

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING  
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI LOST AND FOUND  
MENGUNAKAN LAYANAN UBUNTU LAMPP DAN DALAM BENTUK  
DOCKERILE**



**DISUSUN OLEH:**

**NAMA ANGGOTA : REO SAHOBBY 123170067  
ANDRE JULIO S MARBUN 123170079  
KELAS : A  
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.KOM  
MUHAMMAD IMAM AL FATAH**

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

**2020**

**HALAMAN PENGESAHAN**  
**SISTEM INFORMASI LOST AND FOUND**  
**MENGGUNAKAN LAYANAN UBUNTU LAMPP DAN DALAM BENTUK**  
**DOCKERFILE**

Disusun oleh :

Reo Sahobby

123170067

Andre Julio S Marbun

123170079

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 kepada Tuhan Yang Maha Esa yang telah memberikan berkat-Nya sehingga kami dapat menyelesaikan Laporan Proyek Akhir yang berjudul Sistem informasi lost and found menggunakan layanan ubuntu lamp dan dalam bentuk dcokerfile.

Terima kasih kami ucapkan kepada asisten dosen yang telah membantu kami baik secara moral maupun materi sehingga kami dapat melakukan proyek akhir dengan baik dan lancar. Terima kasih juga kepada teman-teman yang siap membantu kami sehingga kami dapat menyelesaikan tugas ini dengan baik.

Laporan ini mengenai pembuatan website yang memuat layanan informasi mengenai barang yang hilang dan/atau ditemukan di sekitar area stasiun, dimana laporan ini akan dijelaskan tahap-tahap pembuatan dan penggunaan dari website tersebut serta komponen-komponen yang akan digunakan.

Kami menyadari, bahwa Laporan Proyek Akhir kami buat masih jauh dari kata sempurna baik dari segi penyusunan maupun penulisannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang dapat menjadi pendorong kami agar dapat membuat laporan yang lebih baik lagi di masa mendatang.

Semoga Laporan Proyek Akhir ini dapat menjadi pedoman dan menambah wawasan para pembaca dan dapat bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan masyarakat luas, Terima kasih atas perhatiannya, Semoga Tuhan memberkati.

Yogyakarta, 30 Maret 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 .....	3
<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 .....	7
2.4 Tahap Implementasi .....	10
2.5 Hasil Implementasi .....	26
2.6 Pengujian Singkat .....	29
<b>BAB III JADWAL Pengerjaan dan Pembagian Tugas .....</b>	<b>33</b>
3.1 Agenda Pengerjaan .....	33
3.2 Keterangan Pembagian Tugas .....	33
<b>BAB IV KESIMPULAN DAN SARAN .....</b>	<b>34</b>
4.1 Kesimpulan .....	34
4.2 Saran .....	34
<b>DAFTAR PUSTAKA .....</b>	<b>35</b>
<b>LAMPIRAN .....</b>	<b>36</b>

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Istilah *cloud computing* saat ini sedang sering digunakan, dan sering terdengar di dunia perkembangan teknologi. Banyak teknologi dan sistem yang saat ini sudah menggunakan *cloud computing*, namun apa sebenarnya makna dari cloud computing itu sendiri. Menurut Laudon dan Luodon, *Cloud Computing* adalah sebuah model komputasi dimana aktifitas pemrosesan, penyimpanan, perangkat lunak dan layanan lainnya disediakan layaknya sumber virtual terpadu pada suatu jaringan yang umumnya adalah internet. Sumber daya komputasi dari *cloud computing* tersebar dan dapat diakses berdasarkan kebutuhan dari perangkat apapun dan dimanapun terhubung (Fandi, 2017).

Sumber lain juga menyebutkan, cloud computing adalah gabungan pemanfaatan teknologi komputer ('komputasi') dan pengembangan berbasis Internet ('awan'). Awan (*cloud*) adalah metefora dari internet, sebagaimana awan yang sering digambarkan di diagram jaringan computer. Jadi *cloud computing* adalah proses komputasi yang dilakukan diawan, yang bisa di akses oleh semua jaringan, karena tempat pemrosesan atau komputasi tidak hanya internal pada suatu jaringan tertentu lagi (Hutomo, 2015).

Pada saat ini penggunaan cloud computing sudah sangatlah luas, umumnya *cloud computing* digunakan dalam komputasi pada website-website supaya website tersebut dapat diakses oleh seluruh pengguna tidak hanya terbatas dalam satu jaringan yang sama dengan computer server pemroses website tersebut. Selain digunakan dalam komputasi website, *cloud computing* juga biasa digunakan pada alat-alat elektronik yang sudah menerapkan teknologi IoT atau *Internet of Thing*. Dalam peralatan elektronik yang sudah menggunakan IoT, tentunya akan banyak menyimpan data yang dibutuhkan dalam alat tersebut. Oleh karena itu, cloud computing dalam hal tersebut digunakan sebagai tempat penyimpanan data data yang digunakan oleh peralatan elektronik tersebut, supaya bisa diakses oleh semua pengguna alat tersebut. Selain kedua contoh penggunaan *cloud computing* tersebut sebenarnya masih sangat banyak contoh implementasi layanan *cloud computing* dalam kehidupan sehari-hari.

Dalam projek kali ini, kami akan menggunakan layanan *cloud computing* untuk tempat komputasi pemrosesan dari sebuah website, dan tempat penyimpanan data yang digunakan dalam website tersebut. Dalam pembuatan website tersebut dirasa perlu menggunakan layanan *cloud computing* dikarenakan website tersebut nantinya akan menjadi

website suatu layanan informasi milik sebuah tempat transportasi umum yang akan memberikan informasi mengenai barang-barang milik penumpang yang ditemukan oleh petugas di sekitar area stasuin. Dengan adanya website tersebut, diharap para pengumpang atau pengunjung yang merasa kehilangan barang mereka, mereka dapat mencari informasi mengenai barang mereka lewat website tersebut. Dengan begitu pengunjung akan lebih mudah menemukan barang milik mereka, nantinya barang yang ditemukan oleh petugas yang melakukan pengecekan di area stasiun akan menyimpan barang tersebut pada tempat *lost and found*, dan nantinya pengunjung yang merasa memiliki barang tersebut dapat memintanya kepada petugas *lost and found* disertai dengan bukti kepemilikan yang jelas.

Dalam pengembangan website tersebut, tahap tahap yang dilakukan untuk menyelesaikan projek tersebut adalah menyiapkan dan membuat website *lost and found* yang dipastikan sudah dapat dijalankan dengan normal tanpa kendala apa apa. Kemudian menyiapkan server ubuntu yang akan digunakan sebagai tempat untuk *hosting* dari website tersebut. Kemudian, setelah dipastikan website dapat berjalan dengan normal dan layanan *hosting* yang kita buat menggunakan ubuntu LAMPP dapat bekerja dengan normal. Barulah kita membuat website yang kita buat tersebut dihosting menggunakan layanan *hosting* yang telah kita buat. Sehingga nantinya website tersebut dapat diakses oleh masyarakat umum, tidak hanya menjadi website internal yang hanya bisa diakses oleh pembuat website tersebut.

## **1.2 Tujuan Proyek Akhir**

Berhubungan dengan judul proyek akhir yang sedang kita kerjakan dengan judul Sistem Informasi Lost and Found Menggunakan Layanan Uubuntu LAMPP dan Dalam Bentuk Dockerfile, maka laporan ini akan menjelaskan cara penyelesaian permasalahan tersebut dengan menggunakan ubuntu dan lamp sebagai tempat *hosting* dari website yang kita buat. Dalam implementasinya penyelesaian yang terdapat di dalam laporan ini akan menghasilkan *output* berupa website yang berisi layanan informasi tentang kehilangan barang milik penumpang atau pengunjung yang ditemukan oleh petugas stasiun. Yang nantinya, lewat website tersebut akan diberikan informasi mengenai barang barang yang telah ditemukan, dan bagi yang merasa memiliki dari barang tersebut dapat mendatangi stasuin tersebut untuk mengambil barang mereka disertai dengan bukti kepemilikan yang jelas.

## **1.3 Manfaat Proyek Akhir**

Dengan dibangunnya website tersebut, maka pengguna dapat mengetahui dan mencari tau barang yang hilang maupun ditemukan di area stasiun sehingga akan

mempersingkat waktu dan membantu pengguna agar dapat mencari tau ketika ada barang yang ketinggalan di stasiun tanpa perlu berada di stasiun tersebut. Misalkan seorang pengguna kereta api berangkat dari stasiun Jogjakarta dan sampai di stasiun Surabaya ternyata ada barang yang ketinggalan di stasiun Jogja atau ketinggalan di kereta api, akan mudah lebih diketahui barang yang ditemukan secara online tanpa perlu harus balik lagi ke stasiun Jogja.

Kemudian, manfaat dengan adanya *cloud computing* ini dalam menjalankan website lost and found itu adalah website tersebut dapat diakses oleh semua pihak, tidak terbatas pada Batasan akses yang diberikan apabila website tersebut tidak dihosting dan hanya bisa dilakukan pengaksesan oleh *developer* yang membuat website tersebut. Karena memang itulah kelebihan dari layanan hosting yang menggunakan teknologi *cloud computing* dapat menjalankan komputasi dari sebuah website tersebut dimanapun dan kapanpun tidak terbatas oleh jaringan tertentu yang harus sama dengan jaringan yang digunakan *developer* dalam membuat dan mengembangkan website tersebut.

#### **1.4 Tahap Penyelesaian Proyek Akhir**

Tahapan tahapan yang akan dilakukan untuk mengerjakan proyek ini, adalah sebagai berikut :

1. Melakukan analisis kebutuhan sistem yang digunakan dalam web sistem informasi lost and found yang akan ditransformasikan untuk digunakan dalam arsitektur cloud computing dengan basis Saas/Iaas/Paas/DBaaS.
2. Melakukan integrasi *database* yang digunakan saat menggunakan xampp dengan database yang digunakan pada saat menggunakan ubuntu server.
3. Menentukan konfigurasi yang tepat untuk melakukan pengaturan yang dilakukan pada saat hosting file berlangsung. Menyesuaikan kebutuhan yang akan digunakan berupa :
  - a. Alamat IP
  - b. Menggunakan ubuntu server 18.04 LTS
4. Menguji sistem informasi lost and found yang telah dihosting menggunakan ubuntu server, untuk mengetahui kendala dan perbedaan saat menggunakan local hosting lainnya.

## BAB II

### ISI DAN PEMBAHASAN

#### 2.1 Komponen yang Digunakan

Dalam menyelesaikan proyek praktikum teknologi cloud computing kali ini, dibutuhkan beberapa komponen untuk dapat menunjang proyek tersebut terselesaikan. Beberapa komponen tersebut diantaranya adalah laptop dengan spesifikasi prosesor intel core i7, kemudian untuk software yang digunakan adalah software VMWare dan ubuntu OS, LAMP, apache, phpMyAdmin.

##### 2.1.1 Komponen pada Proyek Pertama

Berikut adalah tabel komponen yang digunakan untuk mengerjakan proyek ini.

**Tabel 2.0** Tabel komponen untuk proyek pertama

No	Komponen	Spesifikasi
1.	Laptop	Processor menggunakan intel core-i7
2.	Ram	8Gb
3.	Hardisk	1Tb
4.	Sistem Operasi	Windows 10 Home
5.	VmWare	VmWare Workstation Pro 15.5

Penjelasan mengenai komponen yang digunakan adalah sebagai berikut. Laptop yang digunakan sebagai *host* dalam server tersebut yang digunakan adalah laptop dengan merek Asus, yang memiliki spesifikasi antara lain adalah sebagai berikut. Ditenagai dengan *processor* dari intel yaitu intel core-i7 7700HQ dengan nilai *clock speed* mencapai 2,8Ghz untuk masing-masing corenya. Kemudian RAM yang digunakan pada laptop ini sebesar 8Gb dengan jenis DDR4. Kemudian sistem operasi yang digunakan dalam membuat proyek ini adalah menggunakan windows tepatnya menggunakan windows 10 home 64-bit. Dan perangkat lunak virtualisasi yang digunakan untuk membuat virtualisasi sistem operasi ubuntu adalah menggunakan *software* berupa VmWare Workstation Pro 15.5.

