

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI FIST EFFECT MENGGUNAKAN UBUNTU LAMPP DAN
PEMBUATAN DOCKERFILENYA**



DISUSUN OLEH:

**NAMA ANGGOTA : DIO CAHYO SAPUTRA 123170033
BASRIZAL REZA ASTANA 123170089
KELAS : A
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.KOM.
MUHAMMAD IMAM ALFATAH**

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA
2020**

HALAMAN PENGESAHAN

SISTEM INFORMASI FIST EFFECT MENGGUNAKAN UBUNTU LAMPP DAN DOCKER

Disusun oleh :

Dio Cahyo Saputra

123170033

Basrizal Reza Astana

123170089

Telah diperiksa dan disetujui oleh Asisten Praktikum Teknologi Cloud Computing
pada tanggal :

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

Mangaras Yanu Florestiyanto, S.T., M.Eng.

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum teknologi cloud computing serta laporan proyek akhir praktikum yang berjudul sistem informasi fist effect menggunakan ubuntu lamp dan docker. Adapun laporan ini berisi tentang proyek akhir yang saya pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten dosen yang selalu membimbing dan mengajari saya dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun saya harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, saya ucapkan terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 13 Mei 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	iv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Proyek Akhir	1
1.2 Tujuan Proyek Akhir	2
1.3 Manfaat Proyek Akhir	3
1.4 Tahap Penyelesaian Proyek Akhir	3
BAB II ISI DAN PEMBAHASAN	4
2.1 Komponen yang Digunakan	4
2.1.1 Komponen pada Proyek Pertama.....	4
2.1.2 Komponen pada Proyek Kedua	5
2.2 Rancangan Arsitektur <i>Cloud Computing</i>	6
2.3 Parameter dan Konfigurasi	8
2.4 Tahap Implementasi.....	11
2.4.1 Tahap Implementasi pada Proyek Pertama.....	14
2.4.2 Tahap Implementasi pada Proyek Kedua	21
2.5 Hasil Implementasi	28
2.5.1 Hasil Implementasi pada Proyek Pertama	28
2.5.2 Hasil Implementasi pada Proyek Kedua	28
2.6 Pengujian Singkat	29
2.6.1 Pengujian Singkar pada Proyek Pertama	28
2.6.2 Pengujian Singkat pada Proyek Kedua	30
BAB III JADWAL Pengerjaan dan Pembagian Tugas	33
3.1 Agenda Pengerjaan	33
3.2 Keterangan Pembagian Tugas.....	33
BAB IV KESIMPULAN DAN SARAN.....	34
4.1 Kesimpulan	34
4.2 Saran	34
DAFTAR PUSTAKA.....	35
LAMPIRAN	36

BAB I

PENDAHULUAN

1.1 Latar Belakang Proyek Akhir

Cloud computing mungkin masih samar terdengar bagi orang awam. Tetapi keberadaan cloud computing di era digital kini sebenarnya telah terasa di tengah masyarakat dalam kehidupan sehari-hari seperti penggunaan email dan juga media sosial. Bicara tentang asal usul *cloud computing* sebenarnya tidak jelas, jika dilihat dari bahasa *cloud computing* adalah komputasi awan, sedangkan secara istilah adalah gabungan pemanfaatan teknologi komputer dan pengembangan berbasis pada "awan" (internet) (LaWaveDesign.com, 2020). *Cloud computing* memungkinkan untuk menyimpan data-data dan informasi secara aman dan terpusat pada salah satu server dan dalam mengakses data-data tersebut sangat lah mudah. Selain itu, dalam pembelian media penyimpanan lebih murah dan dapat digunakan sepanjang waktu. Menurut Wikipedia, arsitektur paling umum yang terdapat pada *cloud computing* ini antara lain pertama *Infrastructure as a Service* (IaaS) adalah layanan komputasi awan yang menyediakan infrastruktur IT berupa CPU, RAM, storage, bandwidth dan konfigurasi lain. Semua komponen tersebut digunakan untuk membangun komputer virtual. Kedua *Platform as a Service* (PaaS) adalah layanan yang menyediakan computing platform. Biasanya sudah terdapat sistem operasi, database, web server dan framework aplikasi agar dapat menjalankan aplikasi yang telah dibuat. Dan ketiga *Software as a Service* (SaaS) adalah layanan komputasi awan dimana kita bisa langsung menggunakan aplikasi yang telah disediakan. Penyedia layanan mengelola infrastruktur dan platform yang menjalankan aplikasi tersebut.

Dari sekian banyak *cloud developer* ada beberapa *cloud computing developer* diantaranya adalah ubuntu, microsoft azure, amazon, vmware dan masih banyak yang lain, namun diantara sekian banyak cloud developer ubuntu adalah salah satu cloud developer yang menyediakan *private cloud* secara gratis. Teknologi sistem web hosting yang high availability adalah dengan mengadopsi teknologi virtualisasi sistem operasi berbasis *Container* (Adiputra, 2015). Salah satu software yang mengadopsi konsep ini adalah LXC (*Linux Container*). Implementasi dalam LXC tersebut kemudian diadopsi lebih lanjut oleh Docker. Teknologi ini tidak membangun mesin virtual sendiri, lebih hemat memory, processor dan storage.

Adanya *cloud hosting* yang merupakan cara pengelolaan terhadap beberapa sumber daya server serta sumber daya pendukungnya dalam rangka mengoptimalkan sistem pengantar data ke pengguna sehingga akses menuju web ataupun aplikasi menjadi lebih stabil serta dapat diakses terus-menerus. *Cloud hosting* memungkinkan bagi pengguna untuk melakukan pengelolaan beban puncak secara lebih mudah tanpa menghadapi gangguan kekurangan bandwidth serta dapat mengurangi terjadinya *downtime* ketika terjadi gangguan kerusakan *server* karena sistem *cloud hosting* dapat berpindah ke sumber daya lain. Salah satu contoh teknologi *cloud hosting* adalah *Amazon Web Services*. 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 menyiapkan satu sistem operasi secara utuh, termasuk berbagai aplikasi yang dibawa sistem tersebut. Bisa dibayangkan dengan banyaknya virtualisasi yang berjalan di sebuah server akan memberatkan sistem tersebut. *Container* kemudian datang dan membawa beberapa perubahan. Dengan *container*, sebuah program ‘diikat’ beserta library-nya, file konfigurasi, dan seluruh hal yang dibutuhkannya. Perbedaan yang sangat terlihat dibandingkan dengan virtualisasi adalah *container* memiliki ukuran file yang jauh lebih kecil karena tidak perlu menyiapkan sistem operasi secara penuh. Aplikasi yang berjalan menggunakan *container* pun jauh lebih cepat dan lebih efisien.

Tahap rancangan yang akan dilakukan adalah penginstalan Ubuntu server pada salah satu laptop, dilanjutkan dengan menginstal dan mengkonfigurasi layanan hosting dengan LAMPP seperti Apache sebagai *webserver*, MySQL sebagai database, phpMyAdmin sebagai pengolahan database, dan PHP sebagai bahasa pemrograman web. Setelah terinstal semua file project akan diupload dan dilanjutkan dengan testing aplikasi web. Selanjutnya membuat sebuah *dockerfile*, menambahkan instruksi pada *dockerfile* tersebut, membangun *dockerfile* untuk membuat sebuah image lalu menjalankan image tadi untuk membuat *container*.

1.2 Tujuan Proyek Akhir

Berdasarkan latar belakang proyek akhir yang telah dijelaskan sebelumnya, mengenai tujuan dari pembuatan proyek akhir ini adalah sebagai berikut:

1. Mengimplementasikan arsitektur *cloud computing* untuk Sistem Informasi FIST Effect yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.4 dan LAMPP (Apache 8, PHP 7.2, MySQL 5.2).
2. Mengimplemtasikan platform docker untuk proses pemasangan dan konfigurasi Ubuntu LAMPP.

1.3 Manfaat Proyek Akhir

Manfaat yang dapat diperoleh dari pembuatan proyek akhir ini adalah sebagai berikut:

1. Menghemat biaya dalam penggunaan sistem informasi karena tidak memerlukan pembelian inventaris seperti infrastruktur.
2. Perusahaan tidak perlu memperlakukan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.
3. Memudahkan dan mempercepat dalam proses penginstalan dan konfigurasi Ubuntu LAMPP.

1.4 Tahap Penyelesaian Proyek Akhir

Tahapan secara singkat untuk penyelesaian proyek akhir ini adalah sebagai berikut:

1. Menganalisis kebutuhan dari sistem informasi FIST Effect untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS.
2. Penginstalan dan konfigurasi dua Ubuntu server, yang pertama untuk LAMPP (Apache, PHP, MySQL), dan yang kedua untuk docker.
3. Mengintegrasikan sistem informasi FIST Effect kedalam Ubuntu Server yang berisi LAMPP.
4. Menentukan konfigurasi Dockerfile yang tepat untuk proses penginstalan LAMPP yang digunakan beserta mengintegrasikan sistem informasi FIST Effect dengan Dockerfile tersebut.
5. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi pada setiap Ubuntu server.
6. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dsb... pada setiap Ubuntu Server
7. Menghasilkan Sistem Informasi FIST Effect yang berbasis *cloud computing* sesuai standar ISO 9001.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Yang pertama komponen yang digunakan dalam Sistem Informasi Fist Effect menggunakan Ubuntu LAMP adalah Apache2, PHP, dan MySQL. Yang kedua komponen yang digunakan dalam kontainer Sistem Informasi Fist Effect menggunakan Ubuntu LAMP adalah docker.

2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Sistem Informasi Fist Effect menggunakan Ubuntu LAMP” yang berbasis konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.2.24 dan bahasa HTML dengan versi minimal 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 10.4.6 sehingga dapat digunakan untuk menyimpan berbagai data informasi yang dibutuhkan oleh sistem tersebut.
3. Untuk target dapat diakses pada laptop/pc dengan jaringan yang sama.

Berdasarkan penjelasan poin-poin tersebut, untuk komponen utama penyusun *cloud computing* yang dibutuhkan dapat disimpulkan dalam bentuk tabel sebagai berikut:

Tabel 2.1 Spesifikasi VM *cloud computing* untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Network Address Translation	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.65.129/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.65.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.65.1	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	2 GB	Alokasi RAM untuk <i>guest OS</i>

Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

Tabel 2.2 Spesifikasi Ubuntu OS untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.2.24	Preprosesor Bahasa pemrograman PHP
		MySQL 10.4.6	Sebagai tempat penyimpanan data

2.1.2 Komponen pada Proyek Kedua

Untuk membangun kontainer pada “Sistem Informasi Fict Effect menggunakan Ubuntu LAMPP” yang berbasiskan konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Kontainer yang dibangun menggunakan docker, menggunakan docker versi 19.03.8.
2. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.2.24 dan bahasa HTML dengan versi minimal 4.0.
3. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 8.0 sehingga dapat digunakan untuk menyimpan berbagai data informasi yang dibutuhkan oleh sistem tersebut.
4. Setelah docker berjalan ditargetkan docker tersebut dapat dijalankan pada Ubuntu server yang berbeda..

