

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR

**SISTEM INFORMASI KOST MENGGUNAKAN UBUNTU LAMPP DAN PROSES
PEMBUATAN *DOCKERFILE*NYA**



DISUSUN OLEH:

NAMA ANGGOTA : FAHMI REZA PRASASTIO 123170068
ADIB BACHTIAR 123170075

KELAS : C

ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.
WAHYU AJI NUGROHO, S.Kom.

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

HALAMAN PENGESAHAN

SISTEM INFORMASI KOST MENGGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN *DOCKERFILE*NYA

Disusun oleh :

Fahmi Reza Prasastio

123170068

Adib Bachtiar

123170075

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Wahyu Aji Nugroho, S.Kom

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul Sistem Informasi Kost Menggunakan Ubuntu LAMPP Dan Proses Pembuatan *Dockerfilenya*. Adapun laporan ini berisi tentang proyek akhir yang salah satunya bersifat wajib dan pilihan yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

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

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

Yogyakarta, 31 Maret 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Tujuan Proyek Akhir.....	2
1.3. Manfaat Proyek Akhir.....	2
BAB II ISI DAN PEMBAHASAN	4
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	9
2.5. Hasil Implementasi.....	16
2.5.1 Hasil Implementasi Proyek Pertama	16
2.5.2 Hasil Implementasi Proyek Kedua.....	17
2.6. Pengujian Singkat.....	17
2.6.1 Pengujian Singkat Proyek Pertama	17
2.6.2 Pengujian Singkat Proyek Kedua	18
BAB III JADWAL Pengerjaan dan Pembagian Tugas	20
3.1. Agenda Pengerjaan.....	20
3.2. Keterangan Pembagian Tugas	20
BAB IV KESIMPULAN DAN SARAN	22
4.1. Kesimpulan	22
4.2. Saran.....	22
DAFTAR PUSTAKA	22

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Penggunaan *Cloud Computing* saat ini sangat umum di era teknologi saat ini, berawal dari ide seorang bernama John McCarthy pada tahun 1960 yang bermimpi bahwa komputasi akan menjadi Infrastruktur *public* seperti halnya listrik dan telepon yang terealisasi sekarang bahkan dikembangkan sampai saat ini manfaat yang dihasilkan mampu dirasakan oleh semua kalangan masyarakat. Contoh penggunaan *cloud computing* di masyarakat saat ini yaitu penggunaan *social* media seperti facebook dll, serta berbagai system informasi yang tersedia saat ini pasti menggunakan teknologi *cloud computing*. Pada judul kami “Sistem Informasi Kost Menggunakan Ubuntu LAMP Dan Proses Pembuatan Dockerfilenya” bisa dilihat dalam situs ini terdapat 2 pelaku yaitu antara pemilik dan pencari *kost*, maka dari ini akankah lebih baik jika pemilik dan pencari dapat berhubungan satu sama lain melalui website yang kami buat ini. maka akan dapat memudahkan antara keduanya dalam mencari solusi bisnis, serta dapat menjadi solusi bisnis bagi para pemilik kos dalam menyewakan usahanya. Maka dari ini kami memanfaatkan teknologi *cloud computing* agar bisa diakses antar kedua pelaku tersebut. Dan *website* bisa diakses siapapun dan kapanpun dalam jaringan yang sama.

Cloud computing adalah bentuk khusus dari komputasi terdistribusi yang memperkenalkan model pemanfaatan untuk penyediaan sumber daya yang terukur dari jarak jauh (Erl, Puttini, & Mahmood, 2013).

Sudaryono et al. (1978) mengungkapkan bahwa *cloud computing* atau komputasi awan ialah teknologi yang memanfaatkan layanan internet menggunakan pusat server yang bersifat virtual dengan tujuan pemeliharaan data dan aplikasi. Keberadaan komputasi awan jelas akan menimbulkan perubahan dalam cara kerja sistem teknologi informasi dalam sebuah organisasi.

Sementara *Docker* merupakan sebuah project *open-source* yang menyediakan *platform* terbuka untuk *developer* maupun *sysadmin* agar dapat menjalankan aplikasi dimanapun sebagai sebuah wadah (*container*) yang ringan menurut Asrofil et al (2020). Dengan menggunakan *Docker*, Kami dapat menghemat uang, memindahkan *source code* dengan lancar dan menstandarisasi operasi aplikasi

dengan meningkatkan pemanfaatan sumber daya. Dengan menggunakan *Docker*, kami mendapatkan suatu objek yang dapat dijalankan dimana saja

Pada project ini kami menggunakan ubuntu LAMPP dalam proses *upload source code* ke server *local* agar bisa diakses siapa saja dalam server *local*. Dalam hal ini kami sudah menyiapkan *source code* berbasis web yang siap untuk diunggah ke server *local* dan setelah itu dibuat menjadi *Dockerfile*. Pada saat itu kami sudah melakukan proses *testing*, konfigurasi dan instalasi sesuai dengan dokumentasi yang rutin dilakukan tiap praktikum. Tahapan dalam menyelesaikan project ini yaitu dengan menyiapkan software diantaranya VMWare WorkStation, PuTTY, WinSCP, Docker, dan Dokku serta data mentah berisi Sistem Operasi Linux dalam bentuk Format ISO yang nanti akan diinstall dan dijalankan lewat aplikasi VMWare Workstation

1.2 Tujuan Proyek Akhir

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

1. Mengimplementasikan arsitektur *cloud computing* untuk Sistem Informasi Kost yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Mengimplementasikan arsitektur *cloud computing* untuk membuat *Dockerfile* dari Sistem Informasi Kost

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 terdistribusi dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.

