dari menu phpmyadmin setelah berhasil login. Container dari phpmyadmin tersebut telah diberikan domain dan reverse proxy sehingga container dari phpmyadmin tersebut dapat diakses secara langsung menggunakan domain, dimana akan langsung diarahkan ke alamat IP beserta port yang

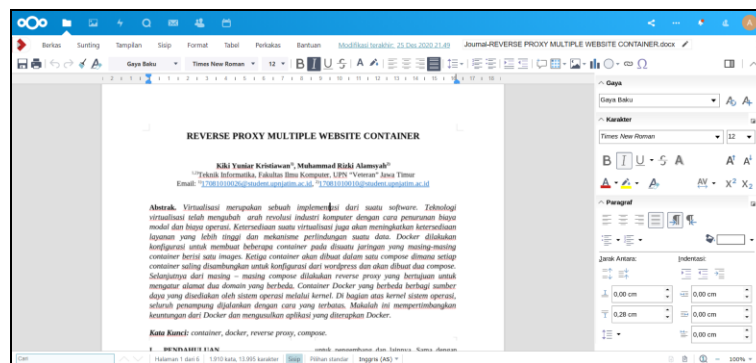
PENERAPAN DOCKER UNTUK MEMBANGUN INFRASTRUKTUR PRIVATE CLOUD STORAGE BERBASIS IAAS

telah didefinisikan saat membangun containernya saat domain diakses. Container ini berfungsi untuk mengatur database dari private cloud storage.



Gambar 5. Tampilan container Portainer

Pada gambar 5 berikut dapat dilihat tampilan dari menu portainer setelah berhasil login. Container dari portainer tersebut telah diberikan domain dan reverse proxy sehingga container dari phpmyadmin tersebut dapat diakses secara langsung menggunakan domain, dimana akan langsung diarahkan ke alamat IP beserta port yang telah didefinisikan saat membangun containernya saat domain diakses. Container ini berfungsi untuk mengatur dan manajemen container yang sedang berjalan maupun tidak.



Gambar 6. Tampilan container collabora office

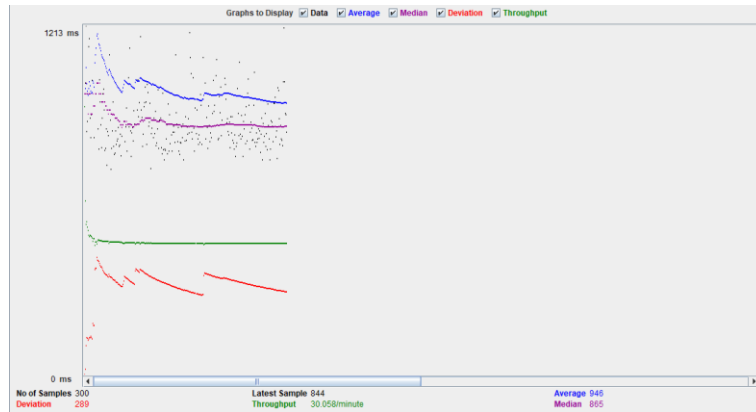
Pada gambar 6 berikut dapat dilihat tampilan dari collabora office pada private cloud storage. Container dari collabora office tersebut telah diberikan domain dan reverse proxy sehingga container dari phpmyadmin tersebut dapat diakses secara langsung menggunakan domain, dimana akan langsung diarahkan ke alamat IP beserta port yang telah didefinisikan saat membangun containernya saat domain diakses. Container ini berfungsi untuk mengakses file dan sebagai text editor secara langsung yang terdapat pada private cloud storage.

3.2 Pengujian

Pengujian ini dilakukan untuk menguji kecepatan laju throughput dari private cloud storage yang telah dibangun dan menguji penggunaan sumber daya yang digunakan saat proses pengujian berlangsung.

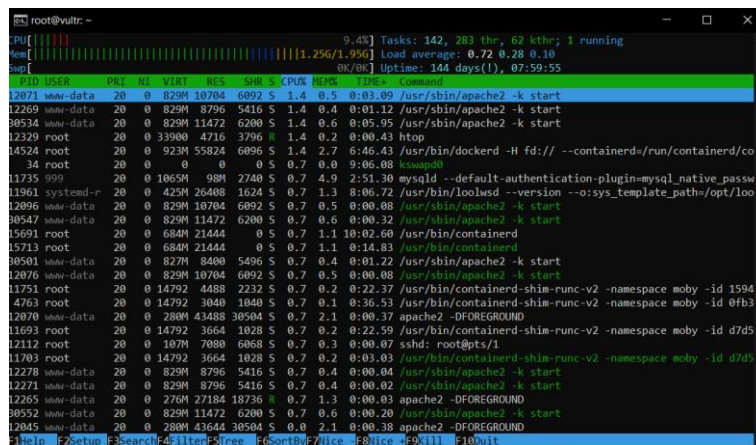
1. Skenario Pertama

Skenario pertama dilakukan dengan upload file sebesar 1 MB dengan jumlah thread 300, dimana pengujian ini mensimulasikan sebanyak 300 client upload file dengan ukuran 1 MB secara bersamaan. Hasil yang didapatkan pada skenario pertama adalah sebagai berikut:



Gambar 7. Laju throughput skenario pertama

Dalam hasil uji coba skenario pertama seperti gambar 7, Apache JMeter mencatat throughput upload data sebesar 30.058/menit, yang berarti server dapat menangani 30.058 permintaan per-menitnya. Sedangkan penyimpangan yang terjadi selama proses upload file adalah 289. Throughput adalah parameter yang mewakili kemampuan server dalam menangani beban berat, yang berarti semakin tinggi nilai throughput, semakin bagus kinerja dari server. Standar penyimpangan (warna merah) semakin kecil nilainya, maka semakin baik.



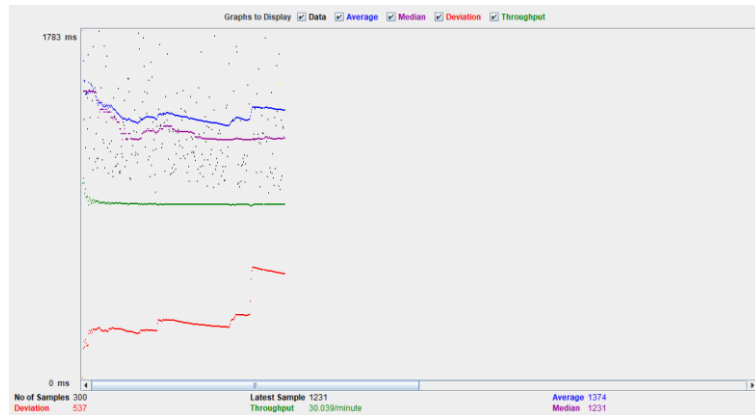
Gambar 8. Penggunaan sumber daya skenario pertama

Hasil yang didapatkan pada gambar 8 adalah kinerja RAM dan CPU selama proses upload data skenario pertama. Dari gambar diatas, dapat dijelaskan bahwa penggunaan CPU mencapai 9,4% , sedangkan penggunaan RAM mencapai 1.25GB dari total RAM 2GB.

2. Skenario kedua

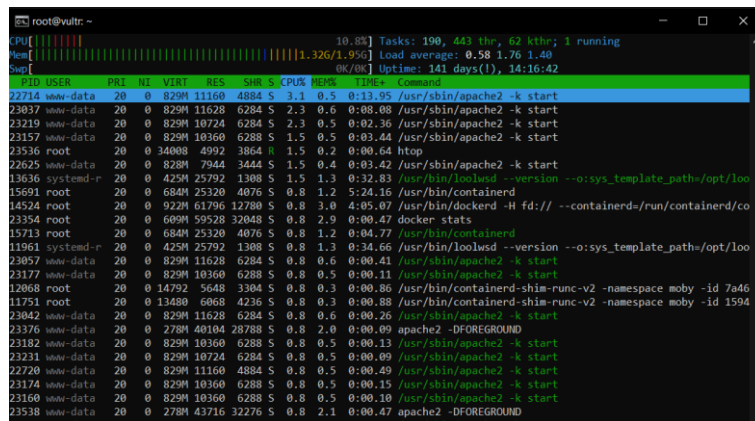
Skenario kedua dilakukan dengan upload file sebesar 2 MB dengan jumlah thread 300, dimana pengujian ini mensimulasikan sebanyak 300 client upload file dengan ukuran 2 MB secara bersamaan. Hasil yang didapatkan pada skenario pertama adalah sebagai berikut:

PENERAPAN DOCKER UNTUK MEMBANGUN INFRASTRUKTUR PRIVATE CLOUD STORAGE BERBASIS IAAS



Gambar 9. Laju throughput skenario kedua

Hasil pengujian skenario kedua dengan menggunakan Apache JMeter pada gambar 9, menjelaskan bahwa throughput yang didapatkan adalah 30.039/menit, yang berarti server pada pengujian skenario kedua dapat menangani 30.039 permintaan per-menitnya. Sedangkan penyimpangan yang terjadi selama proses adalah 537.



Gambar 10. Penggunaan sumber daya skenario kedua

Hasil yang didapatkan pada gambar 10 adalah kinerja RAM dan CPU selama proses upload data skenario kedua. Dari gambar diatas, dapat dijelaskan bahwa penggunaan CPU mencapai 10.8% , sedangkan penggunaan RAM mencapai 1.32GB dari total RAM 2GB.

