

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING  
LAPORAN PROYEK AKHIR**

**SISTEM TOKO ONLINE DENGAN LAMPP PADA UBUNTU SERVER DAN  
PROSES PEMBUATAN DOCKERFILE**



**DISUSUN OLEH:**

**NAMA ANGGOTA : ISNAN RIZQI K 123170041  
ALFRISKA DEVIANE P 123170108  
KELAS : B  
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.  
M IMAM ALFATAH**

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

## HALAMAN PENGESAHAN

### SISTEM TOKO ONLINE DENGAN LAMPP PADA UBUNTU SERVER DAN DOCKER

Disusun oleh :

Isnan Rizqi Kurniawan

123170041

Alfriska Deviane Puspita

123170108

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

M Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

## KATA PENGANTAR

Assalamualaikum wr.wb

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. Adapun laporan ini berisi tentang proyek akhir yang saya pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten laboratorium yang selalu membimbing dan mengajari saya dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena ini kritik dan 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

Wassalamualaikum wr.wb

Yogyakarta, 1 April 2020

Penyusun

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI .....	iv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang Masalah .....	1
1.2. Tujuan Proyek Akhir .....	2
1.3. Manfaat Proyek Akhir .....	2
1.4. Tahap Penyelesaian Proyek Akhir .....	2
<b>BAB II ISI DAN PEMBAHASAN .....</b>	<b>4</b>
2.1. Komponen yang Digunakan .....	4
2.2. Rancangan Arsitektur <i>Cloud Computing</i> .....	5
2.3. Parameter dan Konfigurasi .....	6
2.4. Tahap Implementasi.....	10
2.5. Hasil Implementasi .....	23
2.6. Pengujian Singkat .....	24
<b>BAB III JADWAL Pengerjaan dan Pembagian Tugas .....</b>	<b>27</b>
3.1. Agenda Pengerjaan .....	27
3.2. Keterangan Pembagian Tugas.....	27
<b>BAB IV KESIMPULAN DAN SARAN .....</b>	<b>28</b>
4.1. Kesimpulan .....	28
4.2. Saran .....	28
<b>DAFTAR PUSTAKA .....</b>	<b>29</b>

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Proyek Akhir

Cloud computing merupakan sebuah teknologi yang menyediakan layanan terhadap sumber daya komputasi melalui sebuah jaringan. Sumber daya yang di sediakan di dalam cloud computing meliputi mesin, media penyimpanan data, sistem operasi dan program aplikasi. Fitur dari cloud computing dipercaya akan jauh lebih hemat dan memuaskan (Muslim, M. A et, 2014). Cloud computing adalah model untuk memungkinkan diakses dari manapun, nyaman, sesuai permintaan jaringan sama dengan sumber daya komputasi yang telah dikonfigurasi yang dapat dengan cepat ditetapkan dan dirilis dengan usaha manajemen minimal Dari penyedia layanan. Maka dari itu, *Cloud Computing* menjadi sebuah tren baru untuk membuat sebuah data/file menjadi sangat mudah diakses dan privat.

Penggunaan cloud computing di masyarakat sendiri sangat bermacam-macam variasinya. Layanan yang diberikan oleh penyedia juga ada yang jenis *software*, juga sebuah *platform*, atau infrastruktur. Beberapa contohnya adalah Tokopedia yang merupakan *ecommerce* yang menyediakan barang-barang yang dapat dibeli melalui sistem, bahkan pengguna hanya tinggal memakai dan memilih barang yang akan dibeli, membayarnya juga bisa berbasis *online* seperti *e-banking*, *m-banking*, *virtual money*, dll.

Maka dari itu kami membuat sistem toko online dimana pengguna dapat membeli dan melihat barang yang tersedia secara *real time*. Keuntungan sistem toko yang online ini adalah pengguna atau calon pembeli tidak harus datang langsung ke toko yang dimana akan memberi batas pada jangkauan pasar pada toko. Ketika sebuah toko tersebut dapat memiliki aplikasi untuk menjual barangnya secara *online*, maka pengguna yang akan membeli tidak hanya pada daerah disekitarnya saja tetapi juga orang diseluruh belahan dunia ini. Pada sisi admin juga menguntungkan karena dapat dengan mudah mengecek persediaan yang juga permintaan secara langsung.

Pada penginstalan sistem yang akan digunakan adalah beberapa *software* yang mendukung penginstalan seperti. *VMWare Workstation* yang berfungsi untuk virtualisasi ubuntu *server*. Ubuntu *server* yang merupakan system operasi yang akan digunakan. PuTTY sebagai *remote* pada ubuntu yang telah diinstall. Setelah itu juga akan diinstall PHP-*ApacheMySQL* yang juga berfungsi agar dapat menjalankan file php dan html juga databasenya.

## 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 Penjualan Aksesoris Alfris.co yang telah dibuat sebelumnya dengan menggunakan Ubuntu 10.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Menggunakan docker sebagai PaaS untuk menjalankan Sistem Penjualan Aksesoris Alfris.co

## 1.3 Manfaat Proyek Akhir

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

1. Sistem Monitoring yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.
2. Perusahaan tidak perlu memperlakukan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.
3. Batasan memori penyimpanan multimedia menjadi tidak terbatas dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing*.
4. Permintaan data secara *realtime* dapat dilakukan secara terpusat maupun secara terpilih dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.

## 1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari sistem penjualan aksesoris Alfris.co untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Mengintegrasikan penyimpanan data *stock product* dan pemesanan pada docker dengan Sistem Penjualan Aksesoris Alfris.co yang berada di Ubuntu Server.
3. Menentukan konfigurasi yang tepat untuk pengaturan docker sehingga dapat digunakan sesuai *requirement* yang berupa:

- a. Dapat diakses oleh jaringan/IP tertentu saja.
  - b. Terdapat divisi pusat yang dapat masuk ke semua akun kecuali akun Z.
  - c. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
4. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
5. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Pencatatan Kendaraan ketersediaannya/*availability*-nya maksimal.
6. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dsb...
7. Menghasilkan Sistem Pelacakan Kereta Api yang berbasis *cloud computing* sesuai standar ISO 9001.

## BAB II

### ISI DAN PEMBAHASAN

#### 2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir ini terbagi menjadi dua sesuai dengan judul yang telah dituliskan sebelumnya. Berikut akan dibahas keduanya masing-masing dalam sub bab terkait.

##### 2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Sistem Toko Online dengan Ubuntu LAMPP” 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 5.2 sehingga dapat digunakan untuk menyimpan berbagai data kendaraan bermotor yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunanya ialah calon pembeli. Jadi semua orang yang akan membeli produk dapat mengakses web. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *public*.

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	Intel Core i3-4005U 1.70GHz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.9.129	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.9.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.9.1	Alamat untuk <i>gateway</i> akses keluar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	2GB	Alokasi RAM untuk <i>guest OS</i>
6.	Storage	20GB	Alokasi Harddisk 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

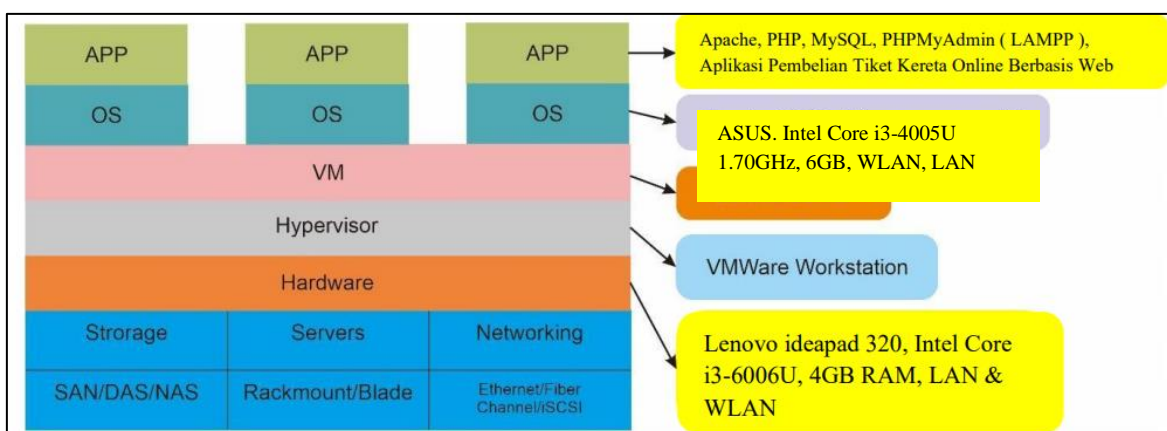
No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.2.24	Prosesor konten Web Server
		MySQL 5.2	Basis layanan manajemen basis data
		PHPMyAdmin	Layanan manajemen database MySQL berbasis Web GUI
2.	HTML	Versi 2.7	Dukungan bahasa pemrograman yang digunakan oleh sistem pelacakan kendaraan.

### 2.1.2 Komponen pada Proyek Kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Docker	Docker Server 19.03	Docker <i>engine server</i> yang digunakan dalam menjalankan <i>container</i>
		Docker Client 19.03	Docker <i>engine client</i> yang digunakan untuk berkomunikasi dengan <i>docker server</i> .
2.	Docker Compose	Docker Compose 1.17	Dukungan alat untuk mendefinisikan dan menjalankan <i>multi-container</i>

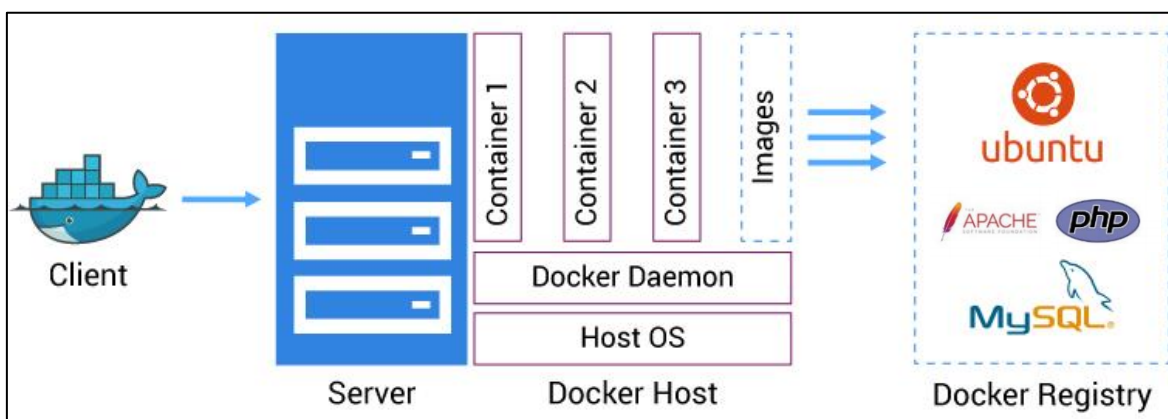
## 2.2 Rancangan Arsitektur Cloud Computing

Pada project akhir ini digunakan bentuk rancangan arsitektur IaaS di mana hardware laptop ASUS sebagai layer utama di bagian bawah, kemudian di atasnya adalah PaaS atau sistem operasi Windows dan aplikasi VMware Workstation yang berjalan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



**Gambar 2.1** Penjelasan layer arsitektur terhadap komponen penyusunnya

Setelah itu, pada proyek ini juga dilakukan proses pembuatan *dockerfile* yang bertujuan untuk membuat aplikasi sebagai sebuah *container* yang dapat diakses. Pada *docker* ini aplikasi web kita akan berada dalam sebuah *image*. *Image* sendiri adalah sebuah istilah dalam *docker* untuk mendefinisikan isi dari *container*. Arsitektur untuk *docker* dapat dilihat pada **Gambar 2.2** berikut. Pada *container* 1 akan berisi *web service*, dan pada *container* 2 akan berisi *database service* dari aplikasi web yang sudah kita buat sebelumnya. *Container-container* tersebut dapat disebut sebagai *image*.



