

ANALISA PENERAPAN *SERVER DEPLOYMENT* MENGGUNAKAN KUBERNETES UNTUK
MENGHINDARI *SINGLE OF FAILURE*Lilik Widyawati^{1*}, Heroe Santoso², Hamdika Budiman³^{1,3}Ilmu Komputer, Universitas Bumigora²Rekayasa Perangkat Lunak Aplikasi, Universitas Bumigora

email: lilikwidya@universitasbumigora.ac.id*

Abstrak: Sistem komputasi terdistribusi menjadi salah satu kebutuhan dalam implementasi aplikasi berbasis server seperti database server dan web server agar tercapainya tingkat performansi tinggi. Masalah yang sering terjadi adalah kegagalan pada server sehingga performa dari sebuah server terganggu, sehingga dibutuhkan suatu teknik deploy dapat digunakan untuk menyediakan sistem terdistribusi dengan performansi tinggi. Virtualisasi berbasis container menjadi pilihan untuk menjalankan sistem terdistribusi karena arsitektur yang ringan, kinerja yang cepat, dan efisiensi sumber daya. Salah satu virtualisasi berbasis container adalah memperkenalkan alat pengembangan sistem terdistribusi yang disebut Kubernetes, yang memungkinkan manajemen deploy server untuk menyediakan sistem dengan availability yang tinggi. Metodologi pengembangan system yang digunakan adalah Network Development Life Cycle (NDLC). Dari 6 tahapan yang ada, hanya digunakan 3 tahapan yaitu Analysis, Design, dan Simulation Prototyping. Uji coba atau scenario pengujian yang dilakukan adalah Ftp Deploy dan Web Server Nginx sehingga dapat menjaga ketersediaan dan sistem mampu melakukan failover saat terjadi kegagalan pada server.

Kata Kunci : Kubernetes, Server Deployment, Server Nginx

Abstract: *istributed computing systems are one of the requirements in implementing server-based applications such as database servers and web servers in order to achieve high levels of performance. The problem that often occurs is failure on the server so that the performance of a server is disrupted, so it takes a deployment technique that can be used to provide a distributed system with high performance. Container-based virtualization is the choice for running distributed systems because of its lightweight architecture, fast performance and resource efficiency. One of the container-based virtualisations is the introduction of a distributed systems development tool called Kubernetes, which allows managing server deployments to provide high availability systems. The system development methodology used is the Network Development Life Cycle (NDLC). Of the 6 stages, only 3 stages are used, namely Analysis, Design, and Simulation Prototyping. The test or test scenario carried out is Ftp Deploy and Nginx Web Server so that it can maintain availability and the system is able to failover when a server failure occurs.*

Keywords : Kubernetes

PENDAHULUAN

Sistem komputasi terdistribusi menjadi suatu kebutuhan dalam implementasi aplikasi web berserta software pendukung seperti database server dan web server. Sistem terdistribusi adalah suatu sistem (hardware + software) yang terdiri atas kumpulan dari komputer yang saling terhubung dan terlihat bagi pengguna sistem sebagai suatu sistem komputer tunggal [1]. Beberapa server yang sering digunakan adalah FTP server dan Web server nginx.

FTP (File Transfer Protocol) adalah suatu standar protocol yang digunakan dalam mempertukarkan file antar komputer di dalam jaringan yang mendukung protokol TCP/IP, seperti Internet [2] Untuk memastikan bahwa file terkirim dan diterima tanpa terjadi loss pada file yang dipertukarkan, FTP menggunakan protokol TCP pada lapisan transport dan Web Server Nginx adalah sistem web server terdistribusi yang terkenal karena stabil dan memiliki tingkat performansi tinggi. Virtualisasi berbasis container menjadi pilihan untuk menjalankan sistem terdistribusi karena arsitektur yang ringan, kinerja yang cepat, dan efisiensi sumber daya [3].

