

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh M. Agung Nugroho dan Rikie Kartadi menguji apakah teknologi container dapat menjadi solusi untuk melakukan *Load Balancing* pada sistem aplikasi web yang menggunakan infrastruktur server yang terdistribusi. Proses pengujian penelitian tersebut dilakukan dengan menggunakan Docker untuk menjalankan container, HAProxy sebagai *software* pada server *load balancer* dan menggunakan bahasa pemrograman NodeJS untuk membangun aplikasi web sebagai media pengujian.

Penelitian yang dilakukan oleh Jihar Al Gifari yaitu pembuatan aplikasi *chat* dengan berbasis web dengan bahasa pemrograman Go, HTML5 dan javascript dimana aplikasi tersebut dijalankan dengan menggunakan *container*.

Penelitian yang dilakukan oleh Jay Shah dan Dushyant Dubaria menjelaskan bagaimana Docker dan Kubernetes dapat digunakan untuk membangun sistem berbasis *cloud* dengan tingkat efisiensi dan fleksibilitas yang cukup tinggi. Dengan menggunakan Docker proses *web service deployment* dapat dilakukan lebih cepat, selain itu untuk skalasi setiap *service* juga dapat dilakukan dengan mudah. Dengan Kubernetes manajemen infrastruktur dapat lebih mudah dilakukan karena setiap *container* yang menjalankan *service* selalu terjaga agar tetap berjalan dan dapat diakses oleh pengguna.

Tabel 2.1 Perbandingan Tinjauan Pustaka

Parameter Penulis	Objek	Teknologi	Bahasa Pemrograman	Hasil
M. Agung Nugroho, M.Kom & Rikie Katardi (2016)	<i>Load balancing</i>	<i>Container, HAproxy</i>	Tidak melakukan proses coding	Penerapan <i>load balancing</i> berhasil dilakukan dan penggunaan CPU terbukti lebih efisien
Jihar Al Gifari (2018)	Aplikasi chatting	<i>Container</i>	<i>Go</i>	Aplikasi berhasil dibuat dan dijalankan dengan menggunakan Docker
Jay Shah & Dushyant Dubaria (2019)	<i>Cloud container cluster</i>	<i>Kubernetes, Container</i>	Tidak melakukan proses coding	Kubernetes dan Docker dapat diterapkan untuk sistem yang mengadopsi konsep <i>Cloud Computing</i> di Google Cloud Platform.
Indra Agus Setiawan (yang diusulkan) (2019)	Otomasi konfigurasi perangkat lunak	<i>Container, Orchestration,</i>	<i>Go, ReactJS</i>	

Dari beberapa tinjauan pustaka, dapat disimpulkan bahwa penggunaan teknologi Container dan Kubernetes merupakan teknologi yang dapat digunakan untuk membangun berbagai macam layanan *cloud computing* yang mengutamakan fleksibilitas dan efisiensi konsumsi sumber komputasi.

2.2. Dasar Teori

2.2.1 Provisioning

Dalam konsep *cloud computing*, menurut Rouse (2018) *provisioning* merupakan sebuah proses untuk mengalokasikan sumber komputasi dalam jumlah tertentu untuk menjalankan sebuah layanan.

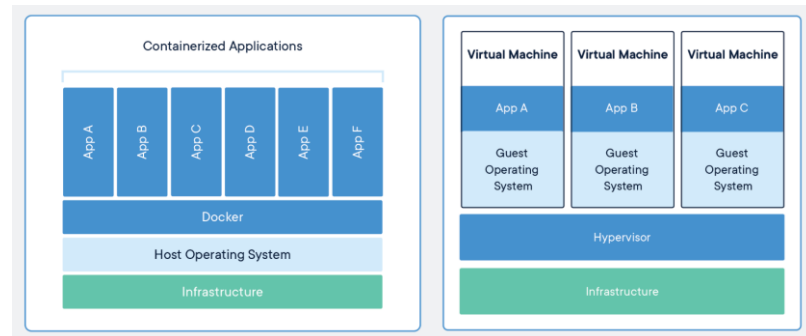
Proses *provisioning* merupakan bagian penting dalam konsep *cloud computing* karena pada setiap transaksi layanan *cloud* pasti akan dijalankan proses *provisioning* untuk mengalokasikan dan menjalankan layanan *cloud* yang akan digunakan.

2.2.2 Docker Container

Dalam situs resminya yaitu <https://www.docker.com/resources/what-container> dijelaskan bahwa Docker merupakan sebuah program komputer yang dapat melakukan virtualisasi di level sistem operasi yang unit virtualisasinya disebut sebagai *container*. *Container* merupakan teknik efisiensi dari virtualisasi dengan konsep *virtual machine (VM)*.

