

# **MAKALAH KUBERNETES METODOLOGI DESAIN PERANGKAT LUNAK PRAKTIK**

Disusun Guna Memenuhi Tugas Pertemuan 14  
Mata Kuliah Metodologi Desain Perangkat Lunak Praktik  
Dosen Pengampu : Endang Anggiratih, S.T., M.Cs.



**Makalah disusun oleh kelompok 7:**

ADI WIJAYA	5200411023
TARSONO	5200411188
BAGAS SATRIA TRI WICAKSANA	5200411302
RIDWAN MAULANA	5200411524

FAKULTAS SAINS DAN TEKNOLOGI  
PROGRAM STUDI INFORMATIKA  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
YOGYAKARTA  
2021/2022

## **KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa. Atas rahmat dan hidayah-Nya, kelompok 7 bisa menyelesaikan makalah yang berjudul "Scrum-ban".

Tidak lupa kelompok 7 mengucapkan terima kasih kepada Ibu Endang selaku Dosen pengampu Mata Kuliah Metodologi Desain Perangkat Lunak Praktik yang telah memberikan penulis tugas guna menyelesaikan mata kuliah ini.

Makalah ini berisi tentang pengetahuan akan Metode Kubernetes . Kelompok 7 menyadari ada kekurangan pada Makalah ini. Oleh sebab itu, saran dan kritik senantiasa diharapkan demi perbaikan karya penulis. Penulis juga berharap semoga karya ilmiah ini mampu memberikan pengetahuan tentang pentingnya penggunaan Kriptografi dalam teknologi dan kehidupan sehari-hari.

# DAFTAR ISI

## KATA PENGANTAR

## DAFTAR ISI

### BAB I Pendahuluan

- 1.1 Latar Belakang
- 1.2 Rumusan Masalah
- 1.3 Tujuan Penelitian

### BAB II Pembahasan

- 2.1 Pengertian Kubernetes
- 2.2 Kelebihan Kubernetes
- 2.3 Mengapa Kubernetes Dibutuhkan

### BAB III CONTOH METODE

- 3.1 Metode Yang Digunakan Kubernetes

### BAB VI Penutup

- 4.1 Kesimpulan
- 4.2 Perbandingan dengan metode

### Daftar Pustaka

# **BAB I**

## **Pendahuluan**

### **1.1 Latar Belakang**

*Container* mentransformasi cara kita memandang *Architecture* aplikasi dan bagaimana kecepatan programming team seharusnya dalam menghasilkan aplikasi yang dibutuhkan untuk mensupport bisnis. *Architecture Container* menjanjikan portabilitas aplikasi di lingkungan cloud hybrid dan memungkinkan programmer untuk fokus kepada pembuatan aplikasi yang hebat, tanpa khawatir dari gangguan *Infrastructure* atau hal-hal yang kecil lainnya.

*Container* dapat membantu menghilangkan kompleksnya sebuah aplikasi dengan cara meminimalisasi packaging *image* aplikasi dan meningkatkan portabilitasnya. Sebaliknya, mengelola *Container* deployment yang besar juga menghadirkan tantangan baru bagi tim operasi.

*Container* juga bisa dihidupkan dalam waktu yang singkat dibanding dengan waktu yang diperlukan untuk menghidupkan Virtual Machine (VM). Selain itu juga di iringi dengan utilisasi penggunaan resource yang lebih efisien. Namun teknologi *Container* ini harus didukung dengan pengelolaan system yang lebih baik karena banyaknya objek yang bisa terpasang, terlebih dengan *lifecycle* yang lebih dinamis. Oleh karena itu, hal ini memerlukan *Management* yang lebih terotomatisasi dan *policy-driven*.

Banyak tim beralih ke Kubernetes yang memiliki rangkaian fitur yang kompleks dan lengkap untuk membantu mereka mengatur dan mengelola *Container*. Baik digunakan untuk production, development atau testing environment. Kubernetes telah terpilih sebagai de facto standar untuk *Orchestration* dan *Management Container*, sehingga penting untuk organisasi memahami teknologi Kubernetes ini.