##### 2.1.2 Komponen pada Proyek Kedua

Komponen-komponen yang digunakan pada proyek kedua, yaitu proyek untuk membuat halaman web yang sudah dibuat pada tugas pertama. Untuk membuat tugas docker tersebut, beberapa komponen yang kami gunakan antara lain adalah VmWare sebagai aplikasi untuk virtualisasi, yang juga digunakan pada tugas pertama, kemudian ubuntu 18.04



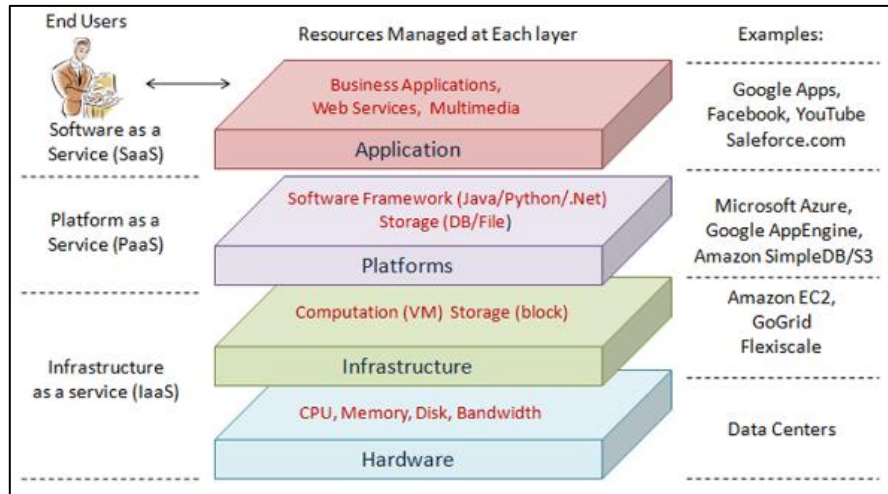
sebagai sistem operasi yang akan divirtualisasikan untuk menjalankan fungsi-fungsi dalam proyek satu dan proyek dua. Kemudian docker yang diinstall pada sistem operasi ubuntu atau linux yang ada pada aplikasi VmWare tersebut. Berikut ini adalah daftar komponen yang kami gunakan untuk mengerjakan tugas proyek kedua.

**Tabel 2.1** Tabel komponen proyek kedua

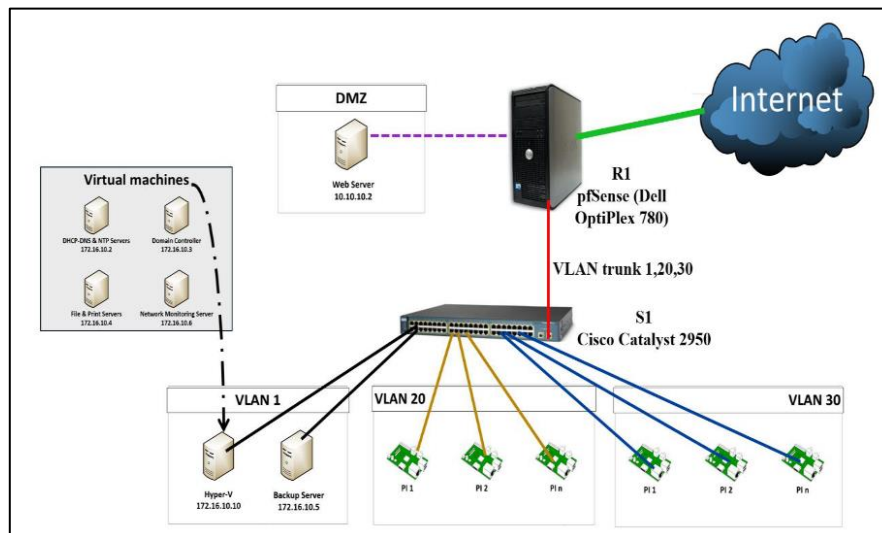
No.	Nama Parameter	Nilai	Keterangan
1.	Merk Server	Virtual Machine dengan VmWare Workstation	Aplikasi untuk melakukan virtualisasi OS
2.	Versi Ubuntu	Ubuntu 18.04 LTS	ISO Ubuntu yang digunakan sebagai <i>guest</i> OS
3.	Versi Docker	Docker Linux	Docker yang digunakan untuk proyek kedua

## 2.2 Rancangan Arsitektur *Cloud Computing*

Dalam mengerjakan dan mengerjakan proyek akhir ini, rancangan arsitektur yang kami buat adalah rancangan arsitektur berbasis Iaas. Hardware yang digunakan untuk menyelesaikan proyek akhir ini adalah laptop ROG yang digunakan sebagai perangkat dasarnya, kemudian di atasnya hypervisor Paas atau sistem operasi yang digunakan, yaitu saya menggunakan Windows 10 Home dan aplikasi VmWare yang kami gunakan sebagai aplikasi untuk media virtualisasi. Layer virtual OS yang kami jalankan menggunakan ubuntu server versi 18.04 LTS. Dengan menggunakan virtual OS tersebut, kami dapat menjalankan beberapa aplikasi atau fitur-fitur yang dapat digunakan untuk menjalankan hosting file seperti Apache, PHP, MySQL, dan PhpMyAdmin. Pada layer application adalah layer yang berisi web yang sudah di hosting, yaitu layer yang berhubungan langsung dengan pengguna yang dapat secara langsung berinteraksi dengan pengguna. Ilustrasi mengenai rancangan arsitektur yang kami gunakan dapat dilihat melalui gambar dibawah ini.

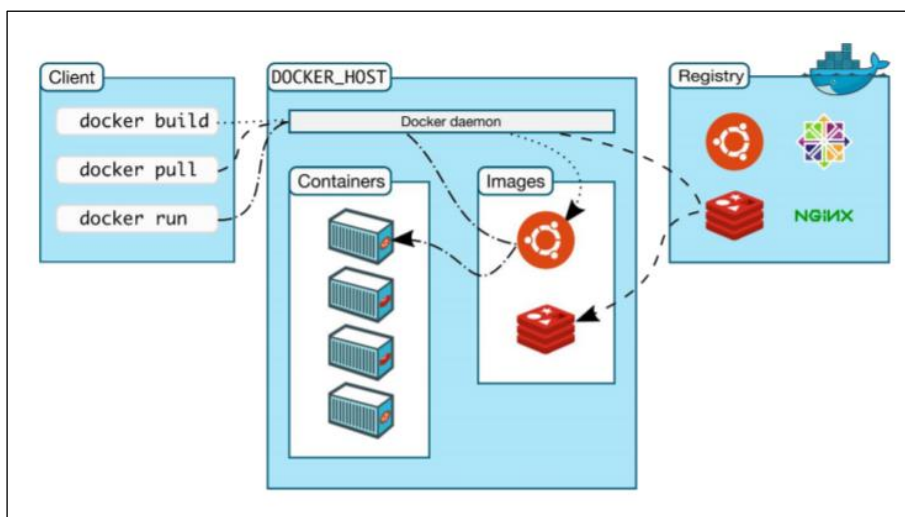


**Gambar 2.1** Penjelasan layer arsitektur terhadap komponen penyusunnya



**Gambar 2.2** Penjelasan Konfigurasi Jaringan

Seperti dijelaskan dalam jurnal (Adiputra, 2015), dalam jurnalnya menjelaskan bahwa pengertian docker adalah sebuah platform terbuka untuk menempatkan berbagai layanan ke dalam sebuah lingkungan bernama *container*, sehingga terbentuk sebuah paket-paket dengan semua layanan dan software yang diperlukan. Di dalam docker terdapat arsitektur yang seperti dijelaskan dalam (Indonesia, 2017), dapat dilihat pada **gambar 2.3** dibawah ini.



**Gambar 2.3** arsitektur docker

Dalam jurnal (ROMADLON BIK, 2017), dijelaskan bahwa dalam arsitektur docker terdapat beberapa komponen diantaranya adalah *images*, *registry*, *container*, dan lain-lain. *Images* adalah sebuah template yang digunakan untuk menjalankan sebuah *container*, *images* dapat berisi beberapa aplikasi yang sudah terinstall. *Registry* adalah sebuah repository dalam docker yang digunakan untuk menyediakan *docker images* yang ingin digunakan. Sedangkan *container* dijelaskan dalam jurnal tersebut berupa directory dalam docker yang digunakan untuk menyimpan segala sesuatu yang diperlukan untuk menjalankan sebuah aplikasi dalam bentuk docker, setiap container dibuat dari sebuah *images*, dan container merupakan platform yang dapat menyediakan wadah yang terisolasi untuk aplikasi dan merupakan *images* yang bersifat *read-write* yang berjalan diatas file docker yang menggunakan *union file system*.

### 2.3 Parameter dan Konfigurasi

Untuk dapat digunakan, maka VMware Workstation perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada **Tabel 2.2** berikut ini :

**Tabel 2.2** Konfigurasi pada VMware Workstation yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	VMware ESXi version	15.5.2	Keterangan versi yang digunakan.
2.	VMware ESXi build	15785264	Keterangan <i>build (patch) number</i> .
3.	IPv4 hypervisor	IP: 192.168.202.128 (Static)	IP number untuk akses manajemen <i>hypervisor</i> melalui <i>web based</i> . Pengaturan IP address menggunakan mode statik.
		SM: 255.255.255.0	Kelas IP/subnet mask yang digunakan.
		DNS: 192.168.202.1	Alamat IP untuk fungsionalitas DNS.
		GW: 192.168.202.255	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.

4.	<i>Processor info</i>	8 x Intel(R) Core(TM) i7-7700 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> .
6.	<i>Harddisk info</i>	20 Gb	Kapasitas HDD pada <i>hypervisor</i>
7.	<i>Network info</i>	NAT	Modul jaringan pada <i>hypervisor</i>
8.	<i>Username info</i>	Reo	Username pada <i>hypervisor</i>
9.	<i>Password info</i>	Reo	Password pada <i>hypervisor</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

#### **Modul 2.1** Parameter instalasi Apache

Parameter yang digunakan untuk proses instalasi MySQL dapat dilihat dalam penjelasan **Modul 2.2** seperti berikut ini :

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

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
- mysql\_secure\_installation: Mengatur keamanan, seperti username dan password pada MYSQL

#### **Modul 2.2** Parameter instalasi MySQL

Parameter yang digunakan untuk proses instalasi PHP dapat dilihat dalam penjelasan **Modul 2.3** seperti berikut ini :

```
$ sudo apt install php libapache2-mod-php php-mysql
$ sudo nano /var/www/html/info.php
```

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
- nano : aplikasi text editor

```
- nano /var/www/html/info.php : membuat file baru bernama info.php di
direktori var/www/html/
```

### Modul 2.3 Parameter instalasi PHP

Parameter yang digunakan untuk proses instalasi PhpMyAdmin dapat dilihat dalam

**Modul 2.4** seperti 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
mysql> UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('123') WHERE User = 'arif;
mysql> FLUSH
PRIVILEGES;

Keterangan:
- mysql -u root : Untuk masuk ke MySQL sebagai user root.
- UPDATE ... : Untuk mengatur password user 'root' menjadi '123
- FLUSH PRIVILEGES : Untuk melakukan restart akun phpmyadmin

$ sudo chown arif /var/www/html

Keterangan:
- chown arif /var/www/ : Untuk memberikan akses ke user arif terhadap
direktori var/www/html
```

### Modul 2.4 Parameter instalasi PhpMyAdmin

Parameter yang digunakan untuk proses instalasi docker dapat dilihat pada **Modul**

**2.5** seperti dibawah ini :

```
$ sudo apt-get update $ sudo apt-get install
\ apt-transport-https
\ ca-certificates
\ curl
\ gnupg-agent
\ software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo aptkey
add - $ sudo apt-key fingerprint 0EBFCD88
$ sudo add-apt-repository
\ "deb [arch=amd64] https://download.docker.com/linux/ubuntu
\ $(lsb_release -cs)
\ stable"

