

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang digunakan untuk melakukan penelitian ini meliputi beberapa penelitian yang sudah dilakukan diantaranya.

Tanjung P, Dr. Ir. Rendy M, dan Danu DS dengan judul Implementasi dan Analisis Computer Clustering System dengan Menggunakan Virtualisasi Docker (2017). Menjelaskan bagaimana cara melakukan implementasi layanan clustering server dengan menggunakan *Docker*. Parameter-parameter yang diuji adalah *latency*, *CPU utilization*, dan *throughput*. Pengujian ini dilakukan pada server yang memiliki spesifikasi 2 *core* CPU dan 4 GB RAM, yang pada tulisan ini disebut server tester. Pengujian ini dibantu oleh perangkat lunak monitoring *resource* server linux SYSTAT, yang berfungsi sebagai perangkat untuk mengambil data kondisi tiap node cluster.

M. Fadlulloh RB, dan Asmunin dengan judul Implementasi *Docker* Untuk Pengelolaan Banyak Aplikasi Web (2017). Menjelaskan *docker* untuk pengelolaan banyak aplikasi web, pada penelitian ini setiap aplikasi web beserta *environment* yang dibutuhkan *di-deploy* ke dalam *containers*. Setiap container akan memiliki IP *Private* yang hanya dapat diakses oleh *Host*. Sehingga dapat diakses dari luar setiap *container* akan dibuatkan *domain*. Dan untuk mengarahkan domain ke *container* tujuan digunakan Apache sebagai *reverse proxy*.

Rakhmi K, Adi M, dan Siti S dengan judul Teknis Kerja *Docker Container* untuk Optimalisasi Penyebaran Aplikasi (2019). Menjelaskan bahwa *docker*

container sangat cocok untuk desain arsitektur sistem dengan pendekatan *microservice*, karena masing-masing service memiliki lingkungan yang terisolasi namun tetap dapat berkomunikasi satu sama lain. *Microservice* sendiri merupakan suatu pendekatan desain arsitektur sistem informasi yang menspesifikasikan dan memecah fungsi dari sistem yang besar menjadi sistem atau service yang kecil dan spesifik. Adapun saran untuk penelitian selanjutnya adalah perlu banyak referensi tentang konfigurasi web server desain dan pengembangan aplikasi berbasis *Container* ini karena pada proses pengembangan aplikasi hal ini dapat mempengaruhi desain arsitektur suatu sistem.

Saleh Dwiyatno, Edi Rakhmat, Oki Gustiawan dengan judul Implementasi Virtualisasi Server Berbasis *Docker Container* (2020) menjelaskan teknologi container cocok dijadikan *alternative deployment* website dengan resource rendah dan cepat. Dependency yang beragam dan dapat dilakukan di environment manapun menjadikan teknologi container dapat memperbaiki kekurangan mesin virtual berbasis *hypervisor*. Parameter pengujian dalam penelitian ini adalah penggunaan CPU dan memori bagi kedua server. Hasil dari penelitian ini menggunakan perangkat lunak apache JMeter membuktikan bahwa kemampuan *docker* dalam memanfaatkan sumber daya hardware sangat baik dan efisien sehingga dapat mengoptimalkan infrastruktur yang ada.

Kisnandar, Rasma Bayu dengan judul Analisis Perbandingan Kinerja Web Server Nginx, Apache, Dan Lighttpd Dengan Metode Stress Test (2019) menjelaskan bahwa layanan dari berbagai layanan web server memiliki kelebihan dan kekurangan. Perbandingan pada penelitian ini bertujuan untuk

menjadikan refrensi web server yang cocok untuk sebuah aplikasi website agar penggunaan sumber daya lebih efisien dan penggunaan web server lebih optimal. Hasil dari penelitian ini disimpulkan dari beberapa web server yang dibandingkan tidak mengalami kegagalan dengan menangani beban 5000 hingga 20000 *request*.

Tinjauan Pustaka dalam bentuk tabel disajikan pada Tabel 2.1 berisi daftar penilitan, judul, metode, dan hasil penelitian oleh peneliti sebelumnya mengenai topik analisis kinerja web server dan juga usulan penelitian dari penulis.