**Gambar 2.2** Arsitektur *Docker*

## 2.3 Parameter dan Konfigurasi

### 2.3.1 Parameter dan Konfigurasi pada Proyek Pertama

Untuk dapat digunakan, maka VMware Workstation perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada **Tabel 2.3** berikut ini

**Tabel 2.3** Konfigurasi pada VMware Workstation yang digunakan

No	Nama Parameter	Nilai	Keterangan
1.	VMWare ESXi <i>version</i>	15.5.2	Keterangan versi yang digunakan
2.	IPv4 <i>hypervisor</i>	IP: 192.168.9.129	IP number untuk akses manajemen hypervisor melalui web based. Pengaturan IP address menggunakan mode statik
		SM: 255.255.255.0	Kelas IP/subnet mask yang digunakan.
		DNS: 192.168.9.1	Alamat IP untuk fungsionalitas DNS.
		DW: 192.168.9.1	Alamat untuk gateway atau gerbang menuju akses jaringan luar.
3.	<i>Processor info</i>	Intel Core i3-4005U 1.70GHz	Jenis processor yang digunakan pada hypervisor.
4.	<i>RAM info</i>	DDR3L 6 Gb	Kapasitas RAM pada hypervisor

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

#### **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

#### **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 php 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

```
- php phpmyadmin php-mbstring php-gettext : nama paket aplikasi
untuk PHPMYAdmin
```

#### **Modul 2.4** Parameter instalasi PHPMYAdmin

Parameter yang digunakan untuk mengatur parameter plugin dari user root dapat dilihat pada penjelasan **Modul 2.5** berikut ini:

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

#### **Modul 2.5** Parameter pengaturan parameter plugin dari user root

### **2.3.2 Parameter dan Konfigurasi pada Proyek Kedua**

Langkah pertama dalam install docker adalah meng update list packages. Parameteryang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

```
sudo apt update
```

#### **Modul 2.6** Parameter update list packages

Langkah kedua dalam install docker adalah install beberapa package yang diperlukan agar apt bisa diakses HTTPS. Parameter yang digunakan untuk mengupdate list packages dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

```
sudo apt install apt-transport-https ca-certificates curl softwareproperties-common
```

#### **Modul 2.7** Parameter package apt

Langkah ketiga dalam install docker adalah menambahkan GPG key dari untuk Docker repository ke sistem kita. Parameter yang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.8** berikut ini:

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

#### **Modul 2.8** Parameter menambahkan GPG key dari untuk Docker repository ke system

Langkah keempat dalam install docker adalah menambahkan Docker repository ke APT. Parameter yang digunakan untuk menambahkan docker repository ke APT dapat dilihat pada penjelasan Modul 2.9 berikut ini:

```
Sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
bionic stable"
```

#### **Modul 2.9** Parameter menambahkan Docker repository ke APT

Langkah kelima dalam install docker adalah update package database. Parameter yang digunakan untuk meng update package database dapat dilihat pada penjelasan **Modul 2.10** berikut ini:

```
sudo apt update
```

**Modul 2.10** Parameter update package database

Langkah keenam dalam install docker pastikan kita akan menginstall Docker repo ke dalam ubuntu. Parameter yang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

```
apt-cache policy docker-ce
```

**Modul 2.11** Parameter memastikan docker repo

Langkah ketujuh dalam install docker adalah install docker itu sendiri. Parameter yang digunakan untuk menginstall docker dapat dilihat pada penjelasan **Modul 2.12** berikut ini:

```
sudo apt install docker-ce
```

**Modul 2.12** Parameter untuk install docker

Langkah kedelapan dalam install docker adalah memastikan bahwa docker sudah berjalan. Parameter yang digunakan untuk memastikan bahwa docker sudah berjalan dapat dilihat pada penjelasan **Modul 2.13** berikut ini:

```
sudo systemctl status docker
```

**Modul 2.13** Parameter untuk memastikan docker sudah berjalan

Setelah docker berjalan, harus menginstall docker compose. Parameter yang digunakan untuk download docker-compose binary file dapat dilihat pada penjelasan **Modul 2.14** berikut ini:

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.24.1/dockercompose-$(uname
-s)-$(uname -m) -o /usr/local/bin/docker-compose
```

**Modul 2.14** Parameter untuk download docker-compose binary file

Parameter yang digunakan untuk install curl dapat dilihat pada penjelasan **Modul 2.15** berikut ini:

```
sudo apt install curl -y
```

**Modul 2.15** Parameter untuk install curl

Parameter yang digunakan untuk install docker compose dapat dilihat pada penjelasan **Modul 2.16** berikut ini:

```
sudo chmod +x /usr/local/bin/docker-compose
```

**Modul 2.16** Parameter untuk install docker compose

Parameter yang digunakan untuk memastikan docker compose sudah berjalan dengan baik dapat dilihat pada penjelasan **Modul 2.17** berikut ini:

```
docker-compose version
```

**Modul 2.17** Parameter untuk memastikan docker compose sudah berjalan

Setelah docker dan docker compose sudah dipastikan berjalan dengan baik, tahap selanjutnya adalah mengatur docker-compose untuk proyek yang akan dikerjakan. Parameter yang digunakan untuk masuk ke folder proyek dapat dilihat pada penjelasan **Modul 2.18** berikut ini:

```
cd /var/www/html/projek/
```

**Modul 2.18** Parameter untuk masuk ke folder proyek

Parameter yang digunakan untuk membuat Dockerfile dapat dilihat pada penjelasan

**Modul 2.19** berikut ini:

```
nano Dockerfile
```

**Modul 2.19** Parameter membuat Dockerfile

Parameter yang digunakan menjalankan web-server, mysql-server, dan phpmyadmin dapat dilihat pada penjelasan **Modul 2.20** berikut ini:

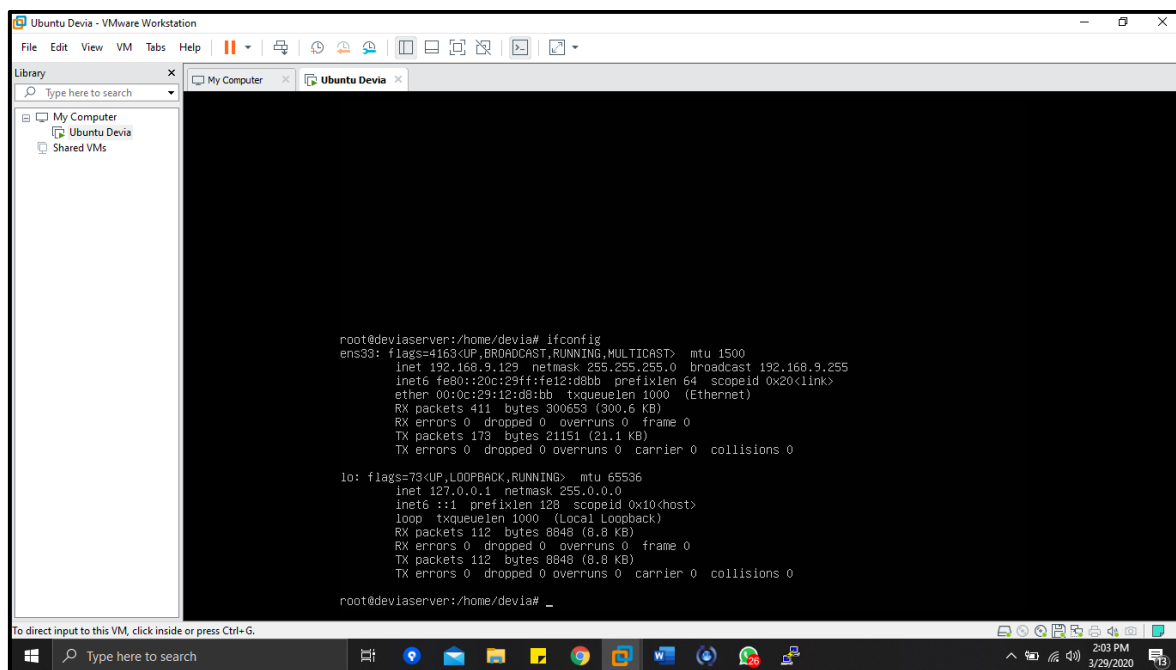
```
docker-compose up
```

**Modul 2.20** Parameter menjalankan web-server, mysql-server, dan phpMyAdmin

## 2.4 Tahap Implementasi

### 2.4.1 Install VMWare Workstation

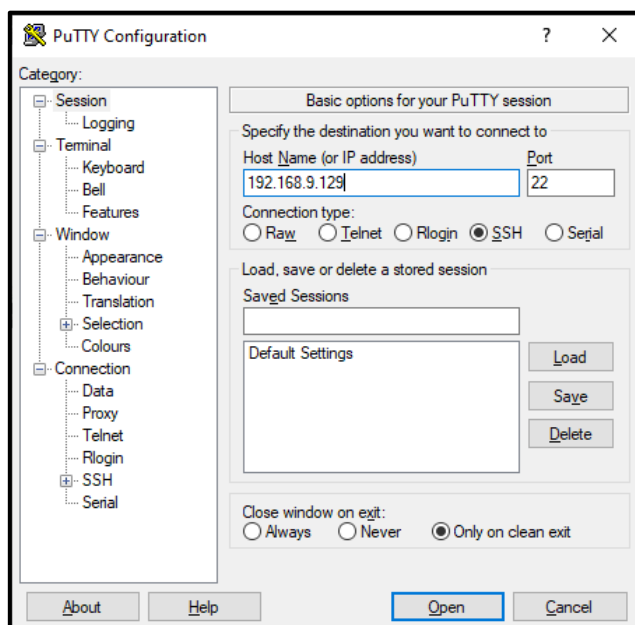
VMWare Workstation ini digunakan sebagai platform virtualisasi untuk menginstall Ubuntu server. Tampilan pada software ini dapat dilihat pada **Gambar 2.1** berikut:



**Gambar 2.1** Tampilan VMWare Workstation

#### 2.4.2 Install PuTTY

Setelah proses install selesai lalu konvigurasikan PuTTY dengan mengisi host name ubuntu yang telah dibuat. Tampilan pada software ini dapat dilihat pada **Gambar 2.2** berikut :



**Gambar 2.2** Tampilan PuTTY

#### 2.4.3 Install apache 2

Penginstallan apache2 menggunakan perintah yang ada pada **Perintah Program 2.1**.