Keterangan:
```

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : parameter tambahan pada apt untuk mengeksekusi perintah mengupdate package yang ada pada OS
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- curl : parameter yang digunakan untuk mengecek konektivitas URL dan sebagai tool transfer data dari URL tersebut
- fsSL : parameter untuk akses SSL
- Key add: parameter yang digunakan untuk menambahkan key
- fingerprint : jenis key yang digunakan
- 0EBFCD88 : nilai key untuk key fingerprint
- Add-apt-repository : parameter yang digunakan untuk menambahkan Reposirot

### Modul 2.5 Parameter instalasi docker repository

Parameter yang digunakan untuk proses instalasi docker setelah proses repository selesai dijelaskan dalam **Modul 2.6** seperti dibawah ini :

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt-get : merupakan package manager pada Ubuntu
- update : parameter tambahan pada apt untuk mengeksekusi perintah mengupdate package yang ada pada OS
- install : untuk mengeksekusi perintah instalasi paket aplikasi
- docker-ce : nama aplikasi untuk docker engine
- docker-ce-cli : nama aplikasi cli pada docker engine
- containerd.io : nama aplikasi untuk container docker
- docker-compose : nama aplikasi untuk menjalankan container docker

### Modul 2.6 Parameter instalasi docker

## 2.4 Tahap Implementasi

Pada tahap ini akan dijelaskan mengenai implementasi yang dilakukan terhadap penyelesaian soal yang diberikan. Untuk menyelesaikan projek ini ada beberapa Langkah yang harus dilakukan seperti instalasi, setting apache, dan lain lain. Untuk lebih lengkapnya akan dijelaskan Langkah-langkah yang dilakukan untuk mengerjakan projek ini:

Tugas 1 :

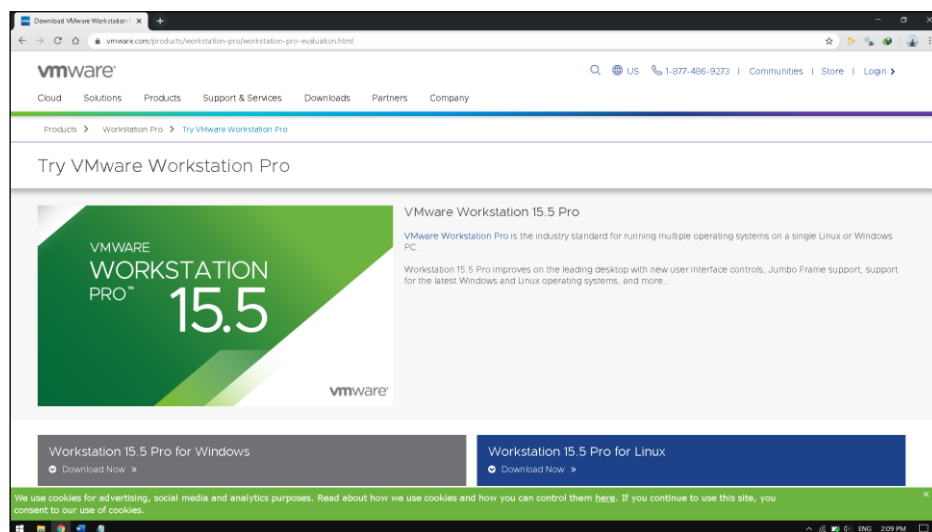
### 1. Instalasi Vm ware

Untuk dapat mengerjakan projek ini, tentunya kita harus mempunyai *software* yang digunakan untuk melakukan virtualisasi sistem operasi, dimana sistem operasi yang akan divirtualisasikan disini adalah sistem operasi ubuntu.

Untuk melakukan virtualisasi tersebut, software yang kamu gunakan adalah aplikasi bernama WmWare Workstation Pro. Berikut adalah Langkah-langkah dalam instalasi aplikasi VmWare.

1. Download file setup Aplikasi VmWare.

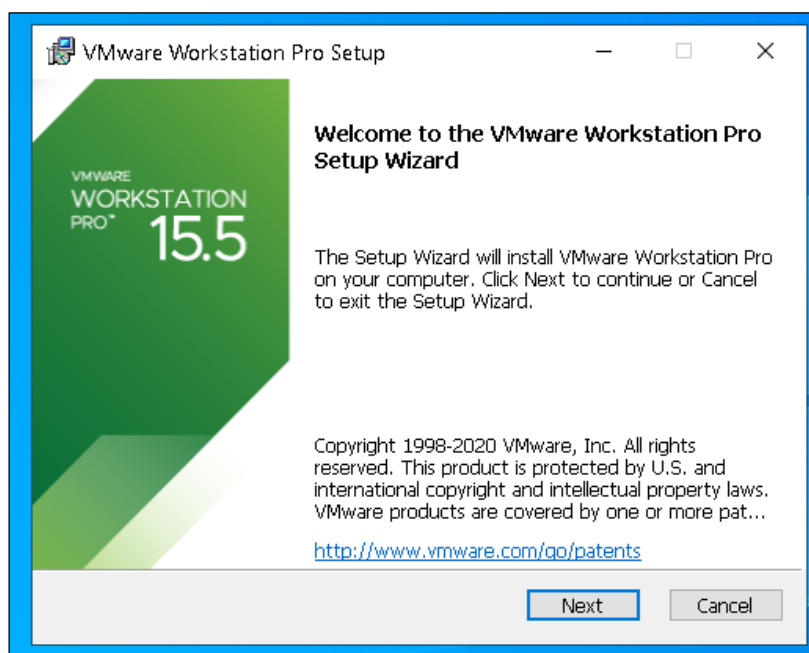
File setup aplikasi VmWare dapat di download pada link berikut, <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>. Dapat dilihat pada **gambar 2.4** berikut.



**Gambar 2.4** Tampilan web download software VmWare

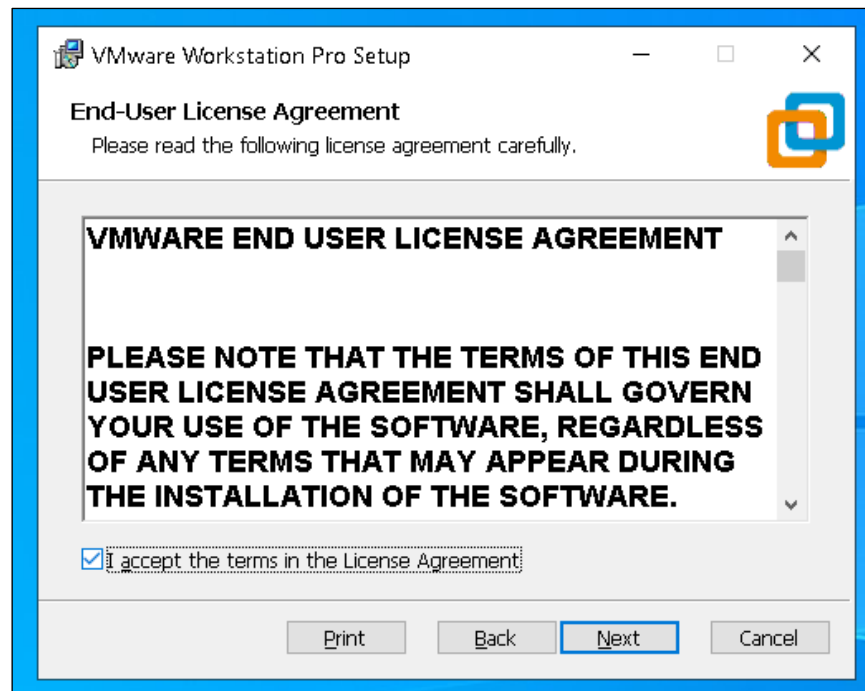
2. Install Aplikasi tersebut

Buka file setup tadi, kemudian klik next, seperti pada **gambar 2.5** berikut.



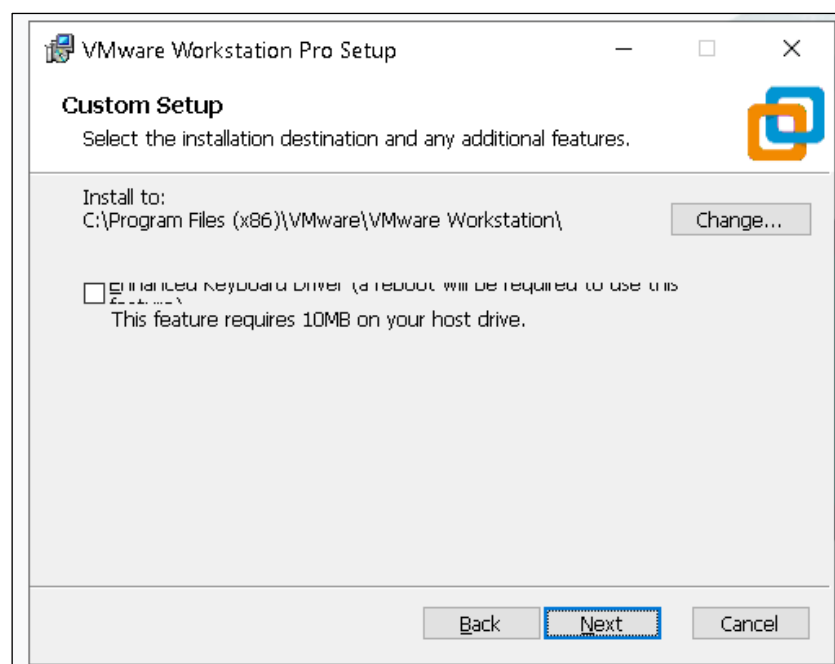
**Gambar 2.5** Tampilan instalasi VmWare

3. Klik next pada *license agreement*. Seperti pada **gambar 2.6** dibawah ini.



**Gambar 2.6** Tampilan license agreement

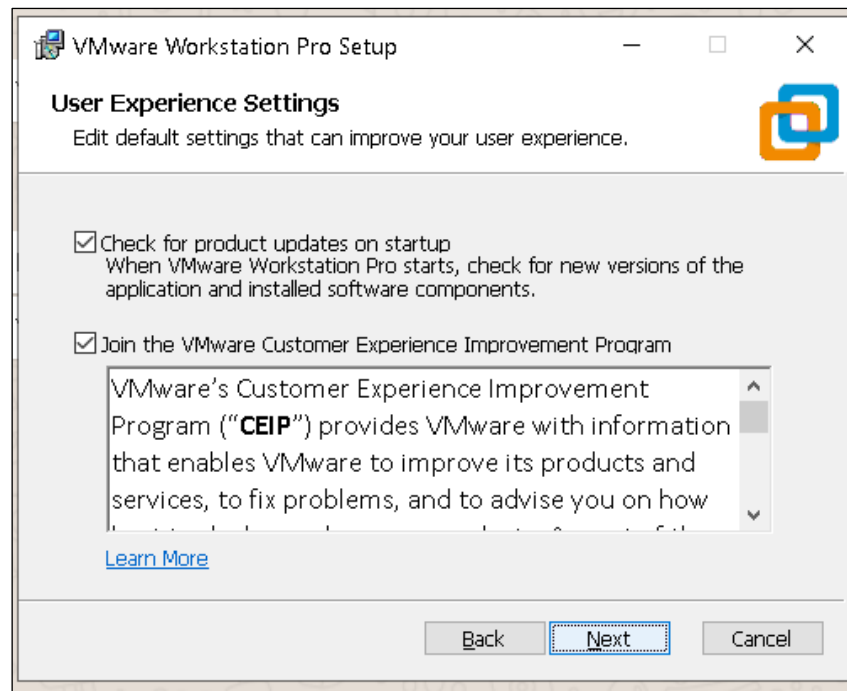
4. Klik next pada custom setup. Sesuai dengan **gambar 2.7** dibawah ini.



**Gambar 2.7** Tampilan custom setup VmWare

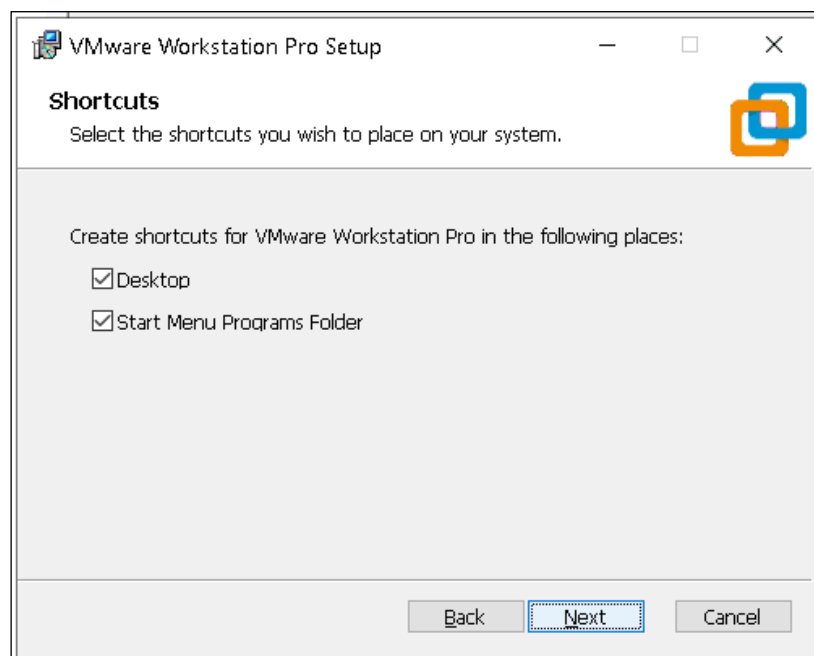


5. Klik next pada user experience setting. Lihat pada **gambar 2.8** dibawah ini.



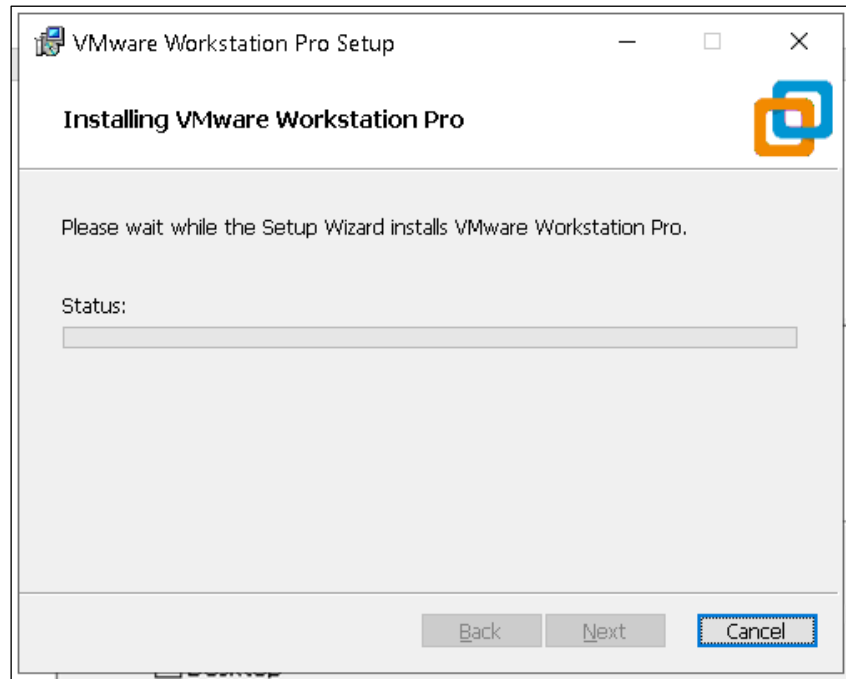
**Gambar 2.8** Tampilan user experience setting

6. Centang semua pilihan *shortcuts* yang diberikan, kemudian klik next, seperti pada **gambar 2.9** dibawah ini.



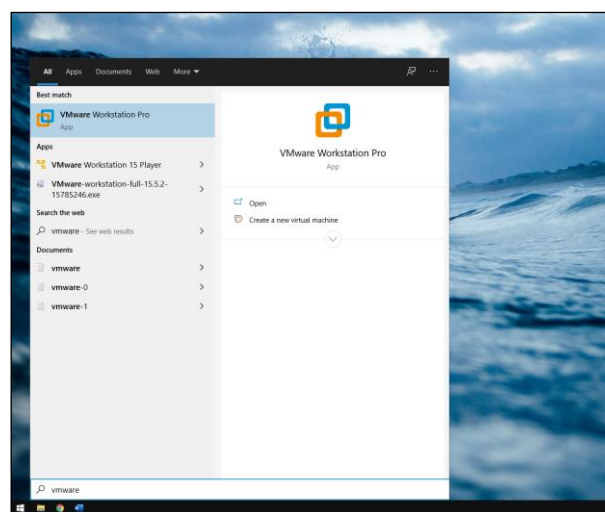
**Gambar 2.9** Tampilan shortcuts pad VmWare

7. Tunggu proses instalasi VmWare sampai selesai. Lihat pada **gambar 3.0** dibawah ini.



**Gambar 3.0** Tampilan proses instalasi VmWare

8. Jika sudah selesai, maka VmWare dapat di cari di desktop atau pada pencarian windows. Seperti pada **gambar 3.1** dibawah ini.

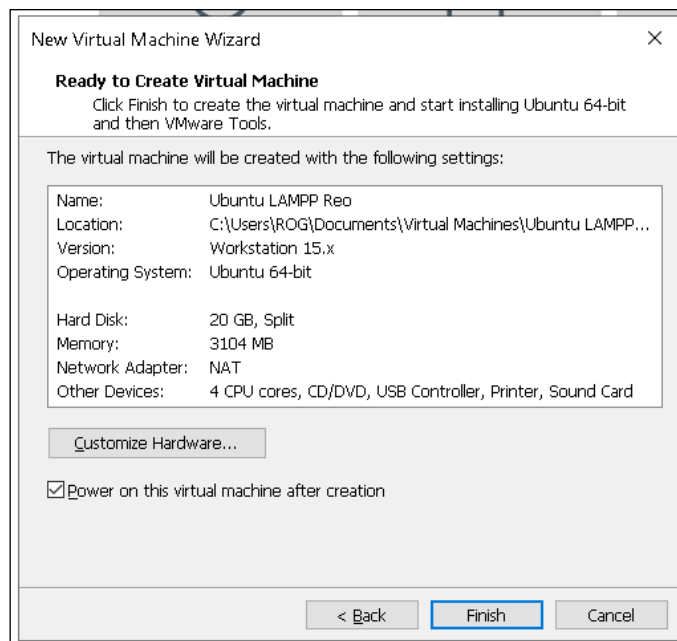


**Gambar 3.1** Tampilan VmWare pada pencarian aplikasi

## 2. Instalasi Ubuntu Server

Selanjutnya, yang akan kita install adalah sistem operasi yang akan kita gunakan untuk mengerjakan proyek. Sistem operasi yang digunakan adalah menggunakan ubuntu dengan versi 18.04 file untuk mendownload dapat dicari dengan menggunakan link berikut. <https://ubuntu.com/download/server>. Setelah mendownload file ISO ubuntu yang akan diinstall, selanjutnya adalah melakukan

instalasi server tersebut dengan menggunakan spesifikasi seperti **gambar 3.2** dibawah ini.

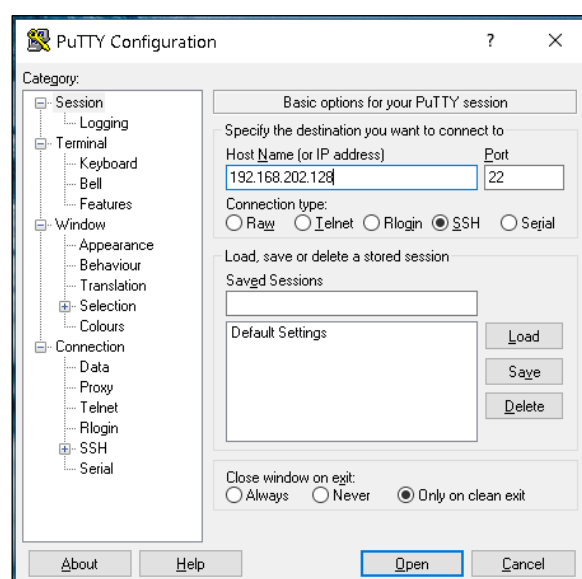


**Gambar 3.2** Spesifikasi ubutnu yang digunakan

### 3. Instalasi Apache

Untuk dapat melakukan *hosting* file secara lokal dengan menggunakan ubuntu tersebut, tentunya harus menggunakan apache supaya dapat membuat server yang akan digunakan untuk hosting file.

- a. Buka aplikasi PuTTY, kemudian masukkan alamat IP yang kita miliki pada ubuntu tersebut. lihat pada **gambar 3.3** dibawah ini.

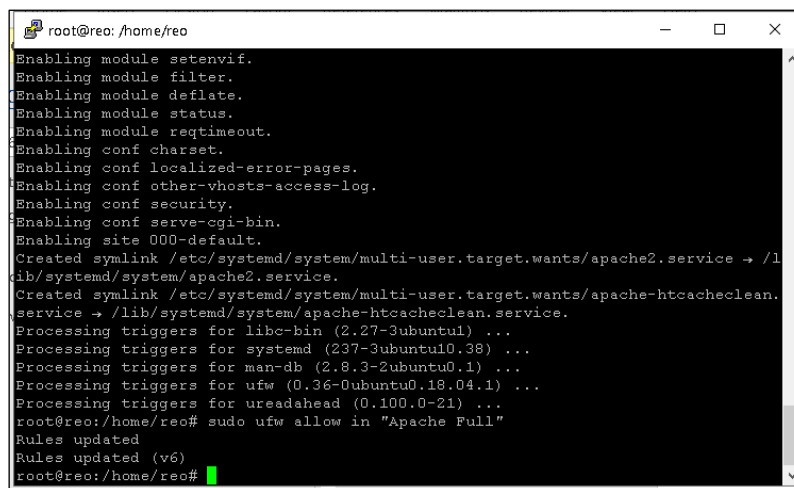


**Gambar 3.3** Tampilan aplikasi PuTTY

- b. Tuliskan sintaks seperti berikut ini, untuk melakukan instalasi. Sesai dengan **listing program 1.0** dan **gambar 3.4** dibawah ini.