Tabel 2. 1. Tinjauan Pustaka

No	Penulis	Judul Penelitian	Metode	Keterangan
1	Tanjung P. K. Dr. Ir. Rendy Munadi, M. T. Danu Dwi, S.,S.T., M.T. (2017)	Implementasi dan Analisis <i>Computer Clustering System</i> dengan Menggunakan Virtualisasi Docker	Docker Swarm NFS HAProxy	Jurnal
2	M. Fadlulloh RB, dan Asmunin (2017)	Implementasi Docker Untuk Pengelolaan Banyak Aplikasi Web	Docker	Jurnal Penelitian
3	Rakhmi K, Adi M, dan Siti S (2019)	Teknis Kerja Docker Container untuk Optimalisasi Penyebaran Aplikasi	Docker	Jurnal Penelitian

4	Saleh Dwiyatno, Edi Rakhmat, Oki Gustiawan (2020)	Implementasi Virtualisasi Server Berbasis Docker Container	Virtualisasi, Docker	Jurnal
5	Kisnandar, Rasma Bayu (2019)	Analisis Perbandingan Kinerja Web Server Nginx, Apache, Dan Lighttpd Dengan Metode Stress Test	Stress Test	Skripsi
4	Usulan Penelitian	Analisa Perbandingan Kinerja Deployment Laravel Native Dengan Laravel Docker Dengan Metode Stres Test	Docker, Virtulisasi, Stress Test	

2.2 Dasar Teori

2.2.1 Server

Server adalah sebuah komputer yang mampu menyediakan sumber daya sebagai pusat penyimpanan data dan layanan khusus untuk para *client* – nya. Penyimpanan server biasanya dipakai untuk menyimpan dokumen dan berbagai informasi. Dari data tersebut server mampu menggunakannya dalam berbagai layanan. Contoh dari sebuah website maka yang berperan adalah data dari layanan web server. Server memiliki berbagai jenis, dan server dibagi untuk memenuhi

kebutuhan *client* – nya, mulai dari DNS, email, hingga website. Server juga bisa berbentuk fisik berupa *hardware*, *software*, atau sebuah mesin virtual. Untuk dapat memenuhi kebutuhan *client*, server harus memiliki perangkat keras yang besar untuk menampung data atau permintaan dari *client*, untuk itu server biasanya memiliki jumlah prosesor dan memori yang lebih dari komputer biasa. (Aprilia, Putri 2021).

2.2.2 Cloud Computing

Cloud Computing (komputasi awan) merupakan gabungan pemanfaatan teknologi komputer (komputasi) dalam suatu jaringan dengan pengembangan berbasis internet (awan) yang mempunyai fungsi untuk menjalankan program atau aplikasi melalui komputer – komputer yang terkoneksi pada waktu yang sama, tetapi tak semua yang terkoneksi melalui internet menggunakan *cloud computing*. Teknologi komputer berbasis sistem *cloud* ini merupakan sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. Teknologi ini mengizinkan para pengguna untuk menjalankan program tanpa instalasi dan mengizinkan pengguna untuk mengakses data pribadi mereka melalui komputer dengan akses internet.

Terdapat tiga buah model dari *cloud computing*, yaitu:

- a. *Software as a Service* (SaaS). Merupakan layanan yang diberikan kepada konsumen untuk menggunakan aplikasi penyedia dapat beroperasi pada infrastruktur *cloud*. Aplikasi dapat diakses dari berbagai *client* melalui antar muka seperti website, email dll. *Customer* tidak dapat mengelola

infrastruktur *cloud* yang mendasar termasuk setting jaringan, server, sistem operasi, penyimpanan, dan fitur dari aplikasi individu. Contoh dari SaaS adalah Google Apps, media sosial seperti *Facebook*.

- b. *Platform as a Service (IaaS)*. Merupakan layanan yang diberikan kepada customer untuk memproses, menyimpan, melakukan konfigurasi jaringan bahkan eksplorasi sumber komputasi, dimana *customer* dapat menyebarkan dan menjalankan perangkat lunak secara bebas, yang dapat menyangkup sistem operasi aplikasi. Contoh dari IaaS adalah *Amazon Elastic Compute, Google Compute*
- c. *Platform as a Service (PaaS)*. Kemampuan yang diberikan kepada konsumen untuk menyebarkan aplikasi yang dibuat konsumen atau diperoleh ke infrastruktur *cloud computing* menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider. Konsumen tidak mengelola atau mengendalikan infrastruktur *cloud* yang mendasar termasuk jaringan, server, sistem operasi, atau penyimpanan, namun memiliki kontrol atas aplikasi yang disebarkan dan memungkinkan aplikasi melakukan *hosting* konfigurasi. Contohnya yang sudah mengimplementasikan ini adalah Force.com dan *Microsoft Azure investment*. (Basari, 2014).