VM akan mem-virtualisasikan hardware komputer di atas *hypervisor* sehingga hardware suatu mesin komputer dibagi ke beberapa VM yang dibuat. Berbeda dengan VM, *container* mengabstraksikan sebuah aplikasi ke dalam satu paket yang berisi kode aplikasi dan perangkat lunak pendukung aplikasi tersebut agar aplikasi dapat berjalan, paket tersebut disebut *image*. *Image* akan dijalankan pada *service* Docker yang berjalan diatas sebuah sistem operasi. Beberapa *container* dengan program dan dependensi yang berbeda – beda dapat berjalan

bersamaan dengan saling berbagi resource *kernel* dari sistem operasi yang menjalankan *service* Docker. Berikut adalah gambaran perbedaan antara VM dan *Container*.



Gambar 2.1 Perbandingan Diagram Container dan VM

Dengan demikian beban hardware untuk menjalankan *container* akan lebih ringan apabila dibandingkan dengan VM karena VM akan menjalankan beberapa sistem operasi dalam sebuah mesin dan menjalankan aplikasi di setiap sistem operasinya sedangkan *container* hanya membutuhkan satu sistem operasi dan dapat menjalankan beberapa aplikasi sekaligus di atasnya dengan kebutuhan dependensi aplikasi yang berbeda-beda terlepas dari sistem operasi yang digunakan.

2.2.3 Container Orchestration

Menurut Bashir (2018) *Container Orchestration* adalah sebuah teknologi otomatisasi untuk menjalankan beberapa *container* dalam sebuah kesatuan sistem dalam beberapa mesin yang berbeda atau biasa disebut dengan *cluster*. Berikut adalah hal-hal yang diotomasikan dalam container orchestration :

- Melakukan pemeriksaan dan memastikan kesesuaian konfigurasi setiap mesin yang terhubung di dalam *cluster*.
- Melakukan pembuatan container sesuai dengan kebutuhan sistem yang akan dijalankan.
- Melakukan pembuatan ulang pada container yang mengalami malfungsi.
- Memastikan *container-contain* yang ada di beberapa mesin yang saling terhubung dapat berkomunikasi satu sama lain.
- Melakukan konfigurasi agar sistem yang dijalankan pada *cluster* dapat diakses dari luar jaringan cluster.
- Menambah atau mengurangi kapasitas beban pemrosesan sistem yang dapat ditangani berdasarkan tingkat beban yang dialami oleh cluster.

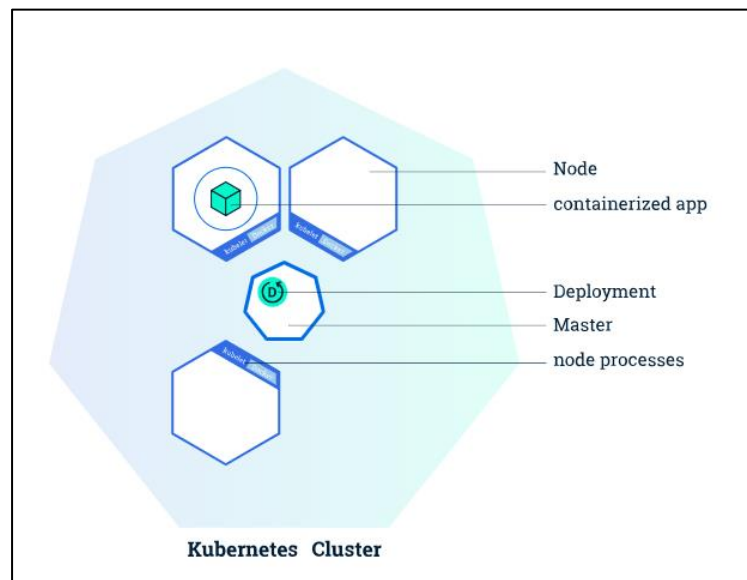
Beberapa contoh aplikasi untuk melakukan *Container Orchestration* adalah Docker Machine, Docker Swarm, Docker Compose dan Kubernetes.

2.2.4 Kubernetes

Dalam website resmi Kubernetes yaitu <https://kubernetes.io/> dijelaskan Kubernetes adalah sebuah program *open source* yang dapat melakukan otomatisasi untuk menjalankan, skalasi dan mengelola sebuah aplikasi yang dijalankan menggunakan *container*. Kubernetes awalnya dikembangkan oleh internal Google untuk kepentingan infrastruktur mereka yang masif dengan judul proyek “Project Seven”. Pada 2014 Google mengumumkan pengembangan Kubernetes ke publik dan pada 2015 Kubernetes v1.0 dirilis. Bersamaan dengan dirilisnya Kubernetes v1.0 Google juga mengumumkan kerjasama dalam pengembangan Kubernetes dengan Linux Foundation dan Cloud Native Computing Foundation yang dengan

kata lain Google menjadikan Kubernetes sebagai sebuah perangkat lunak *open source*.