```
Sudo apt update
Sudp apt install apache2
Sudo ufw allow in "Apache Full"
```

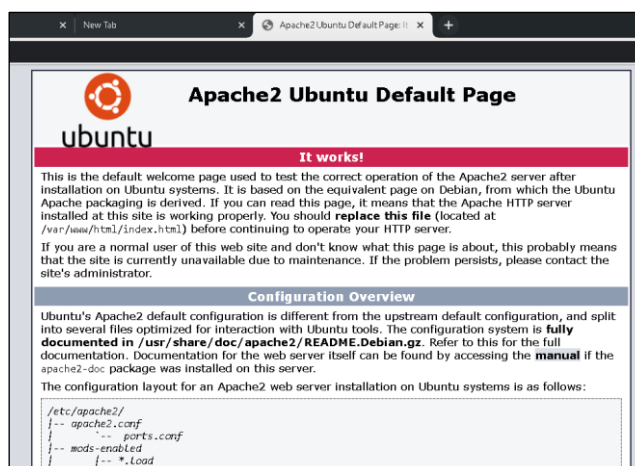
**Listing program 1.0** listing untuk install apache



```
root@reo: /home/reo
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.38) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@reo:/home/reo# sudo ufw allow in "Apache Full"
Rules updated
Rules updated (v6)
root@reo:/home/reo#
```

**Gambar 3.4** Tampilan install apache dan setting firewall

- c. Jika sudah selesai proses instalasi, maka buka alamat ip pada browser, maka akan muncul tampilan seperti ini. Lihat pada **gambar 3.5** dibawah ini.



**Gambar 3.5** Halaman awal apache pada browser

#### 4. Instalasi MySql dan Php Info

Langkan selanjutnya setelah kita menginstall apache adalah mencoba membuat file php sederhana, untuk mengecek apache kita sudah dapat menjalankan file php. Langkah-langkah yang dilakukan adalah sebagai berikut.

- a. Untuk instalasi MySql, tuliskan perintah dibawah ini. Sesuai dengan **listing program 1.1** dibawah ini.



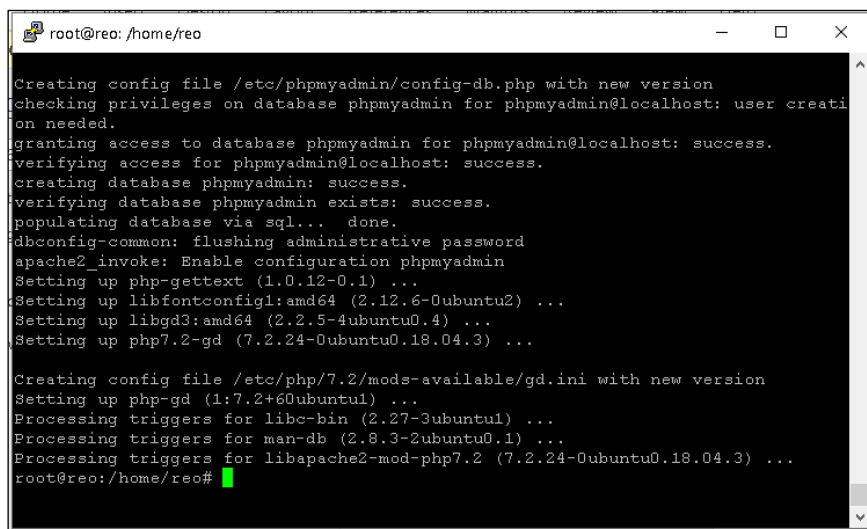
## 5. Instalasi PhpMyAdmin

Untuk dapat membuat dan menggunakan database, nantinya kita perlu menggunakan PhpMyAdmin. Berikut adalah Langkah-langkah untuk melakukan instalasi PhpMyAdmin.

- a. Tuliskan perintah berikut untuk melakukan instalasi. Sesuai dengan **listing program 1.3** dan pada **gambar 3.8** seperti dibawah ini.