Server tunggal tidak cukup untuk mendukung lalu lintas yang tinggi dan dapat menyebabkan *single*

point of failure (SPOF), yaitu kondisi server yang mengalami kegagalan dalam merespon *request client* maka sistem akan tidak berfungsi dikarenakan terlalu banyak menangani *request* dari *client* [4], yang tidak dapat diakses saat terdapat permasalahan pada *server* merupakan hal yang dihindari oleh pengguna. Salah satu teknik untuk mengatasi masalah tersebut adalah menerapkan sistem terdistribusi menggunakan teknik *deploy*. memperkenalkan alat pengembangan sistem terdistribusi yang disebut *Kubernetes*, Kubernetes merupakan platform open-source yang digunakan untuk melakukan manajemen workloads aplikasi yang dikontainerisasi, serta menyediakan konfigurasi dan otomatisasi secara deklaratif [5] sehingga memungkinkan manajemen *deploy server* dan memudahkan dalam proses *container* pada *multihost*.

TINJAUAN PUSTAKA

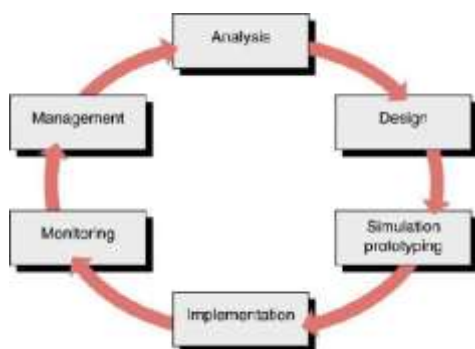
Terdapat beberapa penelitian yang telah dilakukan oleh peneliti terkait sistem terdistribusi. Dawood dkk, (2014) melakukan penelitian membandingkan tiga sistem terdistribusi yaitu *Nginx*, *Apache*, dan *Lighttpd*. Penelitian tersebut menghasilkan bahwa diantara ketiga aplikasi *web*

server tersebut, *Nginx* memiliki kinerja terbaik dan memiliki tingkat performansi tinggi [6]. Sedangkan penelitian yang dilakukan oleh Kusuma dkk, (2017) menyatakan bahwa teknologi virtualisasi *container* dapat menjadi solusi untuk menjalankan sistem terdistribusi yang mudah dikonfigurasi dan mempunyai skalabilitas tinggi [7]. Sebaliknya, penelitian yang dilakukan oleh Apridayanti dkk, (2018) menyampaikan bahwa teknik virtualisasi *container* menggunakan *Kubernetes* untuk manajemen *container* [8].

Penelitian terkait *kubernetes* juga pernah dilakukan oleh yedutun (2019), fokus penelitian yang dilakukan adalah implementasi *kubernetes* untuk mendukung *scalability*, hasil implementasi *scalability* terhadap *container* adalah adanya penghematan *cpu usage* pod pada saat *scalability* diterapkan. Dikarenakan adanya pembagian *container* yang tersebar di tiap *worker* nya [9].

METODE

Metode pengemanagan system yang digunakan adalah NDLC,



Gambar 1. Fase NDLC
 Sumber: Goldmend dan Rawles (2004)

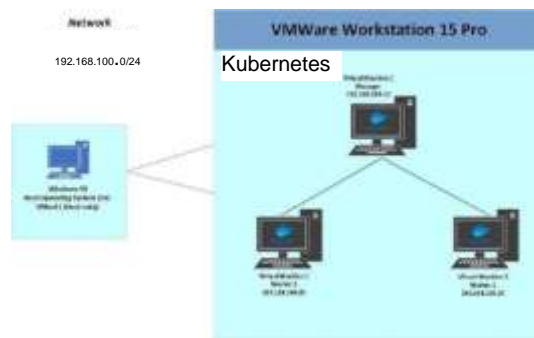
Dari keenam fase yang terdapat pada NDLC, penulis hanya menggunakan 3 (tiga) fase antara lain sebagai berikut: *Analysis*, *design* dan *simulation Prototyping*.

Pada fase *Analysis* penulis melakukan pengumpulan data dengan cara studi literatur, yaitu penulis membaca artikel ilmiah dan jurnal untuk mendapatkan informasi mengenai *Kubernetes*, Virtualisasi berbasis *Container*, *Web Service Nginx* serta *Ftp server*, data yang telah terkumpul kemudian dianalisis.