1.4 Tahap Penyelesaian Proyek Akhir

Tahap Penyelesaian dari Proyek 1 dimulai dari persoalan bisnis yang kami anggap penting untuk diselesaikan dengan teknologi cloud computing, yaitu persoalan Bisnis *Kost*. Lalu Kami menyiapkan bahan yang dibutuhkan untuk menggunakan sarana Cloud Computing untuk Solusi Bisnis ini diantaranya yaitu Server *Ubuntu* untuk memasang dan menggunakan *PHP*, *MySQL* dan *Source Code* dengan Bahasa *PHP* yang sudah kami buat sedemikian rupa untuk membuat suatu website yang bisa diakses banyak orang dalam satu server yang sama. Jadi secara singkat kami menginstall server ubuntu dan menginstall aplikasi didalamnya. Diantaranya *PHP* dan *MySQL* agar bisa berinteraksi dengan source code yang sudah kami buat sebelumnya. Dengan rancangan Topologi di bab 2 maka akan menjelaskan rancangan dari aplikasi yang akan kami buat. Hasilnya maka aplikasi ini akan bisa digunakan oleh user yang dituju yaitu pemilik kos dan pencari kos

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Untuk membangun “Sistem Informasi Kost Menggunakan Ubuntu Lamp Dan Proses Pembuatan Dockerfilenya” yang berbasiskan 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 untuk keperluan website Sistem Informasi *Kost*.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunanya ialah seluruh user dan admin aplikasi Sistem Informasi *Kost* di Indonesia. 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 dan kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode NAT	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.117.130/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.117.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.117.255	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.4 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	<i>Harddisk</i>	20 GB	Alokasi <i>harddisk</i>
6.	<i>Memory</i>	1 GB	Alokasi <i>memory</i> untuk <i>guest OS</i>

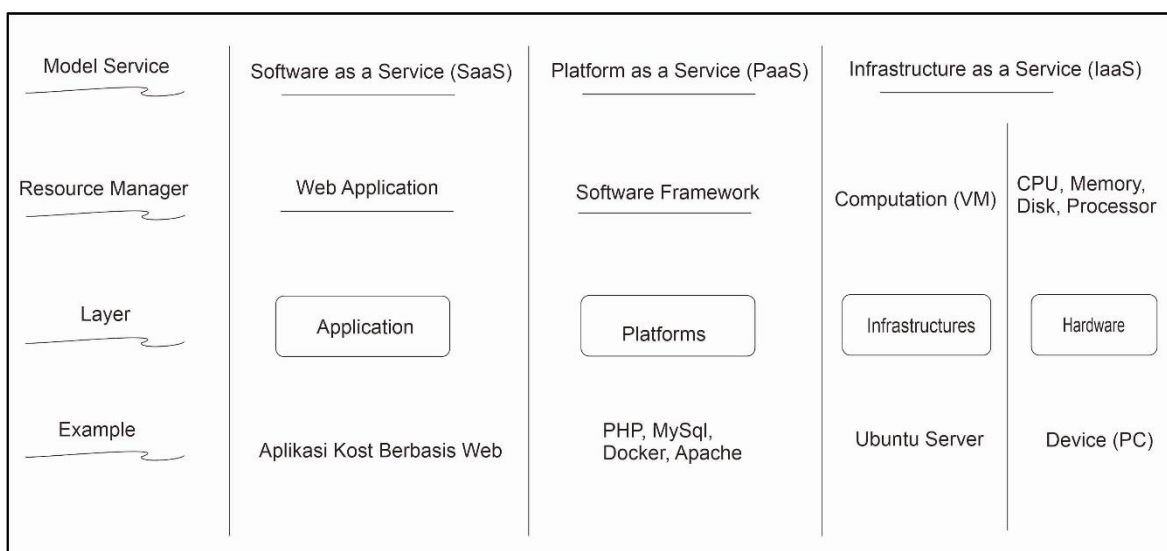
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

Tabel 2.2 Spesifikasi Ubuntu OS untuk proyek pertama dan kedua

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preproesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.2.24	Bahasa Pemrograman yang digunakan dalam membangun sistem informasi kost
		MySQL	Basis Data yang digunakan dalam menyimpan data sistem informasi kost
		PHPMyAdmin	Perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL
2.	Docker	Docker 19.03.8	Platform terbuka agar dapat membangun dan menjalankan aplikasi di berbagai lokasi sebagai sebuah <i>container</i> yang ringan
3.	Docker Compose	Docker Compose 1.24.0	Dukungan alat untuk mendefinisikan dan menjalankan multi-container.

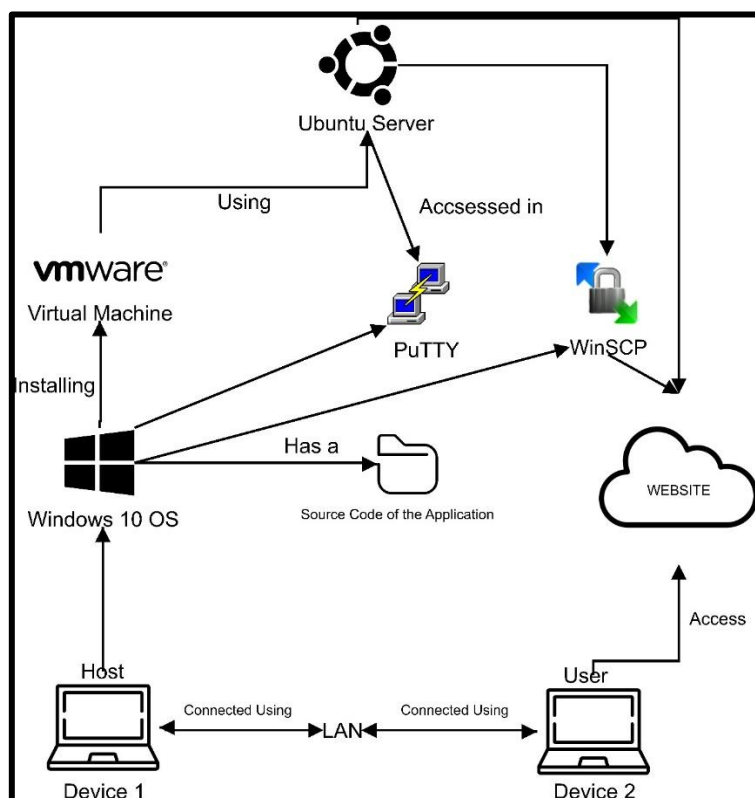
2.2 Rancangan Arsitektur *Cloud Computing*