### **1.2 Rumusan Masalah**

1. Apa itu Metode Kubernetes?
2. Kelebihan Kubernetes?
3. Mengapa Kubernetes Dibutuhkan?

### **1.3 Tujuan Penelitian**

1. Mengetahui apa itu Metode Kubernetes
2. Mengetahui apa saja kelebihan Kubernetes
3. Mengetahui mengapa Kubernetes dibutuhkan

## **BAB II**

### **Pembahasan**

#### **2.1 Pengertian Kubernetes**

Kubernetes adalah platform open source untuk mengelola kumpulan kontainer dalam suatu cluster server. Platform ini pertama kali dikembangkan oleh Google dan kini dikelola oleh Cloud Native Computing Foundation (CNCF) sebagai platform manajemen kontainer yang cukup populer.

Kontainer sendiri adalah environment dengan sumber daya, CPU, dan sistem file untuk satu aplikasi. Jadi, aplikasi tersebut akan memiliki sumber daya sendiri. Keuntungannya, aplikasi jadi tidak mudah downtime. Kubernetes memiliki kemampuan untuk melakukan penjadwalan aplikasi, load balancing server dan peningkatan kapasitas kontainer secara otomatis.

#### **2.2 Kelebihan Kubernetes**

Ada beberapa kelebihan saat menggunakan kubernetes diantaranya adalah.

a. Service Discovery dan Load Balancing

Fitur service discovery memudahkan untuk melacak kontainer dengan otomatis. Hal ini tentu sangat penting di dalam mengembangkan aplikasi microservices. Kubernetes dapat mengenali sebuah service berdasar DNS atau IP address server tersebut. Berkat Load Balancing pengelolaan trafik menjadi lebih mudah saat terjadi trafik yang cukup besar, Kubernetes mampu membagi beban secara merata sehingga membuat aplikasi menjadi stabil.

b. Storage Orchestration

Kubernetes memungkinkan anda melakukan mount pada media penyimpanan (storage) pilihan anda seperti storage lokal atau yang berbasis Cloud seperti AWS dan lainnya.

c. Automated Rollouts and Rollbacks

Fitur automated rollouts dan rollback sangat membantu anda terkait deployment. Pada Kubernetes, deployment bisa menggunakan file YAML yang didalamnya nanti akan berisi ReplicaSet. Apabila terjadi kendala pada deployment, dan anda masih memiliki ReplicaSet dari versi sebelumnya, anda dapat menggunakannya untuk melakukan Rollback.

d. Automatic Bin Packing

Menggunakan fitur Automatic Bin Packing, anda bisa mengatur kapasitas CPU dan sumber daya tiap kontainer secara spesifik, jadi saat limit kapasitas sudah ditentukan, aplikasi terhindar dari berebut sumber daya, selain itu sumber daya jadi lebih hemat.

e. Self Healing

Fitur Self Healing sangat penting bagi anda yang ingin terus menjalankan aplikasi selama 24 jam. Kubernetes memiliki kemampuan untuk memeriksa kontainer yang ada, apakah dalam keadaan running atau mengalami error. Jika terjadi kendala pada kontainer, Kubernetes akan menghentikan proses yang berjalan dan memberikan opsi untuk merestart atau mengganti secara otomatis kontainer yang error tersebut.

f. Secret and Configuration Management

Kubernetes memungkinkan anda menyimpan data-data sensitif seperti password, auth token hingga SSH keys kedalam Kubernetes Secret. Hal ini tentu jauh lebih aman dibanding menyimpannya di dalam container image. Secret bisa dibuat oleh sistem atau oleh user, karena secret sendiri secara default disimpan tanpa terenkripsi, anda bisa melakukan konfigurasi lebih lanjut sesuai dengan kebutuhan, antara lain opaque (secret umum/default), docker registry (untuk otentikasi docker registry), dan TLS (untuk penggunaan dengan public/private keys).