Berdasarkan penjelasan poin-poin tersebut, untuk komponen utama penyusun *cloud computing* yang dibutuhkan dapat disimpulkan dalam bentuk tabel sebagai berikut:

Tabel 2.3 Spesifikasi VM *cloud computing* untuk proyek kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Network Address Translation	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.189.129/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.189.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.189.1	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	2 GB	Alokasi RAM untuk <i>guest OS</i>

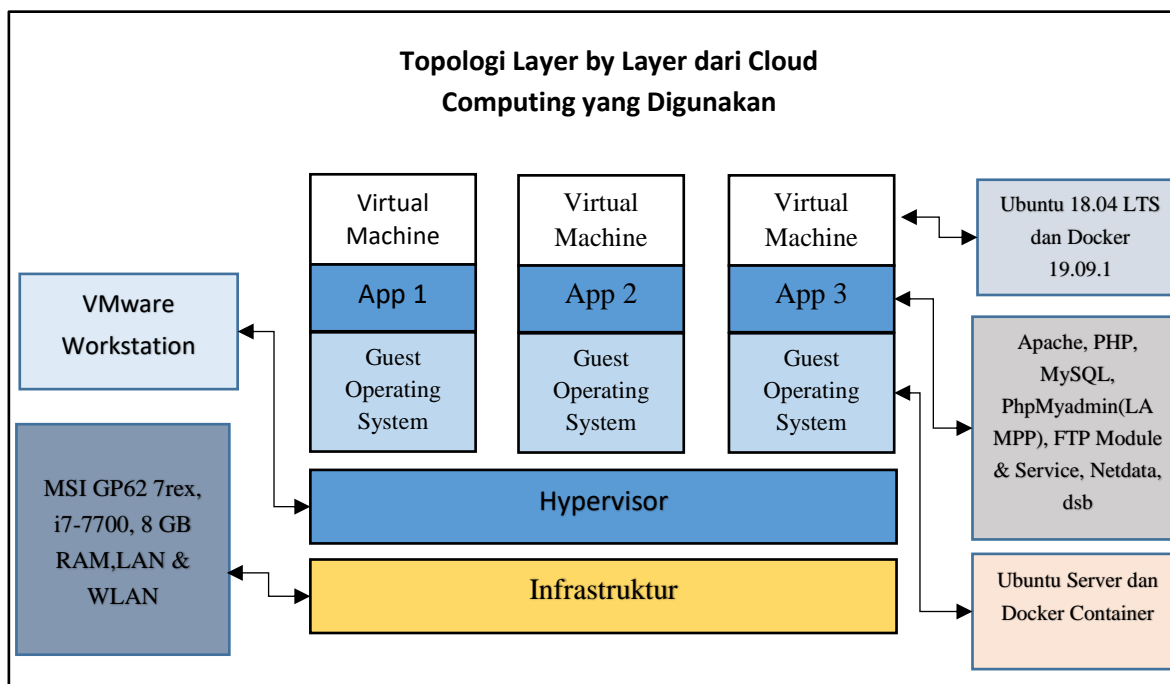
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

Tabel 2.4 Spesifikasi Ubuntu OS untuk proyek kedua

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Preprosesor Bahasa pemrograman PHP
		MySQL 8.0	Sebagai tempat penyimpanan data
		Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
2.	Docker	Versi 19.03.8	Digunakan untuk pembuatan kontainer
3.	Docker compose	Versi 1.21.2	Digunakan untuk menjalankan kontainer secara bersamaan

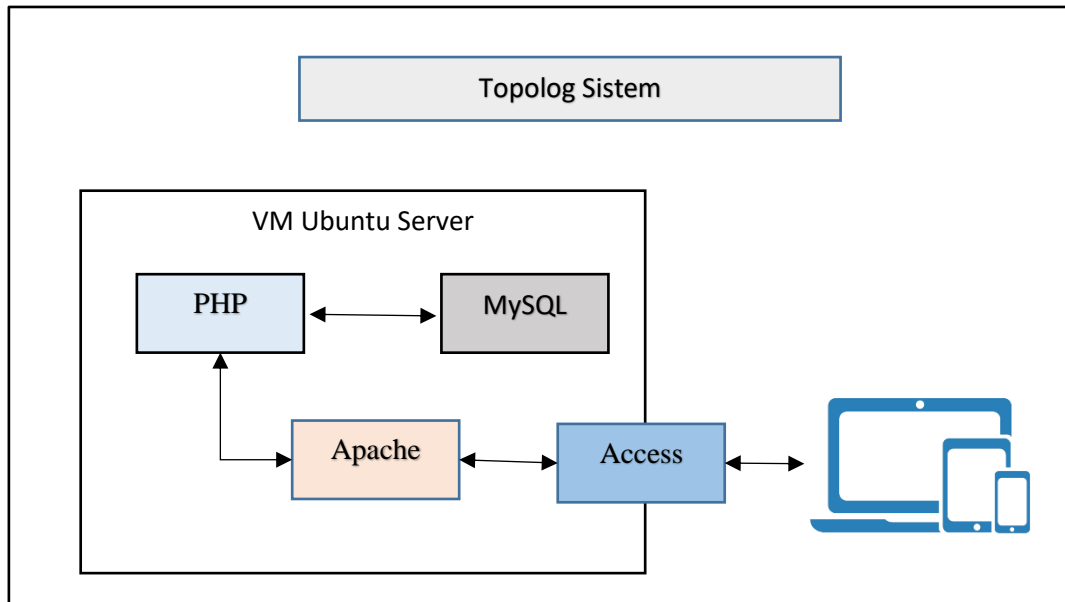
2.2 Rancangan Arsitektur Cloud Computing

Pada project akhir ini digunakan bentuk rancangan arsitektur IaaS di mana infrastruktur berupa hardware laptop MSI GP62 7rex sebagai layer utama berada di bagian paling bawah, Kemudian di atasnya adalah *Hypervisor* berupa VMware Workstation yang merupakan komponen dalam virtualisasi yang memungkinkan beberapa operating system untuk berjalan bersamaan pada sebuah host. Kemudian di atasnya lagi merupakan PaaS atau sistem operasi Ubuntu Server yang berjalan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



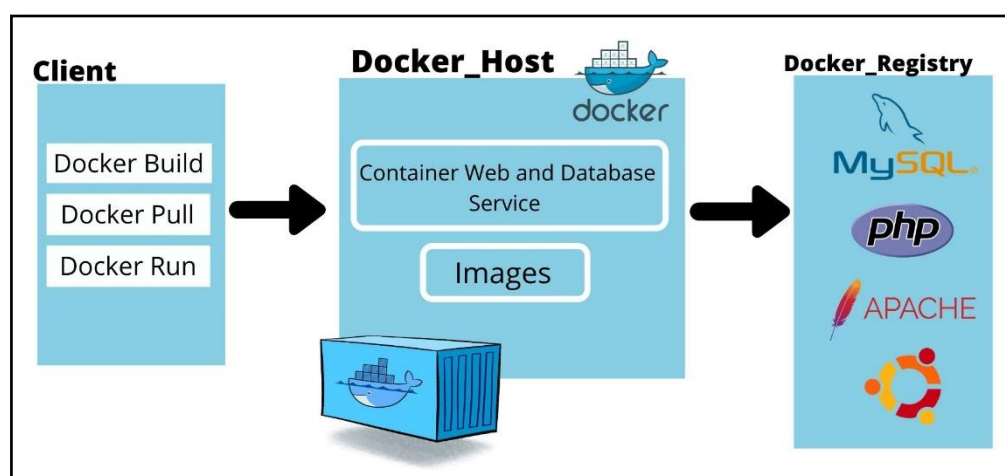
Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya

Virtual machine yang berjalan pada komputer atau laptop memiliki beberapa komponen aplikasi yang digunakan untuk menjalankan Apache, PHP, MySQL, dan PhpMyadmin (LAMPP) yang digunakan sebagai *hosting local*. Layanan secara *local* ini digunakan sebagai simulasi yang menggambarkan kondisi aslinya. Ilustrasi mengenai rancangan topologi sistem dapat dilihat pada **Gambar 2.2** berikut ini:



Gambar 2.2 Rancangan Topologi Sistem

Pada proyek akhir ini juga menggunakan teknologi *Docker*. *Docker* berfungsi sebagai wadah atau *container* untuk membungkus atau memasukkan aplikasi yang digunakan secara lengkap beserta hal-hal yang dibutuhkan lainnya sehingga dapat berjalan di perangkat mana saja. Berikut rancangan arsitektur teknologi *Docker* dapat dilihat pada **Gambar 2.3** berikut ini:



Gambar 2.3 Rancangan arsitektur teknologi Docker

Docker menggunakan arsitektur berbasis *client-server*. Dalam hal ini, *Docker client* mengirimkan permintaan berupa sebuah perintah kepada *Docker daemon* (yang berjalan pada *docker host*) untuk membangun, mendistribusikan, dan menjalankan *container Docker*. *Docker daemon* yang berjalan pada *docker host* tidak dapat berinteraksi secara langsung sehingga untuk mengaksesnya dibutuhkan *Docker client* yang merupakan tampilan utama untuk pengguna, sehingga pengguna dapat berkomunikasi dengan *Docker daemon*.

2.3 Parameter dan Konfigurasi

Untuk dapat digunakan, maka VMware Workstation perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada **Tabel 2.5** berikut ini:

Tabel 2.5 Konfigurasi pada VMware Workstation yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	VMware Workstaion <i>version</i>	15.5.2	Keterangan versi yang digunakan.
2.	VMware Workstation <i>build</i>	10737736	Keterangan <i>build (patch) number</i> .
3.	IPv4 <i>hypervisor</i>	IP: 192.168.65.1	IP <i>number</i> untuk akses manajemen <i>hypervisor</i> melalui <i>web based</i> . Pengaturan IP <i>address</i> menggunakan mode statik.
		SM: 255.255.255.0	Kelas IP/ <i>subnet mask</i> yang digunakan.
		DNS: 192.168.65.1	Alamat IP untuk fungsionalitas DNS.
4.	<i>Processor info</i>	12 x Intel (R) Core (TM) i7-8700 CPU @ 4.2GHz	Jenis <i>processor</i> yang digunakan pada <i>hypervisor</i> .
5.	<i>RAM info</i>	8 GiB Memory	Kapasitas RAM pada <i>hypervisor</i> .

Untuk penginstalan Ubuntu melalui VMware Workstation menggunakan konfigurasi seperti pada **Table 2.6** berikut ini:

Tabel 2.6 Konfigurasi pada Ubuntu yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	Ubuntu <i>version</i>	18.04.3	Keterangan versi yang digunakan.
2.	RAM <i>info</i>	2 GiB Memory	Kapasitas RAM pada Ubuntu.
3.	Disk <i>info</i>	20 GiB Memory	Kapasitas Disk yang digunakan pada Ubuntu.
4.	<i>Processor info</i>	2 buah <i>Processor</i>	Jumlah <i>Processor</i> yang digunakan pada Ubuntu.

Untuk penggunaan Ubuntu melalui PuTTY menggunakan konfigurasi seperti pada **Table 2.7** berikut ini:

Tabel 2.7 Konfigurasi pada PuTTY yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	PuTTY <i>version</i>	0.73	Keterangan versi yang digunakan.

2.	<i>Category info</i>	<i>Session</i>	<i>Category yang digunakan untuk penggunaan PuTTY.</i>
3.	<i>Host Name (or IP address)</i>	192.168.65.129 (<i>Depend</i>)	<i>IP address yang dimasukkan merupakan IP address saat Ubuntu server login.</i>

Parameter yang digunakan untuk instalasi Apache dapat dilihat pada penjelasan

Modul 2.1 berikut ini:

```
$ sudo apt install apache2
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- apache2 : nama paket aplikasi untuk Apache

```
$ sudo ufw allow in "Apache Full"
```

Keterangan:

Perintah ini menunjukkan lalu lintas dari Apache Web Server dengan tipe protokol HTTP (port 80) dan HTTPS (port 443) diijinkan

Modul 2.1 Parameter instalasi Apache

Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan

Modul 2.2 berikut ini:

```
$ sudo apt install mysql-server
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- mysql-server : nama paket aplikasi untuk MySQL

```
$ sudo mysql_secure_installation
```

Keterangan:

Perintah ini menunjukka proses pengaturan dasar pengamanan MySQL yang berupa users, kata sandi, hak akses, dan sebagainya.