```
$ sudo apt install apache2
```

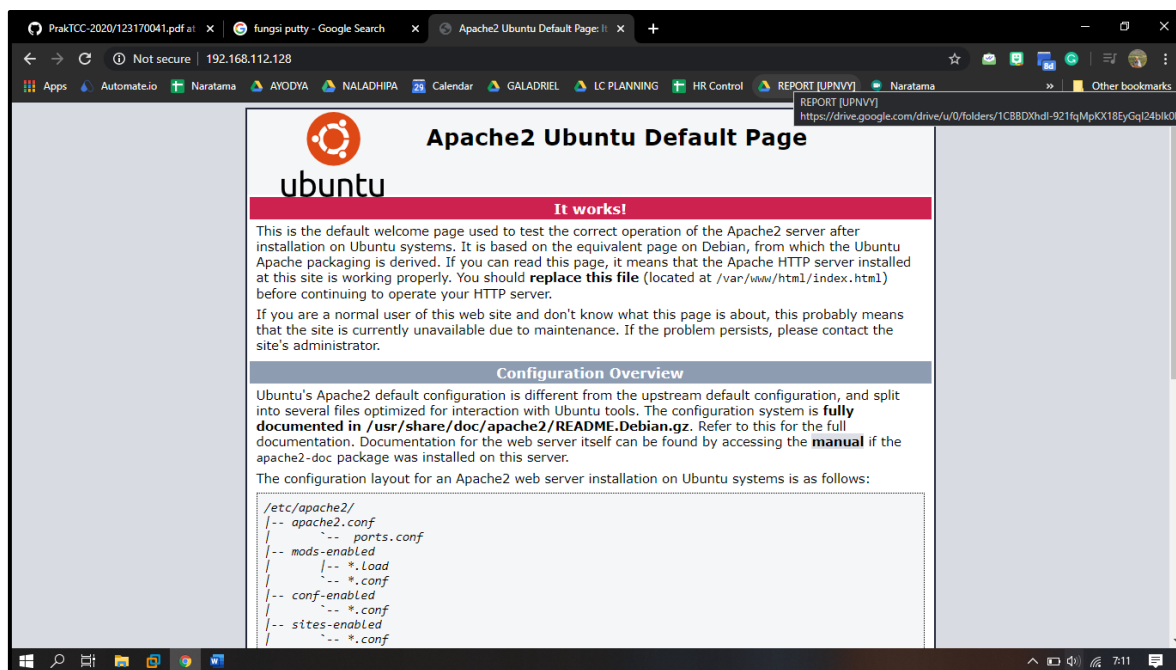
### Perintah Program 2.1 Operasi instalasi paket aplikasi *apache2*

Kemudian akan mengkonfigurasi pengaturan firewall, dengan perintah pada **Perintah Program 2.2.**

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

### Perintah Program 2.2 Konfigurasi Firewall

Dimana akan menaktifkan status pada port yang bernama Apache Full. Tampilan ketika sudah berhasil terinstall Apache adalah dengan mengetikan IP server yang sebelumnya diketahui dari WMWare kedalam browser. Tampilan pada Apache2 ini dapat dilihat pada **Gambar 2.3** berikut :



**Gambar 2.3** Tampilan Apache2

### 2.4.4 Install MySQL dan PHP

Pada instalasi mySQL perintah yang digunakan dengan perintah pada **Perintah Program 2.3.**

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

### Perintah Program 2.3 Instalasi mySQL



Setelah itu, akan diatur konfigurasi berupa user, password, hak akses dan sebagainya pada mysql sebagai pengaturan dasar untuk pengamanan. Hal tersebut dilakukan dengan perintah pada **Perintah Program 2.4**.

```
$ sudo mysql_secure_installation
```

**Perintah Program 2.4** Konfigurasi keamanan dasar mySQL

Setelah itu akan ada beberapa pernyataan untuk konfigurasi dibawah ini :

- a) VALIDATE PASSWORD plugin? No
- b) Remove anonymous user? Yes
- c) Disallow root login remotely? Yes
- d) Remove test database and access to it? Yes
- e) Reload privilege tables now? Yes

Selanjutnya akan melakukan instalasi pada PHP. Perintah yang digunakan dengan perintah pada **Perintah Program 2.5**.

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

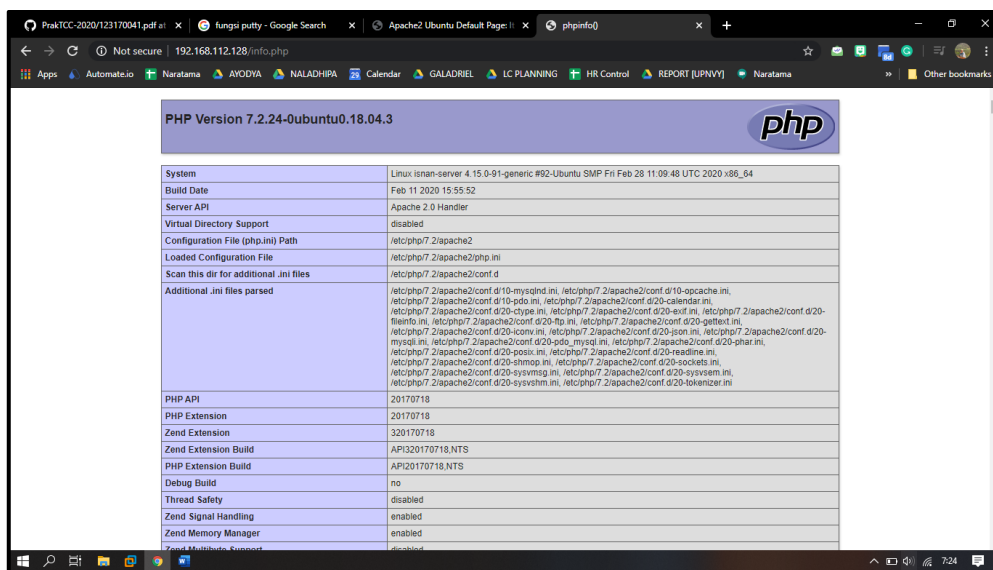
**Perintah Program 2.5** Instalasi PHP

Setelah itu akan membuat syntax sederhana untuk membuka phpinfo yang dibuat pada aplikasi nano. Pada perintah dibawah ini, memungkinkan kita untuk membuat file bernama info.php yang berada di direktori /var/www/html/ seperti pada **Perintah Program 2.6**.

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

**Perintah Program 2.6** Membuat file info.php

Hasilnya ketika dicek pada browser dengan [http://\(IPAddress\)/info.php](http://(IPAddress)/info.php) akan muncul halaman seperti pada **Gambar 2.4**.



**Gambar 2.4** Tampilan PHPinfo

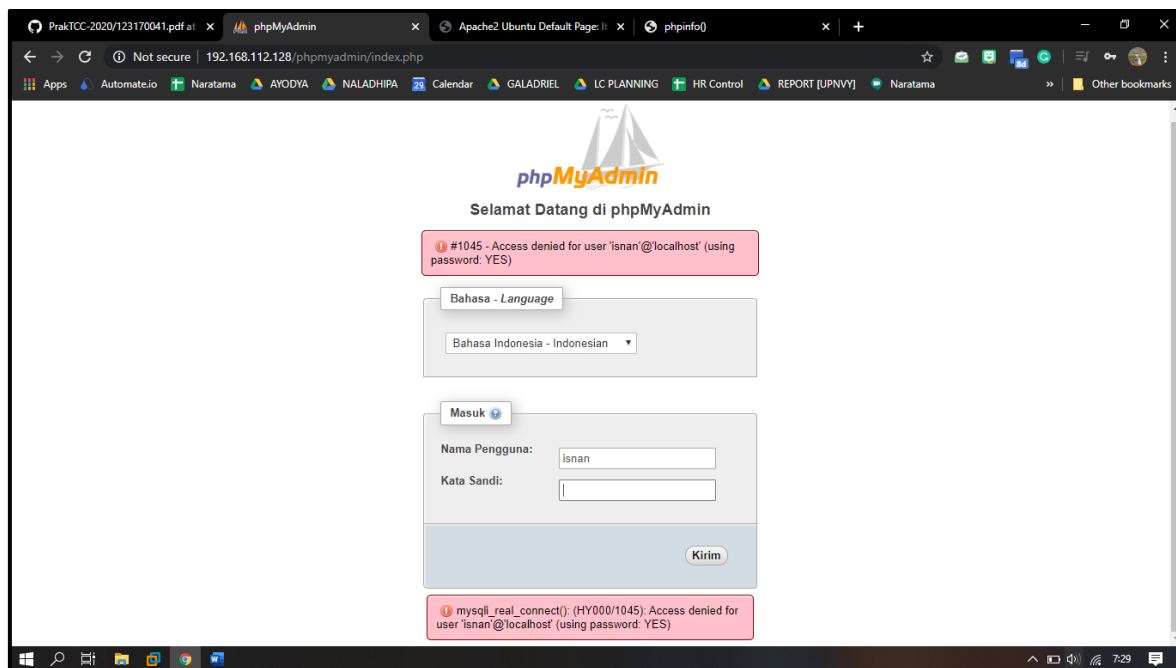
### 2.4.5 Install php myadmin

Pada penginstallan phpMyAdmin perintah yang digunakan adalah sebagai berikut pada **Perintah Program 2.7**.

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

#### **Perintah Program 2.7** Install phpMyAdmin

Hasilnya kemudian akan tampil seperti pada **Gambar 2.5**



**Gambar 2.5** Tampilan phpMyAdmin

Setelah instalasi selesai maka akan muncul error seperti pada gambar di atas. Untuk memperbaiki error tersebut diperlukan konfigurasi MYSQL sebagai user root dengan code pada **Perintah Program 2.8**.

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

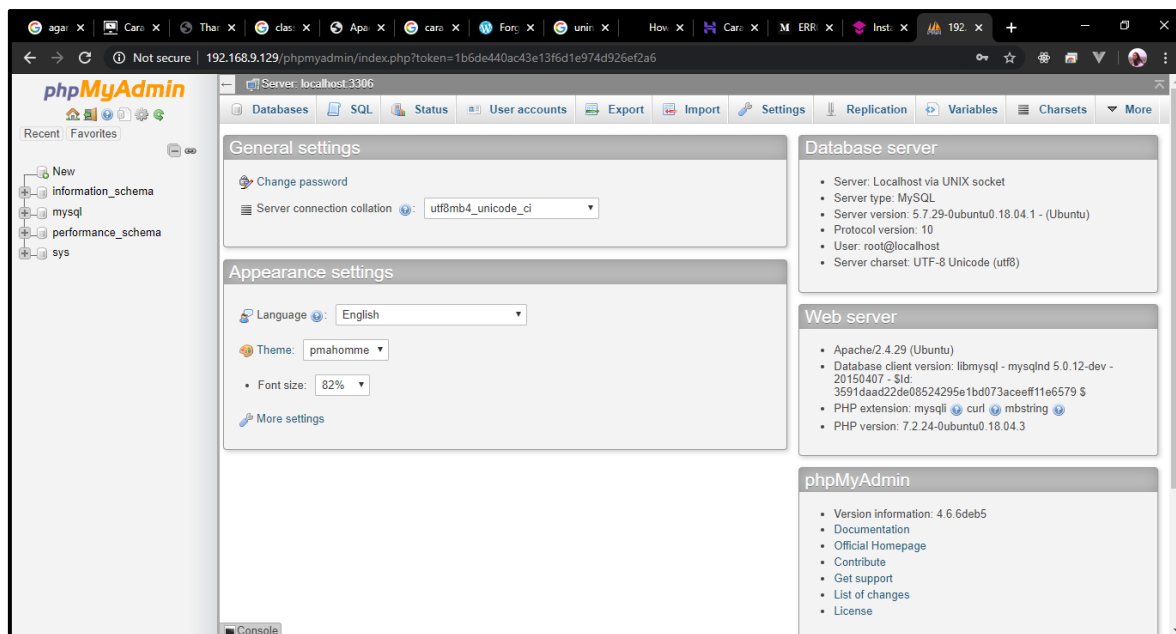
**Perintah Program 2.8** Masuk kedalam mysql, untuk memperbaiki error

Setelah berhasil akan muncul tampilan mysql lalu ketik query seperti pada **Perintah Program 2.9**.