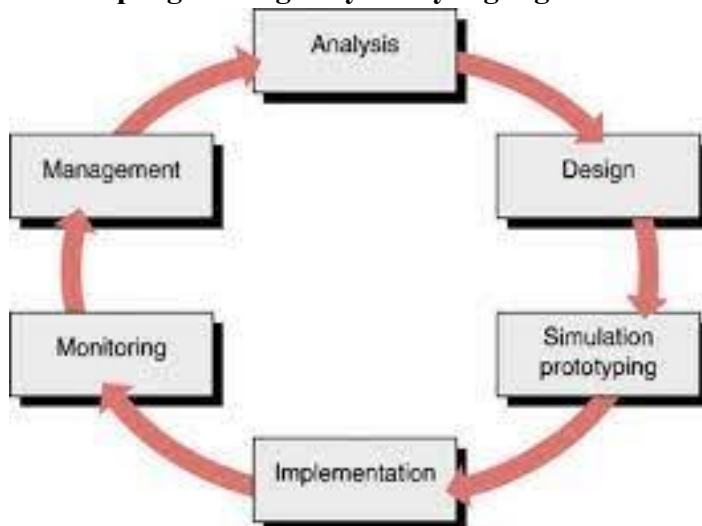
## **2.3 Mengapa Kubernetes Dibutuhkan**

Kubernetes bisa melakukan penjadwalan serta menjalankan container pada aplikasi di kelompok mesin virtual juga fisik. Tetapi, aplikasi cluster management open source ini dapat dijadikan pengembangan terhadap “cut the cord” menuju mesin virtual maupun fisik. Pergerakan dimulai dari infrastruktur host-sentris menuju infrastruktur container. Kubernetes ini juga menyediakan atau memberikan infrastruktur demi membantu membangun berbagai lingkungan untuk pengembangan dalam kontainer-sentris. Disini anda sebagai pengguna sudah tidak perlu menginstall banyak server yang didalamnya ada banyak environment, aplikasi, juga kebutuhan yang lain, sebab semuanya yang dibutuhkan oleh aplikasi sudah dipasang pada kontainer.

## BAB III CONTOH METODE

Metode yang digunakan oleh kubernetes disini adalah Network Development Life Cycle (NDLC). NDLC merupakan sebuah metode yang bergantung pada proses pembangunan sebelumnya seperti perencanaan strategi bisnis, daur hidup pengembangan aplikasi, dan analisis pendistribusian data.

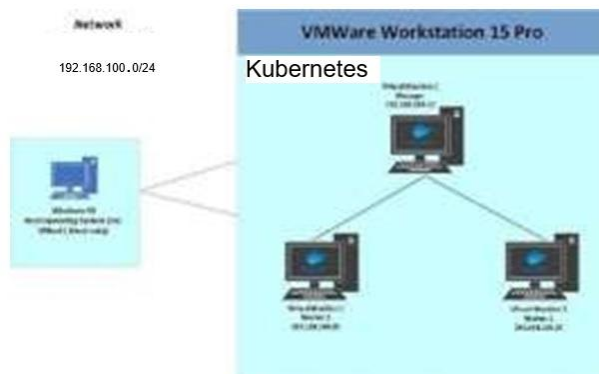
### 3.1 Metode pengembangan system yang digunakan adalah NDLC



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

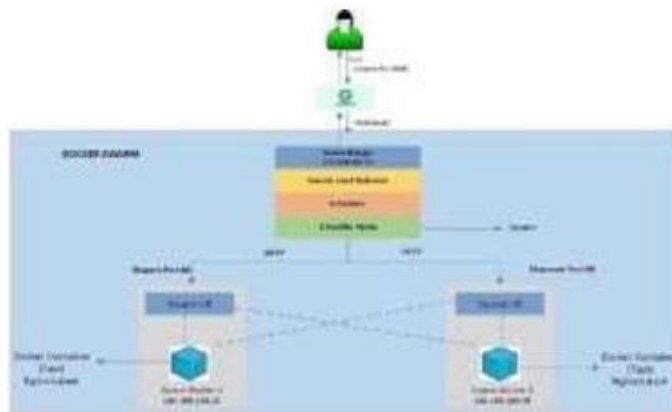
Pada fase Analysis penulis melakukan pengumpulan data dengan cara studi literatur, yaitu penulis membaca artikel ilmiah dan jurnal untuk mendapatkan informasi mengenai Kubernetes, Virtualisasi berbasis Container, Web Service Nginx serta Ftp server, data yang telah terkumpul kemudian dianalisis.