```
Sudo apt install phpMyAdmin php-mbstring php-gettext
```

**Listing Program 1.3** Listing program untuk install phpmyadmin



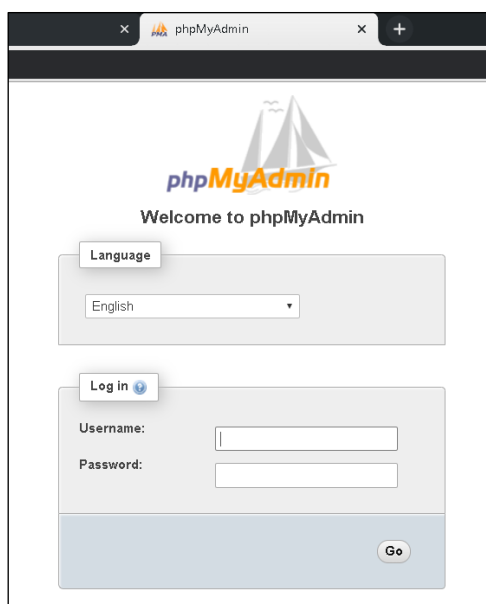
```
root@reo: /home/reo

Creating config file /etc/phpmyadmin/config-db.php with new version
Checking privileges on database phpmyadmin for phpmyadmin@localhost: user creation needed.
granting access to database phpmyadmin for phpmyadmin@localhost: success.
verifying access for phpmyadmin@localhost: success.
creating database phpmyadmin: success.
verifying database phpmyadmin exists: success.
populating database via sql... done.
dbconfig-common: flushing administrative password
apache2_invoke: Enable configuration phpmyadmin
Setting up php-gettext (1.0.12-0.1) ...
Setting up libfontconfig1:amd64 (2.12.6-0ubuntu2) ...
Setting up libgd3:amd64 (2.2.5-4ubuntu0.4) ...
Setting up php7.2-gd (7.2.24-0ubuntu0.18.04.3) ...

Creating config file /etc/php/7.2/mods-available/gd.ini with new version
Setting up php-gd (1:7.2+60ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libapache2-mod-php7.2 (7.2.24-0ubuntu0.18.04.3) ...
root@reo:/home/reo#
```

**Gambar 3.8** Tampilan instalasi PhpMyAdmin

- b. Setelah itu, buka browser dan tuliskan alamat ip/phpMyAdmin. Lihat **gambar 3.9** dibawah ini.



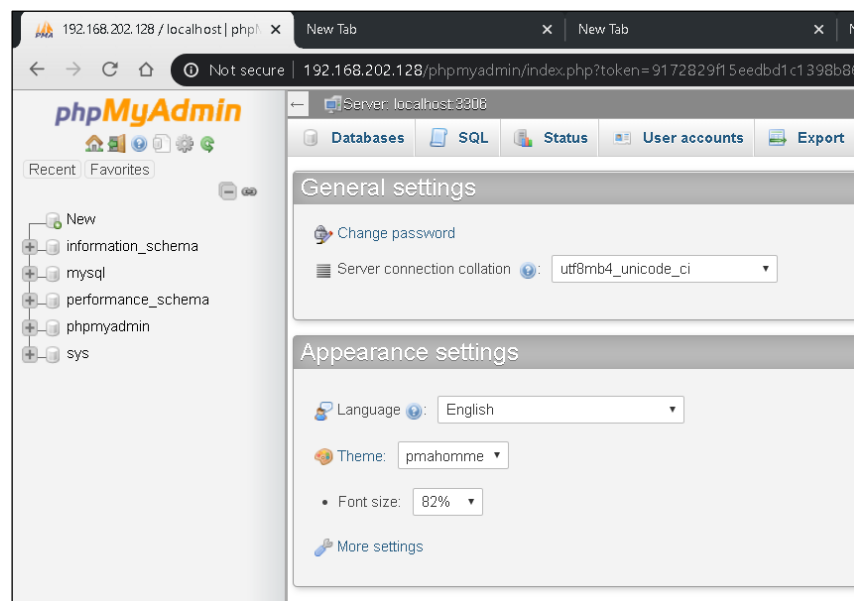
**Gambar 3.9** Tampilan login PhpMyAdmin

- c. Kemudian tuliskan perintah tersebut untuk mengganti password root dan dapat login ke dalam PhpMyAdmin tersebut. sesuai dengan **listing program 1.4** dibawah ini.

```
Sudo mysql -u root
UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('KATA_SANDI_ROOT_USER')
WHERE User = 'root';
```

**Listing program 1.4** Listing program untuk setting password

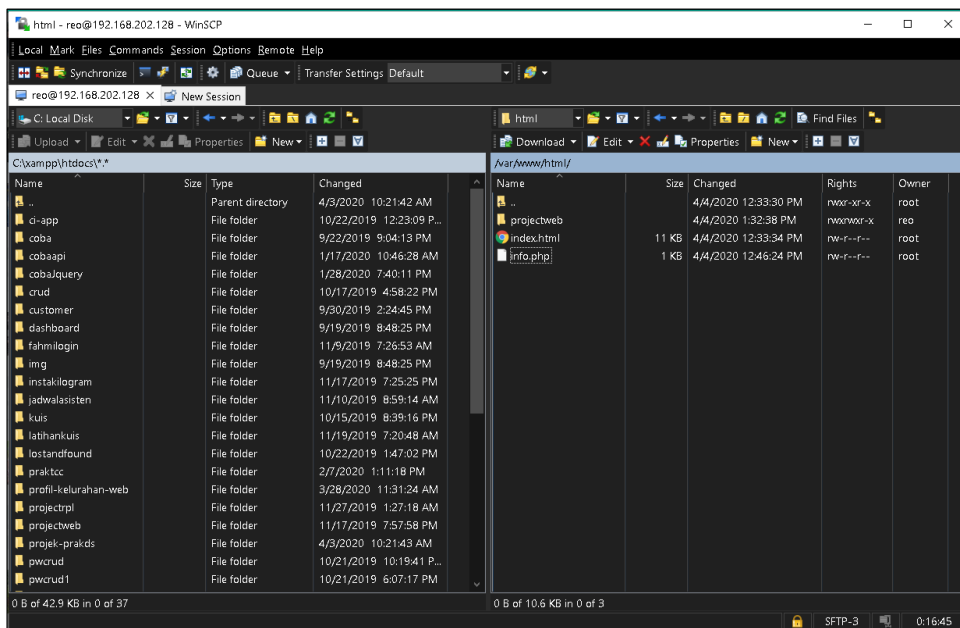
- d. Setelah selesai, masuk lagi dan dapat login sesuai dengan password yang dibuat. Seperti pada **gambar 4.0** dibawah ini.



**Gambar 4.0** Halaman dashbor PhpMyAdmin

6. Upload berkas menggunakan WinScp.

Setelah semua proses diatas selesai, Langkah selanjutnya adalah melakukan upload berkas menggunakan aplikasi bernama winscp. Dalam aplikasi tersebut terdapat dua buah bilah kanan dan kiri, arahkan bagian kanan pada bagian /var/www/html/ dan pada bilah kiri arahkan pada file yang telah kita buat. Lihat pada **gambar 4.1** dibawah ini.



**Gambar 4.1** Tampilan pindah berkas dengan winscp

## Tugas 2:

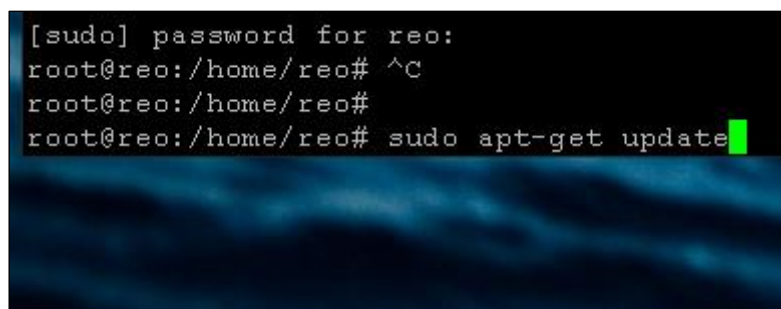
### 1. Instalasi Docker

Untuk dapat menjalankan docker yang memuat tugas yang sudah dihosting tersebut, dalam menggunakan docker kita perlu melakukan instalasi docker pada linux/server yang kita gunakan. Cara instalasi adalah sebagai berikut :

1. Melakukan update ubuntu terlebih dahulu. Seperti **listing program 1.5** dan **gambar 4.2** dibawah ini.

```
sudo apt-get update
```

**Listing Program 1.5** Listing program update server



**Gambar 4.2** Update server

2. Instalasi Docker, dengan menuliskan sintask seperti dibawah ini. Seperti **listing program 1.6** dan **gambar 4.3** dibawah ini.

```
sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
```



```
software-properties-common
```

#### Listing Program 1.6 Listing program install docker

```

Reading package lists... Done
root@reo:/home/reo# sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common

```

**Gambar 4.3** sintaks instalasi docker

3. Add Docker Official Gpg Key, dengan menuliskan sintaks dibawah ini. Seperti **listing program 1.7** dan **gambar 4.4** dibawah ini.

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

#### Listing Program 1.7 listing program gpg key

```

Preparing to unpack .../gnupg-agent_2.2.4-1ubuntu1.2_amd64.deb ...
Unpacking gnupg-agent (2.2.4-1ubuntu1.2) ...
Setting up apt-transport-https (1.6.12) ...
Setting up gnupg-agent (2.2.4-1ubuntu1.2) ...
root@reo:/home/reo# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

```

**Gambar 4.4** sintaks untuk add GPG key docker

4. Add Apt Key Fingerprint, dengan menuliskan sintaks dibawah ini. Sesuai dengan **listing program 1.8** dan **gambar 4.5** dibawah ini.

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

#### Listing Program 1.8 Listing program key fingerprint

```

root@reo:/home/reo# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
root@reo:/home/reo# sudo apt-key fingerprint 0EBFCD88
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]

root@reo:/home/reo#

```

**Gambar 4.5** sintaks add apt key fingerprint

5. Install Docker Repository. Sesuai **listing program 1.9** dan **gambar 4.6** dibawah ini.

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

**Listing program 1.9** Listing program docker repository

```
root@reo:/home/reo# sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:2 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.0 kB]
Hit:3 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:6 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease
Fetched 75.5 kB in 2s (34.5 kB/s)
Reading package lists... Done
root@reo:/home/reo#
```

**Gambar 4.6** sintaks install docker repository

6. Melakukan Update Repository, dengan menuliskan sintaks dibawah ini. Sesuai **listing program 2.0** dan **gambar 4.7** dibawah ini.