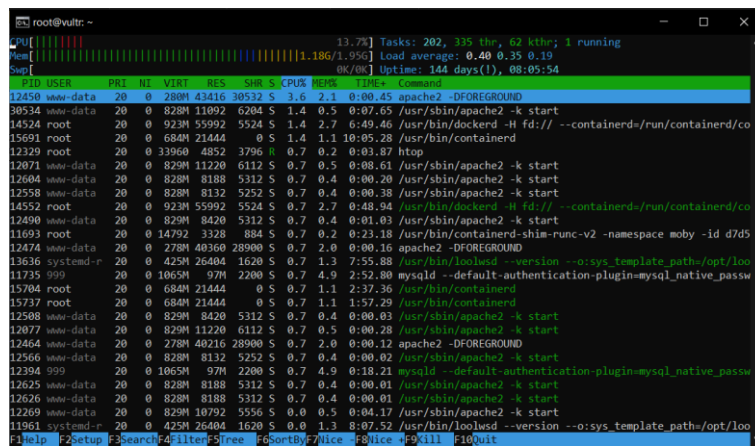
3. Skenario ketiga

Skenario ketiga dilakukan dengan upload file sebesar 3 MB dengan jumlah thread 300, dimana pengujian ini mensimulasikan sebanyak 300 client upload file dengan ukuran 3 MB secara bersamaan. Hasil yang didapatkan pada skenario pertama adalah sebagai berikut:



Gambar 11. Laju throughput skenario ketiga

Hasil pengujian skenario kedua dengan menggunakan Apache JMeter pada gambar 11, menjelaskan bahwa throughput yang didapatkan adalah 30.028/menit, yang berarti server pada pengujian skenario kedua dapat menangani 30.028 permintaan per-menitnya. Sedangkan penyimpangan yang terjadi selama proses adalah 587



Gambar 12. Penggunaan sumber daya skenario ketiga

Hasil yang didapatkan pada gambar 12 adalah kinerja RAM dan CPU selama proses upload data skenario ketiga. Dari gambar diatas, dapat dijelaskan bahwa penggunaan CPU mencapai 13.7% , sedangkan penggunaan RAM mencapai 1.18GB dari total RAM 2GB.

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Berdasarkan hasil dari percobaan yang dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Docker container dapat digunakan sebagai tempat untuk mengemas konfigurasi, kode, beserta resource yang dibutuhkan oleh sebuah aplikasi dan dapat dijalankan dimana saja.
2. Hasil pengujian kecepatan laju throughput dan deviasi menggunakan Apache JMeter menunjukkan bahwa kecepatan laju throughput dan deviasi bergantung kepada besar file yang diupload.
3. Hasil perbandingan pengujian laju throughput dan deviasi pada skenario pertama, kedua, dan ketiga adalah skenario pertama dapat menangani 30.058 permintaan, skenario kedua dapat menangani 30.039 permintaan dan skenario ketiga dapat menangani 30.028 permintaan.

4. Hasil perbandingan pengujian laju throughput dan deviasi pada skenario pertama, kedua, dan ketiga adalah skenario pertama dapat menangani 30.058 permintaan, skenario kedua dapat menangani 30.039 permintaan dan skenario ketiga dapat menangani 30.028 permintaan. Hasil pengujian sumber daya perangkat keras yang dilakukan dengan menggunakan Htop dan Docker Stats menunjukkan bahwa semakin besar ukuran file yang diupload, maka semakin besar juga sumber daya perangkat keras yang digunakan.

4.2. Saran

Adapun saran yang dapat diberikan dari penelitian mengenai penerapan docker untuk membangun infrastruktur private cloud storage berbasis IaaS ini adalah:

1. Gunakan banyak kontainer untuk membangun infrastruktur private cloud storage, ini dimaksudkan untuk membagi kinerja resource yang ada.
2. Gunakan docker compose untuk mempermudah membangun banyak container pada infrastruktur private cloud storage.

5. DAFTAR RUJUKAN

- [1] IdCloudhost, "Mengenal apa itu cloud computing," 2019. [Online]. Available: <https://idcloudhost.com/mengenal-apa-itu-cloud-computing-defenisi-fungsi-dan-cara-kerja/>. [Accessed 27 November 2020].
- [2] J. Shen, T. Zhou, X. Chen, J. Li and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," vol. 13, no. 4, pp. 912-925, 2018.
- [3] D. Hernowo, R. Y. Rachmawati and U. Lestari, "Penerapan Private Cloud Storage Sebagai Media Sharing Dan Backup Data di PT. Telkom Indonesia Kandatel Kisanan," vol. 4, no. 2, pp. 9-19, 2016.
- [4] H. M. Abdullah and H. Nugroho, "Rancang Bangun Cloud Storage Untuk Coworking Space Berbasis Raspberry PI," pp. 75-87, 2018.
- [5] D. D. Team, "Docker Overview," 2018. [Online]. Available: <https://docs.docker.com/>. [Accessed 26 April 2021].
- [6] C. Janto, "Apa itu docker dan mengapa docker sangat populer," 2020. [Online]. Available: <https://exabytes.co.id/blog/apa-itu-docker-dan-mengapa-docker-sangat-populer/>. [Accessed 27 Mei 2021].