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.





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



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 disebarakan pada nodenode worker kemudian pada node master terdapat

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

## **BAB VI**

### **Penutup**

#### **4.1 Perbandingan**

##### *Metode WaterFall*

metode ini, kita membutuhkan banyak dokumentasi serta struktur di awal pembuatan. Prosesnya dibagi dalam beberapa langkah dan tahapan yang mandiri. Tahap pertama merupakan tahapan yang sangat penting, Pemahaman penuh dari pengembang dan pengguna mengenai ruang lingkup dan tuntutan proyek sangat dibutuhkan sebelum memulai segala sesuatu.

##### *Metode Prototype*

metode yang memicu pengembang hanya membuat contoh resolusi guna secara resmi menunjukkan esensi fungsional produk kepada pengguna. Pengembang akan melakukan berbagai perubahan yang diperlukan sesuai dengan permintaan pengguna. Setelah sampel disepakati, pengembang baru akan membuat produk aslinya sebagai hasil akhir dari proyek

##### *Metode RAD*

pengembangan yang ringkas untuk menghasilkan sistem dengan kualitas tinggi dengan biaya investasi rendah. Biaya pada metode ini dapat ditekan karena memiliki kemampuan untuk menyesuaikan diri dengan cepat. Untuk tahapan desain pengguna dan konstruksi akan terus diulang sampai pengguna menyatakan bahwa hasilnya sudah sesuai dengan keinginan mereka

##### *Metode Agile*

Lebih menghargai hubungan dan interaksi antar pribadi, tidak hanya memperdulikan sarana (tools). Menampilkan kerjasama dengan pengguna selama proses pengembangan berlangsung. Memberi tanggapan terhadap munculnya perubahan; tidak hanya melulu mengikuti rencana yang sudah ditetapkan. Fokus untuk menampilkan perangkat lunak yang benar-benar berfungsi; bukan hanya sekedar mementingkan dokumentasi.

##### *Metode DevOps*

Metode ini berpusat pada perubahan yang meningkatkan kerjasama antar departemen yang bertanggung jawab terhadap segmen yang berbeda pada pengembangan siklus hidup organisasi. Siklus tadi meliputi pengembangan, jaminan kualitas serta pelaksanaan atau operasi.

## **4.2 Kesimpulan**

Dari pembahasan dalam penelitian ini maka kesimpulan yang dapat diambil adalah sebagai berikut : Agile bermanfaat bagi proses pengembangan proyek dan mendorong komunikasi baik secara horizontal maupun vertikal di sepanjang siklus pengembangan proyek tersebut. Di dalam agile pula dapat meningkatkan inovasi melalui performa yang tinggi , lalu disiplin yang kuat antar tim yang memastikan nilai nilai bisnis dengan keterlibatan client secara langsung. Scrum membantu dalam mengefektifkan komitmen dalam sprint ini. Ini juga memperkenalkan batas WIP sebagai mekanisme untuk siklus perubahan yang selanjutnya. Dengan metode ini membantu banyak perangkat proyek untuk lebih terkontrol seperti biaya dll

## **Daftar Pustaka**

- **(ANALISA PENERAPAN SERVER DEPLOYMENT MENGGUNAKAN KUBERNETES UNTUK MENGHINDARI SINGLE OF FAILURE) Lilik Widyawati<sup>1\*</sup> , Heroe Santoso<sup>2</sup> , Hamdika Budiman<sup>3</sup>, JINTEKS (Jurnal Informatika Teknologi dan Sains) ISSN 2686-3359 (Online) Vol. 3 No. 1, Februari 2021, hlm. 267 – 271, 1,3 Ilmu Komputer, Universitas Bumigora 2Rekayasa Perangkat Lunak Aplikasi, Universitas Bumigora email: lilikwidya@universitasbumigora.ac.id\***