Berdasarkan hasil penelusuran artikel jurnal ilmiah yang terkait, maka dapat diketahui bahwa belum terdapat penelitian tentang sistem web server *Nginx* dan *Ftp Server* berbasis *fauria* menggunakan *Kubernetes* agar tercapainya sistem terdistribusi untuk menyediakan ketersediaan tinggi serta memberikan toleransi kesalahan. Dari permasalahan dan hasil analisa tersebut maka mendorong penulis untuk melakukan penelitian tentang Analisa Penerapan *Web Server Nginx* dan *Ftp Server* *fauria*

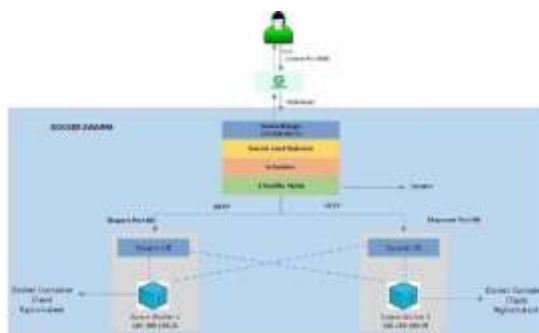
Menggunakan *Kubernetes*.

Pada tahap *Design* penulis membuat rancangan yang meliputi rancangan jaringan uji coba, rancangan system menggunakan *Kubernetes*, pengalamatan IP, serta kebutuhan perangkat keras dan perangkat lunak. Adapun rancangan system dapat dilihat pada gambar dibawah ini.



Gambar 2. Rancangan Sistem *Kubernetes*

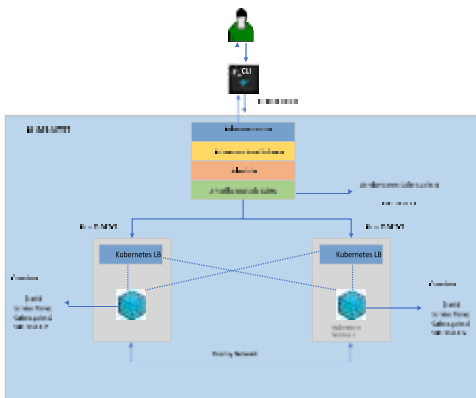
Dalam rancangan uji coba akan disimulasikan secara virtualisasi menggunakan *VMware Workstation* yang dijalankan pada 1 (satu) laptop. Di dalam *VMware Workstation* terdapat 3 (tiga) virtual machine yang difungsikan sebagai master dengan IP 192.168.100.10, worker 1 dengan IP 192.168.100.20 dan worker 2 dengan IP 192.168.100.30. Sistem operasi yang digunakan adalah *Linux CentOS release 7.7.1908* dan di *deploy* menggunakan *Kubernetes*. Pada komputer client telah terinstal sistem operasi *windows 10* dan aplikasi browser *Google Chrome* untuk mengakses web server *Nginx*.



Gambar 3. Rancangan Sistem *Web Server Nginx*

Sistem dari web server *Nginx* di *deploy* menggunakan *Kubernetes*. Terdapat 3 (tiga) node yaitu master, worker 1 dan worker 2. Node master bertugas untuk mengatur status *deploy*, membagikan tugas ke node worker, membuat replika proses, serta memperbaiki kesalahan pada node worker. Pada node master dilakukan pembuatan web server *Nginx* lalu tersebut direplikasi atau disebarkan pada node-worker kemudian pada node master terdapat

scheduler untuk menjadwalkan atau merutekan pada node dalam deploy sebagai satu atau lebih tugas replika.



Gambar 4. Rancangan Sistem Ftp Server

Sistem dari Ftp Server di deploy menggunakan Kubernetes. Terdapat 3 (tiga) node yaitu master, worker 1 dan worker 2. Node master bertugas untuk mengatur status deploy, membagikan tugas ke node worker, membuat replika proses, serta memperbaiki kesalahan pada node worker. Pada node master dilakukan pembuatan Ftp Server lalu tersebut direplikasi atau disebarkan pada node-node worker kemudian pada node master terdapat scheduler untuk menjadwalkan atau merutekan pada node dalam deploy sebagai satu atau lebih tugas replika.

Tahap berikutnya adalah melakukan simulasi dan membuat prototype. Pada fase ini dilakukan instalasi dan konfigurasi serta uji coba system virtualisasi container yang telah di deploy menggunakan Kubernetes.

Uji coba dilakukan dengan mengakses ftp atau web server baik dalam kondisi seluruh server node aktif, dalam kondisi salah satu server node tidak aktif serta dalam kondisi seluruh container down lalu melihat kinerja dari hasil penerapan Kubernetes apakah mampu server atau host yang lainnya untuk menangani request dari client sehingga tercapai fungsi ketersediaan (availability).