```
mysql> UPDATE mysql.user SET plugin= 'mysql_native_password', authentication_string=
PASSWORD ('devia') WHERE User='root';
mysql> flush privileges
```

**Perintah Program 2.9** SET password yang sudah dibuat,

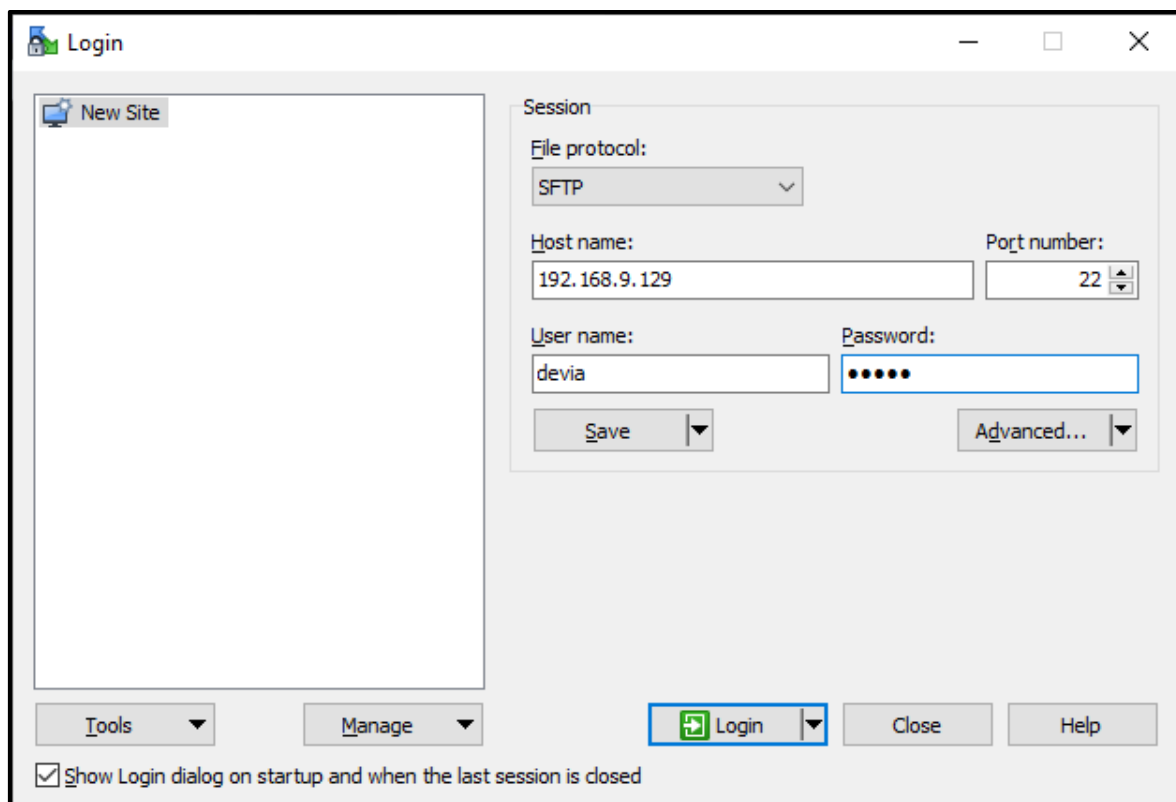
Setelah berhasil coba kembali login ke phpMyAdmin, jika berhasil maka akan menampilkan seperti pada gambar dibawah ini pada **Gambar 2.6**.



**Gambar 2.6** Tampilan dashboard phpMyAdmin

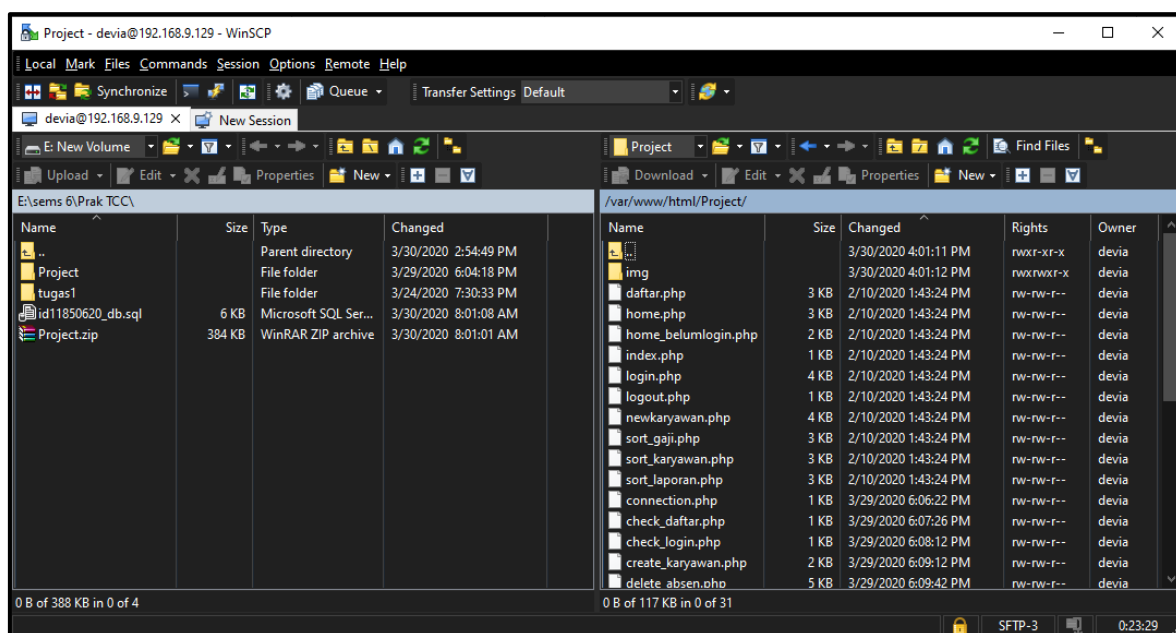
#### 2.4.6 Install WinSCP

Pilih new site lalu pastikan file protocol yang digunakan adalah SFTP kemudian masukan hostname, user dan password sesuai yang telah dibuat di ubuntu. pada **Gambar 2.7**.



**Gambar 2.7** Tampilan WinSCP

Jika berhasil login akan tampil seperti **Gambar 2.8**. Bagian kanan merupakan tampilan dari berkas yang berada pada server, sedangkan yang merupakan berkas dari workstation ke dalam server.



**Gambar 2.8** Dashboard WinSCP

Untuk mengunggah file yang akan digunakan, pastikan kita sudah mengubah hak akses. Cara mengubah hak akses yaitu dengan masuk ke PuTTY lalu menuliskan code pada

**Perintah Program 2.10.**

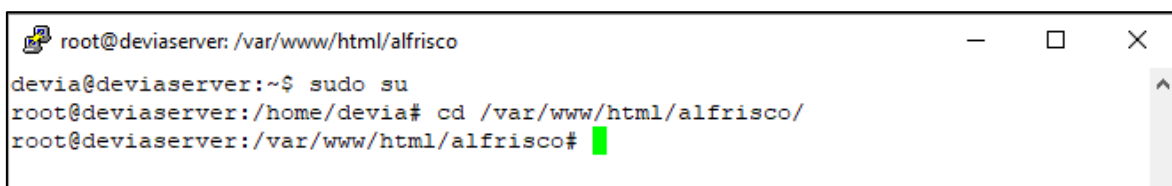
```
sudo chown devia /var/www/html
```

**Perintah Program 2.10** Ubah hak akses

Devia merupakan nama user sedangkan /var/www/html merupakan directory yang akan dituju. Setelah itu buka kembali WinSCP lalu arahkan kolom bagian kanan. Untuk mengunggah file bisa dilakukan dengan drag and drop atau klik kanan file lalu pilih upload. Test hasilnya dengan mengetik url 192.168.9.129/alfrico sesuaikan dengan ip dan nama berkas yang ada.

#### 2.4.7 Masuk ke folder yang berisi project

Sebelumnya sudah membuat layanan yang berbasis dokku dimana setelah itu dapat langsung digunakan untuk membuat sebuah dockerfile sebagai *image* pada dokker. Pada **Gambar 2.9** adalah proses untuk masuk kedalam folder yang berisi proyek.

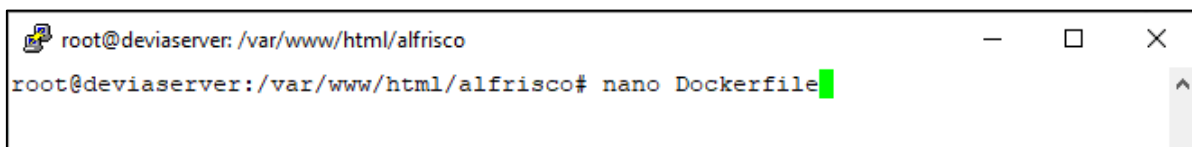


```
root@deviaserver: /var/www/html/alfrisco
devia@deviaserver:~$ sudo su
root@deviaserver:/home/devia# cd /var/www/html/alfrisco/
root@deviaserver:/var/www/html/alfrisco#
```

**Gambar 2.9** Folder *project*

#### 2.4.8 Membuat dockerfile

Pada **Gambar 2.10** adalah perintah yang digunakan untuk membuat *dockerfile*. Setelah masuk kedalam program nano, maka selanjutnya adalah mengetikkan perintah atau code yang berfungsi agar kita dapat membuat *dockerfile* dari program yang sudah kita buat.

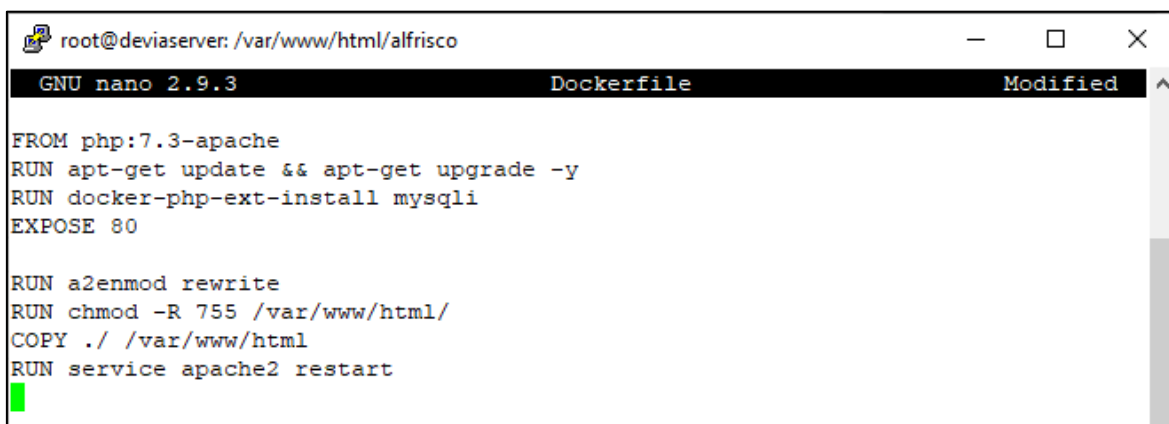


```
root@deviaserver: /var/www/html/alfrisco
root@deviaserver:/var/www/html/alfrisco# nano Dockerfile
```

**Gambar 2.10** Membuat *dockerfile*

#### 2.4.9 Isi dockerfile dengan code ini:

Pada **Gambar 2.11** adalah sebuah kode yang ditulis dalam *nano*, berfungsi untuk memberikan perintah agar program yang kita buat menjadi *dockerfile*.