Dalam implementasinya Kubernetes memiliki beberapa unit operasi utama yaitu *cluster*, *node*, *Pods* dan *deployment* dan *environment* dapat dipisahkan menggunakan *namespace*.



Gambar 2.2 Diagram Cluster Kubernetes

Cluster

Cluster merupakan sebuah unit keseluruhan dari sebuah sistem yang berjalan dengan menggunakan Kubernetes. Dalam sebuah cluster Kubernetes terdapat beberapa node.

Node

Node merupakan satu unit mesin komputer dalam cluster Kubernetes yang digunakan untuk menjalankan *container* dari aplikasi yang ingin dijalankan dalam Kubernetes.

Pods dan Deployment

Pods adalah unit untuk sebuah aplikasi yang berjalan dalam cluster. *Pod* berisi konfigurasi dari *container* yang digunakan untuk menjalankan aplikasi dan perilaku dari *pod* dapat diatur melalui konfigurasi *deployment*.

Service

Unit *service* dalam kubernetes adalah sebuah konfigurasi *endpoint* yang memungkinkan aplikasi yang berjalan pada *pod* dapat diakses melalui jaringan di internal maupun eksternal *cluster*. Dengan demikian perangkat lunak yang berjalan di dalam *pod* dapat diakses oleh *pod* lain dan melalui internet.

Unit-unit yang ada di dalam sebuah *cluster* Kubernetes dikelola pada sebuah mesin yang dikonfigurasi dengan *tool* kubectl, yaitu sebuah software yang dapat melakukan berbagai macam konfigurasi Kubernetes yang akan diterapkan di seluruh node dengan menggunakan *Kubernetes API* yang berjalan disetiap *daemon service* kubernetes pada setiap node.

Namespace

Namespace merupakan suatu metode pada Kubernetes untuk melakukan isolasi *environment* sehingga objek-objek yang ada dalam suatu *namespace* tidak saling berpengaruh dengan objek-objek pada *namespace* lain.

2.2.5 Go

Dalam situs “Kodingin” pada link <https://dasarpemrogramangolang.novalagung.com/> dijelaskan bahwa bahasa pemrograman Go atau juga biasa disebut Golang merupakan bahasa pemrograman

yang dikembangkan oleh Robert Griesemer, Rob Pike, dan Ken Thompson pada tahun 2007 di Google pada tahun 2007. Go dikembangkan berdasarkan bahasa pemrograman C dan C++ sehingga struktur kodenya mirip dengan kedua bahasa pemrograman tersebut.

Tergolong bahasa pemrograman yang baru, popularitas Go meningkat cukup pesat dari tahun ke tahun karena kelebihan – kelebihan yang dimilikinya, diantaranya adalah sebagai berikut.

- a. Go mendukung konkruensi di level bahasa dengan pengaplikasian yang cukup mudah.
- b. Mendukung pemrosesan data dengan di beberapa unit prosesor dalam waktu yang bersamaan (*parallel processing*).
- c. Memiliki *garbage collector* dimana nilai dari variabel-variabel dalam kode yang tidak terpakai dapat dibuang sehingga penggunaan memori menjadi lebih efisien.
- d. Proses kompilasi yang cepat.
- e. Merupakan bahasa pemrograman hirarkial sehingga developer tidak perlu memikirkan segmen *object-class* seperti pada bahasa pemrograman dengan paradigma *object-oriented*.
- f. Paket modul yang lengkap. Karena bahasa Go merupakan bahasa pemrograman *open source* maka banyak *developer* yang mengembangkan modul untuk berbagai macam fungsi yang sudah tersedia.

Go sudah teruji sampai ke level *production* di kalangan industri. Perusahaan – perusahaan besar seperti Google, Microsoft dan lain-lain menggunakan Go di berbagai sektor layanan yang mereka miliki.

2.2.6 ReactJS

Dalam situs website resmi ReactJS yaitu <https://reactjs.org/> dijelaskan bahwa ReactJS merupakan bahasa pemrograman yang dirancang khusus digunakan pada *frontend*. Dengan model deklaratif dan *component based* ReactJS memudahkan *developer* untuk melakukan *debugging* dan enkapsulasi komponen *frontend* untuk merancang UI yang lebih kompleks dengan lebih mudah.

ReactJS memiliki berbagai macam fitur, salah satunya ReactJS sangat mendukung untuk sebuah sistem yang mengadopsi *microservices* dimana programmer dapat mengakses data dari sebuah *web service* dan mengolah data tersebut pada perangkat *client*.

2.2.7 MySql

Dalam website resmi MySql yaitu <https://www.mysql.com/> dijelaskan bahwa MySql merupakan software yang digunakan untuk menjalankan database server. MySql dioperasikan dengan SQL (*Structured Query Language*) yaitu sebuah bahasa manajemen data yang digunakan untuk mengolah data yang saling berelasi (RDBMS).