HASIL DAN PEMBAHASAN

Instalasi dan konfigurasi dilakukan pada 1 (satu) server yang difungsikan sebagai node master untuk kontroler pada komputer deploy, 2 (dua) server yang difungsikan sebagai node worker untuk mengeksekusi proses yang telah dibagi oleh node master, serta 1 (satu) client yang difungsikan untuk menggunakan yang telah ditentukan.

Pada server yang difungsikan sebagai node master dilakukan instalasi Linux CentOS 7 menggunakan DVD ISO images minimal,

konfigurasi pengelamatan IP, instalasi , instalasi Kubernetes sebagai metode server.

Pada server yang difungsikan sebagai node worker dilakukan instalasi Linux CentOS 7 menggunakan DVD ISO images minimal, konfigurasi pengelamatan IP, serta instalasi dan konfigurasi Kubernetes sebagai metode server.

Pada komputer client yang difungsikan untuk menggunakan dilakukan konfigurasi pengelamatan IP agar komputer client dan server dapat terhubung.

Skenario uji coba dilakukan dengan pembuatan web server berbasis Nginx dan Ftp Server Fauria serta dilakukan verifikasi konfigurasi untuk deploy service untuk mengetahui apakah tersebut berhasil tereplikasi pada tiap-tiap node worker.

Uji coba selanjutnya dengan mengakses web server berbasis Nginx dan Ftp Server Fauria serta menganalisa hasil kinerja dari penerapan Kubernetes.

Tabel 1. Pengujian Web server NGINX

N	Skenario	Hasil	Keterangan
1	Akses Web Server Nginx dalam kondisi seluruh server node aktif	Sistem mampu melayani request	Sistem server menggunakan Kubernetes berjalan dengan baik
2	Akses Web Server Nginx dalam kondisi salah satu server node mengalami kegagalan	Berhasil diakses walaupun salah satu server mengalami kegagalan	Container pada server node yang down berpindah pada server node yang tersedia/aktif.
3	Akses Web Server Nginx dalam keadaan seluruh container down.	berhasil diakses walaupun seluruh container down.	Sistem Kubernetes melakukan Self Healing dengan pembuatan container baru

Tabel 1 menunjukkan bahwa sistem mampu melayani request dari client baik dalam kondisi seluruh server node aktif maupun dalam kondisi salah satu server node tidak aktif termasuk dalam kondisi seluruh container down. Hal tersebut dapat tercapai sebagai dampak dari fungsi dari ketersediaan yang diberikan oleh Kubernetes. Jika salah satu host down maka service akan digantikan dengan host yang sedang aktif.

Ketika salah satu node dalam keadaan down, maka container yang berada pada node tersebut melakukan self-healing dengan membuat container

baru yang secara otomatis muncul pada salah satu server node yang aktif. Selain itu ketika container pada seluruh server node *down*, sistem otomatis akan melakukan *self-healing* dengan pembuatan container baru. Hal tersebut dapat tercapai karena Kubernetes mengimplementasikan Algoritma Konsensus Raft pada node manager untuk mengelola status deploy. Raft adalah protokol untuk mengimplementasikan konsensus terdistribusi yang toleran terhadap kesalahan.

Dengan algoritma Konsensus Raft, node manager dapat memindahkan ke node yang masih aktif sehingga apabila salah satu node mengalami kegagalan, hal ini tidak akan mengganggu sistem yang sedang berjalan sehingga tercapainya ketersediaan dan user tetap mendapatkan balesan dari system.

Tabel 2. Pengujian FTP server Fauria Deploy

No	Skenario	Hasil	keterangan
1	Membuat terkoneksi menggunakan aplikasi FileZilla Client	Node yang telah dibuat berhasil di sinkronisasi ke seluruh node pada deploy	Sinkronisasi Ftp server berhasil
2	Mengirimkan file .txt ke setiap node kubernetes	File .txt yang dikimkan berhasil sampai ke node master, worker 1 dan worker 2	Sinkronisasi File .txt berhasil.
3	Mengirimkan file .pdf ke setiap node kubernetes	File .pdf yang dikimkan berhasil sampai ke node master, worker 1 dan worker 2	Sinkronisasi file .pdf berhasil.

