

## **TUGAS UAS JARKOM**

### **INSTALL WORDPRESS DI DOCKER DENGAN LEMP (Linux, Nginx, Mariadb, Php)**



Disusun Oleh :

Dio Aji Rismawan (1914311001)  
Fachri Nurcahyo (1914311007)  
Aga Bagus Firmansah (1914311010)  
Rivaldi Maulana Arfandi (1914311021)  
Alfian Pangestu (1914311027)  
Baghas Wahyu Pangestu (1914311046)  
Moch Saffana w.p (1914311074)

**FAKULTAS TEKNIK**

**PRODI INFORMATIKA**

**UNIVERSITAS BHAYANGKARA SURABAYA**

**2019/202**

## ABSTRAK

Wordpress merupakan sebuah situs yang terdiri dari halaman-halaman yang memuat berbagai tulisan dalam format teks yang dapat disisipi gambar dan konten lainnya. Manajemen sistem konten / artikel secara gratis ini tersedia cukup banyak, beberapa diantaranya yang paling populer adalah blogger dan wordpress. Blog digemari banyak orang karena gartis dan kemudahan untuk melakukan konfigurasinya. Secara umum, artikel ini terbagi dalam dua bagian, yaitu bagian 1 menjelaskan instalasi membuat kontainer pada Docker di terminal windows (Command Prompt). 2 menjelaskan cara instalasi OS pada terminal windows dan terminal pada docker atau CLI, Bagian 1 diawali dengan proses instalasi untuk kontainer pada docker.

Bagian 2 diawali dengan cara menjalankan docker. Lalu dijelaskan mengenai cara menjalankan dengan perintah perintah pada terminal windows dan CLI pada kontainer . Berikutnya dibahas cara instalasi linux,nginx,php,mariadb dan cara konfigurasinya untuk memberikan akses hanya kepada user tertentu saja. Proses instalasi merupakan suatu proses yang penting dalam awal pembuatan suatu sistem, oleh karena itu diperlukan suatu keamanan yang dilakukan dari awal instalasi. Untuk mengamankan segala aplikasi desktop yang tidak diperlukan, maka proses instalasi menggunakan Ubuntu server yang hanya terdiri dari terminal saja. Hasil pengujian diperoleh kesimpulan bahwa tujuan dari perancangan aplikasi ini tercapai yaitu aplikasi ini layak digunakan untuk pengguna umum yang khususnya untuk pengguna dengan latar belakang baru mengenal web server.

# DAFTAR ISI

|  |     |
|--|-----|
| <b>ABSTRAK</b> .....   | i   |
| <b>Daftar Isi</b> .....  | ii  |
| <b>KATA PENGANTAR</b> .....  | iii |
| <b>BAB I PENDAHULUAN</b> .....                                       | 1   |
| <b>Latar Belakang</b> .....  | 1   |
| <b>Rumusan Masalah</b> .....   | 3   |
| <b>Tujuan Pembahasan</b> .....                                       | 3   |
| <b>BAB II PEMBAHASAN</b> .....                                       | 4   |
| <b>Pengertian Docker</b> .....                                       | 4   |
| <b>Langkah-langkah install ubuntu di kontainer pada Docker</b> ..... | 4   |
| <b>Langkah-langkah install LEMP (Linux,Nginx,Mariadb,Php)</b> .....  | 5   |
| <b>Membuat Database untuk Wordpress</b> .....                        | 6   |
| <b>BAB III PENUTUP</b> .....   | 7   |
| <b>Kesimpulan</b> .....  | 7   |
| <b>Saran</b> .....   | 7   |

## **KATA PENGANTAR**

Segala puji bagi Tuhan yang telah menolong hamba-Nya menyelesaikan makalah ini dengan penuh kemudahan. Tanpa pertolongan Dia mungkin penyusun tidak akan sanggup menyelesaikan dengan baik makalah ini. Makalah ini disusun agar pembaca dapat memperluas ilmu dalam menggunakan fasilitas internet yaitu blog yang kami sajikan berdasarkan pengamatan dari berbagai sumber. Makalah ini di susun oleh penyusun dengan

berbagai rintangan. Baik itu yang datang dari diri penyusun maupun yang datang dari luar. Namun dengan penuh kesabaran dan terutama pertolongan dari Tuhan akhirnya makalah ini dapat terselesaikan.