Modul 2.2 Parameter instalasi MySQL

Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan

Modul 2.3 berikut ini:

```
$ sudo apt install php libapache2-mod-php php-mysql
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)

- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- php libapache2-mod-php php-mysql : nama paket aplikasi untuk PHP

Modul 2.3 Parameter instalasi PHP

Parameter yang digunakan untuk instalasi PHPMyAdmin dapat dilihat pada penjelasan **Modul 2.4** berikut ini:

```
$ sudo apt install phpmyadmin php-mbstring php-gettext
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- phpmyadmin php-mbstring php-gettext : nama paket aplikasi untuk PHPMyAdmin

```
$ sudo mysql -u root -p
```

Keterangan:

Perintah Diatas untuk login ke mysql, -u merupakan user dan -p merupakan password dari user

```
UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('MASUKKAN_KATA_SANDI_ROOT') WHERE User
= 'root';
```

Keterangan:

Perintah ini digunakan untuk mengupdate password pada user root, pada PASSWORD('MASUKKAN_KATA_SANDI_ROOT') masukkan password yang dibuat saat penginstalan PHPMyAdmin

```
FLUS PRIVILAGES;
```

Keterangan:

Perintah ini menunjukkan melakukan flush kepada sistem untuk membaca ulang tanpa harus merestart server mysql

Modul 2.4 Parameter instalasi PHPMyAdmin

Untuk mengupload berkas menggunakan WinSCP menggunakan konfigurasi seperti pada **Table 2.8** berikut ini:

Tabel 2.8 Konfigurasi pada WinSCP yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	WinSP <i>version</i>	5.17	Keterangan versi yang digunakan.
2.	<i>File protocol</i>	SFTP	Jenis <i>File Protocol</i> yang digunakan.
3.	<i>Host Name</i>	192.168.65.129 (<i>Depend</i>)	<i>IP address</i> yang dimasukkan merupakan <i>IP address</i> saat Ubuntu server login.
4.	Port	22	Port yang digunakan untuk login melalui WinSCP.

Parameter yang digunakan untuk instalasi Docker dapat dilihat pada penjelasan **Modul 2.5** berikut ini:

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

Keterangan: perintah diatas untuk Perbarui indeks paket apt dan instal paket untuk memungkinkan apt menggunakan repositori melalui HTTPS

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Keterangan: Perintah Datas untuk menambahkan kunci GPG resmi docker

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Keterangan: Perintah diatas digunakan untuk mengatur repositori stable

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Keterangan: Perintah ini digunakan untuk menginstal docker engine versi terbaru

Modul 2.5 Parameter instalasi Docker

Parameter yang digunakan untuk instalasi Docker Compose dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

```
$ sudo apt install docker docker-compose
```

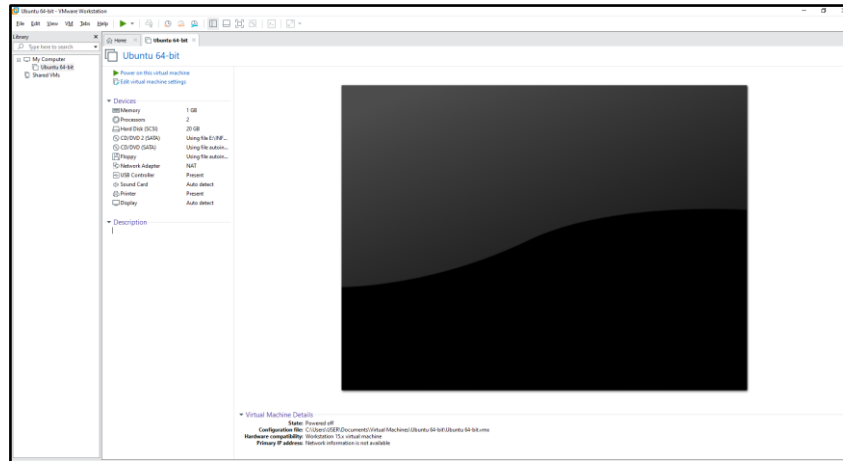
Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- docker docker-compose : nama paket aplikasi untuk docker-compose

Modul 2.5 Parameter instalasi Docker-compose

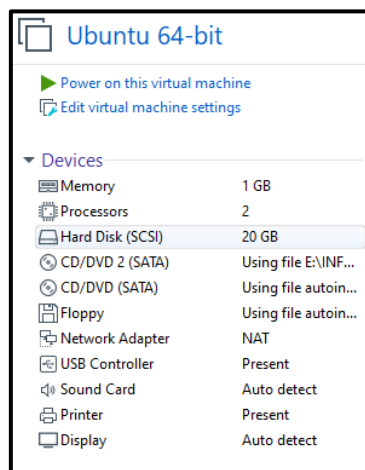
2.4 Tahap Implementasi

Berikut ini adalah tampilan dari awal *VMware* yang sudah terinstal sebuah *virtual machine*



Gambar 2.4 Tampilan Ubuntu yang sudah terinstal

Pada Virtual machine tersebut, berikut spesifikasi dari *virtual machine*



Gambar 2.5 Spesifikasi dari *virtual machine*

Selanjutnya adalah tampilan awal dari *virtual machine* setelah kita *power on* atau hidupkan.

Sebelumnya *login* terlebih dahulu ke server yang telah kita buat dan masukkan password

```

Apr 2020 09:09:16 +0000. DataSource DataSourceNoCloud [seed=/var/lib/cloud/seed/nocloud-net] [dsmode
-net]. Up 22.72 seconds

Ubuntu 18.04.3 LTS reza1-server tty1
reza1-server login: reza1
Password:
Last login: Wed Apr  1 02:13:48 UTC 2020 from 192.168.65.1 on pts/0
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Wed Apr  1 09:10:48 UTC 2020

System load: 0.59          Processes:    198
Usage of /:  22.7% of 19.56GB    Users logged in: 0
Memory usage: 42%              IP address for ens33: 192.168.65.129
Swap usage:  0%

 * Kubernetes 1.18 GA is now available! See https://microk8s.io for docs or
   install it with:
     sudo snap install microk8s --channel=1.18 --classic

 * Multipass 1.1 adds proxy support for developers behind enterprise
   firewalls. Rapid prototyping for cloud operations just got easier.
     https://multipass.run/

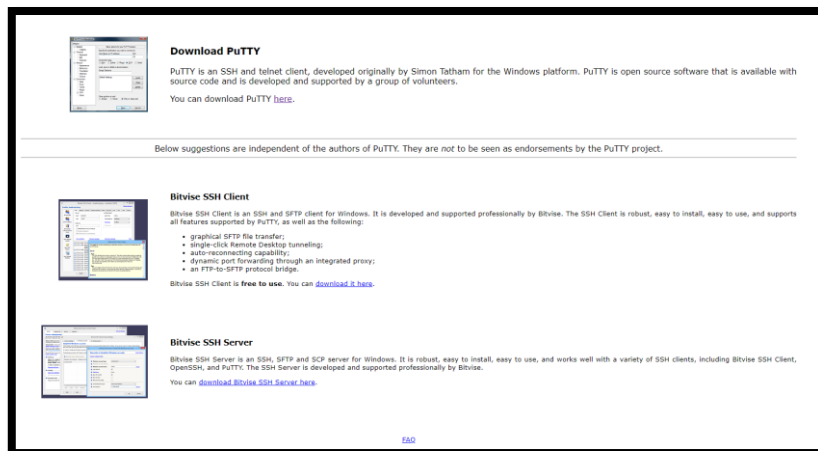
65 packages can be updated.
0 updates are security updates.

reza1@reza1-server:~$

```

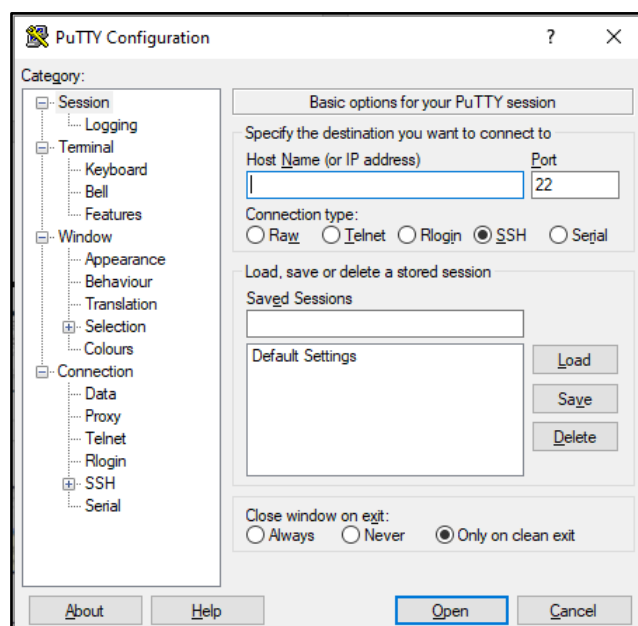
Gambar 2.6 Spesifikasi dari *virtual machine*

Langkah selanjutnya untuk *hosting* dengan LAMPP, kita akan melakukan instalasi dan konfigurasi aplikasi PuTTY.



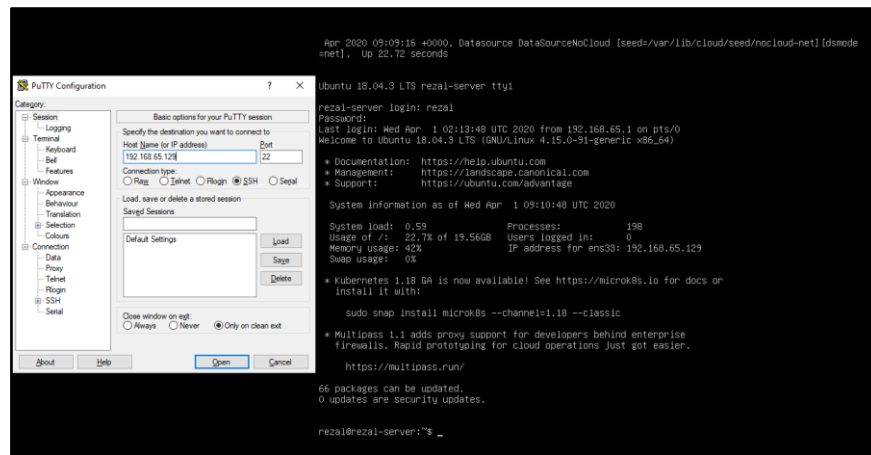
Gambar 2.7 Tampilan untuk mendownload aplikasi PuTTY

Setelah selesai mengunduh, lakukan instalasi dan masuk pada sebuah tampilan awal pada PuTTY, sebagai berikut



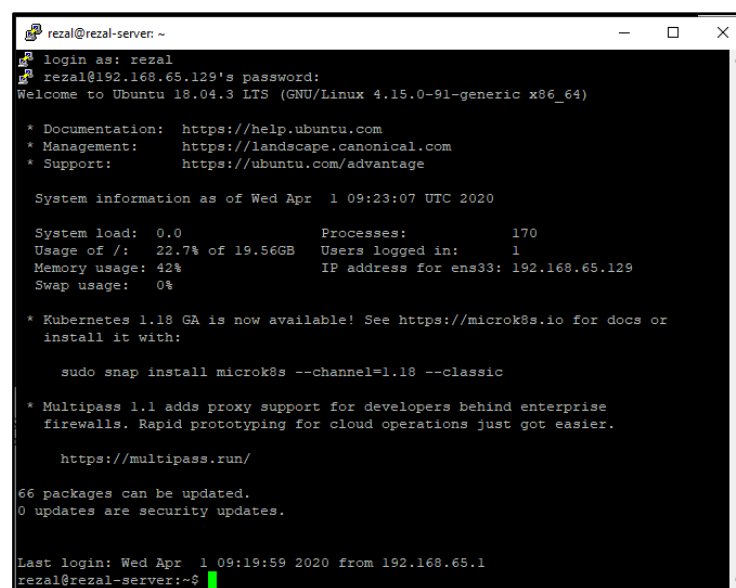
Gambar 2.8 Tampilan awal aplikasi PuTTY

Masukkan *Host Name (or IP address)* pada aplikasi PuTTY dengan *IP Address* yang ada pada *virtual machine* yang telah dibuat sebelumnya. Port tidak perlu diubah, tetap dengan 22. Kemudian klik tombol *open* pada aplikasi PuTTY :



Gambar 2.9 Tampilan untuk melihat IP address pada Ubuntu server

Setelah akan muncul *pop-up* tampilan *certificate approval*, tekan yes. Lalu anda akan masuk pada tampilan selanjutnya pada aplikasi PuTTY ini merupakan hasil implementasi **Table 2.3** yang dapat dilihat pada **Gambar 2.35** dan masukkan user yang dibuat pada VMware dan passwordnya, berikut tampilannya



Gambar 2.10 Hasil konfigurasi PuTTY dan login Ubuntu server

Proyek pertama dan kedua untuk penginstalan dan konfigurasi Ubuntu server sama hanya saja dibuat dibeda Ubuntu server.