```
sudo apt-get update
```

**Listing program 2.0** Listing program update server ubuntu

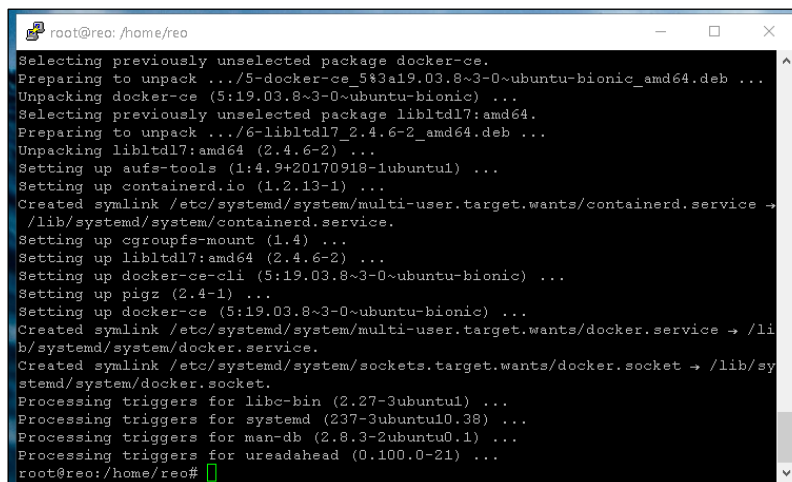
```
Reading package lists... Done
root@reo:/home/reo# sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 252 kB in 6s (38.8 kB/s)
Reading package lists... Done
root@reo:/home/reo#
```

**Gambar 4.7** sintaks update repository docker

7. Install Latest Version Docker. Sesuai **listing program 2.1** dan **gambar 4.8** dibawah ini.

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

**Listing Program 2.1** Listing program install docker



```

root@reo:/home/reo
Selecting previously unselected package docker-ce.
Preparing to unpack .../5-docker-ce_5:19.03.8~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5:19.03.8~3-0~ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../6-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.2.13-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up cgroupfs-mount (1.4) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Setting up docker-ce-cli (5:19.03.8~3-0~ubuntu-bionic) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (5:19.03.8~3-0~ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.38) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@reo:/home/reo#

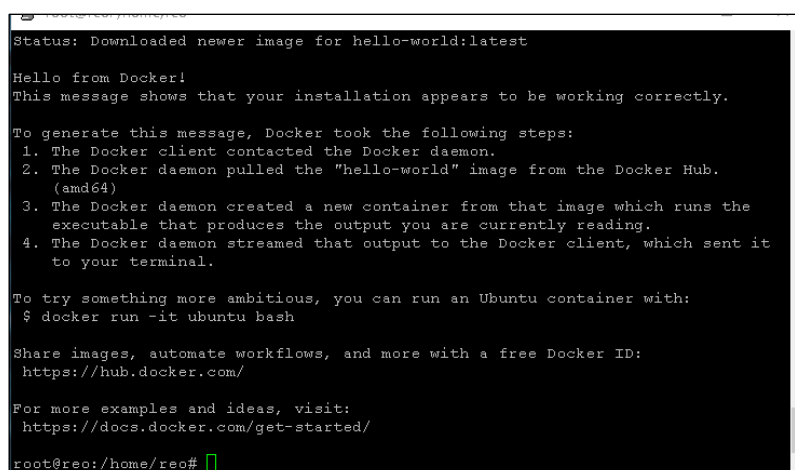
```

**Gambar 4.8** sintaks install docker

8. Verify Docker Engine. Sesuai **listing program 2.2** dan **gambar 4.9** dibawah ini.

```
sudo docker run hello-world
```

**Listing Program 2.2** Listing program run images hello word



```

Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

root@reo:/home/reo#

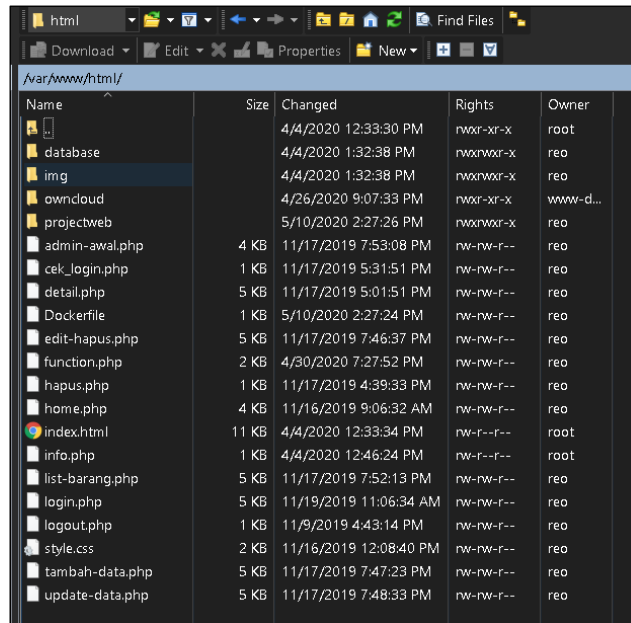
```

**Gambar 4.9** sintaks verify docker

## 2. Membuat Docker File

Untuk membuat file web yang sudah kita hosting ada menjadi docker, ada beberapa tahapan sebagai berikut :

1. Buka winscp, kemudian copy kan file yang ada di dalam folder projectweb keluar dari folder tersebut. Lihat **gambar 5.0** dibawah ini.



Name	Size	Changed	Rights	Owner
database		4/4/2020 12:33:30 PM	rw-r--r--	root
img		4/4/2020 1:32:38 PM	rw-r--r--	reo
owncloud		4/4/2020 1:32:38 PM	rw-r--r--	reo
projectweb		4/26/2020 9:07:33 PM	rw-r--r--	www-d...
admin-awal.php	4 KB	5/10/2020 2:27:26 PM	rw-r--r--	reo
cek_login.php	1 KB	11/17/2019 7:53:08 PM	rw-r--r--	reo
detail.php	5 KB	11/17/2019 5:31:51 PM	rw-r--r--	reo
Dockerfile	1 KB	11/17/2019 5:01:51 PM	rw-r--r--	reo
edit-hapus.php	5 KB	5/10/2020 2:27:24 PM	rw-r--r--	reo
function.php	2 KB	11/17/2019 7:46:37 PM	rw-r--r--	reo
hapus.php	1 KB	4/30/2020 7:27:52 PM	rw-r--r--	reo
home.php	4 KB	11/17/2019 4:39:33 PM	rw-r--r--	reo
index.html	11 KB	11/16/2019 9:06:32 AM	rw-r--r--	root
info.php	1 KB	4/4/2020 12:33:34 PM	rw-r--r--	root
list-barang.php	5 KB	4/4/2020 12:46:24 PM	rw-r--r--	reo
login.php	5 KB	11/17/2019 7:52:13 PM	rw-r--r--	reo
logout.php	1 KB	11/19/2019 11:06:34 AM	rw-r--r--	reo
style.css	2 KB	11/9/2019 4:43:14 PM	rw-r--r--	reo
tambah-data.php	5 KB	11/16/2019 12:08:40 PM	rw-r--r--	reo
update-data.php	5 KB	11/17/2019 7:47:23 PM	rw-r--r--	reo
	5 KB	11/17/2019 7:48:33 PM	rw-r--r--	reo

**Gambar 5.0** isi file folder html

2. Buka putty, dan arahkan direktori pada /var/www/html. Seperti **listing program 2.3** dibawah ini.

```
$ cd /var/www/html/
```

**Listing Program 2.3** Listing program pindah direktori

3. Buat file Dockerfile di dalam folder html, satu folder dengan file php websie kita. Seperti **listing program 2.4** dibawah ini.

```
FROM php:7.3-apache
RUN apt-get update && apt get upgrade -y
RUN docker-php-ext-intall mysqli

EXPOSE 80

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

COPY ./ /var/www/html
RUN service apache 2 restart
```

**Listing Program 2.4** Listing program isi Dockerfile

4. Buat docker image dari Dockerfile yang sudah kita buat tadi. Saya buat Dengan nama webweb dan tag versi-1. Seperti **listing program 2.5** dan pad a **gambar 5.1** dibawah ini.

```
$ docker images
```

### Listing Program 2.5 Listing program untuk melihat daftar images

```

reo@reo: ~
reo@reo:~$ docker images
REPOSITORY          TAG                 IMAGE ID
SIZE
webweb               versi-1            a6e358eacee7
515MB
web_app_web         latest             b98fc098ac00
412MB
php                  7.3-apache        0285d3ded552
411MB
mysql                8.0               a7a67c95e831
541MB
reo@reo:~$

```

**Gambar 5.1** daftar images yang dibuat

- Setelah image terbuat, jalankan image tersebut menjadi container. Sesuai dengan **listing program 2.6** dan **gambar 5.2** di bawah ini.

```
$ sudo docker run -d -p 8000:80 webweb:versi-1
```

### Listing Program 2.6 Listing program run images

```

reo@reo: ~
reo@reo:~$ sudo docker run -d -p 8000:80 webweb:versi-1
7b4da30fbd17a5643cfa6583beedb711c9bd1684a1c5a9685c21e4c33fb742b
reo@reo:~$

```

**Gambar 5.2** sintaks menjalankan images webweb

- Cek container yang berjalan apakah sudah sesuai. Seperti **listing program 2.7** dan **gambar 5.3** di bawah ini.

```

$ docker ps
Atau..
$ docker container ls

```

### Listing Program 2.7 Listing Program cek container yang berjalan

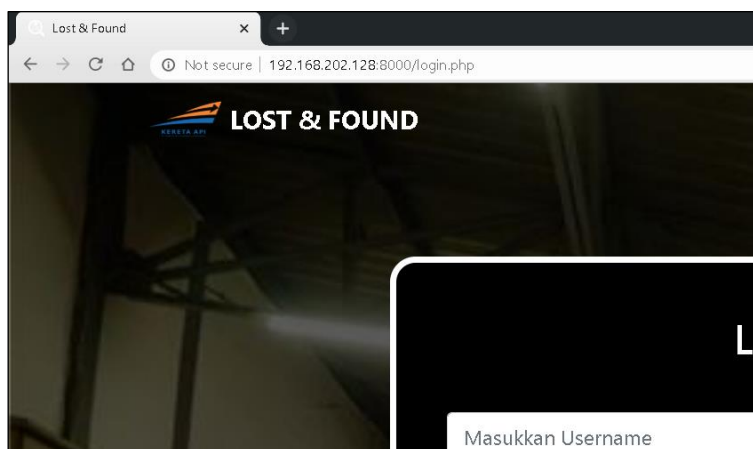
```