Pada proyek akhir ini digunakan arsitektur SaaS untuk Sistem Informasi Kost, selanjutnya di bawahnya terdapat layer PaaS dimana sistem operasi Ubuntu berjalan. Kemudian pada layer paling bawah yaitu IaaS terdapat hardware laptop yang melakukan virtualisasi pada hardware yang diperlukan. Arsitekturnya dapat dilihat pada **gambar 2.1** dibawah ini:

**Gambar 2.1** Layer Arsitektur

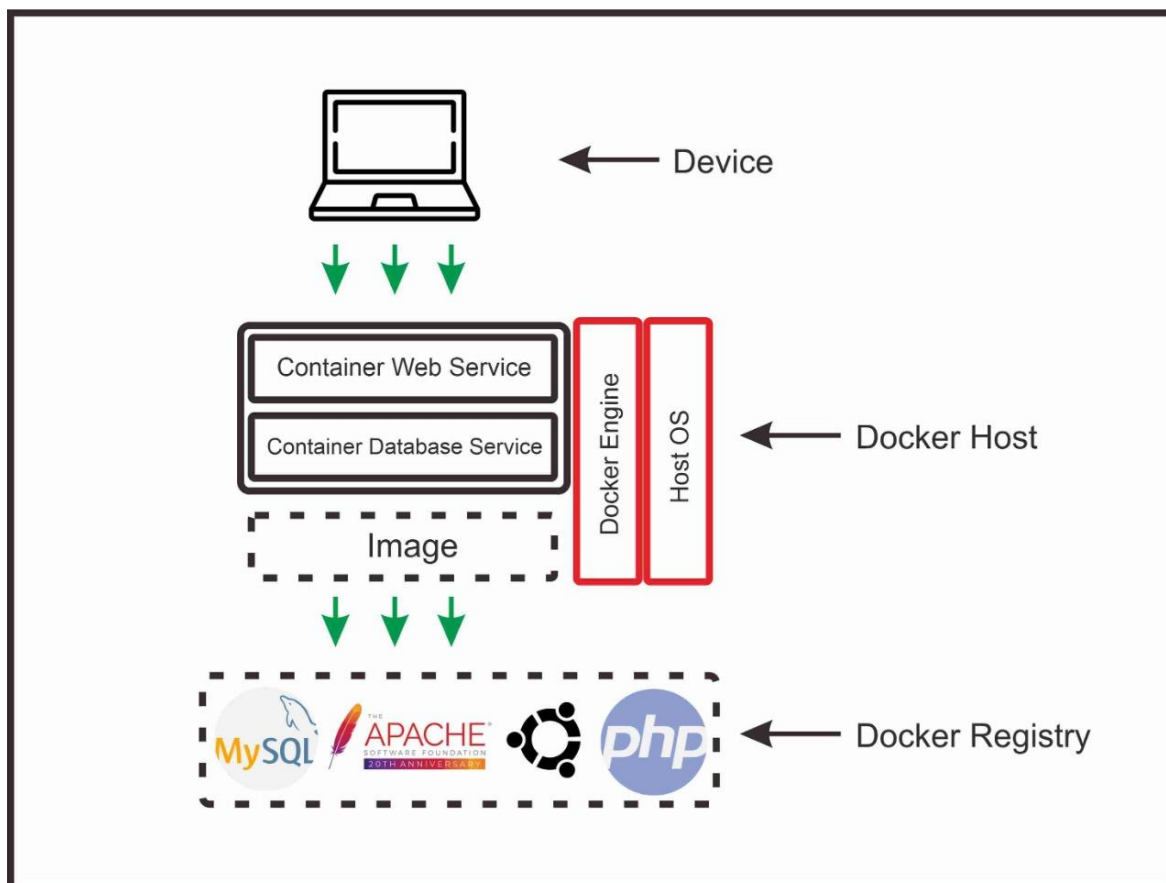
Pada proyek pertama koneksi LAN (Indihome) digunakan untuk menghubungkan Device 1 yang bertugas sebagai host dan Device 2 yang

bertugas sebagai User. Tujuan menghubungkan kedua device dengan LAN adalah agar website juga dapat diakses oleh Device 2 tidak hanya Device 1 saja. Device 1 yang bertugas sebagai host yang memakai OS Windows kemudian dilakukan instalasi VMware Workstation, PuTTY dan WinSCP. Kemudian pada VMWare Workstation dilakukan instalasi Ubuntu Server yang akan dilakukan remote oleh software PuTTY. Kemudian dalam Ubuntu Server akan dilakukan instalasi Apache, MySQL, PHPMyAdmin sehingga dapat menjalankan website. Source Code dari website yang akan digunakan kemudian di copy ke dalam directory yang terdapat pada Ubuntu Server dengan menggunakan WinSCP yang sudah tersambung dengan IP Address dari Ubuntu Server. Arsitekturnya dapat dilihat pada **gambar 2.2** dibawah ini:



Gambar 2.2 Topologi Proyek Pertama

Pada proyek kedua menerapkan teknologi Docker yang akan membungkus proyek web yang sudah dibuat beserta dependency yang diperlukan web tersebut. Arsitekturnya dapat dilihat pada **gambar 2.3** dibawah ini:



Gambar 2.3 Topologi Proyek Kedua

2.3 Parameter dan Konfigurasi

Parameter yang digunakan untuk instalasi Apache dapat dilihat pada penjelasan

Modul 2.1 berikut ini:

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

$ sudo ufw allow in "Apache Full"
Keterangan: Untuk mengatur Firewall agar port 80 dan port 443
diizinkan oleh sistem Ubuntu.
```

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 pada apt untuk mengeksekusi perintah instalasi paket
- mysql-server : nama paket untuk MySQL

$ sudo mysql_secure_installation
Keterangan: Untuk mengatur keamanan pada MySQL contohnya username dan password
```