2.4.1 Tahap Implementasi Pada Proyek Pertama

Pertama instalasi *Apache2* dan pengaturan *Firewall*, lakukan dengan perintah berikut :

```
$ sudo apt update
$ sudo apt install apache2
```

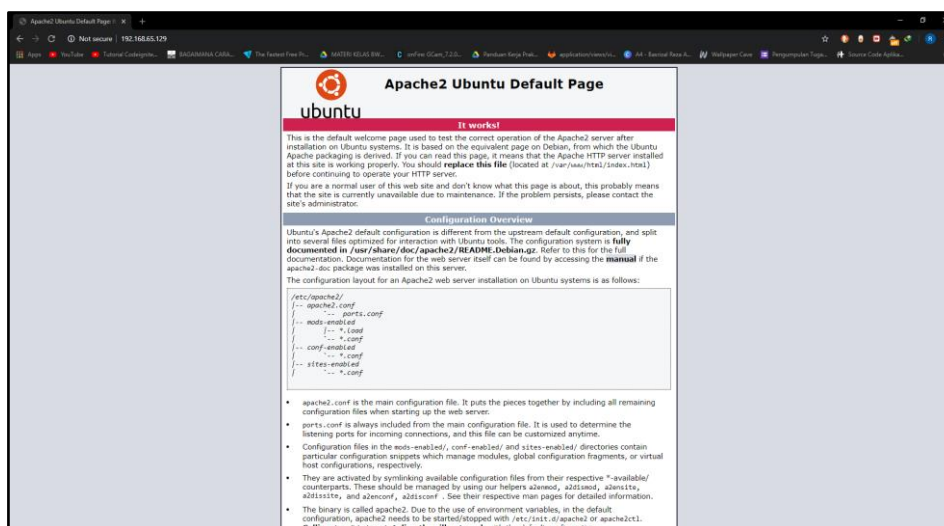
Perintah Program 2.1 Operasi instalasi paket aplikasi *apache2*

Setelah proses berhasil, lanjutkan dengan menekan huruf Y dan tunggu hingga proses selanjutnya berhasil tanpa adanya error. Selanjutnya adalah pengaturan *Firewall*, tuliskan perintah ini selanjutnya :

```
$ sudo ufw allow in "Apache Full"
```

Perintah Program 2.2 Operasi pengaturan *Firewall*

Jika proses ini telah selesai, maka kita dapat melihat *Apache2 Ubuntu Default Page* dengan cara menuliskan IP Server, dalam laporan ini, maka menggunakan IP Server kita, yaitu <http://192.168.65.129/> . Bila telah berhasil terbuka sempurna seperti pada gambar berikut ini, maka proses instalasi dan pengaturan *Firewall* telah berhasil dijalankan dengan sempurna:



Gambar 2.11 Tampilan apache sudah terinstal

Selanjutnya adalah Instalasi layanan MySQL yang akan kita gunakan nantinya, langkah pertama yaitu menuliskan perintah berikut:

```
$ sudo apt install mysql-server
```

Perintah Program 2.3 Operasi instalasi paket aplikasi MySQL

Setelah itu tekan huruf Y untuk proses pada langkah selanjutnya. Selanjutnya tuliskan sintak berikut :

```
$ sudo mysql_secure_installation
```

Perintah Program 2.4 Operasi pengamanan MySQL

Setelah menuliskan sintak berikut, pada aplikasi PuTTY akan menampilkan beberapa pertanyaan perihal validasi, jawaban untuk pertanyaan berikut yang kami lakukan adalah :

- No

- Lalu lakukan proses pembuatan kata sandi dan ingat kata sandi tersebut
- Y
- Y
- Y
- Y

Hasil akhir tersebut adalah informasi "All done" dan berakhirnya beberapa pertanyaan perihal validasi

Proses selanjutnya adalah melakukan instalasi untuk layanan PHP. Yang pertama dilakukan adalah menuliskan sintak berikut :

```
$ sudo apt install php libapache2-mod-php php-mysql
```

Perintah Program 2.5 Operasi instalasi paket aplikasi PHP

Perintah tersebut akan melakukan instalasi layanan PHP, pada pertanyaan tersebut jawab dengan menekan huruf Y dan menuliskan sintak selanjutnya yaitu sebagai berikut :

```
$ sudo nano /var/www/html/info.php
```

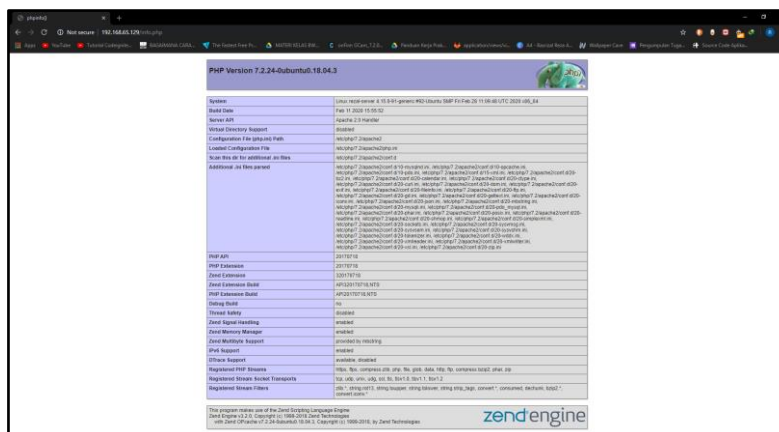
Perintah Program 2.6 Operasi pembuatan file info.php

Lalu akan muncul sebuah *GNU nano* yang berisi lembar kerja dari *info.php*, untuk menyipan berkas tersebut dengan menekan gabungan tombol yaitu CTRL + O untuk proses *save*. Selanjutnya untuk keluar dari *GNU nano* dengan menekan gabungan tombol yaitu CTRL + X. Dalam lembar kerja tersebut tuliskan kodingan berikut :

```
<?php
phpinfo();
?>
```

Perintah Program 2.7 Operasi php untuk mengetahui info php

Langkah selanjutnya adalah mengecek pada web browser yang anda gunakan ketikkan IP Server/info.php disini saya menggunakan IP server yaitu <http://192.168.65.129/info.php>, apakah layanan PHP berhasil aktif dan dapat diproses dengan benar. Bila pada web browser tampilannya seperti gambar dibawah, maka menandakan bahwa instalasi berhasil dilakukan:



Gambar 2.12 Tampilan phpinfo pada web browser

Langkah selanjutnya adalah melakukan instalasi PHPMyAdmin layanan *database* yang akan kita gunakan berbasis WEB GUI. Langkah pertama yang kita lakukan adalah menginstal package *phpMyAdmin* dengan sintak berikut :

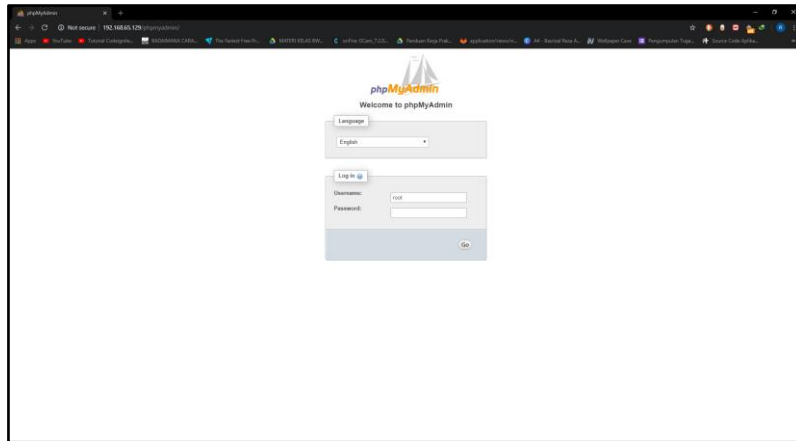
```
$ sudo apt install phpmyadmin php-mbstring php-gettext
```

Perintah Program 2.7 Operasi instalasi paket aplikasi PHPMyAdmin

Lalu tekan tombol huruf Y untuk melanjutkan prosesnya. Pada proses selanjutnya akan muncul sebuah window baru untuk menyelesaikan proses instalisasi. Perhatikan benar-benar dalam pengisian.

- Langkah pertama, tekan tombol SPASI, pastikan ada symbol asterisk (*) pada opsi *apache2* lalu kebawah menekan tombol OK
- Selanjutnya klik tombol OK dan melakukukan konfigurasi *database* untuk phpMyAdmin yang akan kita gunakan, tekan tombol YES dengan tombol enter
- Selanjutnya masukkan password yang anda inginkan, dan ingat password tersebut.

