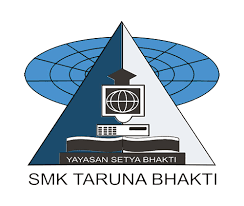
**MODUL DOCKER**

**Guru Pembimbing : Pak Fajar**

****

**Disusun Oleh :**

**M. Sandi Haikal**

**XI RPL 2**

**SMK TARUNA BHAKTI**

**2021**

Jln. Pekapuran Kel. Curug . Kec. Cimanggis Depok

# KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa. Atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan tugas yang berjudul "DOCKER" dengan tepat waktu.

Modul disusun untuk memenuhi tugas Mata Pelajaran Komputer Jaringan Dasar. Selain itu, tugas ini bertujuan menambah wawasan tentang computer jaringan dasar bagi para pembaca dan juga bagi penulis.

Penulis mengucapkan terima kasih kepada Pak Fajar selaku Guru Komputer Jaringan Dasar. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu diselesaikannya tugas ini.

Penulis menyadari makalah ini masih jauh dari sempurna. Oleh sebab itu, saran dan kritik yang membangun diharapkan demi kesempurnaan makalah ini.

Depok, 13 Januari 2022

Penulis

# DAFTAR ISI

KATA PENGANTAR i

DAFTAR ISI i

BAB I PENDAHULUAN 1

1.1 Latar Belakang Masalah 1

1.2 Rumusan Masalah 1

1.3 Tujuan 1

BAB II PEMBAHASAN 2

2.1 Pengertian Docker 2

2.2 Cara Kerja Docker 2

2.3 Kelebihan Docker 3

2.4 Kekurangan Docker 3

2.5 Alasan Penggunaan Docker 3

2.6 Kegunaan Docker 4

PENUTUP 5

3.1 Kesimpulan 5

# BAB I PENDAHULUAN

## Latar Belakang Masalah

Di era globalisasi ini semakin pesatnya perkembangan teknologi di dunia. Teknologi ini dimanfaatkan dan dikembangkan untuk dapat membantu pekerjaan mereka sehingga dapat menyelesaikan pekerjaan dengan mudah dan efisien.

Dalam membangun program, pengembang biasanya menjalankan virtualisasi pada server sehingga proses pembuatan program dapat berjalan pada berbagai platform maupun konfigurasi hardware. Masalah yang dihadapi dengan virtualisasi adalah perlunya penyiapan satu sistem operasi secara utuh, dan juga berbagai aplikasi bawaan sistem tersebut. Bisa dibayangkan dengan tak terhingganya virtualisasi yang berjalan di sebuah server dapat memberatkan kinerja sistem tersebut.

Terciptanya teknologi container dapat membawa beberapa perubahan, sebuah program berserta library-nya, file konfigurasi dan seluruh hal yang dibutuhkannya dapat terikat dalam container.

Docker adalah salah satunya platform yang dibangun berdasarkan teknologi container. Docker juga meupakan project open-source yang menyediakan platform terbuka bagi developer dan sysadmin untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun sebagai sebuah wadah(container) yang ringan. Mengenal Portainer merupakan satu platform open-source yang berguna sebagai manajemen Docker API, portainer adalah aplikasi ringan yang memungkinkan developer dapat mengelolah container, Docker image, jaringan dan volume masingmasing container melalui web dashboard sederhana.

## 1.2 Rumusan Masalah

1. Pengertian Docker  
2. Cara Kerja Docker  
3. Kelebihan Docker  
4. Kekurangan Docker

5. Alasan Penggunaan Docker

6. Kegunaan Docker

## 1.3 Tujuan

1. Mengetahui Pengertian Docker  
2. Mengetahui Cara Kerja Docker  
3. Mengetahui Kelebihan Docker  
4. Mengetahui Kekurangan Docker

5. Mengetahui Alasan Penggunaan Docker

6. Mengetahui Kegunaan Docker

# BAB II PEMBAHASAN

## 2.1 Pengertian Docker

Dilansir dari Docker.com, Docker adalah platform untuk mengembangkan, mengirim, dan menjalan aplikasi yang mulai berkembang pada tahun 2013. Docker memungkinkan pengembang perangkat lunak untuk memisahkan aplikasi dari infrastruktur sehingga pengembang dapat mengirimkan perangkat lunak dengan cepat.

Docker dapat membantu pengembang untuk mengelola infrastruktur teknis dengan cara yang sama seperti saat pengembang mengelola aplikasi. Pemanfaatan metode docker untuk pengiriman, pengujian, dan penerapan kode dengan cepat. Selain itu, pengembang dapat secara signifikan mengurangi penundaan antara penulisan kode dan proses menjalankannya dalam produksi perangkat lunak

Secara umum, Docker adalah proyek open source yang menyediakan platform terbuka untuk para website developer atau pengembang agar dapat membangun,mengemas, dan menjalankan aplikasi di berbagai lokasi dan bertindak sebagai sebuah container yang ringan. Docker memiliki fungsi utama dalam penyederhanaan konfigurasi yang dibangun berdasarkan teknologi container.