2.2.3 Web server

Web server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui browser kemudian memberikan tanggapan permintaan dalam bentuk halaman situs web atau lebih umumnya dalam dokumen HTML. Namun, web server dapat mempunyai dua pengertian berbeda, yaitu

sebagai bagian dari perangkat keras (*hardware*) maupun sebagai bagian dari perangkat lunak (*software*).

Saat mengambil halaman website, browser mengirimkan permintaan ke server yang kemudian diproses oleh web server. HTTP *request* dikirimkan ke web server. Sebelum memproses HTTP *request*, web server juga melakukan pengecekan terhadap keamanan. Pada web server, HTTP *request* diproses dengan bantuan HTTP server. HTTP server merupakan perangkat lunak yang bertugas menerjemahkan URL (alamat situs website) serta HTTP (protokol yang digunakan browser untuk menampilkan halaman website). Kemudian web server mengirimkan HTTP *response* ke browser dan memprosesnya menjadi halaman situs website. Pada saat web server menerima HTTP *request* dari browser, jika diperlukan web server akan mengirimkan *query* ke database untuk memenuhi permintaan HTTP *request* yang dikirimkan oleh browser (Yasin, 2018)

2.2.4 Database

Database atau yang dikenal juga dengan istilah basis data adalah sekumpulan data yang dikelola dengan sedemikian rupa berdasarkan ketentuan tertentu yang saling berkaitan sehingga memudahkan dalam pengelolaannya. Lewat pengelolaan itulah pengguna bisa mendapatkan kemudahan dalam mencari sebuah informasi, membuang informasi, maupun menyimpan informasi. Selain pengertian *database* diatas, terdapat pengertian lain dari *database* yaitu sebuah sistem yang berguna untuk mengumpulkan file, arsip, atau tabel yang disimpan dan terhubung dalam berbagai media elektronik yang ada. (Lely Azizah, 2021).

2.2.5 Laravel

Laravel merupakan *framework* yang menggunakan Bahasa pemrograman PHP dalam proses pengembangannya karena PHP menjadi Bahasa pemrograman yang dinamis menjadikan laravel sebuah *framework* yang lebih *powerfull*, cepat, aman dan *simple*. Setiap perilisan versi laravel terbaru selalu menghadirkan fitur atau teknologi terbaru menjadikan *framework* yang paling banyak diminati oleh para *developer* di github (tahun 2015). Laravel fokus dibagian *end-user* yang berarti fokus pada kejelasan dan kesederhanaan, baik penulisan maupun tampilan, serta menghasilkan fungsionalitas aplikasi web yang bekerja sama sebagaimana mestinya. Hal ini membuat para *developer* maupun perusahaan menggunakan *framework* ini untuk membangun sebuah proyek website dari skala kecil hingga skala atas. Laravel mengubah pengembangan website menjadi lebih elegan, ekspresif, dan menyenangkan. Selain itu laravel lebih mudah dalam proses pengembangannya dengan menerapkan konsep MVC (*Model, View, Control*) (Yasin K 2019).

2.2.6 Docker

Docker pertama kali diperkenalkan oleh Solomon Hykes, pendiri dan CEO ofdotCloud dalam presentasi singkatnya di *Python Developers Conference* di Santa Clara, California, pada tanggal 15 Maret 2013. Pada saat diperkenalkannya Docker, hanya sekitar 40 orang diberi kesempatan untuk mencoba Docker. Beberapa minggu kemudian proyek ini mengejutkan beberapa kalangan karena Docker adalah sebuah proyek *open – source* dan tersedia untuk umum pada GitHub, di mana orang bisa mengunduh dan berkontribusi dalam proyek ini. Docker

merupakan *project open-source* yang menyediakan *Platform* terbuka dalam bentuk teknologi virtualisasi berbasis *container*, ditujukan bagi para developer maupun *sysadmin* untuk dapat membangun, *membundel* dan menjalankan aplikasi dimanapun dalam satu *container* yang ringan. Mirip seperti *virtual machine* namun lebih ringan karena Docker tidak membawa keseluruhan sistem operasi, melainkan berbagi sistem dengan *host* induknya.