Selanjutnya jika proses instalasi berhasil dapat kita lakukan pengecekan dengan menuliskan IP server/phpmyadmin pada web browser yang anda gunakan, yang saya gunakan maka dengan IP berikut <http://192.168.65.129/phpmyadmin/> maka jika proses instalasi benar, tampilan pada web browser anda akan menampilkan sebagai berikut :



Gambar 2.13 Tampilan phpmyadmin pada web browser

Namun pada awal tampilan jika kita masukkan user dan password akan mengalami kegagalan. Untuk memperbaiki error tersebut, berikutnya kita tuliskan sintaknya, yaitu :

```
$ sudo mysql -u root
ATAU
$ sudo mysql -u root -p
```

Perintah Program 2.8 Operasi untuk login ke mysql

Kemudian untuk mengatur parameter dari user root, atur parameter password dengan kata sandi pada akun root pada MySQL yang telah kita buat sebelumnya, tuliskan sintaknya sebagai berikut :

```
UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('MASUKKAN_KATA_SANDI_ROOT') WHERE User
= 'root';
```

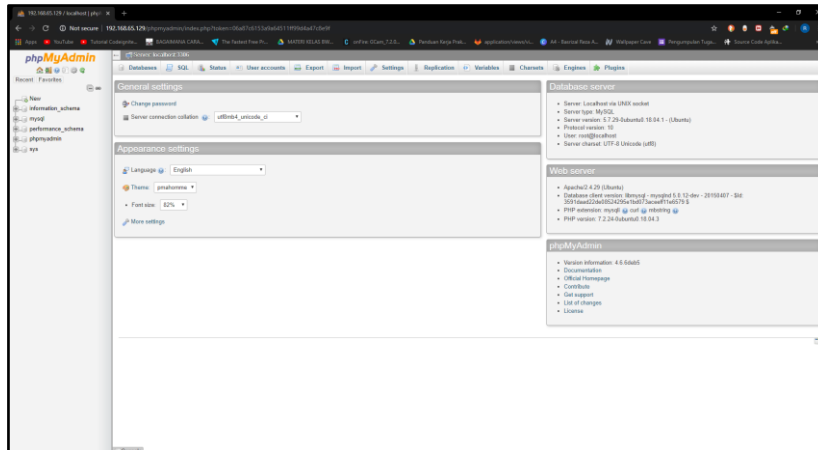
Perintah Program 2.9 Operasi mengatur parameter password pada user root

Setelah selesai melakukan pengaturan kata sandi, berikutnya kita melakukan flush yang berguna untuk flush kepada sistem untuk membaca ulang tanpa harus merestart server mysql. Dengan sintak berikut :

```
FLUSH PRIVILEGES;
```

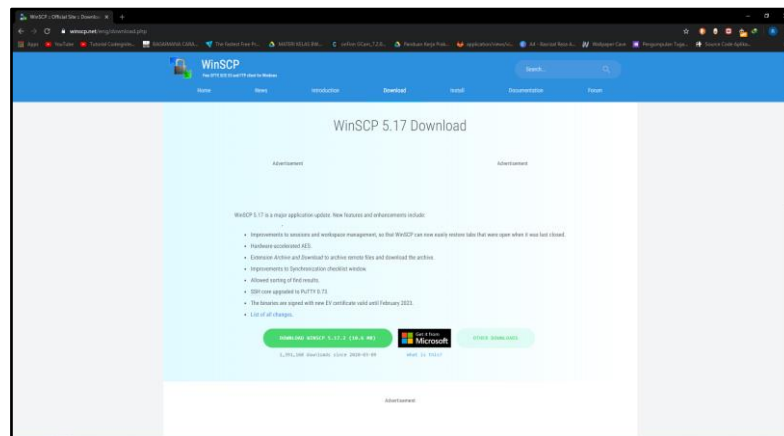
Perintah Program 2.10 Operasi flush sistem mysql

Lalu selanjutnya lakukan *refresh* pada web browser yang berada pada laman *phpmyadmin* tadi. Lalu lakukan proses login dengan user dan password, maka akan masuk pada halaman/tampilan awal *phpmyadmin* tadi, yaitu sebagai berikut :



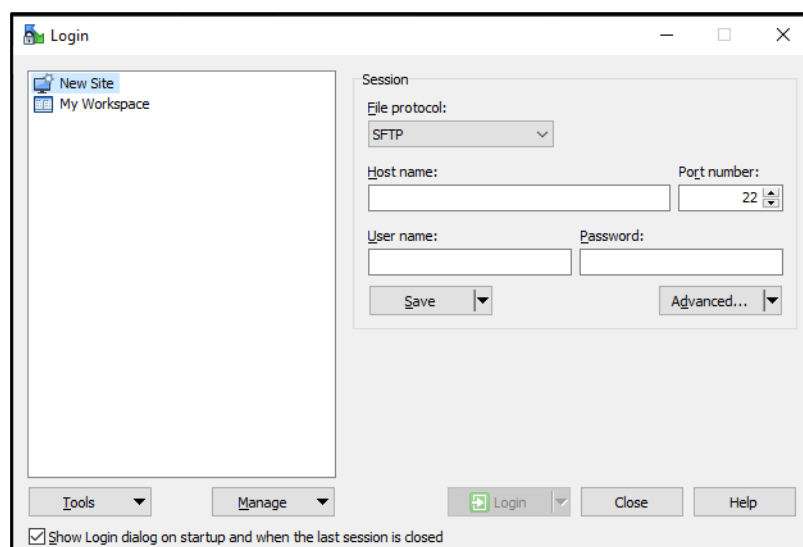
Gambar 2.14 Tampilan phpmyadmin setelah login

Langkah selanjutnya adalah proses *Upload* berkas dengan aplikasi WinSCP.



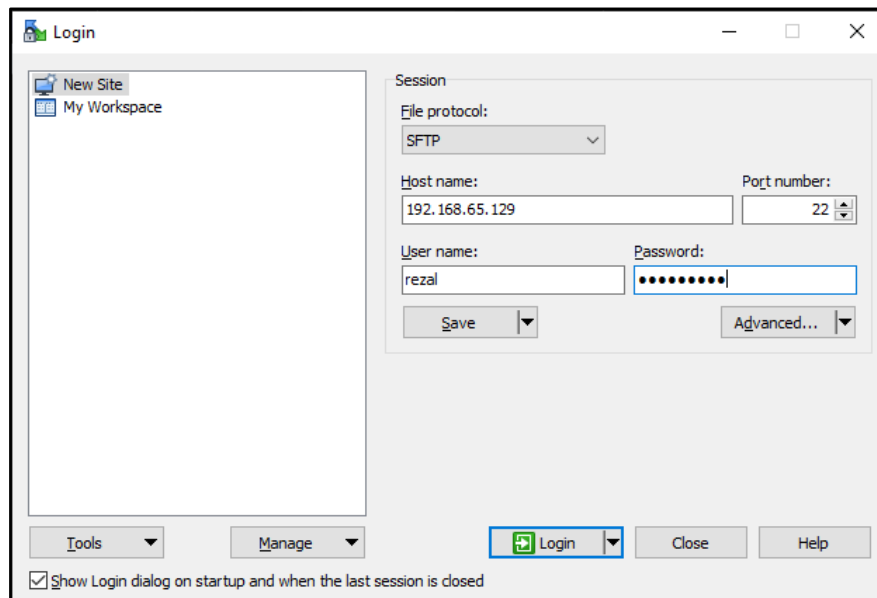
Gambar 2.15 Tampilan halaman download WinSCP

Berikut adalah laman awal dari aplikasi WinSCP :



Gambar 2.16 Tampilan konfigurasi WinSCP

Gunakan *File Protocol* dengan SFTP pada opsi pertama selanjutnya pada opsi *Host Name* sesuai IP dari server kita tadi, yaitu 192.168.65.129 dan pada *Port Number* gunakan 22. Masukkan username dan password sesuai pada *virtual machine*, berikut gambarnya :



Gambar 2.17 Tampilan konfigurasi WinSCP yang terisi

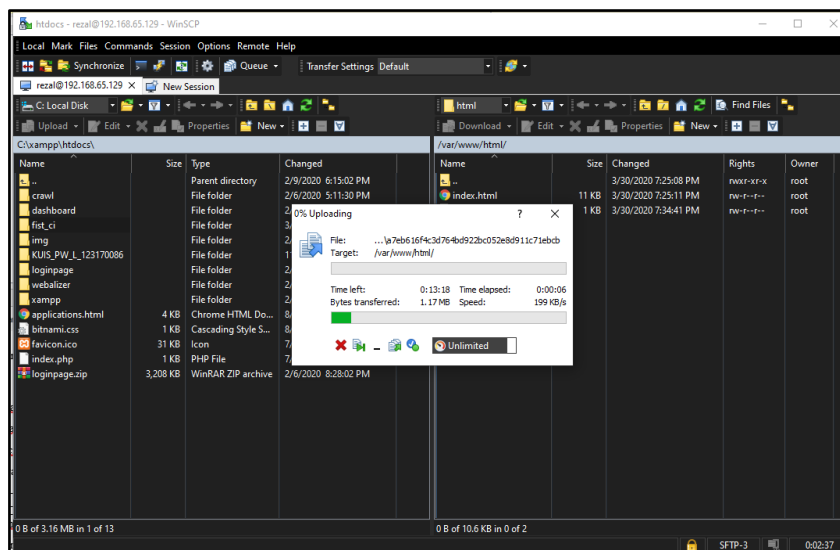
Kemudian klik tombol login, selanjutnya ada *pop-up window* konfirmasi sertifikat dan tekan tombol YES.

Kemudian pada tampilan kanan adalah tampilan berkas yang berada pada server. Sedangkan yang kiri merupakan yang ada Workstation. Sebelum memindahkan kita melakukan berkas, kita harus melakukan perintah *chown*, perintah ini digunakan untuk mengganti *owners* dari file/folder. Berikut sintak yang digunakan :

```
$ sudo chown NAMA_USER /var/www/html
```

Perintah Program 2.11 Operasi mengganti *owners* dari file/folder

Setelah melakukan perintah tersebut, anda dapat memindahkan berkas yang berada pada server. Jangan lupa direktori server yaitu */var/www/html*. Berikut gambar proses pemindahan berkas ;



Gambar 2.18 Proses pemindahan berkas

Setelah proses *upload* selesai dapat kita cek apakah web sudah dapat di gunakan atau belum dengan mengetik IP server/nama_web_untuk_akses.

2.4.2 Tahap Implementasi Pada Proyek Kedua

Pertama instalasi Docker dengan mengikuti langkah langkah dibawah ini:

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

Perintah Program 2.12 Instalasi paket untuk dapat menggunakan repository melalui HTTPS

Perintah ini digunakan untuk pembaruan indeks paket apt dan instalasi paket untuk memungkinkan apt dapat menggunakan repositori melalui HTTPS. Selanjutnya adalah menambahkan kunci GPG resmi docker dengan menggunakan perintah :

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Perintah Program 2.13 Operasi penambahan kunci GPG

Setelah melakukan perintah ini dapat dicek dengan memasukkan perintah ini

```
$ sudo apt-key fingerprint 0EBFCD88
```

Perintah Program 2.14 Operasi mengecek kunci GPG

```
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid   [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

Gambar 2.19 Hasil pemasukkan kunci GPG

Jika output dari perintah yang dimasukkan seperti ini maka terbukti kunci yang dimasukkan sukses. Selanjutnya mengatur repositori stable menggunakan perintah:

```
$ sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Perintah Program 2.15 Operasi mengatur repository stable

Setelah proses selesai dilanjutkan untuk menginstal docker engine dengan menggunakan perintah:

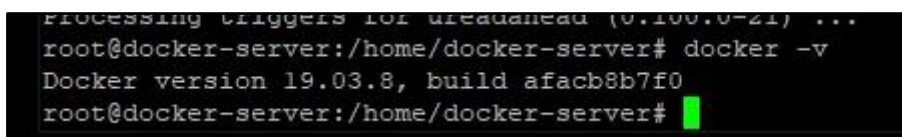
```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Perintah Program 2.16 Operasi instalasi docker engine

Setelah proses selesai maka proses penginstalan docker sudah selesai. Untuk mengeceknya dapat lakukan perintah pengecekan versi docker:

```
$ docker -v
```

Perintah Program 2.17 Operasi mengecek docker versi



```
Processing triggers for ureadahead (0.100.0-21) ...
root@docker-server:/home/docker-server# docker -v
Docker version 19.03.8, build afacb8b7f0
root@docker-server:/home/docker-server#
```

Gambar 2.20 Hasil pengecekan docker versi

Selanjutnya menginstal docker composer yang digunakan untuk menjalankan kontainer docker secara bersamaan. Perintahnya dibawah ini:

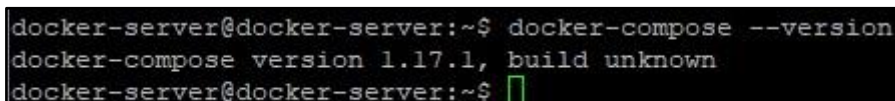
```
$ sudo apt install docker docker-compose
```

Perintah Program 2.18 Operasi instalasi docker-compose

Tunggu sampai proses selesai. Setelah selesai untuk mengeceknya dengan mengecek versi docker composenya dengan perintah:

```
$ docker-compose -version
```

Perintah Program 2.19 Operasi mengecek docker-compose versi



```
docker-server@docker-server:~$ docker-compose --version
docker-compose version 1.17.1, build unknown
docker-server@docker-server:~$
```

Gambar 2.21 Hasil pengecekan docker-compose versi

Dapat dicek terlebih dahulu apakah sudah mendapatkan *permission* untuk docker. Jika belum maka lakukan perintah ini:

```
$ sudo groupadd docker
$ sudo usermod -aG docker $USER
```

```
$ newgrp docker
```

Perintah Program 2.20 Operasi pemberian *permission* untuk docker

Jika perintah sukses dijalankan tidak terjadi error, maka tidak akan muncul *error permission* saat memanggil perintah docker, lalu untuk mencobanya dengan menampilkan list image docker dengan menggunakan perintah:

```
$ docker image ls
```

Perintah Program 2.21 Operasi menampilkan list image docker

Sebelum melanjutkan ada beberapa penyesuaian terlebih dahulu karena disini kami menggunakan *framework* ci. Pertama adalah mensetting config.php, ubah la *base_url* menjadi IP:8000 dan *sess_save_path* menjadi *sys_get_temp_dir()*. Disini 8000 adalah port yang akan digunakan nanti

```

/*
 * If you need to allow multiple domains, remember that this file is still
 * a PHP script and you can easily do that on your own.
 */
$config['base_url'] = 'http://192.168.189.128:8000';

/*
|-----
| Index File
|-----

```

Gambar 2.22 config.php

```

/*
 * Other session cookie settings are shared with the rest of the application,
 * except for 'cookie_prefix' and 'cookie_httponly', which are ignored here.
 */
$config['sess_driver'] = 'files';
$config['sess_cookie_name'] = 'ci_session';
$config['sess_expiration'] = 7200;
$config['sess_save_path'] = sys_get_temp_dir();
$config['sess_match_ip'] = FALSE;
$config['sess_time_to_update'] = 300;
$config['sess_regenerate_destroy'] = FALSE;

/*
|-----
| Cookie Related Variables
|-----

```

Gambar 2.23 config.php

Kedua adalah database.php. Sesuaikan host, username, password, namadb dengan database yang akan digunakan. Pada project ini menggunakan

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'db',
    'username' => 'latihan',
    'password' => 'rahasia',
    'database' => 'sifist',
    'dbdriver' => 'mysqli',
    'dbprefix' => ''

```

Gambar 2.24 database.php

Ketiga mensetting .htaccess yang berada diluar *application*. Untuk yang ini tidak harus karena pada .htaccess project ini bukan hanya *default setting* dari ci saja

```
RewriteEngine on

RewriteBase ./
```

Gambar 2.25 .htaccess diluar *application*

Setelah selesai menseting project ci maka selanjutnya kita membuat *directory* baru, disini nama *directory*nya web_app dilanjutkan masuk ke *directory* tersebut dengan perintah:

```
$ mkdir web_app
$ cd web_app/
```

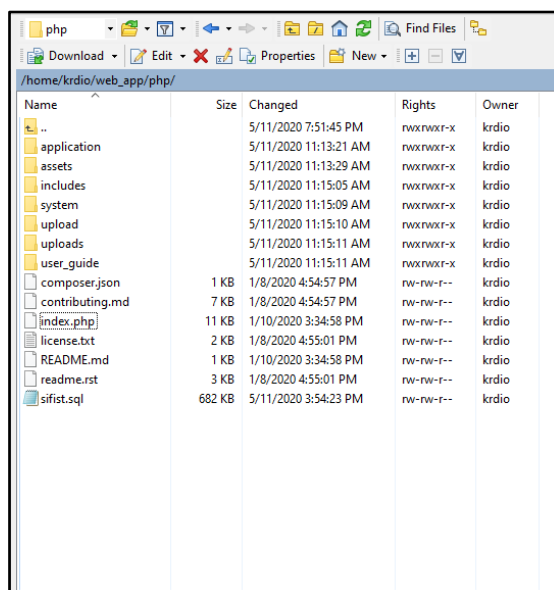
Perintah Program 2.22 Operasi membuat *directory* baru web_app

Selanjutnya membuat *directory* baru lagi didalam *directory* tadi dengan nama php

```
$ mkdir php
```

Perintah Program 2.23 Operasi membuat *directory* baru php

Setelah selesai membuat *directory* baru selanjutnya copykan project kalian ke *directory* yang sudah di buat dengan aplikasi WinSCP seperti gambar dibawah ini:



Gambar 2.26 Hasil perpindahan project kedalam *directory* php

Setelah itu masuk ke *directory* php jika sebelumnya belum masuk dan membuat file Dockerfile dengan perintah:

```
$ cd php
$ nano Dockerfile
```

Perintah Program 2.24 Operasi membuat sekaligus mengedit file Dockerfile

Disini hanya menggunakan `cd php` karena posisi sudah didalam *directory* web_app tadi, jika belum masuk pada *directory* tersebut nanti akan disesuaikan. Selanjutnya isikan file Dockerfile seperti ini

```
FROM php:7.3-apache

RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli

EXPOSE 80

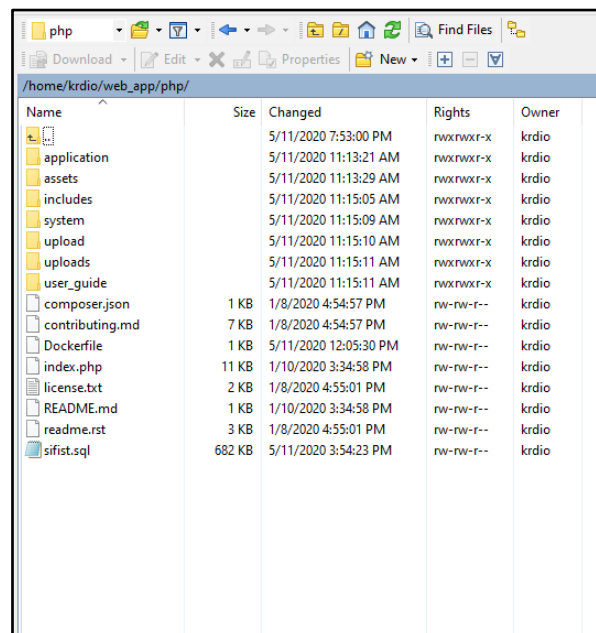
RUN a2enmod rewrite
RUN chmod -R 755 /var/www/html

COPY ./ /var/www/html

RUN service apache2 restart
```

Perintah Program 2.25 Operasi mengedit file dockerfile

Setelah diisikan klik CTRL + O untuk write out dan CTRL + X untuk exit. Tampilan Dockerfile yang terbuat di dalam *directori* php



Name	Size	Changed	Rights	Owner
application		5/11/2020 7:53:00 PM	rw-rw-r--	krdio
assets		5/11/2020 11:13:21 AM	rw-rw-r--	krdio
includes		5/11/2020 11:13:29 AM	rw-rw-r--	krdio
system		5/11/2020 11:15:05 AM	rw-rw-r--	krdio
upload		5/11/2020 11:15:09 AM	rw-rw-r--	krdio
uploads		5/11/2020 11:15:10 AM	rw-rw-r--	krdio
user_guide		5/11/2020 11:15:11 AM	rw-rw-r--	krdio
composer.json	1 KB	5/11/2020 11:15:11 AM	rw-rw-r--	krdio
contributing.md	7 KB	1/8/2020 4:54:57 PM	rw-rw-r--	krdio
Dockerfile	1 KB	1/8/2020 4:54:57 PM	rw-rw-r--	krdio
index.php	11 KB	5/11/2020 12:05:30 PM	rw-rw-r--	krdio
license.txt	2 KB	1/10/2020 3:34:58 PM	rw-rw-r--	krdio
README.md	1 KB	1/8/2020 4:55:01 PM	rw-rw-r--	krdio
readme.rst	3 KB	1/10/2020 3:34:58 PM	rw-rw-r--	krdio
sifist.sql	682 KB	1/8/2020 4:55:01 PM	rw-rw-r--	krdio

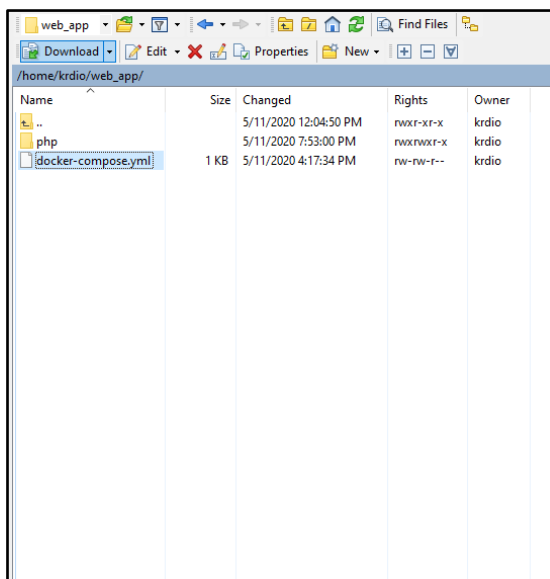
Gambar 2.27 Hasil pembuatan dan mengedit file Dockerfile

Selanjutnya kembali ke *directory* web_app untuk membuat dan mengedit file docker-compose.yml dengan perintah

```
$ cd web_app
$ nano docker-compose.yml
```

Perintah Program 2.26 Operasi membuat dan mengedit file docker-compose.yml

Lalu isikan file docker-compose.yml dengan perintah seperti pada Lampiran 2.1 Listing docker-compose.yml. Tampilan jika file docker-compose terbuat pada *directory* web_app



Gambar 2.28 Tampilan file docker-compose.yml

Selanjutnya menjalankan docker-compose dengan perintah:

```
$ docker-compose up
```

Perintah Program 2.28 Operasi menjalankan docker-compose

Jangan lupa selalu tersambung keinternet karena saat menjalankan docker-compose akan mendownload file-file yang dibutuhkan sesuai dengan perintah yang ada didalam file docker-compose

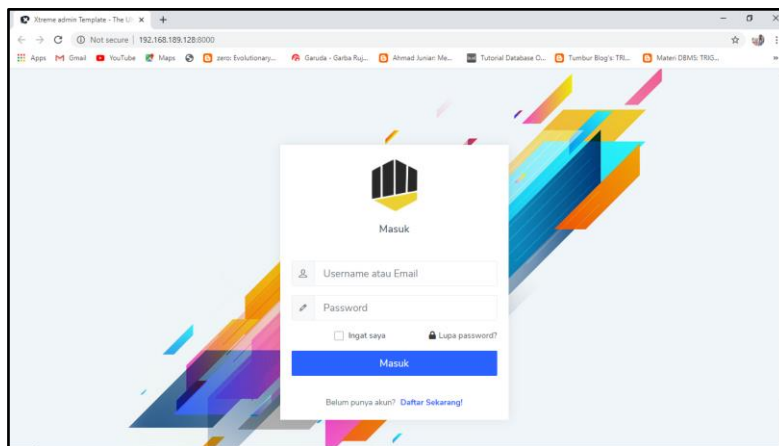
```
docker-server@docker-server:~/web_app$ docker-compose up
Creating network "webapp_default" with the default driver
Building web
Step 1/4 : FROM php:7.3-apache
7.3-apache: Pulling from library/php
54fec2fa59d0: Already exists
cc4504446071: Pull complete
f8d5b99827d: Downloading [=====] 17.72MB/76.65MB
6bf7f8bea876: Download complete
81ecb6839daa: Downloading [=====] 6.65MB/18.68MB
94884188472c: Download complete
1d3e02b28167: Download complete
2da3148e7e6: Downloading [==>] 399.8kB/12.45MB
2b41fad541f4: Waiting
dcf4d822b25e: Waiting
0e69fc8086e4: Waiting
d8f2bc770a95: Waiting
3c38040e6c74: Waiting
7c94a258a015: Waiting
```