## 2.2 Cara Kerja Docker

Docker bertindak sebagai alat yang digunakan untuk menjalankan container. Container ini bertindak seperti mesin virtual, yang seperti simulasi komputer yang berjalan di dalam komputer asli pengembang. Pada mesin virtual ini nantinya semua kode sistem tersimpan untuk menjalankan simulasi seolah-olah adalah operasi sistem utama. Docker bertindak melakukan virtualisasi sistem operasi di dalam sistem operasi host.

Docker membangun container berdasarkan gambar yang berisi kode program. Gambar atau images ini ditumpuk satu sama lain untuk lantas membangun pengaturan yang lengkap. Gambar bertumpuk dapat berbagi gambar inti yang sama, seperti cabang-cabang dari batang pohon yang sama.

Proses ini dapat dimisalkan ketika seorang pengembang ingin menguji tampilan situs web baru di sebuah browser web berbeda, tetapi pengembang tidak ingin langsung melakukan instalasi setiap browser ke komputernya. Melakukan hal ini dapat menyebabkan masalah dengan peramban pribadi pengembang. Dalam pengujian inilah nantinya Docker berguna sebagai tempat uji coba.

## 2.3 Kelebihan Docker

1. Penghematan biaya dan melancarkan return of investment.
2. Penyebaran cepat.
3. Keamanan tinggi.
4. Kesederhanaan dan konfigurasi lebih cepat.

## Kekurangan Docker

1. Ada fitur yang hilang.
2. Data dalam container masih perlu di-back up.
3. Kompatibilitas lintas platform tidak fleksibel, jika dirancang di Windows maka tidak dapat dijalankan di Linux.
4. Grafis antarmuka dan jalankan aplikasi tidak fleksibel karena berbasis aplikasi hosting.

## Alasan Penggunaan Docker

Docker memungkinkan penggunaan sumber daya sistem yang lebih efisien. Contoh aplikasi dalam container menggunakan memori yang jauh lebih sedikit daripadai mesin virtual (VM’’s). Docker akan memulai dan berhenti lebih cepat karena dapat dikemas jauh lebih padat pada perangkat keras host mereka. Alasan inilah yang menjadi basis utama mengapa Docker dapat membuat tim pengembang/teknologi mengeluarkan biaya lebih sedikit.

Penghematan biaya merupakan alasan utama untuk menggunakan Docker karena variasi biayanya tergantung pada aplikasi apa yang sedang dipakai dan seberapa intensif sumber dayanya. Sebagai tambahan, kinerja container juga lebih efisien. Tak hanya itu, biaya lisensi perangkat lunak juga lebih hemat saat menggunakan Docker.

Docker memungkinkan siklus pengiriman perangkat lunak lebih cepat dan juga penempatan versi perangkat lunak terbaru dengan fitur bisnis terbaru akan lebih cepat membantu produksi. Bahkan, Docker memungkinkan pengembang untuk mengembalikan versi terbaru ke versi sebelumnya. Hal ini tentunya juga mendukung portabilitas aplikasi karena docker merangkum semua yang dibutuhkan aplikasi untuk dijalankan dan dapat dipindahkan dengan mudah.

## 2.6 Kegunaan Docker

1. Produktivitas pengembang
2. Manajemen pipeline kode
3. Penyederhanaan konfigurasi
4. Kecepatan Penyebaran
5. Multi-tenance
6. Dapat digunakan untuk Debugging
7. Melakukan konsolidasi Server

# PENUTUP

## 3.1 Kesimpulan

Docker dapat meng-optimalisasi infrastruktur IT perusahaan anda dan juga meningkatkan kinerja pada programer serta sistem administrator. Dengan sistem kontainer, seluruh pekerjaan pebuatan aplikasi, modul, monitoring, serta manajemen infrastruktur dapat lebih cepat, efisien dan lebih aman.Dalam hubungannya dengan komputasi awan, manfaat yang disebutkan di atas menunjukkan bagaimana Docker adalah sebuah platform open source yang efektif.

Perusahaan di Indonesia semakin banyak yang mengadopsi Docker, terutama para penyedia jasa teknologi informasi dan perusahaan besar yang memiliki data center sendiri. Demikian diharapkan untuk instansi pemerintahan, keunggulan docker merupakan solusi penghematan anggaran belanja negara. Organisasi selalu mengevaluasi basis platform terhadap kemampuan mereka untuk memberikan fleksibilitas, kebebasan dan mengatasi siklus hidup aplikasi, baik untuk pengembangan maupun tim operasional TI.