```

root@deviaserver: /var/www/html/alfrisco
GNU nano 2.9.3 Dockerfile Modified
FROM php:7.3-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli
EXPOSE 80

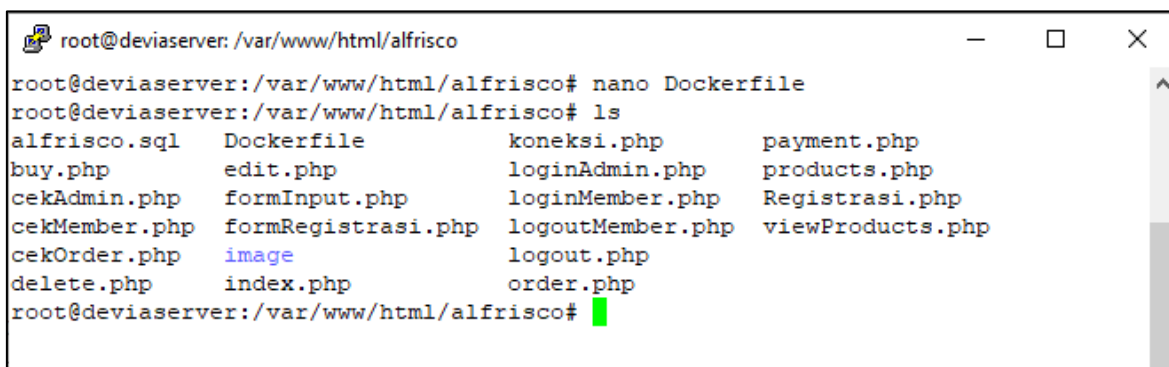
RUN a2enmod rewrite
RUN chmod -R 755 /var/www/html/
COPY ./ /var/www/html
RUN service apache2 restart

```

**Gambar 2.11** Dockerfile

#### 2.4.10 Cek apakah file sudah tersimpan

Pada **Gambar 2.12** adalah langkah untuk mengecek, apakah proyek sudah berhasil dibuat *dockerfile*-nya. Dapat dilihat pada **Gambar 2.12** proyek kita berhasil dibuat dalam bentuk *dockerfile* atau *image*.



```

root@deviaserver: /var/www/html/alfrisco
root@deviaserver:/var/www/html/alfrisco# nano Dockerfile
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  Dockerfile      koneksi.php      payment.php
buy.php       edit.php        loginAdmin.php   products.php
cekAdmin.php  formInput.php   loginMember.php  Registrasi.php
cekMember.php formRegistrasi.php logoutMember.php  viewProducts.php
cekOrder.php  image           logout.php
delete.php    index.php       order.php
root@deviaserver:/var/www/html/alfrisco#

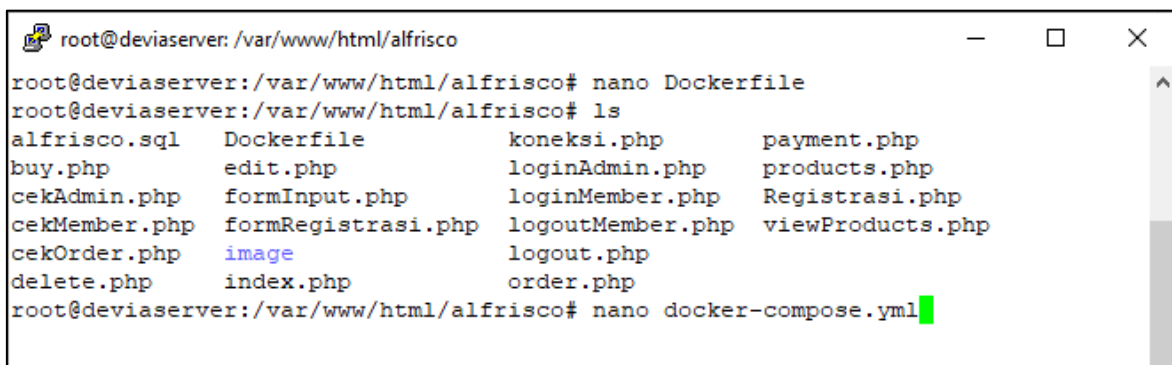
```

**Gambar 2.12** Cek Dockerfile

#### 2.4.11 Membuat file docker-compose

Pada **Gambar 2.13** adalah langkah untuk membuat *docker-compose* yang berfungsi sebagai menjalankan container docker secara bersamaan. *docker-compose* akan dibuat dengan nano dan ekstensinya adalah *yml*. Untuk membuat sebuah *docker-compose* dalam

nano kita harus menuliskan beberapa perintah seperti pada **Gambar 2.14** dan **Gambar 2.15**. Jika berhasil maka dapat dicek dengan menampilkan seperti pada **Gambar 2.16**.

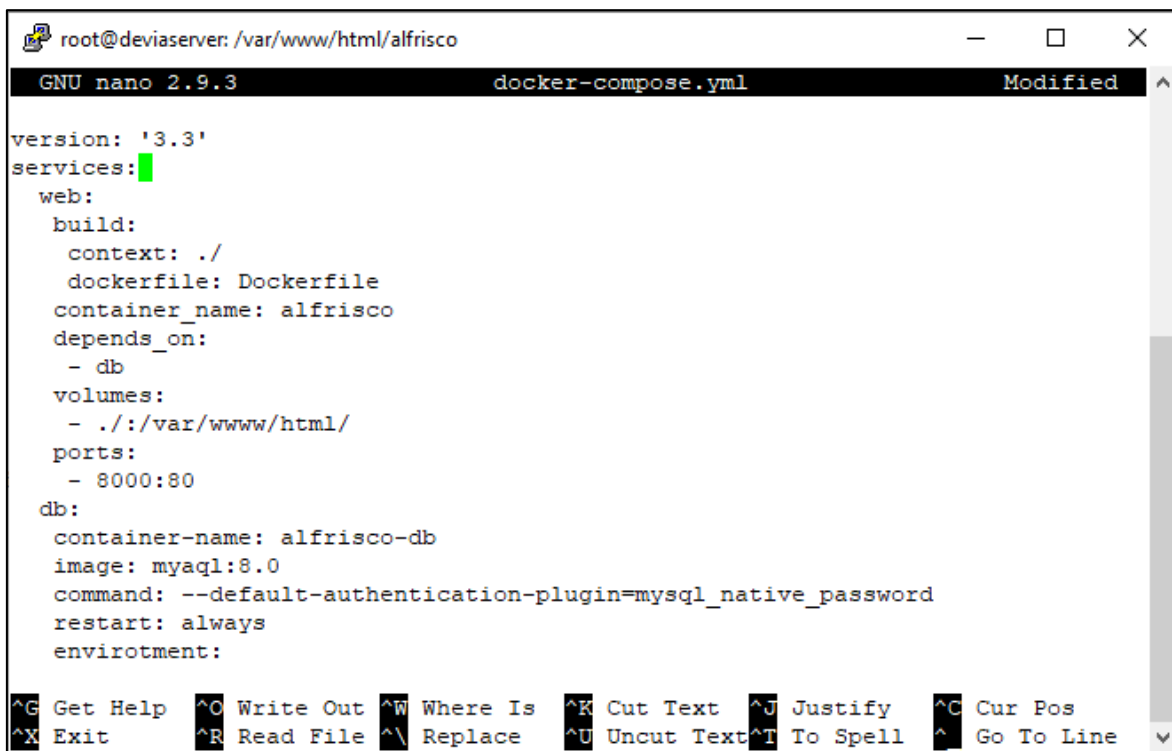


```

root@deviaserver: /var/www/html/alfrisco
root@deviaserver:/var/www/html/alfrisco# nano Dockerfile
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  Dockerfile      koneksi.php      payment.php
buy.php       edit.php        loginAdmin.php   products.php
cekAdmin.php  formInput.php   loginMember.php  Registrasi.php
cekMember.php formRegistrasi.php logoutMember.php  viewProducts.php
cekOrder.php  image           logout.php
delete.php    index.php       order.php
root@deviaserver:/var/www/html/alfrisco# nano docker-compose.yml

```

**Gambar 2.13** Membuat Docker-compose



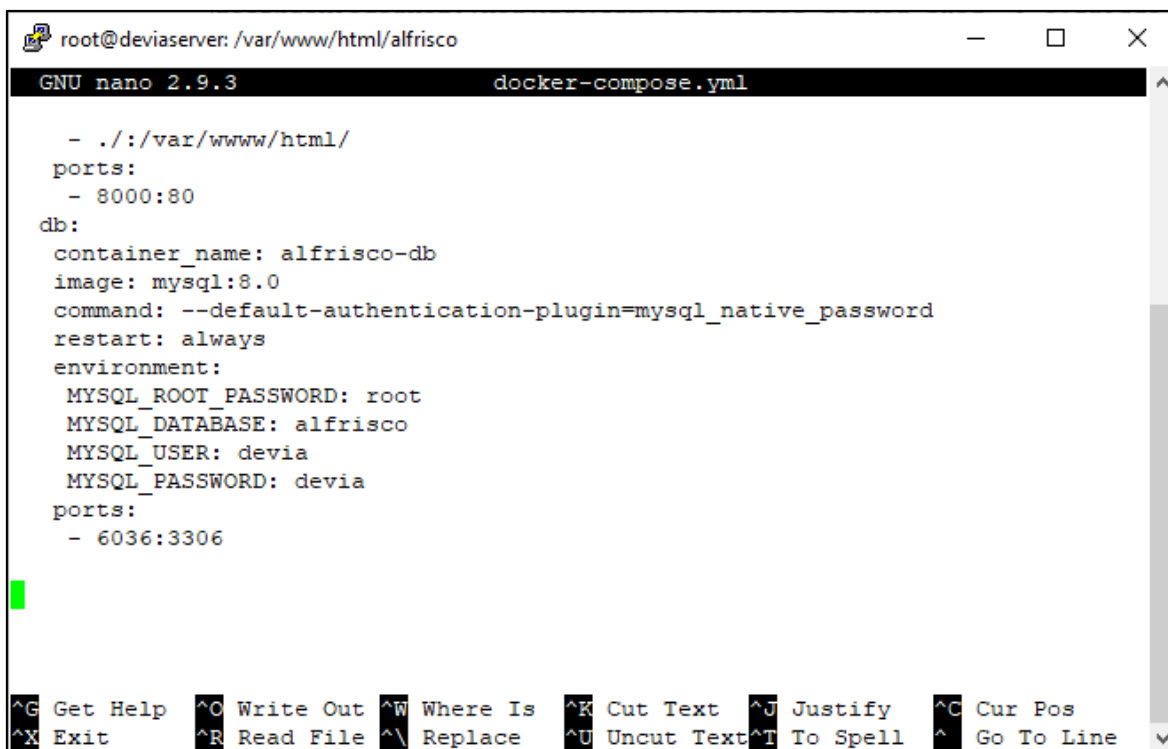
```

GNU nano 2.9.3      docker-compose.yml      Modified
version: '3.3'
services:
  web:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: alfrisco
    depends_on:
      - db
    volumes:
      - ./:/var/www/html/
    ports:
      - 8000:80
  db:
    container-name: alfrisco-db
    image: myaql:8.0
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    envirotment:

```

<sup>^</sup>G Get Help    <sup>^</sup>O Write Out    <sup>^</sup>W Where Is    <sup>^</sup>K Cut Text    <sup>^</sup>J Justify    <sup>^</sup>C Cur Pos  
<sup>^</sup>X Exit        <sup>^</sup>R Read File    <sup>^</sup>\ Replace    <sup>^</sup>U Uncut Text <sup>^</sup>T To Spell    <sup>^</sup> Go To Line

**Gambar 2.14** Isi Docker-compose