Gambar 2.29 Tampilan file docker-compose.yml yang sedang berjalan

```
docker-server@docker-server:~/web_app$ docker-compose up
Creating network "webapp_default" with the default driver
Building web
7.3-apache: Pulling from library/php
54fec2fa59d0: Already exists
cc4504446071: Pull complete
f8d5b99827d: Pull complete
6bf7f8bea876: Pull complete
81ecb6839daa: Pull complete
94884188472c: Pull complete
1d3e02b28167: Pull complete
2da3148e7e6: Pull complete
2b41fad541f4: Pull complete
dcf4d822b25e: Pull complete
0e69fc8086e4: Pull complete
d8f2bc770a95: Pull complete
3c38040e6c74: Pull complete
7c94a258a015: Pull complete
Digest: sha256:1a5e676753a254d53085154f91a0bce77d14894d4ca717418e1c500cd1f446877
Status: Downloaded newer image for php:7.3-apache
--> f0d98b8dd96e
Step 2/4 : RUN apt-get update && apt-get upgrade -y
--> Running in d014375a3475
Get:1 http://security.debian.org/debian-security buster/updates InRelease [65.4 kB]
Get:2 http://security.debian.org/debian-security buster/updates/main amd64 Packages [189 kB]
Get:3 http://deb.debian.org/debian buster InRelease [122 kB]
Get:4 http://deb.debian.org/debian buster-updates InRelease [49.3 kB]
Get:5 http://deb.debian.org/debian buster/main amd64 Packages [7907 kB]
Get:6 http://deb.debian.org/debian buster-updates/main amd64 Packages [7380 B]
Fetched 8839 kB in 2s (920 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
The following package was automatically installed and is no longer required:
  sensible-utils
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Removing intermediate container d014375a3475
--> 79e5d5ae1e6e
Step 3/4 : RUN docker-php-ext-install mysqli
--> Running in f19bb890528
```

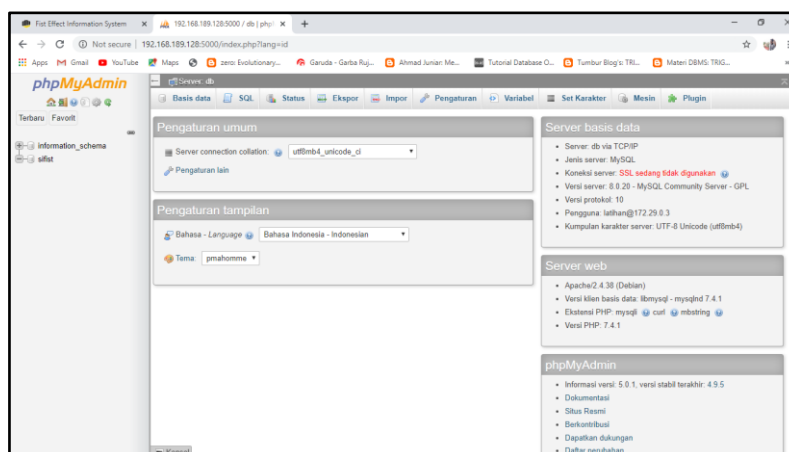
Gambar 2.30 Tampilan file docker-compose.yml yang sedang berjalan

Setelah semua selesai dijalankan, selanjutnya buka browser dan masukkan url IP-SERVER:8000 maka akan tampil project yang diproses oleh docker tadi



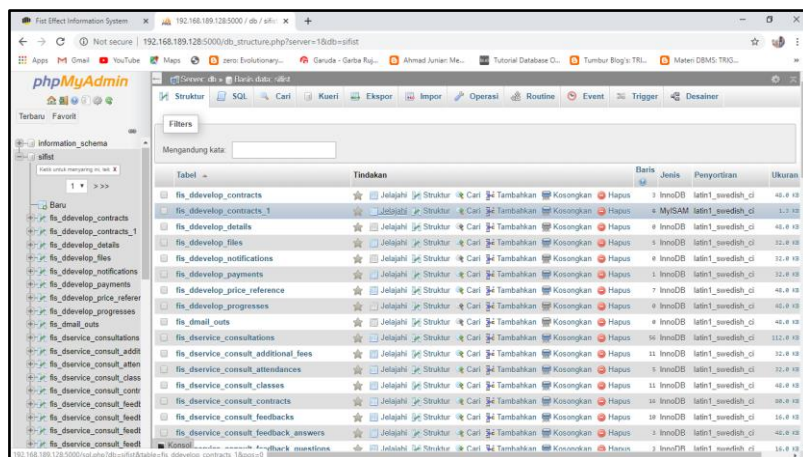
Gambar 2.31 Tampilan web setelah selesai proses docker-compose

Setelah web berhasil di jalankan, langkah selanjutnya adalah import database yang akan kita gunakan. Import database dengan cara masuk pada browser lalu ketikkan IP:5000 untuk membuka phpmyadmin



Gambar 2.32 Tampilan phpmyadmin

Disitu sudah terdapat database sifist database yang akan digunakan, maka klik saja database tersebut lalu import databasenya. Disini nama filenya adalah sifist.sql. Jika sudah selesai maka akan seperti ini



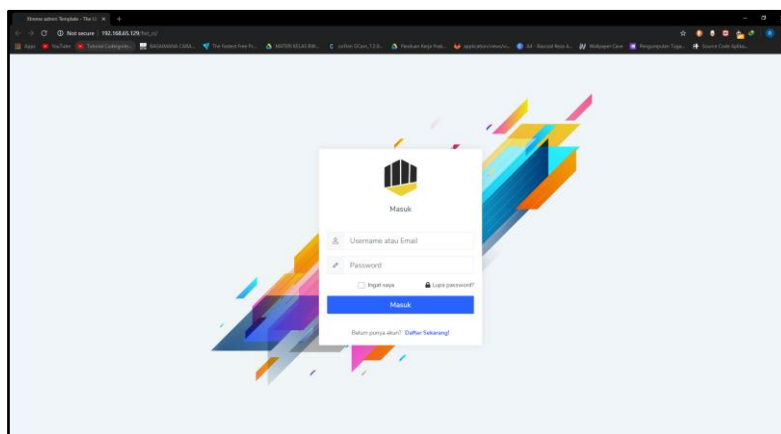
Gambar 2.33 Tampilan database yang sudah import

Setelah selesai, cek kedalam browser tadi disini mengeceknya dengan cara login kedalam aplikasi web

2.5 Hasil Implementasi

2.5.1 Hasil Implementasi pada Proyek Pertama

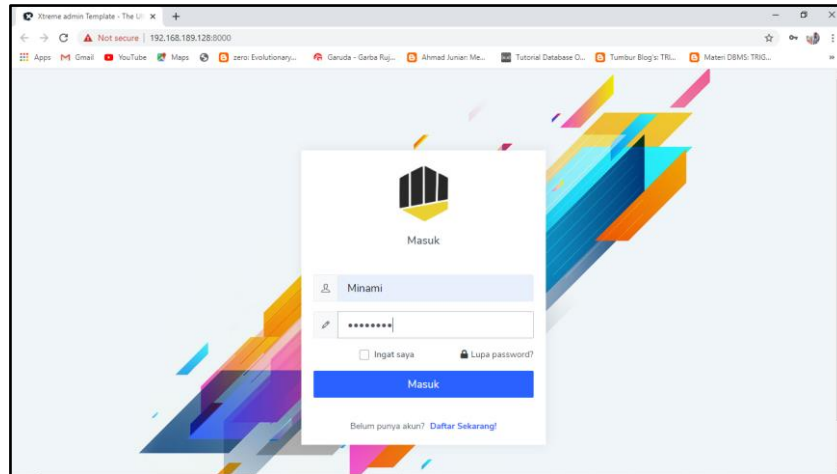
Dengan adanya koneksi internet, maka dapat diakses dari lokal server. Berikut caranya menggunakan IP server yang kita gunakan , yaitu http://192.168.65.129/fist_ci/ berikut adalah tugas KP dari kelompok kami yang digunakan, maka tampilannya akan seperti gambar dibawah:



Gambar 2.34 Tampilan berkas project

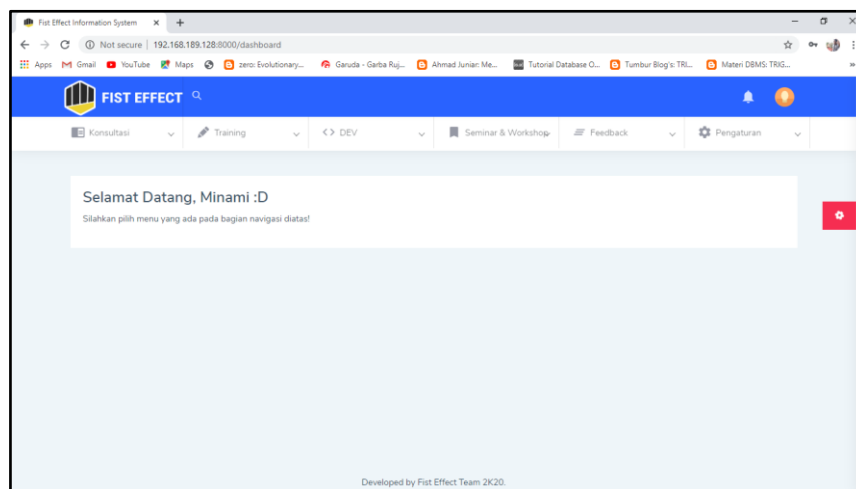
2.5.2 Hasil Implementasi pada Proyek Kedua

Setelah selesai melakukan tahap implementasi pada bab sebelumnya disini akan memberikan tampilan hasil dari pada bab sebelumnya. Dengan adanya internet, maka dapat diakses aplikasi web tadi dengan cara memasukkan url IP-SERVER:8000 lalu akan tampil seperti ini



Gambar 2.35 Tampilan web login

Setelah itu untuk mengecek apakah database sudah berjalan dengan login pada aplikasi web tersebut. Jika berhasil maka akan masuk kedalam aplikasi web setelah login



Gambar 2.36 Tampilan web berhasil login

2.6 Pengujian Singkat

2.6.1 Pengujian Singkat pada Proyek Pertama

Terhadap sistem informasi yang telah dirancang, akan dilakukan pengujian terhadap permasalahan kecepatan proses yang dalam percobaannya pada manajemen *feedback* dan *submit feedback*.

Pertama-tama sebagai admin membuat sebuah pertanyaan untuk form feedback pelatihan pada menu manajemen feedback seperti yang terlihat pada **Gambar 2.37** berikut.

The screenshot shows the 'Feedback Pelatihan' (Training Feedback) page. At the top, there's a navigation bar with 'FIST EFFECT' logo and various menu items like 'Konsultasi', 'Training', 'DEV', 'Seminar & Workshop', 'Feedback', and 'Pengaturan'. Below the navigation bar, a success message 'Sukses' (Success) states 'Penambahan Feedback berhasil disimpan' (Feedback addition was successfully saved). The main section is titled 'Manajemen Feedback Pelatihan' (Training Feedback Management) and contains a form with fields for 'Nama Lengkap' (Full Name), 'Nama Universitas' (University Name), and 'Kritik dan Saran' (Criticism and Suggestions). There are also dropdown menus for 'Jawaban Singkat' (Short Answer) and 'Jawaban Panjang' (Long Answer). A green button labeled 'Tambah Partisipan' (Add Participant) is at the bottom left, and a green button labeled 'Simpan' (Save) is at the bottom right.