Makalah ini memuat tentang blog yang saat ini sedang menjamur diberbagai kalangan masyarakat sebagai fasilitas internet secara gratis. Penyusun juga mengucapkan terima kasih kepada dosen Jaringan Komputer yang telah membimbing penyusun agar dapat menyelesaikan makalah ini. Semoga makalah ini dapat memberikan wawasan yang lebih luas kepada pembaca. Walaupun makalah ini memiliki kelebihan dan kekurangan. Penyusun mohon untuk saran dan kritiknya. Terima kasih.

Surabaya, 1 juli 2020

# BAB I PENDAHULUAN

## Latar Belakang

Perkembangan aplikasi berbasis web sangat pesat, seiring dengan perkembangan komputer dan internet. Selain itu, aplikasi berbasis web juga semakin banyak digunakan karena dapat diakses di berbagai platform komputer hanya dengan menjalankan peramban web. Sehingga, kemudahan proses deployment (penyebaran) aplikasi web beserta perangkat lunak pendukung seperti server web, server basis data, dependencies dan environment lain ke server sangat dibutuhkan. Pengguna yang memiliki layanan aplikasi berbasis server-klien (Aplikasi Web) banyak menggunakan layanan cloud computing untuk hosting aplikasinya, terutama aplikasi yang bersifat publik yang dapat diakses setiap saat. Dengan kata lain pengguna tersebut menyewa layanan cloud pada sebuah Penyedia Layanan Internet (PLI) yang berupa infrastruktur (Pusat Data) atau platform. Hal yang jadi pertimbangan untuk menggunakan cloud computing dibandingkan komputasi tradisional adalah efektivitas, keamanan, skalabilitas, kolaborasi dan lebih murah.

Secara umum ada dua metode deployment aplikasi web ke dalam server. Pertama menginstal aplikasi web beserta environment yang dibutuhkan ke dalam server tunggal, kelebihanannya adalah melakukan konfigurasi terhadap server lebih mudah, sederhana dan cepat dalam proses deployment. Tetapi metode tersebut memiliki kekurangan yaitu setiap aplikasi tidak terisolasi, sehingga apabila mendeploy beberapa aplikasi yang masing-masing memiliki ketergantungan dengan paket versi tertentu maka dapat menimbulkan Dependencie Hell (Cross-platform Dependencies, Conflicting Dependencies dan Custom Dependencies) .

Metode yang kedua yaitu dengan memanfaatkan teknologi virtualisasi berbasis hypervisor, jadi setiap aplikasi dan dependency yang dibutuhkan di-deploy ke dalam mesin virtual yang berbeda. Dengan metode ini dapat meningkatkan skalabilitas, karena 2 setiap aplikasi berjalan pada sumber daya (CPU, memori, penyimpanan data) yang berbeda sehingga dapat dengan mudah ditambahkan sesuai kebutuhan .

Akan tetapi masalah klasik menjalankan mesin virtual berbasis hypervisor adalah membutuhkan sumber daya yang besar. Karena setiap mesin virtual menjalankan sistem operasi guest beserta kernel-nya sendiri terpisah dari host.

Sehingga ketika menjalankan aplikasi yang mungkin besarnya hanya puluhan MB, mesin virtual juga harus menjalankan sistem operasi guest yang besarnya bisa mencapai 10 GB .

Sebab itu dibutuhkan teknologi yang dapat menyediakan virtualisasi yang mengisolasi aplikasi beserta environment yang dibutuhkan dengan kebutuhan sumber daya minimal yang dapat berjalan di berbagai infrastruktur untuk memudahkan proses deployment aplikasi. Muncul sebuah perangkat lunak Platform as a Service yang dapat mengatasi permasalahan tersebut yaitu Docker. Docker merupakan perangkat lunak virtualisasi jenis Operating systemlevel virtualization berbasis LXC (Linux Container) yang memungkinkan pengembang aplikasi membuat, menguji dan menjalankan aplikasi dalam sebuah lingkungan yang berbeda, terisolasi dan fleksibel. Tujuan utama dari virtualisasi ini adalah untuk memudahkan para pengembang dan sistem administrator aplikasi berbasis server-klien (Aplikasi Web) dalam proses pengembangan, uji coba dan pemaketan kode program dan hosting .