```

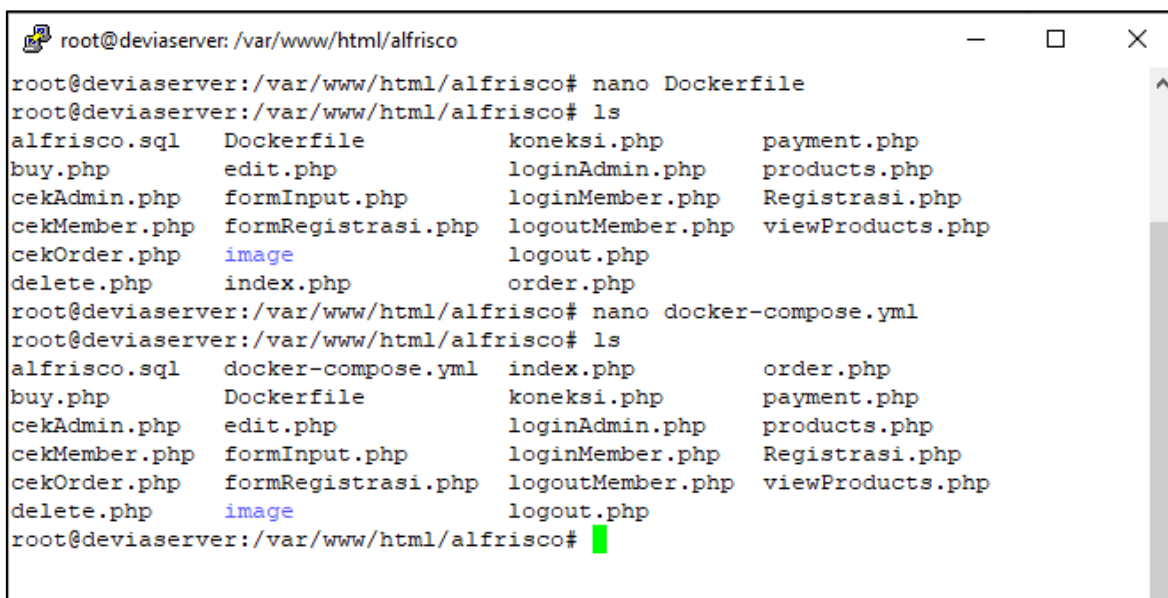
root@deviaserver: /var/www/html/alfrisco
GNU nano 2.9.3 docker-compose.yml

- ./:/var/www/html/
ports:
- 8000:80
db:
  container_name: alfrisco-db
  image: mysql:8.0
  command: --default-authentication-plugin=mysql_native_password
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: root
    MYSQL_DATABASE: alfrisco
    MYSQL_USER: devia
    MYSQL_PASSWORD: devia
  ports:
    - 6036:3306

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Gambar 2.15 Isi Docker-compose



```

root@deviaserver: /var/www/html/alfrisco
root@deviaserver:/var/www/html/alfrisco# nano Dockerfile
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  Dockerfile      koneksi.php      payment.php
buy.php       edit.php        loginAdmin.php   products.php
cekAdmin.php  formInput.php   loginMember.php  Registrasi.php
cekMember.php formRegistrasi.php logoutMember.php  viewProducts.php
cekOrder.php  image           logout.php
delete.php    index.php       order.php
root@deviaserver:/var/www/html/alfrisco# nano docker-compose.yml
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  docker-compose.yml  index.php      order.php
buy.php       Dockerfile          koneksi.php     payment.php
cekAdmin.php  edit.php            loginAdmin.php  products.php
cekMember.php formInput.php       loginMember.php Registrasi.php
cekOrder.php  formRegistrasi.php logoutMember.php viewProducts.php
delete.php    image              logout.php
root@deviaserver:/var/www/html/alfrisco#

```

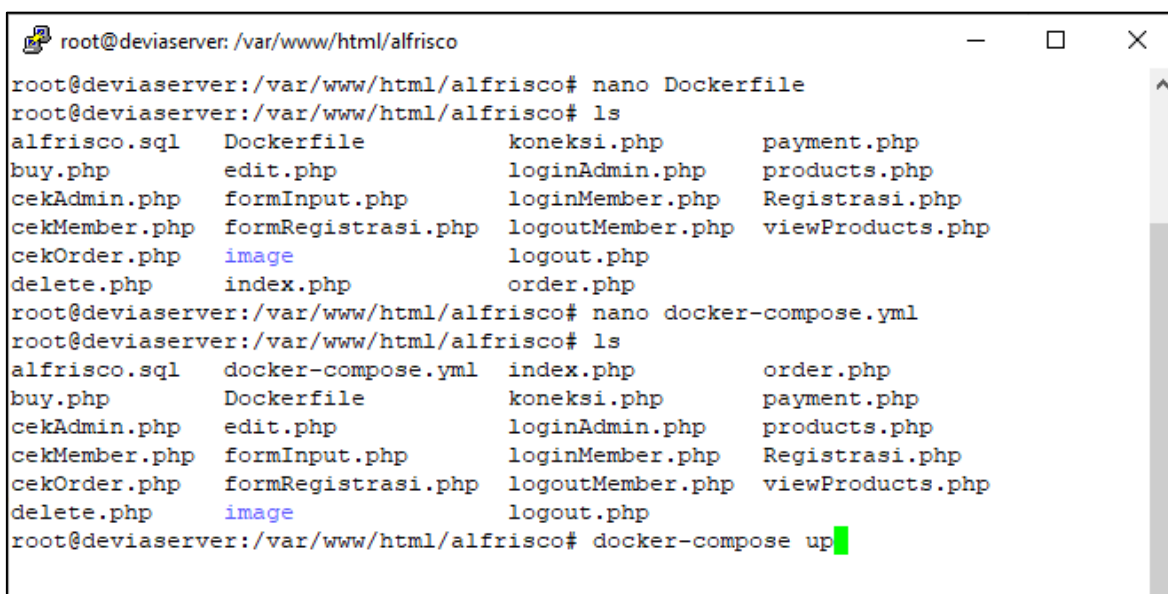
Gambar 2.16 Cek Docker-compose

#### 2.4.12 Docker-compose up

Pada **Gambar 2.17** adalah langkah agar *container* dapat mulai digunakan. Jika sudah diupload selanjutnya akan menggunakan perintah *pull* untuk finalisasi seperti pada **Gambar**



2.18. Ketika sudah selesai maka akan muncul tampilan seperti pada **Gambar 2.19** yang menandakan *docker* sudah selesai.

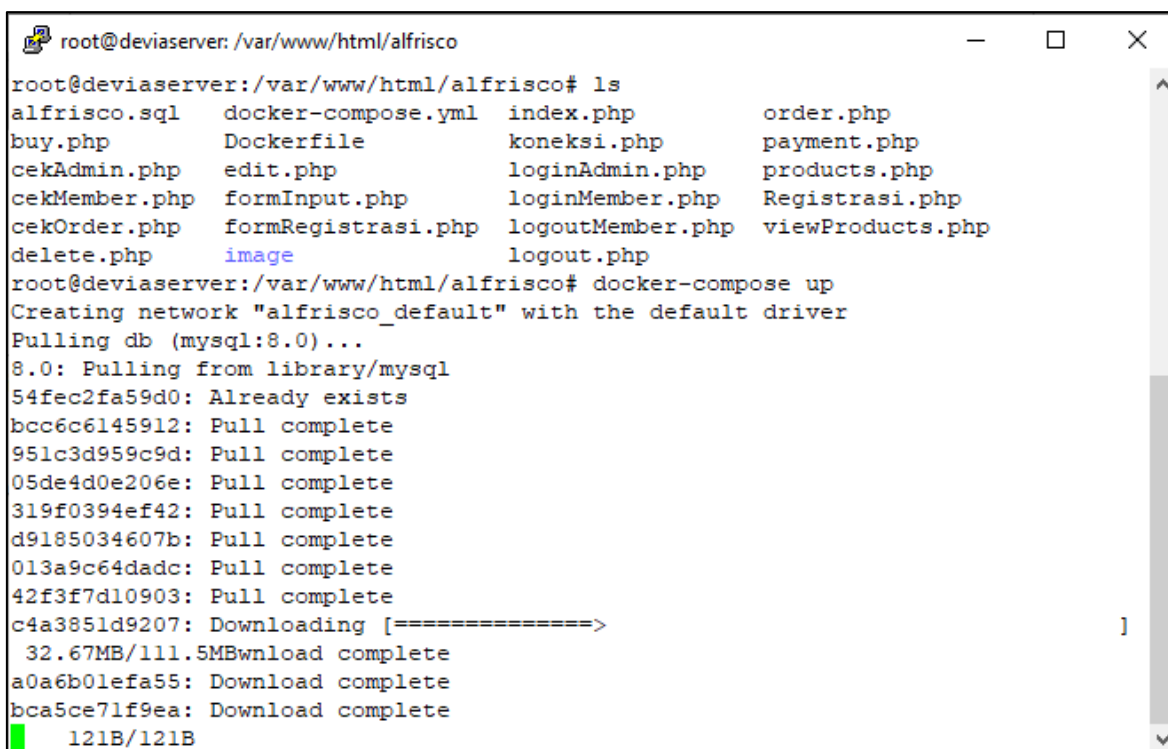


```

root@deviaserver: /var/www/html/alfrisco
root@deviaserver:/var/www/html/alfrisco# nano Dockerfile
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  Dockerfile      koneksi.php      payment.php
buy.php       edit.php        loginAdmin.php   products.php
cekAdmin.php  formInput.php   loginMember.php  Registrasi.php
cekMember.php formRegistrasi.php logoutMember.php  viewProducts.php
cekOrder.php  image          logout.php
delete.php    index.php      order.php
root@deviaserver:/var/www/html/alfrisco# nano docker-compose.yml
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  docker-compose.yml  index.php      order.php
buy.php       Dockerfile          koneksi.php     payment.php
cekAdmin.php  edit.php            loginAdmin.php  products.php
cekMember.php formInput.php       loginMember.php Registrasi.php
cekOrder.php  formRegistrasi.php  logoutMember.php viewProducts.php
delete.php    image              logout.php
root@deviaserver:/var/www/html/alfrisco# docker-compose up

```

**Gambar 2.17** Cek Docker-compose upload



```

root@deviaserver: /var/www/html/alfrisco
root@deviaserver:/var/www/html/alfrisco# ls
alfrisco.sql  docker-compose.yml  index.php      order.php
buy.php       Dockerfile          koneksi.php     payment.php
cekAdmin.php  edit.php            loginAdmin.php  products.php
cekMember.php formInput.php       loginMember.php Registrasi.php
cekOrder.php  formRegistrasi.php  logoutMember.php viewProducts.php
delete.php    image              logout.php
root@deviaserver:/var/www/html/alfrisco# docker-compose up
Creating network "alfrisco_default" with the default driver
Pulling db (mysql:8.0)...
8.0: Pulling from library/mysql
54fec2fa59d0: Already exists
bcc6c6145912: Pull complete
951c3d959c9d: Pull complete
05de4d0e206e: Pull complete
319f0394ef42: Pull complete
d9185034607b: Pull complete
013a9c64dad3: Pull complete
42f3f7d10903: Pull complete
c4a3851d9207: Downloading [=====>]
 32.67MB/111.5MBwnload complete
a0a6b01efa55: Download complete
bca5ce71f9ea: Download complete
121B/121B