Gambar 2.37 Tampilan Manajemen Feedback Pelatihan

Selanjutnya sebagai public memberikan feedback melalui link yang sudah disediakan oleh system, untuk link feedback pelatihan IP:8000/feedback/training, bisa dilihat pada **Gambar 2.38** berikut.

The screenshot shows the 'Feed Back Pelatihan' (Training Feedback) submission page. It features a header with 'FIST EFFECT' logo and navigation links 'Feed Back', 'About', and 'Contact'. Below the header, there's a progress bar labeled 'PROGRESS' showing '1/3' completion. The main section is titled 'Feed Back Pelatihan' and contains a form with a field for 'Nama Lengkap' (Full Name). A blue button labeled 'Selesaikan' (Finish) is at the bottom right.

Gambar 2.38 Tampilan Submit Feedback Pelatihan

Setelah selesai mengisi feedback pelatihan, sebagai admin dapat melihat hasil submit feedback yang diberikan seperti pada **Gambar 2.39** berikut.

The screenshot shows the 'Detail Jawaban Pelatihan' (Training Answer Detail) page. It displays a table with two columns: 'Pertanyaan' (Question) and 'Jawaban' (Answer). The table contains three rows of data.

Pertanyaan	Jawaban
Nama Lengkap	Dio
Nama Universitas	UPN
Kritik dan Saran	Tingkatkan lagi pelatihannya

Gambar 2.39 Tampilan Detail Feedback Pelatihan

Hasil dari apa yang telah dilakukan role sebagai admin dan public tersebut dapat manajemen feedback dan submit feedback sama seperti cara konvensional namun dari segi

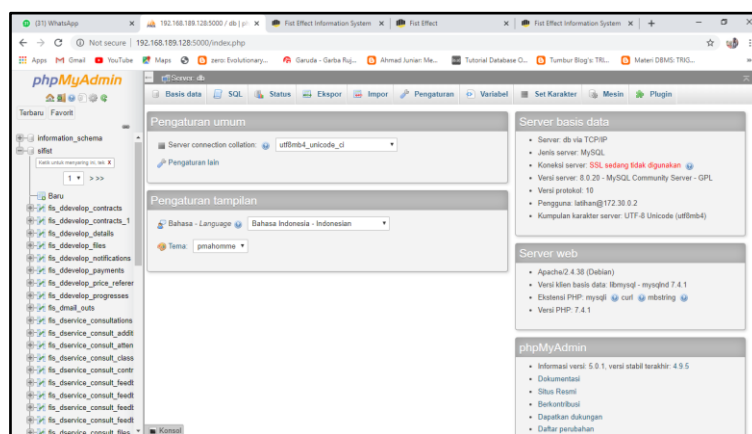
waktu dan tenaga jauh lebih cepat. Hal ini dapat terlihat bahwa waktu yang dibutuhkan hanya kurang lebih 5 menit dan dapat dilakukan dari mana saja selama ada koneksi internet.

Kesimpulan dari percobaan tersebut bahwa sistem yang dibuat mampu memberikan layanan yang sangat bermanfaat bagi admin maupun orang diluar perusahaan, selain dapat diakses dari mana saja, biaya penyewaan hosting berkala jauh lebih murah.

Untuk pengujian singkat kami menggunakan VM proyek docker karena VM proyek 1 tidak sengaja kehapus.

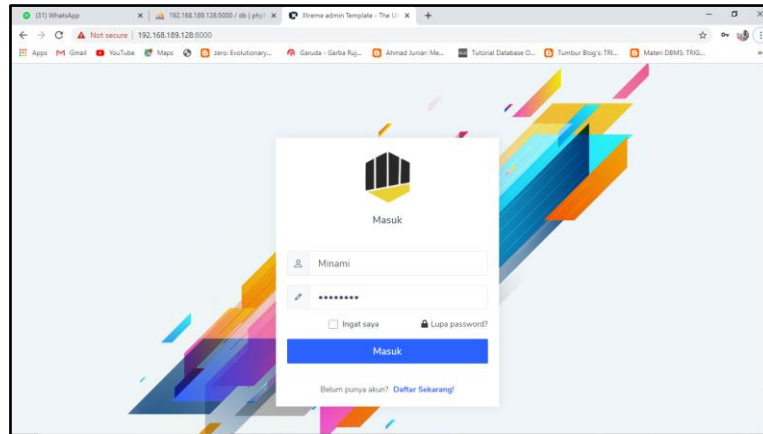
2.6.2 Pengujian Singkat pada Proyek Kedua

Terhadap proyek yang dirancang, akan dilakukan pengujian terhadap permasalahan kecepatan dan kemudahan dalam pemasangan dan konfigurasi LAMPP dan Aplikasi web menggunakan docker. Pada pengujian prosesnya dapat dilihat melalui **2.4.1 Tahap Implementasi Pada Proyek Kedua**, disitu sudah jelas untuk proses – proses dalam pemasangan dan konfigurasi. Kita hanya tinggal membuat dockerfile yang isi dari dockerfile dapat dilihat pada **Listing Program 2.25** selajutnya membuat docker-compose.yml yang isi dari file tersebut dapat dilihat pada **Lampiran 2.1** dan menyesuaikan file file seperti config.php, database.php, dan .htaccess pada project web seperti pada **Gambar 2.22**, **Gambar 2.23**, **Gambar 2.24**, dan **Gambar 2.25**. Selanjutnya hanya menjalankan docker-compose.yml, dan setelah selesai dapat dijalankan phpmyadmin melalui web browser dengan memasukkan IP : 5000 seperti pada **Gambar 2.40** berikut.



Gambar 2.40 Tampilan PHPMyadmin

Dan dapat menjalankan aplikasi web melalui web browser dengan memasukkan IP : 8000 seperti pada **Gambar 2.41** berikut.



Gambar 2.41 Tampilan Aplikasi Web Login

Hasil dari apa yang telah dilakukan sama seperti cara pemasangan dan konfigurasi secara manual namun lebih mudah dan cepat dengan menggunakan docker. Hal ini dapat terlihat bahwa hanya membuat Dockerfile dan docker-compose.yml sesuai dengan apa yang dibutuhkan lalu menyesuaikan konfigurasi aplikasi web dengan docker-compose.yml yang dibuat tanpa perlu memikirkan konfigurasi yang dibutuhkan untuk pemasangan LAMP ini.

Kesimpulan dari percobaan tersebut bahwa docker yang dibuat mampu memberikan kemudahan bagi para developer, selain kemudahan yang diberikan, proses pemasangan jauh lebih cepat.

BAB III

JADWAL Pengerjaan dan Pembagian Tugas

3.1 Agenda Pengerjaan

Berikut pada **Tabel 3.1** merupakan pembagian jenis tugas proyek akhir terhadap alokasi waktu pengerjaan pada bulan Maret dan April tahun 2020:

Tabel 3.1 Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Penginstalan dan Konfigurasi Ubuntu LAMPP								
4.	Pemasukkan Project KP ke Ubuntu LAMPP								
5.	Pembuatan Dockerfile dan pemasukkan project								
6.	Laporan Bab 1								
7.	Laporan Progress 2								
8.	Laporan Final Project								

3.2 Keterangan Pembagian Tugas

Berikut pada **Tabel 3.2** merupakan pembagian tugas-tugas pada proyek akhir terhadap anggota pada tim pembuatan proyek akhir:

Tabel 3.2 Pembagian Tugas Proyek

No.	Keterangan Tugas	Penanggung Jawab
1.	Analisa Persoalan	Dio & Basri
2.	Pembagian Tugas	Dio & Basri
3.	Penginstalan dan Konfigurasi Ubuntu LAMPP	Basri
4.	Pemasukkan Project KP ke Ubuntu LAMPP	Basri & Dio
5.	Pembuatan Dockerfile dan pemasukkan project	Dio & Basri
6.	Laporan Bab 1	Dio & Basri
7.	Laporan Pogress 2	Dio & Basri
8.	Laporan Final Project	Dio & Basri

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan implementasi dan pengujian singkat yang telah dilakukan kami menyimpulkan bahwa

- a. Berdasarkan masalah dalam kecepatan manajemen dan akses pada system informasi fist effect, kemudian dilakukan hosting dengan menggunakan Ubuntu server dan mendapatkan hasil yang cukup memuaskan dan sesuai dengan harapan, aplikasi web dapat diakses dan dijalankan sesuai dengan sistem yang dibangun. Berdasarkan pengujian singkat yang dilakukan pada masalah secara acak bahwa sistem yang dibuat mampu memberikan layanan yang sangat bermanfaat bagi admin maupun orang diluar perusahaan, selain dapat diakses dari mana saja, biaya penyewaan hosting berkala jauh lebih murah.
- b. Berdasarkan masalah dalam kemudahan pemasangan dan konfigurasi LAMPP dan, sistem informasi fist effect, kemudian dilakukan dengan menggunakan Docker dan mendapatkan hasil sesuai dengan harapan. LAMPP dapat berjalan dan sistem informasi fist effect dapat diakses sesuai sistem yang dibangun. Berdasarkan pengujian singkat yang dilakukan bahwa docker yang dibuat mampu memberikan kemudahan bagi para developer, selain kemudahan yang diberikan, proses pemasangan jauh lebih cepat. Tetapi mendapatkan kendala yaitu ada beberapa proses yang tidak berjalan dikarenakan versi yang dipasang tidak sesuai dengan versi saat sistem dibangun.
- c. Berdasarkan pembagian tugas proyek pada bab 3, hasil tidak terlalu sesuai dengan jadwal yang diharapkan akan tetapi proyek yang kami kerjakan dapat terselesaikan.

4.2 Saran

Dalam pemasangan LAMPP disamakan dengan versi pada sistem yang dipasang, HDD laptop yang digunakan diharapkan dalam kondisi masih bagus, belum pernah terjadi masalah pada HDD tersebut, dan dalam pembuatan laporan ini masih jauh dari kata sempurna, dimohon pada pembaca dapat memberikan kritik dan saran untuk pembuatan laporan dan proyek kedepannya.

DAFTAR PUSTAKA

Adiputra, F. (2015). Container dan Docker: Teknik Virtualisasi dalam Pengelolaan Banyak Aplikasi Web. *Jurnal SimanteC*, 4(3)

_____, <<https://www.lawavedesign.com/berita/33/pengertian-manfaat-cara-kerja-dan-contoh-cloud-computing.html/>>, (2020 accessed 31 Maret 2020)

_____, <<https://www.codepolitan.com/mengenal-teknologi-docker/>>, (18 Jul 2016 accessed 31 Maret 2020)

_____, <<https://docs.docker.com/get-started/>>, (2013 - 2020 accessed 13 April 2020)

_____, <https://id.wikipedia.org/wiki/Komputasi_awan#Layanan_Komputasi_Awan/>, (6 April 2020 accessed 29 April 2020)

Use the "Insert Citation" button to add citations to this document.

LAMPIRAN

```
version: '3.3'
services:
  web:
    build:
      context: ./php
      dockerfile: Dockerfile
    container_name: php73
    volumes:
      - ./php:/var/www/html/
    ports:
      - 8000:80
  db:
    container_name: mysql8
    image: mysql:8.0
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: sifist
      MYSQL_USER: latihan
      MYSQL_PASSWORD: rahasia
    ports:
      - 6033:3306
  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: db
      PMA_USER: latihan
      PMA_PASSWORD: rahasia
    ports:
      - 5000:80
```

Lampiran 2.1 Listing docker-compose.yml