reo@reo: ~
reo@reo:~$ docker ps
CONTAINER ID        IMAGE               PORTS              COMMAND              NAMES              CREATED
7b4da30fbd17       webweb:versi-1     0.0.0.0:8000->80/tcp  "docker-php-entrypoi..."  zen_nightingale    About a minute ago
Up About a minute
reo@reo:~$
reo@reo:~$ docker container ls
CONTAINER ID        IMAGE               PORTS              COMMAND              NAMES              CREATED
7b4da30fbd17       webweb:versi-1     0.0.0.0:8000->80/tcp  "docker-php-entrypoi..."  zen_nightingale    About a minute ago
Up About a minute
reo@reo:~$

```

**Gambar 5.3** Container yang sedang berjalan

7. Buka browser, ketikkan alamat\_ip:8000/login.php. Lihat pada **gambar 5.4** dibawah ini.

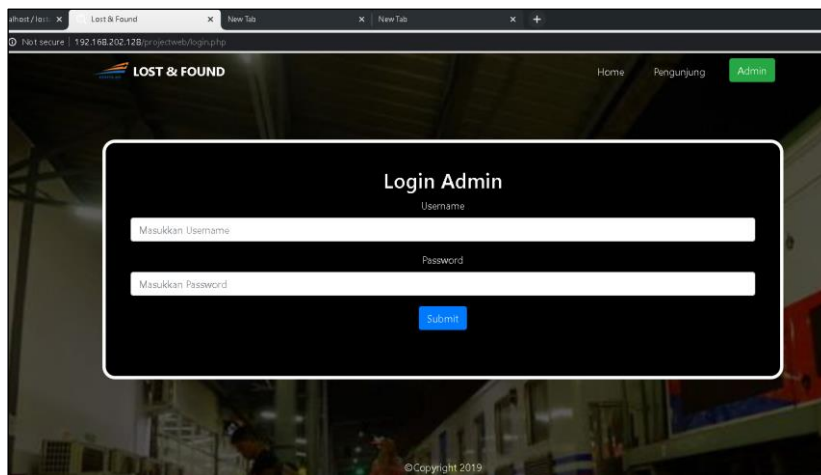


**Gambar 5.4** sudah dapat dijalankan

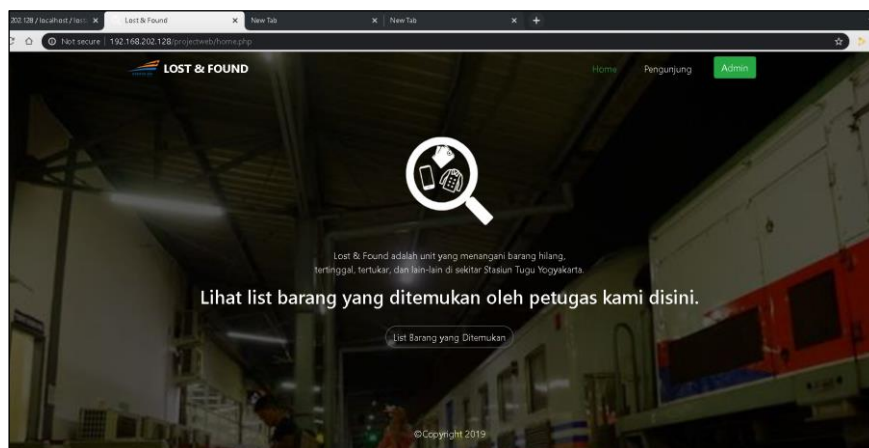
## 2.5 Hasil Implementasi

Dari hasil apa yang telah di kerjakan pada langkah-langkah sebelumnya, kita sudah dapat melakukan hosting file dengan menggunakan ubuntu lamp. File yang kita pindahkan pada winscp tadi adalah sekumpulan file yang berisi kumpulan file php, apabila dijalankan akan memuat halaman website lost and found. Untuk dapat membuka halaman tersebut kita tuliskan pada browser yang kita pakai, alamat\_ip/projectweb/login.php, dengan mengetikkan url tersebut nantinya browser akan menjalankan file yang ada di dalam server lokalnya, file yang dijalankan adalah file yang ada di dalam folder projectweb dengan nama file yang akan diakses adalah login.php, hostingan ini sama halnya apabila kita menggunakan xampp dan menuliskan localhost untuk dapat menjalankan semua file php kita yang ada di dalam file htdocs. Namun, pada kasus ini kita melakukan hostingan tersebut tanpa menggunakan xampp, tapi menggunakan apache yang dijalankan dalam sistem operasi ubuntu, menggunakan bantuan VmWare sebagai aplikasi untuk membantu virtualisasinya.

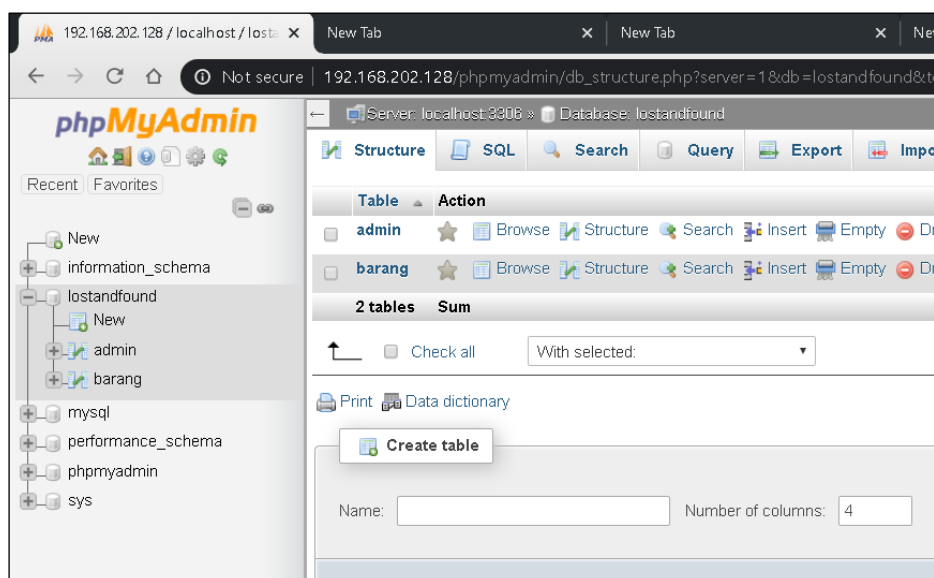
Berikut ini adalah tampilan web yang sudah kami hosting menggunakan apache, dan menggunakan ubuntu untuk menjalankan server apache dan hal-hal lain yang diperlukan untuk melakukan *hosting* file. Lihat pada **gambar 5.5** sampai **gambar 5.7** di bawah ini.



**Gambar 5.5** Halaman awal web yang di hosting

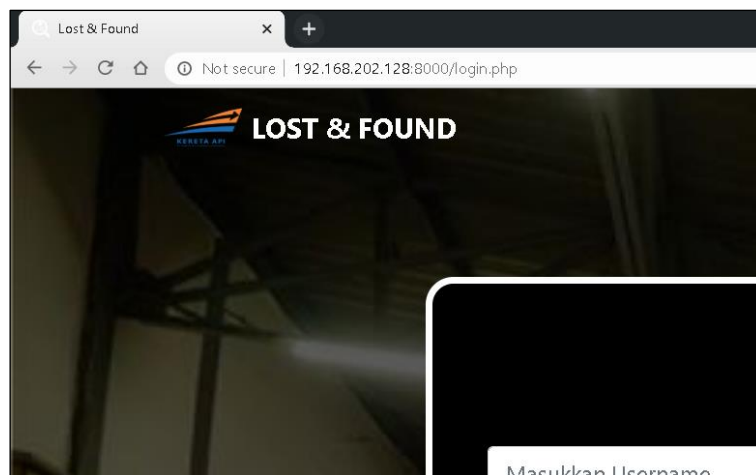


**Gambar 5.6** Halaman web sudah dapat diakses

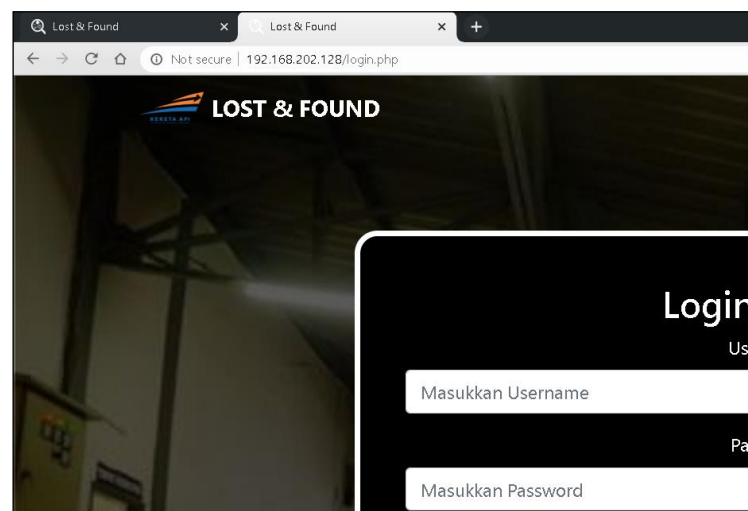


**Gambar 5.7** Tampilan database yang berisi data lost and found

Untuk hasil dari docker yang sudah kita kerjakan pada tahap sebelumnya, dapat kita jalankan dengan membuka browser dan mengetikkan `alamat_ip:8000/login.php`. Maksud dari docker tersebut adalah membuat port baru yang kemudian akan kita isikan atau kita fungsikan port tersebut menjadi sebuah website yang sudah kita hosting pada tugas 1. Dalam tahapan sebelumnya kita menuliskan `run -d -p 8000:80` itu bermaksud port 80 yang awalnya digunakan sebagai port standar kita untuk menjalankan file website yang sudah kita hosting pada tugas 1, akan kita pindahkan atau kita buat duplikatnya kemudian kita gunakan port 8000 untuk menjalankan file duplikat website tersebut, yang berjalan dengan bantuan docker. Sehingga, kita bisa membuka dua halaman web yang sama dengan menuliskan `alamat_ip/login.php` atau `alamat_ip:8000/login.php`. Dokumentasi dari hasil dapat dilihat pada **gambar 5.8** dan **gambar 5.9** di bawah ini.



**Gambar 5.8** Halman web dengan docker



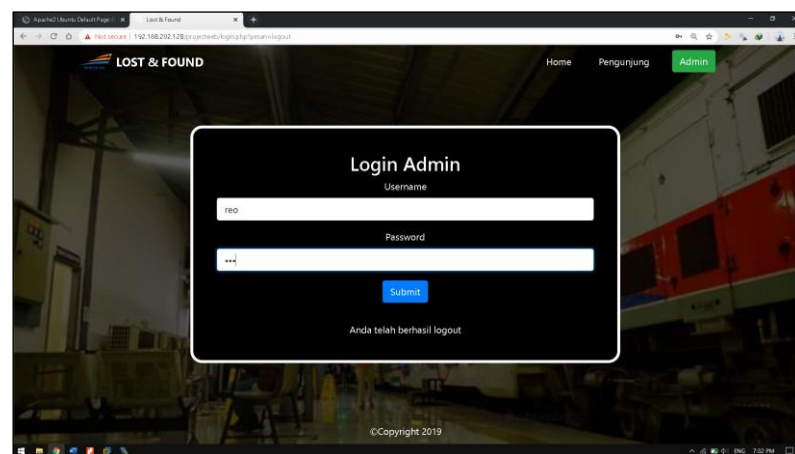
**Gambar 5.9** Halaman web tanpa docker



## 2.6 Pengujian Singkat

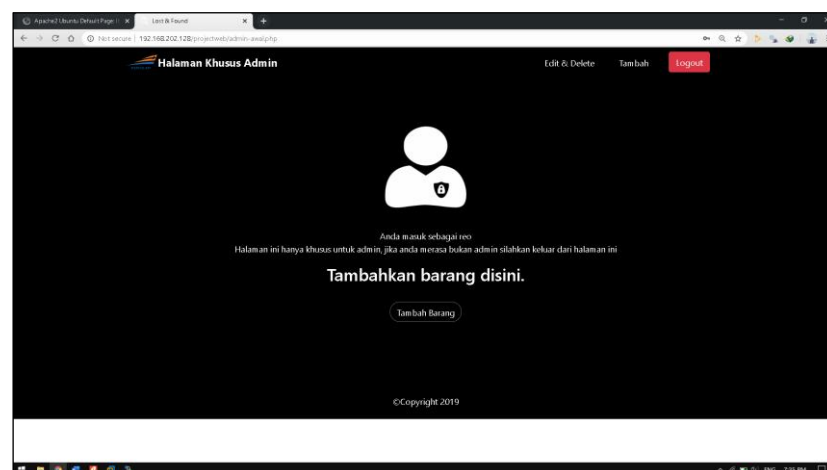
Pada bagian ini akan dijelaskan hasil dari apa yang sudah dikerjakan dalam tahap tahap sebelumnya, untuk mengetahui apakah file yang dihosting sudah dapat berjalan normal, atau malah memiliki Kendala atau file yang dihosting dapat dijalankan namun, terdapat beberapa fitur yang tidak berjalan secara normal.

Pengujian pertama yang dilakukan adalah, pengujian seorang admin untuk masu kedalam sistem website tersebut. dengan catatan, hanya adminlah yang memiliki username dan password yang sudah terdaftar dalam sistem informasi tersebut. melakukan proses login, maka perlu mengisi form username dan password, seperti dapat dilihat pada **gambar 6.0** seperti dibawah ini.



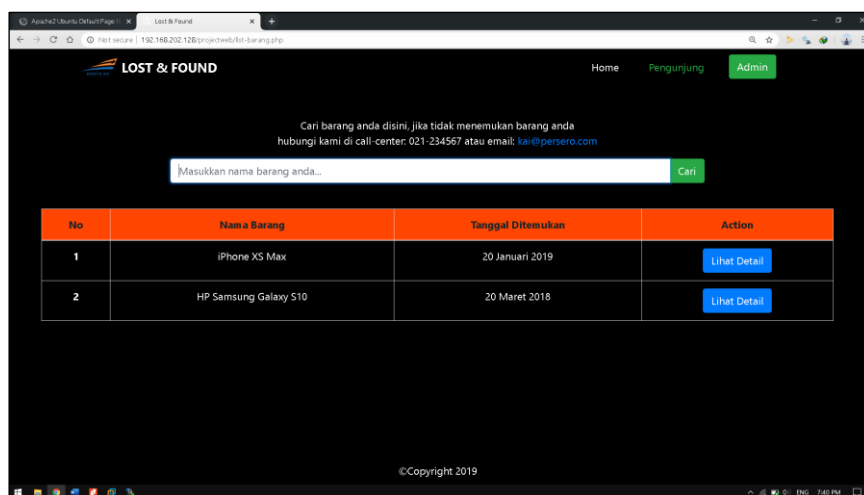
**Gambar 6.0** Tampilan login sistem informasi lost and found

Setelah login maka halaman langsung akan berpindah kepada, halaman selamat datang untuk admin, dan proses login pun sukses. Dapat dilihat dari **Gambar 6.1** dibawah ini.

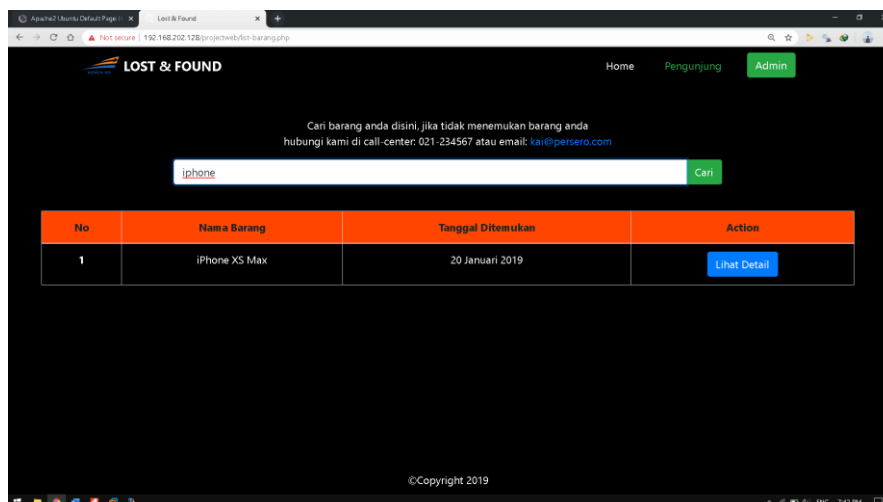


**Gambar 6.1** Tampilan sukses login

Pengujian selanjutnya adalah pengujian yang diasumsikan sebagai pengunjung untuk melihat dan mencari barang sesuai dengan nama. Pertama mengunjungi halaman web yang dikhususkan kepada pengunjung, terdapat button berwarna hijau bertuliskan pengunjung, maka akan berpindah ke tampilan seperti pada **Gambar 6.2** berikut ini.

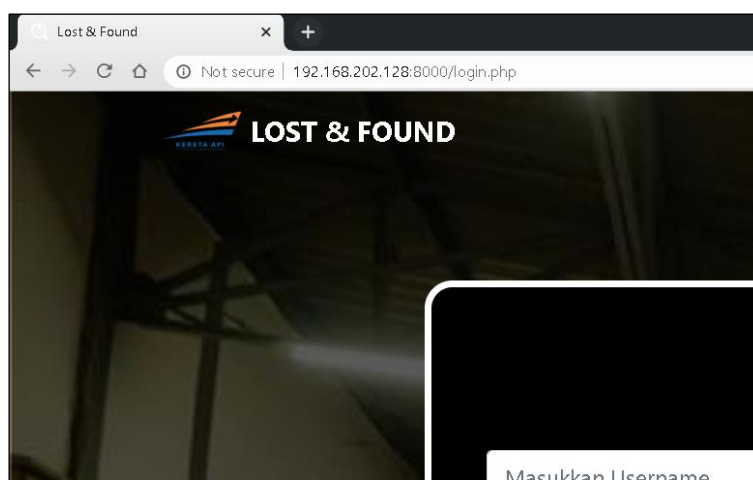
**Gambar 6.2** Tampilan list barang yang ditemukan

Kemudian pada halaman tersebut terdapat form untuk mencari barang sesuai dengan nama barang tersebut, maka menuliskan nama barang kemudian kita klik button cari yang berwarna hijau tersebut. Seperti yang terlihat pada **Gambar 6.3** berikut ini.

**Gambar 6.3** Tampilan saat mencari iphone

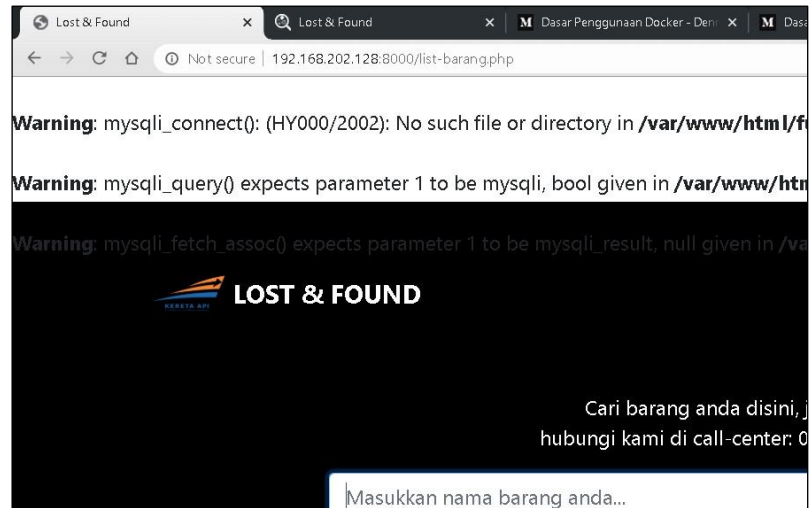