Docker menyatukan perangkat lunak dalam *filesystem* lengkap yang berisi semua yang diperlukan menjalankannya: *source code*, paket sistem untuk runtime, perangkat sistem, sistem pustaka *software* apapun yang dapat diinstal pada server. Hal ini menjamin bahwa perangkat lunak akan selalu berjalan sama, tidak tergantung pada lingkungannya. Pada perkembangan terkini, Docker bahkan bisa dijalankan diatas sistem *Windows* maupun *MacOS*(Nugroho, 2018).

2.2.7 Hypervisor

Hypervisor adalah sebuah teknik virtualisasi yang memungkinkan beberapa *operating system* untuk berjalan bersamaan pada sebuah *host*. Dikatakan teknik virtualisasi karena OS yang ada bukanlah sebuah OS yang sesungguhnya, hanya sebuah *virtual machine* saja. Tugas dari *hypervisor* adalah untuk mengatur setiap *operating system* tersebut sesuai dengan gilirannya agar tidak mengganggu satu dengan yang lainnya. Terkadang, *hypervisor* juga disebut sebagai *Virtual Machine Management* (VMM), sesuai dengan tugasnya dalam mengatur beberapa *virtual machine*.

Pada setiap jenis komputer, seperti *cluster computing*, *grid computing*, PC ataupun *mainframe*, memiliki OS yang berbeda satu sama lain karena memiliki

sistem yang juga berbeda. Setiap OS tersebut di desain sesuai dengan kebutuhan dari sistem masing masing. Untuk *Hypervisor* sendiri, didesain lebih mirip OS untuk *mainframe* dari pada *Windows OS*. Hal ini dikarenakan sebuah *hypervisor*, harus bisa mengatur beberapa sistem sekaligus, layaknya sebuah host melayani beberapa *client* pada *mainframe*(Ilham Efendi, 2016).

2.2.8 Benchmarking

Benchmarking adalah proses evaluasi *performance* dari suatu sistem pada kondisi tertentu. *Benchmarking* dilakukan dengan menjalankan sebuah atau kumpulan program yang dinamakan *benchmarking tools*. Program tersebut akan menjalankan operasi – operasi tertentu untuk menguji kemampuan dari suatu sistem. *Benchmarking tools* dalam dunia komputasi secara umum menguji beberapa aspek dalam komputer seperti kemampuan *CPU*, *storage*, *graphics*, maupun sistem secara keseluruhan(Tjandra dkk, 2016).

2.2.9 Apache JMeter

Apache JMeter merupakan *software open source* yang dikembangkan untuk dapat melakukan serangkaian pengujian yang berkaitan dengan performa. Jmeter merupakan 100% aplikasi Java murni, untuk pengujian beban dan kinerja. JMeter dirancang untuk melakukan beberapa kategori tes seperti beban, fungsional, kinerja, regresi, dll. Apache JMeter membutuhkan JDK (Java Development Kits) 5 atau lebih tinggi agar bisa dijalankan pada sebuah komputer.

2.2.10 Throughput

Throughput adalah jumlah bit atau data yang diterima dengan sukses perdetik melalui sebuah sistem atau media komunikasi dalam selang waktu pengamatan tertentu. Aspek utama *throughput* yaitu berkisar pada ketersediaan *bandwith* yang cukup untuk menjalankan aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh suatu aplikasi saat melewati jaringan, oleh sebab itu semakin besar *throughput* yg dihasilkan akan semakin baik. (Muhammad Syaiful Adnan, 2017).

2.2.11 Response Time

Waktu Tanggap (*Response Time*) adalah waktu tanggap yang diberikan oleh antar muka *interface* ketika user merequest permintaan ke server. Secara umum, waktu tanggap yang baik adalah waktu tanggap yang sependek – pendeknya. (Linda, 2015).

2.2.12 Git

Git adalah salah satu sistem pengontrol versi (*Version Control System*) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds. Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri. Git dikenal juga dengan distributed revision control (VCS terdistribusi), artinya penyimpanan *database* Git tidak hanya berada dalam satu tempat saja. Dalam Git terdapat merge untuk menyebut aktifitas penggabungan kode. Sedangkan pada VCS (*Version Control System*) yang terpusat *database* disimpan dalam satu tempat dan setiap perubahan disimpan ke sana. (Ahmad Muhardian, 2021).