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
- php libapache2-mod-php php-mysql : nama paket untuk PHP

$ sudo nano /var/www/html/info.php
Keterangan: Untuk membuat file baru bernama info.php yang berada di direktori var/www/html
```

Modul 2.3 Parameter instalasi PHP

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

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

$ sudo mysql -u root
Keterangan: Untuk masuk ke MySQL sebagai user root.

mysql> UPDATE mysql.user SET plugin = 'mysql_native_password', authentication_string = PASSWORD('123') WHERE User = 'root';
Keterangan: Untuk mengubah password dari user 'root' menjadi '123'

mysql> FLUSH PRIVILEGES;
Keterangan: Untuk merefresh akun yang terkoneksi dengan phpmyadmin

$ sudo chown fahmi /var/www/html
Keterangan: Untuk memberikan akses ke User fahmi agar dapat mengakses direktori var/www/html
```

Modul 2.4 Parameter instalasi PHPMyAdmin

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

```
$ sudo apt-get update
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
Keterangan: Untuk memperbarui indeks paket apt dan instal
paket untuk memungkinkan apt menggunakan repositori melalui HTTPS:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg
| sudo apt-key add -
$ sudo apt-key fingerprint 0EBFCD88
Keterangan: Untuk menambahkan Docker Official GPG Key

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

$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Keterangan: Untuk memperbarui indeks paket apt dan install
versi terbaru Docker Engine
```

Modul 2.5 Parameter instalasi Docker

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

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

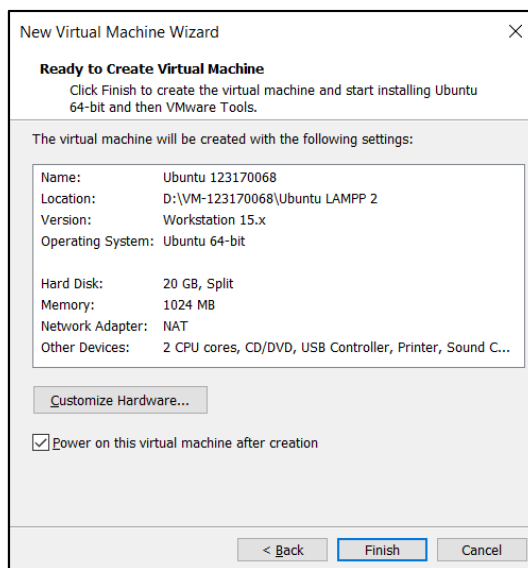
Keterangan:
- sudo : perintah untuk eksekusi suatu command dengan hak akses
tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah
instalasi paket
- docker docker-compose: nama paket untuk docker-compose
-docker-compose --version : untuk mengecek versi dari docker-compose yang
telah di install
```

Modul 2.6 Parameter instalasi Docker-compose

2.4 Tahap Implementasi

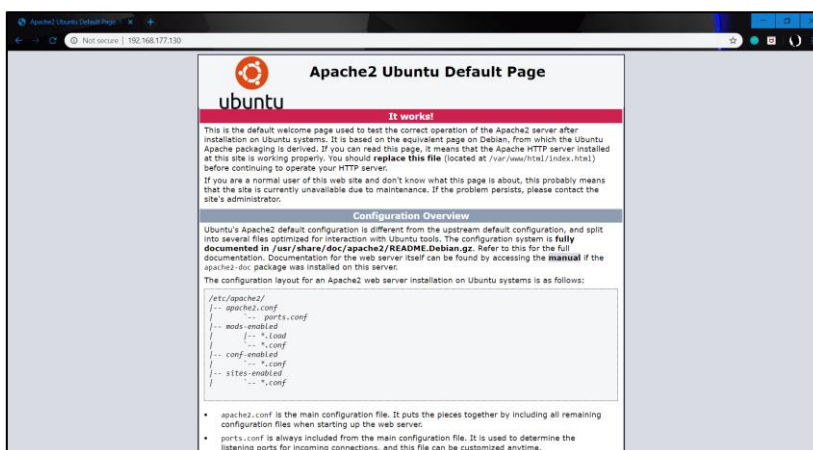
2.4.1 Tahap Implementasi Proyek Pertama

Hasil dari implementasi **Tabel 2.3** tentang konfigurasi untuk VMware Workstation untuk Virtualisasi Ubuntu yang dibuat dapat dilihat pada **Gambar 2.4** berikut ini:



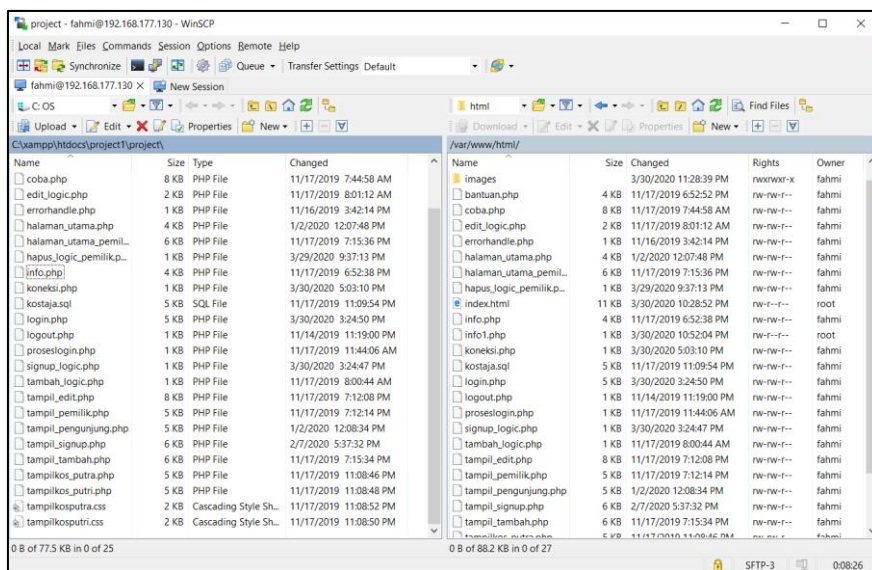
Gambar 2.4 Hasil Pembuatan Virtual Machine dengan OS Ubuntu

Hasil dari implementasi **Modul 2.1** tentang cara instalasi Apache dapat dilihat pada **Gambar 2.5** berikut ini:



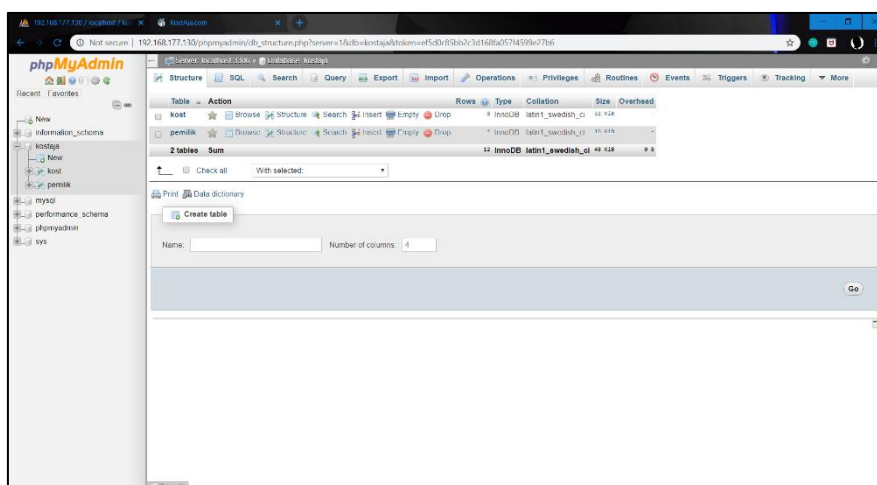
Gambar 2.5 Hasil Instalasi Apache

Hasil dari implementasi **Modul 2.3** tentang cara instalasi PHP dapat dilihat pada **Gambar 2.6** dibawah ini:



Gambar 2.8 Proses Pemindahan Source Code

Setelah dilakukan source code berhasil dimasukkan ke dalam direktori var/www/html maka langkah selanjutnya adalah melakukan import database website yang digunakan seperti pada **gambar 2.9** dibawah ini:



Gambar 2.9 Hasil Import Database

Setelah semua tahapan selesai kita dapat langsung mengeceknya dengan membuka browser dan mengetikkan IP-SERVER. Untuk hasil implementasi dapat dilihat pada bagian Hasil Implementasi Proyek Pertama.

2.4.2 Tahap Implementasi Proyek Kedua

Hasil dari implementasi **Modul 2.5** tentang instalasi Docker dapat dilihat pada **Gambar 2.10** berikut ini:


```
fahmi@fahmi-server:~$ docker -v
Docker version 19.03.8, build afacb8b7f0
```

Gambar 2.10 Hasil Instalasi Docker

Hasil dari implementasi **Modul 2.6** tentang instalasi Docker-compose dapat dilihat pada **Gambar 2.11** berikut ini:

```
fahmi@fahmi-server:~$ docker-compose --version
docker-compose version 1.24.1, build 4667896b
```

Gambar 2.11 Hasil Instalasi Docker-compose

Setelah itu langkah selanjutnya akan diberikan perintah permission untuk Docker agar dapat digunakan, maka lakukan perintah seperti pada **Perintah Program 2.1** berikut ini:

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

Perintah Program 2.1 Perintah Pemberian *Permission* untuk *Docker*

Jika perintah tersebut sudah berhasil dijalankan, maka Docker sudah dapat digunakan dan tidak muncul *error permission* saat akan melakukan perintah docker. Selanjutnya kita akan membuat *directory* baru di ubuntu kita bernama docker dan akan masuk ke *directory* nya dengan perintah berikut:

```
$ mkdir docker
$ cd docker
```

Perintah Program 2.2 Perintah membuat *directory* baru

Selanjutnya setelah selesai membuat *directory* baru dan masuk ke dalamnya kita akan *copy* file *source code* dan file database dari *project* yang sudah dibuat dan masukkan ke *directory* baru tersebut menggunakan WinSCP. Setelah itu di dalam *directory* docker kita akan membuat file Dockerfile dengan perintah seperti pada **Perintah Program 2.3** berikut:

```
$ nano Dockerfile
```

Perintah Program 2.3 Perintah membuat dan mengedit file Dockerfile

Selanjutnya kita isikan isi dari file Dockerfile tadi dengan isian seperti pada **Perintah Program 2.4** berikut:

```
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 775 /var/www/html

COPY ./ /var/www/html

RUN service apache2 restart
```

Perintah Program 2.4 Isi file Dockerfile

Setelah itu klik CTRL+O untuk save file tersebut dan klik CTRL+X untuk keluar dari perintah nano.

Selanjutnya kita akan membuat file docker-compose.yml dengan perintah seperti pada **Perintah Program 2.5** berikut:

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

Perintah Program 2.5 Perintah membuat dan mengedit file docker-compose.yml

Selanjutnya kita isikan isi dari file docker-compose.yml tadi dengan isian seperti pada **Perintah Program 2.6** dan **Perintah Program 2.7** berikut:

```
Version: '3.3'

services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: kostaja
    depends on:
      - db
    volumes:
      - ./:/var/www/html
```

Perintah Program 2.6 Isi file docker-compose.yml

```

ports:
  - 8000:80

db:

  container_name: kostajadb

  image: mysql:8.0

  command: --default-authentication-
plugin=mysql_native_password

  restart: always

  environment:

    MYSQL_ROOT_PASSWORD: root

    MYSQL_DATABASE: kostaja

    MYSQL_USER: fahmi

    MYSQL_PASSWORD: 123170068

  ports:
    - 6036:3306

```

Perintah Program 2.7 Lanjutan Isi file docker-compose.yml

Hasil dari copy source code, membuat file Dockerfile, dan membuat file docker-compose.yml dapat kita lihat pada aplikasi WinSCP seperti pada **gambar 2.12** berikut ini:

/home/fahmi/docker/				
Name	Size	Changed	Rights	Owner
images		3/30/2020 11:28:39 PM	rw-rw-r--	fahmi
bantuan.php	4 KB	4/25/2020 7:36:48 PM	rw-rw-r--	fahmi
docker-compose.yml	1 KB	5/6/2020 1:34:18 PM	rw-rw-r--	fahmi
Dockerfile	1 KB	5/5/2020 7:31:13 PM	rw-rw-r--	fahmi
edit_logic.php	2 KB	11/17/2019 8:01:12 AM	rw-rw-r--	fahmi
errorhandle.php	1 KB	11/16/2019 3:42:14 PM	rw-rw-r--	fahmi
halaman_utama.php	4 KB	4/25/2020 7:44:33 PM	rw-rw-r--	fahmi
halaman_utama_pemil...	6 KB	11/17/2019 7:15:36 PM	rw-rw-r--	fahmi
halaman_utama_peng...	5 KB	4/25/2020 7:43:37 PM	rw-rw-r--	fahmi
hapus_logic_pemilikp...	1 KB	3/29/2020 9:37:13 PM	rw-rw-r--	fahmi
index.html	5 KB	4/25/2020 7:45:27 PM	rw-rw-r--	fahmi
info.php	4 KB	4/25/2020 7:42:22 PM	rw-rw-r--	fahmi
info.php.php	1 KB	3/30/2020 10:52:04 PM	rw-rw-r--	fahmi
koneksi.php	1 KB	5/11/2020 10:51:19 AM	rw-rw-r--	fahmi
kostaja.sql	5 KB	11/17/2019 11:09:54 PM	rw-rw-r--	fahmi
login.php	5 KB	4/25/2020 7:41:31 PM	rw-rw-r--	fahmi
logout.php	1 KB	11/14/2019 11:19:00 PM	rw-rw-r--	fahmi
proseslogin.php	1 KB	11/17/2019 11:44:06 AM	rw-rw-r--	fahmi
signup_logic.php	1 KB	3/30/2020 3:24:47 PM	rw-rw-r--	fahmi
tambah_logic.php	1 KB	11/17/2019 8:00:44 AM	rw-rw-r--	fahmi
tampil_edit.php	8 KB	11/17/2019 7:12:08 PM	rw-rw-r--	fahmi
tampil_pemilik.php	5 KB	5/6/2020 8:31:02 PM	rw-rw-r--	fahmi
tampil_signup.php	6 KB	5/6/2020 8:30:39 PM	rw-rw-r--	fahmi
tampil_tambah.php	6 KB	11/17/2019 7:15:34 PM	rw-rw-r--	fahmi
tampilkos_putra.php	5 KB	5/6/2020 8:58:14 PM	rw-rw-r--	fahmi
tampilkos_putri.php	5 KB	5/6/2020 8:28:58 PM	rw-rw-r--	fahmi
tampilkosputra.css	2 KB	11/17/2019 11:08:52 PM	rw-rw-r--	fahmi
tampilkosputri.css	2 KB	11/17/2019 11:08:50 PM	rw-rw-r--	fahmi

Gambar 2.12 Hasil copy source code dan pembuatan file

Selanjutnya kita akan melakukan *build* file docker-compose.yml seperti pada **Perintah Program 2.8** berikut:

```
$ docker-compose up -d
```

Perintah Program 2.8 Build docker-compose.yml

Setelah itu langkah selanjutnya kita akan *import file database kostaja.sql* yang terdapat pada directory yang sama dengan source code kita dan mengimport ke dalam *database kostaja* yang sudah dideklarasikan dan dibuat sebelumnya pada *file docker-compose.yml*. Perintah untuk *mengimport file sql* dapat dilihat pada Perintah Program 2.9 berikut ini:

```
$ docker exec -i kostajadb mysql -ufahmi -p123170068 kostaja < ./kostaja.sql
```

Perintah Program 2.9 Import file sql ke database

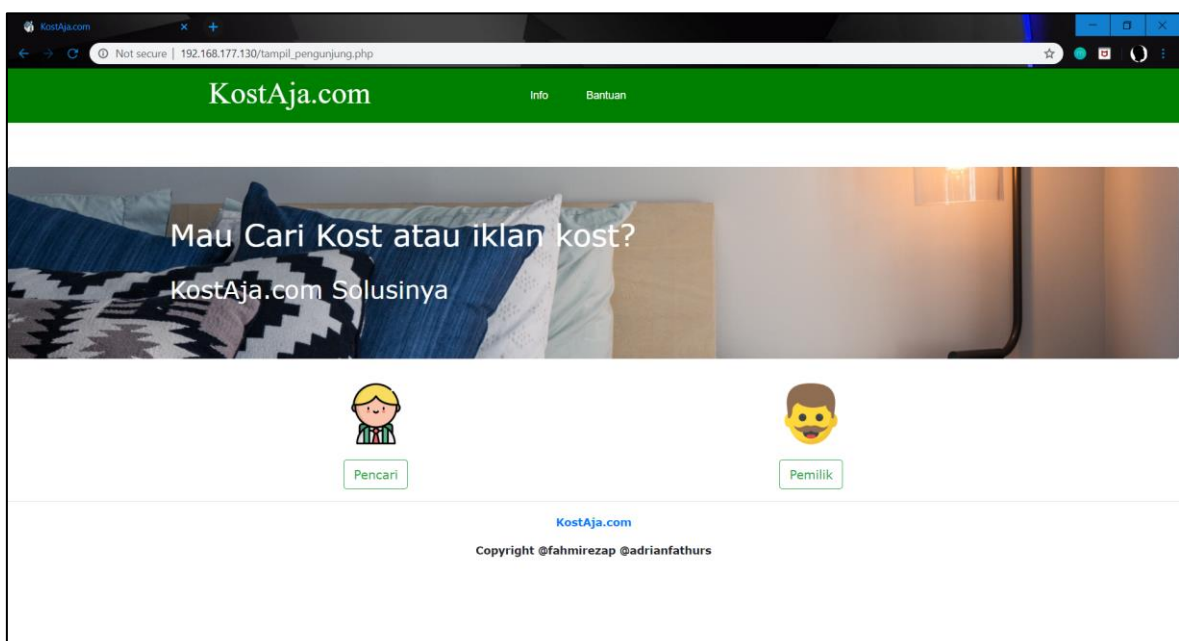
Maksud dari -ufahmi adalah user database yaitu fahmi dan -p123170068 maksudnya password yang dipakai adalah 123170068.

Setelah semua tahapan selesai kita dapat langsung mengeceknya dengan membuka browser dan mengetikkan IP-SERVER:8000. Untuk hasil implementasi dapat dilihat pada bagian Hasil Implementasi Proyek Kedua.

2.5 Hasil Implementasi

2.5.1 Hasil Implementasi Proyek Pertama

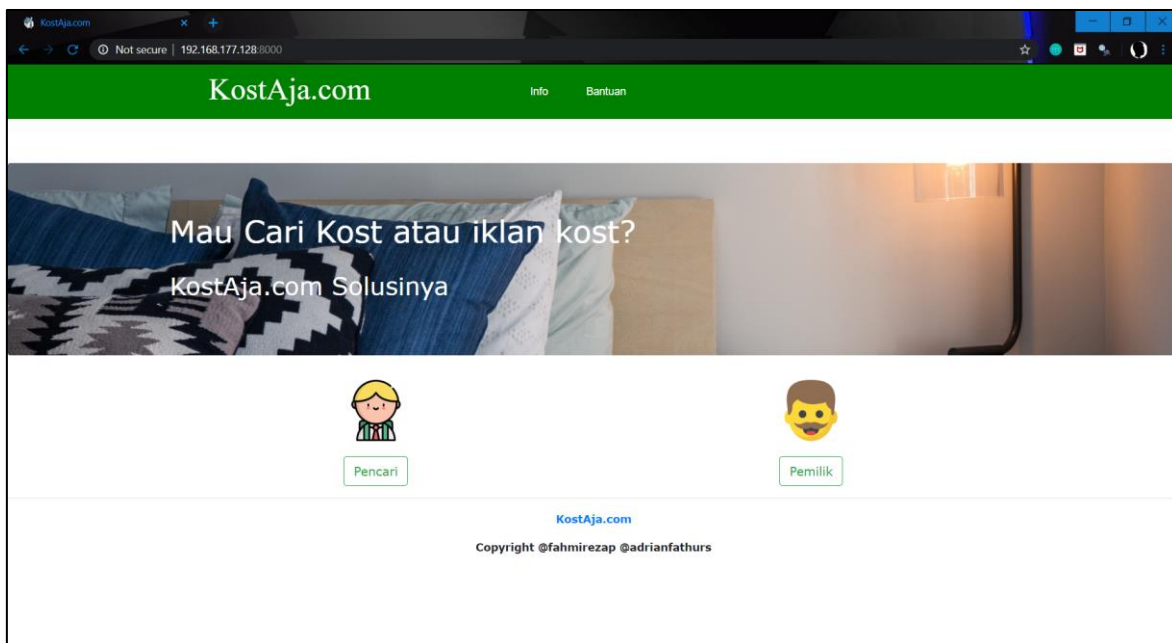
Setelah tahap implementasi proyek pertama selesai, hasil dari hosting website menggunakan LAMP pada Ubuntu seperti pada **Gambar 2.13** dibawah ini:



Gambar 2.13 Hasil website yang sudah dihosting dengan LAMP

2.5.2 Hasil Implementasi Proyek Kedua