```

**Gambar 2.18** Proses pull

```

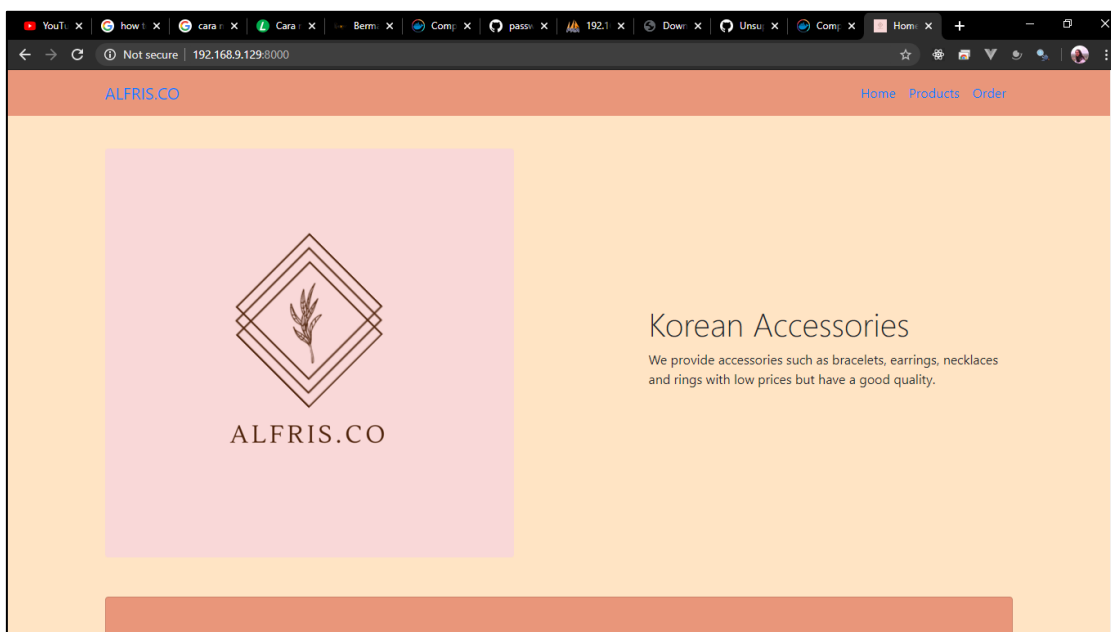
root@deviaserver: /var/www/html/alfrisco
fari/537.36"
alfrisco | 192.168.9.1 - - [29/Apr/2020:03:14:56 +0000] "GET /image/tp.png HTTP/
1.1" 200 35501 "http://192.168.9.129:8000/index.php" "Mozilla/5.0 (Windows NT 10
.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safa
ri/537.36"
alfrisco | 192.168.9.1 - - [29/Apr/2020:03:14:56 +0000] "GET /image/ig.png HTTP/
1.1" 200 47135 "http://192.168.9.129:8000/index.php" "Mozilla/5.0 (Windows NT 10
.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safa
ri/537.36"
alfrisco | 192.168.9.1 - - [29/Apr/2020:03:14:56 +0000] "GET /image/B001.JPG HTT
P/1.1" 200 128161 "http://192.168.9.129:8000/index.php" "Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 S
afari/537.36"
alfrisco | 192.168.9.1 - - [29/Apr/2020:03:15:12 +0000] "GET /products.php HTTP/
1.1" 200 1724 "http://192.168.9.129:8000/index.php" "Mozilla/5.0 (Windows NT 10.
0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safar
i/537.36"
alfrisco | 192.168.9.1 - - [29/Apr/2020:03:17:01 +0000] "GET /phpmyadmin HTTP/1.
1" 404 494 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH
TML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
alfrisco | 192.168.9.1 - - [29/Apr/2020:03:17:08 +0000] "GET / HTTP/1.1" 200 197
7 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/81.0.4044.129 Safari/537.36"

```

**Gambar 2.19** Proses pull berhasil

#### 2.4.12 Cek URL dengan IP:PORT

Pada **Gambar 2.20** adalah langkah untuk mengecek apakah sudah berhasil dockerfile sudah berhasil atau belum. Jika sudah maka, proyek akan bisa mulai digunakan.

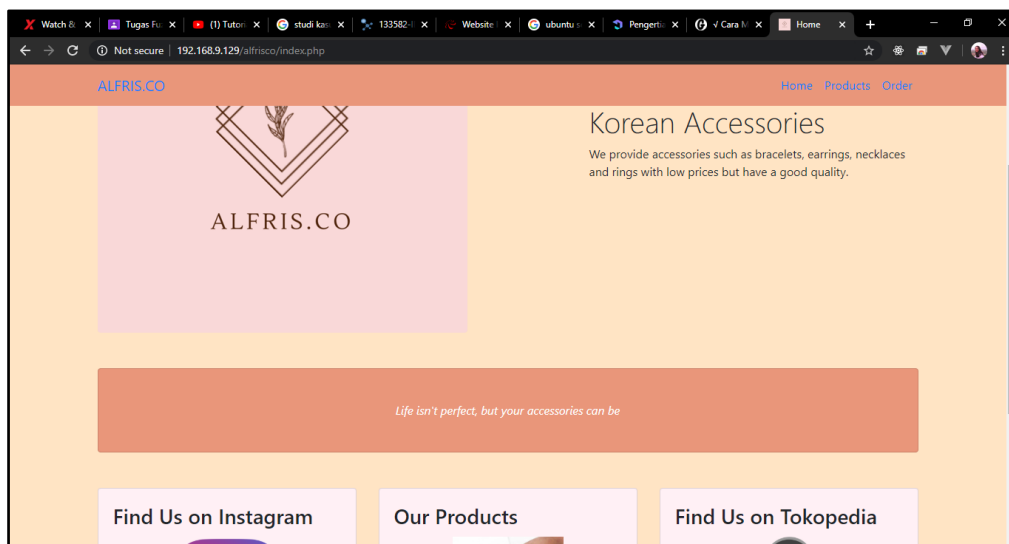


**Gambar 2.20** Tampilan

## 2.5 Hasil Implementasi

### 2.5.1 Hasil Implementasi pada Proyek Pertama

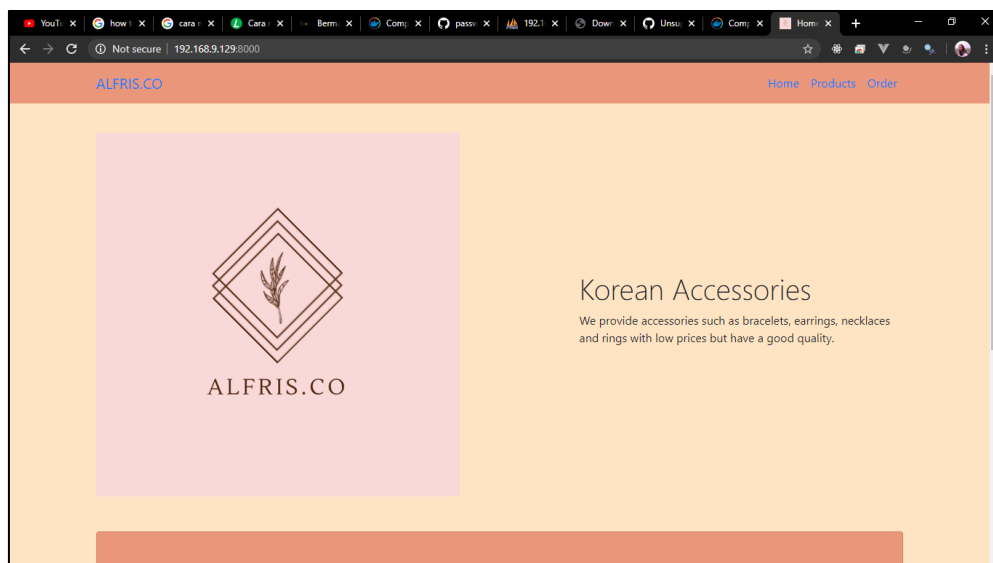
Dengan mengetik 192.168.9.129/alfrico maka akan muncul web seperti pada gambar pada **Gambar 2.21**.



**Gambar 2.21** Tampilan awal program

### 2.5.2 Hasil Implementasi pada Proyek Kedua

Dengan mengetik 192.168.9.129:8000 maka akan muncul web seperti pada gambar pada **Gambar 2.22**.