Docker sangat ringan dan mempunyai mekanisme yang lebih maju jika dibandingkan dengan perangkat lunak virtualisasi berbasis hypervisor seperti VirtualBox atau KVM. Indikasinya adalah adanya efektivitas lebih pada Docker dalam hal penggunaan sumber daya mesin host, karena dalam proses deploymentnya Docker akan menjalankan sebuah container menggunakan base image dengan metode file system as a layer yang berarti Docker hanya akan menyalin lapisan perubahannya saja untuk dijalankan sebagai duplikasi container yang berbeda dengan base image yang sama. Berbeda dengan virtualisasi tradisional di mana kita harus menyalin seluruh container beserta kernel dan library yang berada di dalamnya, hal ini mengakibatkan adanya penghematan penyimpanan dan memori pada Docker .

Kelebihan yang lain adalah Docker merupakan perangkat lunak yang berlisensi open source. 3 Saat ini pengguna yang memiliki aplikasi berbasis web banyak menggunakan Content Management System (CMS) sebagai perangkat lunak yang dapat mempermudah pengelolaan konten dalam sebuah website. CMS sendiri adalah perangkat lunak yang digunakan untuk menambah atau memanipulasi (mengubah) isi dari suatu situs web. Diambil dari laman web perusahaan BuiltWith persentase pengguna website yang menggunakan CMS hingga tanggal 10 Oktober 2016 adalah sebagai berikut: WordPress 38%; Drupal 8%; Google search Appliance 3%; Adobe CQ 3%; DNN Software 1%; CPANEL 1%; Joomla! 1%; Blogger 1%; vBulletin 1%; Lain-lain 43% .

### **Rumusan Masalah**

- A. Bagaimana konfigurasi membuat web pada docker itu sendiri?
- B. Apa saja yang digunakan didalam kontainer pada docker itu?
- D. Bagaimana langkah membuat wordpress pada sistem itu?

### **Tujuan Pembahasan**

- A. Memahami tentang Docker
- B. Mengetahui cara menggunakan docker
- C. Dapat membuat web (wordpress) di docker

## BAB II PEMBAHASAN

### Pengertian Docker

Memahami konsep mesin virtual akan membantu kita mengenal container dengan mudah. Sama seperti mesin virtual, container menawarkan keamanan dengan secara serentak menjalankan instance terpisah dari sistem operasi tanpa adanya interaksi antara satu sama lain. Pun sama halnya dengan mesin virtual, container meningkatkan portabilitas & fleksibilitas proyek, sebagaimana Anda tidak bergantung pada hardware spesifik mana pun, dan dapat bermigrasi ke cloud, local environment mana saja, dan lain-lain.

Berbeda dengan mesin virtual, yang membutuhkan instalasi sistem operasi kaya fitur bersamaan dengan besarnya overhead, container membagikan kernel sistem operasi tunggal selagi tetap mempertahankan isolasi dengan mematuhi container lainnya. Singkatnya, manfaat yang sama dari mesin virtual dapat Anda rasakan tanpa mengeluarkan biaya tambahan.

Docker menggunakan skema yang sama untuk membuat container di VM berbasis Linux. Di container Docker tunggal, Anda akan mendapatkan akses ke seluruh keperluan pengembangan: source code, dependency, dan runtime.

Docker sangat berguna bagi developer WordPress. Tak hanya sebagai inti untuk membuat lingkungan tes WordPress (database, dependency, server), tetapi juga membuang sistem resource yang bernilai, seperti ruang server, memori, dan lain-lain. Secara cepat, Docker mampu meningkatkan performanya dengan membuat lingkungan minimal dan bisa dibawa ke mana pun Anda suka.

### Langkah-langkah install ubuntu di kontainer pada Docker

1. Membuat container di docker dengan memasukkan perintah di command prompt, perintahnya `docker run -td --name wp -p 80:80 ubuntu` lalu tekan enter