Kesimpulan dari dua pengujian kasus sederhana tersebut adalah, sistem yang dijalankan dengan hostingan menggunakan ubuntu dapat berjalan normal, *database* dapat terhubung tanpa ada kendala sama sekali.

Dalam proyek kedua, yaitu penggunaan docker kami mencoba membuat pengujian dengan sederhana untuk mengakses dan memastikan sistem yang terdapat di dalam docker tersebut berjalan normal tanpa kendala, hal yang pertama kami coba untuk memastikan docker telah berjalan normal adalah kami uji dengan mengakses halaman awal dari website lost and found, apakah ditemukan kendala atau tidak. Pengujian ini dilakukan dengan mengakses alamat ip dari server yang kita miliki, menggunakan port 8000, port yang sudah disesuaikan dengan konfigurasi docker yang digunakan dalam proyek kedua ini. Hasilnya, website lost and found dapat berjalan normal, menampilkan halaman yang benar, seperti pada saat tidak menggunakan docker. Dapat dilihat dalam **gambar 6.4** dibawah ini.



**Gambar 6.4** tampilan halaman awal menggunakan docker

Kemudian, percobaan kedua dilakukan dengan mengakses halaman pengujung, untuk melihat apakah database yang digunakan sudah dapat berjalan normal atau tidak. Pengujian tersebut dilakukan dengan cara mengakses halaman awal website dan memilih *button* pengujung, dan nantinya yang akan tampil adalah daftar barang yang ditemukan dan diupload dalam website tersebut, sehingga halaman yang diakses adalah alamat ip menggunakan port 8000/list-barang.php. Namun, halaman yang diakses ternyata masih memiliki kekurangan, yaitu database yang digunakan dalam docker ini tidak dapat bekerja secara normal, sehingga tidak dapat menampilkan data yang ada di database. Dapat dilihat dari **gambar 6.5** dibawah ini.



**Gambar 6.5** tampilan akses database web dengan docker

### 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			
		1	2	3	4	1	2	3	4
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Instalasi Software untuk hosting								
4.	Konfigurasi Hosting								
5.	Instalasi Software untuk Docker								
6.	Konfigurasi Docker								
7.	Dokumentasi Pengerjaan								
8.	Pembuatan Laporan								
9.	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	Andre
2.	Pengujian Singkat	Reo
3.	Latar Belakang Masalah	Andre
4.	Instalasi Software untuk hosting	Reo
5.	Konfigurasi hosting	Reo
6.	Instalasi Software untuk Docker	Reo
7.	Konfigurasi Docker	Reo
8.	Kesimpulan dan Saran	Andre

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berdasarkan masalah dan tugas yang dikerjakan dalam laporan ini, terdapat dua terdapat dua tugas yaitu tugas pertama untuk melakukan hosting, dan program dua memasukkann hostingan tersebut ke dalam paket-paket yang dijalankan menggunakan docker. Dari dua tugas tersebut tugas pertama, dapat berjalan dengan normal, baik tampilan *interface* ataupun database dari website kami buat tidak berubah saat kami hosting menggunakan layanan LAMPP ubuntu tersebut, database yang digunakan juga dapat berjalan normal, data-data dapat ditampilkan dan proses crud dari database berjalan normal tidak ada masalah, tugas pertama kami simpulkan dapat berjalan semestinya tanpa kendala apapun. Tugas kedua adalah memasukkan paket-paket file yang terdapat pada tugas pertama untuk dijalankan dalam bentuk dockerfile dan dapat dijalankan melalui port 8000. Dalam tugas kedua ini, port 800 yang kami pilih pada saat diakses dapat menampilkan interface yang sama, tidak memiliki kendala dalam hal halaman yang ditampilkan. Namun, pada tugas kedua ini terdapat kendala yaitu pada bagian database tidak dapat diakses, database tidak dapat menampilkan data yang ada pada saat halaman website dijalankan dengan menggunakan docker.

#### **4.2 Saran**

Saran yang dapat kami tambahkan dalam pelaksanaan proses pengerjakan tugas ini adalah sebaaiknya diberikan penjelasan mengenai komponen-komponen yang digunakan terlebih dahulu sehingga tidak memakan waktu yang lama dalam melakukan pengerjakan tugas, sebaiknya untuk pembagian tugas lebih merata sehingga tugas dapat lebih cepat diselesaikan, dan untuk mahasiswa yang sekiranya ke depan akan mengerjakan tugas ini, lebih diperhatikan lagi bagian docker dan database dari docker tersebut sehingga docker yang digunakan dapat berjalan normal tanpa ada kendala apapun.

## DAFTAR PUSTAKA

- Adiputra, F. (2015). Container dan Docker: Teknik Virtualisasi dalam Pengelolaan Banyak Aplikasi Web. *Jurnal SimanteC*, 4(3).
- Fandi. (2017). *Pengertian Cloud Computing Menurut Ahli*. Fandi Media.  
<https://www.fandimedia.com/2017/11/pengertian-cloud-computing-menurut-ahli.html>
- Hutomo, I. (2015). *Apa itu Cloud Computing?* Kompasiana.  
<https://www.kompasiana.com/ianhutomo/5500f4dfa333115318510435/apa-itu-cloud-computing>
- Indonesia, C. F. (2017). *Dasar Manajemen Docker Dengan Portainer*.  
[https://www.cilsy.id/assets/source/filepraktek/Modul dasar docker portainer.pdf](https://www.cilsy.id/assets/source/filepraktek/Modul%20dasar%20docker%20portainer.pdf)
- ROMADLON BIK, M. (2017). IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB (Studi Kasus : Jurusan Teknik Informatika UNESA). *Jurnal Manajemen Informatika*, 7(2), 46–50.

## **LAMPIRAN**

Lampiran pada bagian ini dapat berupa screenshoot, listing program yang terlalu panjang, dan sebagainya, atau dapat juga tugas bilamana diminta oleh asisten praktikum.