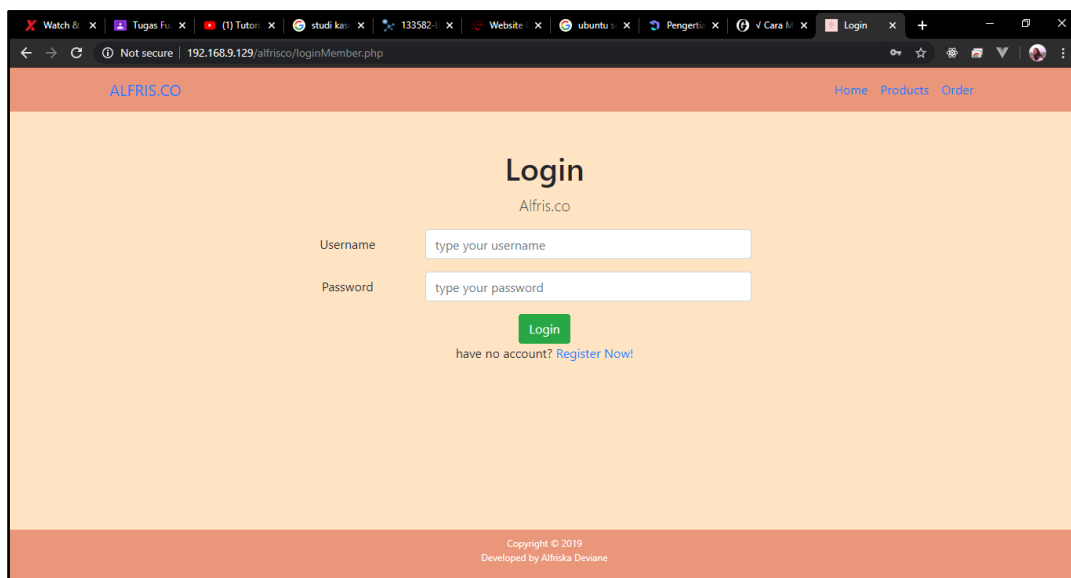


**Gambar 2.22** Tampilan awal program pada tugas kedua

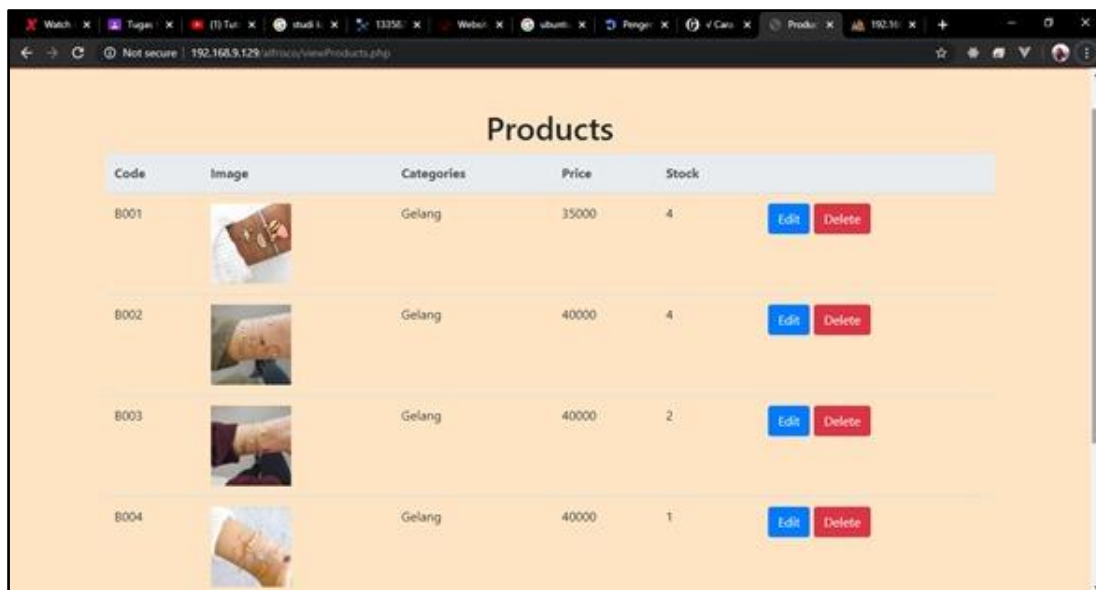
## 2.6 Pengujian Singkat

### 2.6.1 Pengujian Singkat pada Proyek Pertama

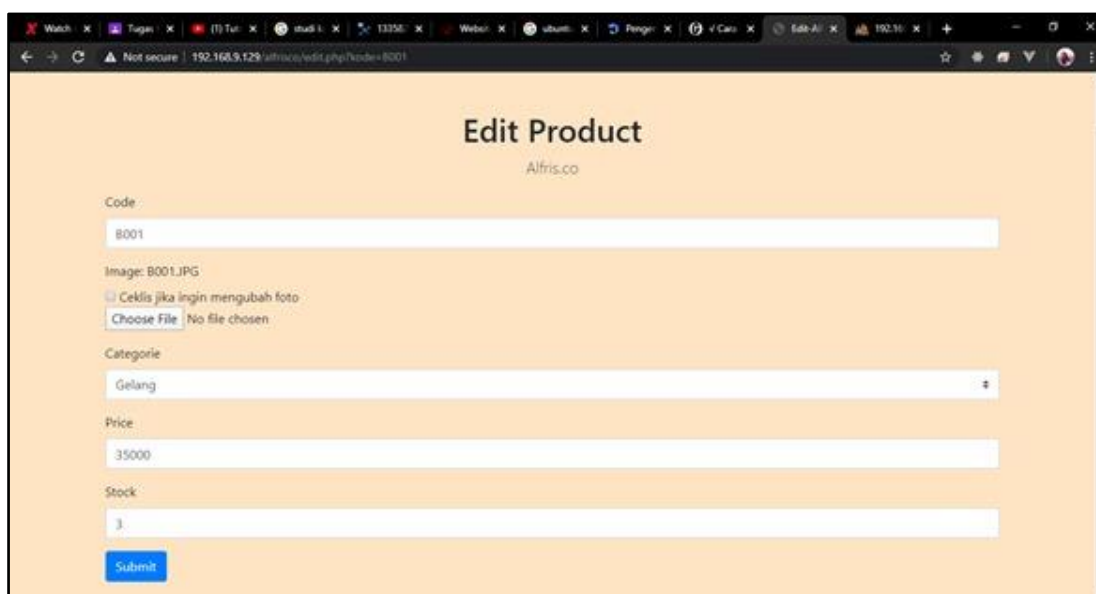
Pada tahap ini buktikan bahwa rancangan yang Anda buat telah dapat digunakan terhadap beberapa pengujian singkat. Buat sendiri dua masalah yang sesuai dengan penjelasan di latar belakang/tujuan proyek akhir, tentunya yang terkait dengan cloud computing. Bagian ini akan dibuktikan secara langsung pada saat presentasi proyek akhir. Tuliskan pada bagian ini dengan format: masalahnya atau hasil yang diharapkan, eksekusi penyelesaian masalah, hasil yang didapat. Bila hasil yang didapat tidak sesuai, maka jelaskan apa kendalanya, tidak diwajibkan hasil harus sesuai dengan harapan. Berikan juga screenshootnya. Login sebagai member, akan dapat dilihat seperti pada **Gambar 2.23**. Pada pengujian singkat proyek pertama kami juga berhasil untuk mengkonfigurasi *database*, dapat dilihat pada **Gambar 2.24** dan **Gambar 2.25**.



**Gambar 2.23** Tampilan Login untuk Member



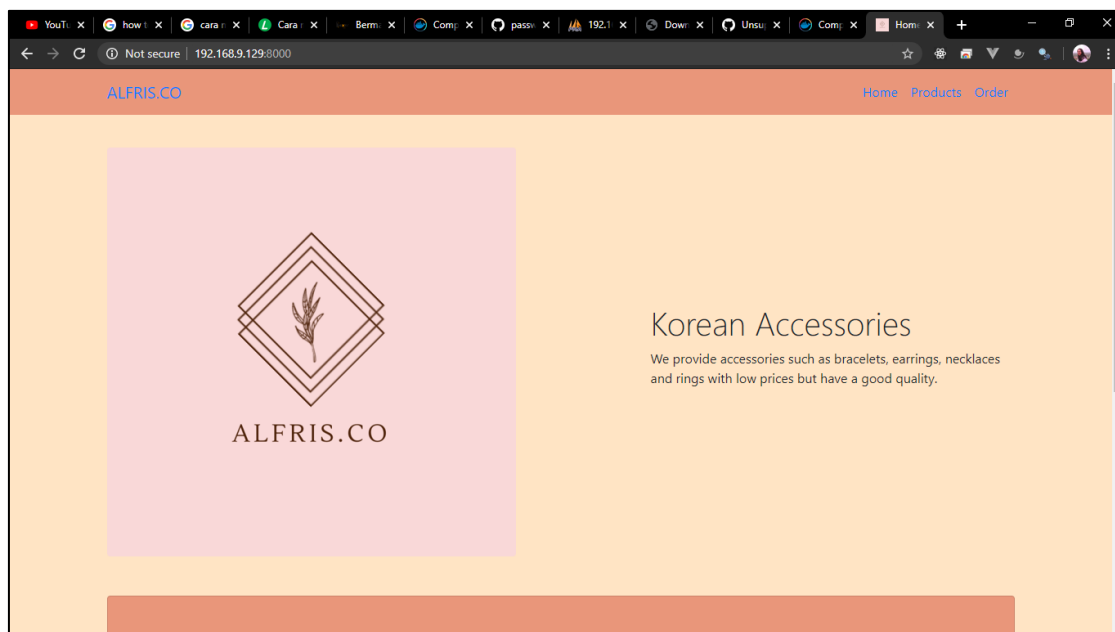
**Gambar 2.24** Tampilan *database* berhasil diimplementasikan



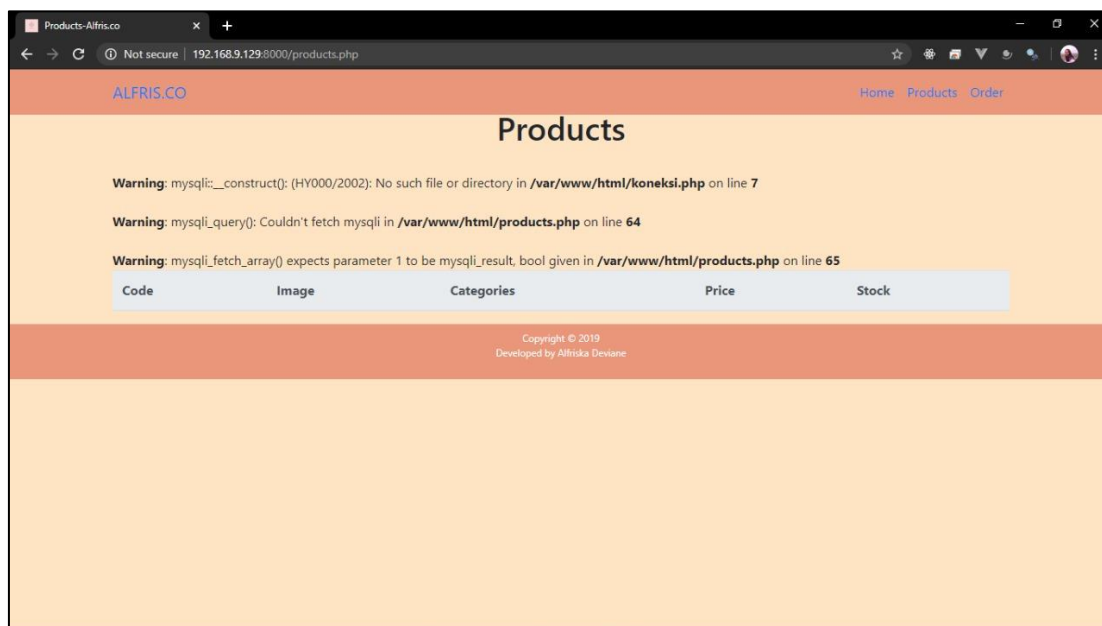
**Gambar 2.25** Tampilan *database* berhasil diimplementasikan

## 2.6.2 Pengujian Singkat pada Proyek Kedua

Pada pengujian tugas kedua, kami berhasil menampilkan program kami pada *browser* dengan menggunakan IP Address 192.168.9.129:8000. Pada aplikasi web kami sudah berjalan seperti pada **Gambar 2.26**, akan tetapi dalam implementasinya kami menemukan kendala untuk mengkonfigurasi *database* yang digunakan, sehingga masih terdapat *error* yang muncul seperti pada **Gambar 2.27**.



**Gambar 2.26** Pengujian hasil *docker*



**Gambar 2.27** Error pada implementasi *database*

### 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 Mei tahun 2020:

**Tabel 3.1** Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan									
		Maret				April				Mei	
		1	2	3	4	1	2	3	4	1	2
1.	Analisa Persoalan										
2.	Pembagian Tugas										
3.	Perancangan database										
4.	Perancangan program										
5.	Instalasi VMware & Ubuntu, dan LAMP										
6.	Konfigurasi hosting lokal										
7.	Instalasi Docker										
8.	Pembuatan Laporan BAB I										
9.	Pembuatan Laporan BAB II - Selesai										
10.	Presentasi Proyek Akhir										

##### 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.	Perancangan Arsitektur Cloud Computing	Isnan
2.	Pengujian Singkat	Isnan
3.	Latar Belakang Masalah	Isnan
4.	Agenda Pengerjaan Proyek	Devia
5.	Perancangan aplikasi dan database	Devia
6.	Dokumentasi	Devia
7.	Instalasi VMware & Ubuntu dan LAMP	Isnan & Devia
8.	Konfigurasi hosting lokal	Devia
9.	Instalasi Docker	Devia
10.	Pembuatan Laporan BAB I	Devia
11.	Pembuatan Laporan BAB II - Selesai	Isnan

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Pada proyek yang sudah kita laksanakan, maka kami mendapat beberapa kesimpulan sebagai berikut:

- a. Proyek aplikasi web sudah dapat berjalan tanpa kesalahan dengan menggunakan layanan LAMPP pada Ubuntu server yang berjalan pada VMWare Workstation. Proyek aplikasi web juga sudah berhasil dijalankan dengan arsitektur *docker* yang menggunakan system *container*, akan tetapi kami menemukan masalah yaitu adalah menghubungkan *database* dengan *container*. Jadi proyek aplikasi web yang sudah berjalan tidak bisa berhubungan dengan *database* yang ada.
- b. Proyek yang menggunakan layanan LAMPP dapat diakses dengan *device* yang sudah berada dalam jaringan lokal yang sama, dan diakses melalui *IP Address*. Sedangkan pada arsitektur *docker* yang menggunakan sistem *container* akan dapat langsung mengakses ke *hardware*. Keuntungan menggunakan *docker* adalah proyek tidak terlalu berat.
- c. Pembagian tugas yang ada pada BAB III, sudah dapat dijalankan dengan mulus. Tetapi kami menemukan beberapa kendala dimana setiap laptop yang kami gunakan memiliki *error* yang berbeda-beda juga, hal tersebut menghambat kecepatan kami dalam menyelesaikan tugas akhir.

#### **4.2 Saran**

Saran untuk proyek ini adalah seharusnya dapat membuat file *backup* untuk setiap proses implementasinya. Juga untuk dapat memperhatikan spesifikasi hardware yang dipakai agar dapat maksimal untuk menjalankan sebuah *virtual machine*.



## DAFTAR PUSTAKA

- Muslim, M. A., & Retno, N. A. (2014). Implementasi Cloud Computing Menggunakan Metode Pengembangan Sistem Agile. *Scientific Journal of Informatics*, 1(1), 29-37.
- Dasar Penggunaan Docker, <https://medium.com/@denmasyarikin/dasar-penggunaan-docker-f80d9602186c> (diakses pada 29 April 2020)
- Fauziah, Y. (2014). *Arsitektur Cloud Computing Pada Sistem Informasi Desa Sebagai Layanan Akses Informasi Desa*. 2014(semnasIF), 224–`230.
- Apridayanti, S., Saputra, R. A., Informatika, J. T., Teknik, F., Oleo, U. H., Kunci, K., & Web, A. (2018). Desain dan Implementasi Virtualisasi Berbasis Docker untuk Deployment Aplikasi Web. *Semantik*, 4(2), 37–46.
- Romadlon Bik, M. (2017). Implementasi Docker Untuk Pengelolaan Banyak Aplikasi Web (Studi Kasus : Jurusan Teknik Informatika UNESA). *Jurnal Manajemen Informatika*, 7(2), 46–50.