**Note** :Wp bisa diganti dengan nama bebas

2. Memulai container dengan cara memasukkan perintah di command prompt, perintahnya : `docker exec -it wp /bin/bash`

**Note**: nama container disesuaikan dengan nama yang telah dibuat pada step nomor 1

3. Ketik `apt update` lalu enter



## Langkah-langkah install LEMP (Linux, Nginx, Mariadb, Php)

4. Menginstall LEMP dengan perintah `apt install mariadb-server mariadb-client php-fpm php-mysql nano wget zip net-tools -y` lalu enter
5. Install Nginx dengan input perintah `apt install nginx` lalu enter
6. Lalu pilih kode server benua dengan kode `6` lalu enter
7. Kemudian pilih kode server kota dengan kode `35` lalu enter
8. Memulai service nginx dengan perintah `service nginx start`
9. Untuk mengecek apakah nginx berjalan apa tidak, silahkan buka browser
10. ketik `localhost` lalu enter, maka terbuka homepage nginx
11. Mengubah konfigurasi nginx dengan cara `nano /etc/nginx/sites-available/default` lalu enter
12. Masukkan `index.php` pada line `add index.php to the list if you are using php` setelah syntax `index` dan sebelum `index.html`
13. Uncomment pada line
  - `location ~ \.php${`
  - `include snippets/fastcgi-php.conf;`
  - `fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;`
  - `}`
  - `location ~ /\.ht {`
  - `deny all;`
  - `}`
14. Lalu keluar dari text editor, dengan cara ketik perintah `ctrl+x` lalu `y` lalu tekan enter
15. Untuk mengecek apakah konfigurasi nginx berhasil ketik `nginx -t`
16. Restart nginx service dengan cara `service nginx restart`
17. Memulai layanan php dengan cara `service php7.4-fpm start` lalu tekan enter
18. Memulai service mariadb dengan perintah `service mysql start`
19. Untuk mengamankan mariadb dengan cara input perintah `mysql_secure_installation`
20. Ketika ada pemberitahuan ketik sesuai yang ada di bawah ini
  - Enter current password for root (enter for none): tekan Enter
  - Set root password? [Y/n]: `Y`
  - New password: `masukkan password bebas`
  - Re-enter new password: `ulangi password`
  - Remove anonymous users? [Y/n]: `Y`
  - Disallow root login remotely? [Y/n]: `Y`

- Remove test database and access to it? [Y/n]: **Y**
- Reload privilege tables now? [Y/n]: **Y**

## Membuat Database untuk Wordpress

21. Kemudian membuat database untuk wordpress dengan cara input perintah **mysql -u root -p** lalu enter dan masukkan password yang telah dibuat tadi

22. Langkah pertama membuat database yaitu dengan perintah **CREATE DATABASE wpdb;** tekan enter

**Note** : wpdb bisa diganti dengan nama bebas

23. Langkah selanjutnya membuat user pada database dengan perintah **CREATE USER wpuser@localhost IDENTIFIED BY 'password';** tekan enter

**Note** : wpuser dan password bisa diganti dengan nama bebas

24. Selanjutnya memberikan izin user ke database dengan perintah **GRANT ALL PRIVILEGES ON wpdb.\* TO wpuser@localhost;** lalu tekan enter

25. Lalu ketik **FLUSH PRIVILEGES;** lalu tekan enter

26. Terakhir keluar dari mariadb dengan perintah **EXIT;** lalu tekan enter

27. Restart service mariadb dengan cara **service mysql restart**

28. Kemudian membuka direktori tmp dengan cara perintah **cd /tmp**

29. Lalu download wordpress dengan cara input perintah **wget <https://wordpress.org/latest.zip>** lalu tekan enter

30. Setelah download selesai ekstrak file tersebut dengan cara input perintah **unzip latest.zip** lalu tekan enter

31. Setelah file ter-ekstrak ubah kepemilikan folder dengan cara **chown -R www-data:www-data /tmp/wordpress/** lalu tekan enter

32. Kemudian ubah perizinan folder dengan cara input perintah **chmod -R 755 /tmp/wordpress/** lalu tekan enter

33. Selanjutnya membuat direktori baru dalam folder wordpress dengan perintah **mkdir /tmp/wordpress/wp-content/uploads** lalu tekan enter

34. Selanjutnya ubah kepemilikan folder dengan cara **chown -R www-data:www-data /tmp/wordpress/wp-content/uploads** lalu tekan enter

35. Kemudian menghubungkan folder wordpress ke folder html dengan cara input perintah **ln -s /tmp/wordpress/ /var/www/html/** lalu tekan enter

36. Lalu silahkan buka browser lalu ketik **localhost/wordpress** lalu tekan enter

## **BAB III PENUTUP**

### **Kesimpulan**

Wordpress merupakan fasilitas internet dimana setiap orang bisa saling menerima dan mendapatkan informasi secara mudah. Wordpress merupakan hal yang bersifat umum dan terbuka. Kita sebagai pengguna perlu memperhatikan etika dalam menggunakan fasilitas web ini. Bagaimanapun banyaknya fasilitas internet yang meluas jangan sampai menambah sifat individualisme seseorang.

### **Saran**

Jika memberi tugas diberi tolong cara penggunaan