Tabel 2 menunjukkan bahwa sistem mampu melakukan terhadap ftp server yang sudah di deploy pada node master dan di sebarakan ke node worker 1 dan node worker 2. Pada uji coba Ftp server Fauria akan ditampilkan analisa hasil ujicoba meliputi terkoneksi dengan FileZilla client, mengirim file .txt dan file .pdf yang dijalankan pada deploy server menggunakan Kubernetes baik ketika kondisi seluruh server node aktif, kondisi salah satu server node tidak aktif, maupun saat seluruh container pada server node tidak aktif.

Jika salah satu host down maka pada host tersebut akan digantikan atau dipindahkan ke host yang sedang aktif. Tugas pemindahan tersebut di handle oleh node master karena salah satu fungsi dari node manager adalah memperbaiki kesalahan pada node worker. Dengan algoritma Konsensus Raft, node manager dapat memindahkan ke node yang masih aktif sehingga apabila salah satu node mengalami kegagalan, hal ini tidak akan mengganggu sistem yang sedang berjalan sehingga

tercapainya ketersediaan dan user tetap mendapatkan balesan dari system.

KESIMPULAN DAN SARAN

Berdasarkan hasil instalasi, konfigurasi dan ujicoba serta analisa terhadap hasil ujicoba maka dapat diperoleh kesimpulan sebagai berikut:

Kubernetes dapat digunakan untuk melakukan container pada web berbasis Nginx dan Ftp server sehingga dapat terjaga ketersediaannya berdasarkan skenario uji coba mengaktifkan seluruh node, menonaktifkan salah satu node dan menonaktifkan seluruh node pembentuk deploy.

Teknik deploy menggunakan kubernetes dapat menghindari single point of failure dan ketika salah satu atau seluruh node pembentuk deploy mengalami gangguan (tidak aktif) maka sistem secara otomatis akan melakukan self healing untuk membentuk container baru.

Mekanisme failover yang terdapat pada kubernetes berjalan dengan baik, sehingga sistem memiliki ketersediaan tingkat tinggi (high availability) Saran memuat berbagai usulan atau pendapat yang sebaiknya dikaitkan oleh penelitian sejenis. Saran dibuat berdasarkan kelemahan, pengalaman, kesulitan, kesalahan, temuan baru yang belum diteliti dan berbagai kemungkinan arah penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] A. Bima, M. Wijawa, J. Soeprapto and Y. Tegar, "Analisa Perancangan Arsitektur Sistem terdistribusi menggunakan metode Nested Transaction pada Lingkungan Kerja Perkantoran Difficulty or Complexity of Bussiness Integration," vol. 7, no. 2, pp. 2-9, 2015.
- [2] Yuhfizar, "File Transfer Protokol," in *10 Jam Meng. Internet, teknologi & Apl*, Jakarta, PT Elex media Komputindo, 2008.
- [3] A. Terry, "Take Control of Kubernetes," Nginx, 9 5 2019. [Online]. Available: <https://www.nginx.com/products/nginx-service-mesh>. [Accessed 2 12 2020].
- [4] R. Julianto, W. Yahya and S. R. Akbar, "Implementasi Load Balancing Di Web Server Menggunakan Metode Berbasis Sumber Daya CPU Pada Software Defined Networking," *JPTIIK*, 2017.
- [5] Editor, "Apa itu Kubernetes," *Kubernetes.io*, 9 10 2020. [Online]. Available: <https://kubernetes.io/id/docs/concepts/overview/what-is-kubernetes/>. [Accessed 2 12 2020].
- [6] R. Dawood, S. F. Qiana and S. Muchallil, "Kelayakan Raspberry Pi Sebagai Web Server: Perbandingan Kinerja Nginx, Apache, Dan Lighttpd Pada Platform Raspberry Pi," *Jurnal*

- Rekayasa Elektrika*, vol. 11, no. 1, pp. 30-35, 2014.
- [7] T. P. Kusuma, R. Munadi and D. D. Sanjoyo, "Implementasi Dan Analisis Computer System Dengan Menggunakan Virtualisasi," in *e-proceeding of engineering*, 2017.
- [8] S. Apridayanti, "Desain Dan Implementasi Virtualisasi Berbasis Container," *Semantik*, vol. 4, no. 2, pp. 37-46, 2018.
- [9] K. Yedutun, A. Noertjahyana and H. N. Palit, *Jurnal Infra*, vol. 7, no. 2, 2019.