Setelah tahap implementasi proyek kedua selesai, hasil dari pembuatan *dockerfile* dapat dilihat pada **Gambar 2.14** berikut ini:

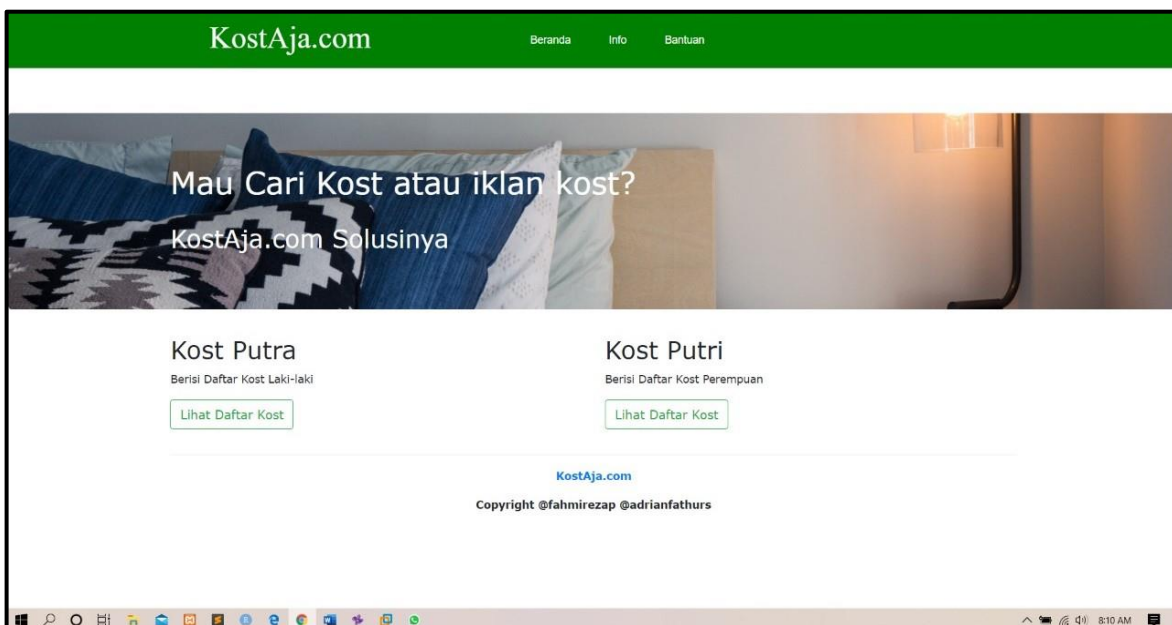


Gambar 2.14 Hasil dari pembuatan *dockerfile*

2.6 Pengujian Singkat

2.6.1 Pengujian Singkat Proyek Pertama

Pada proyek pertama ini sekaligus menjadi solusi kemudahan proses bisnis bagi pemilik *kost* dan pencari *kost* agar menghemat waktu dan tenaga dengan adanya teknologi *cloud computing* ini. Pada **Gambar 2.15** dan **Gambar 2.16** dibawah ini terlihat bahwa terdapat



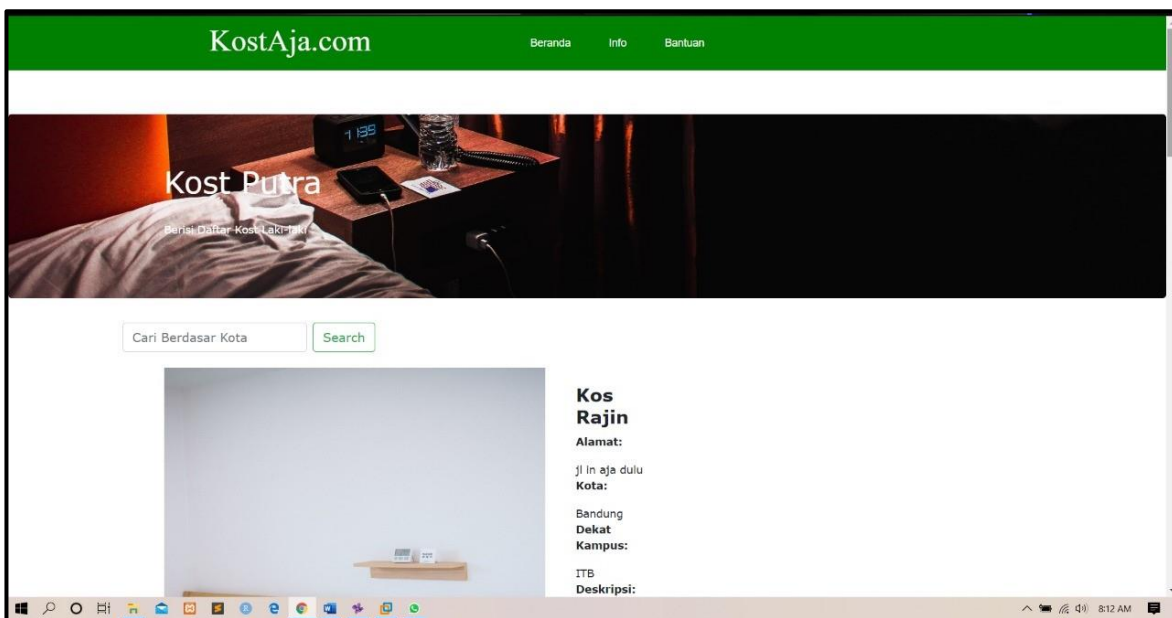
Menu dan fitur untuk Pencari kost agar bisa berkomunikasi dengan pemilik *kost* lewat pencarian *kost*.

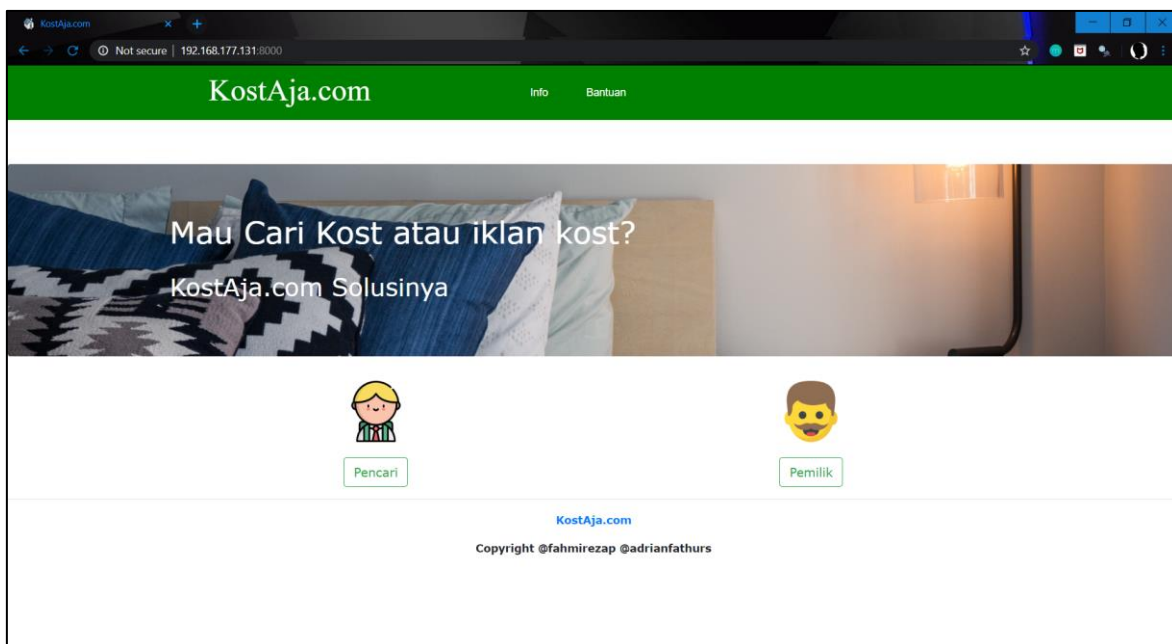
Gambar 2.15 Hasil website yang sudah dihosting dengan LAMPP

Gambar 2.16 Hasil website yang sudah dihosting dengan LAMPP

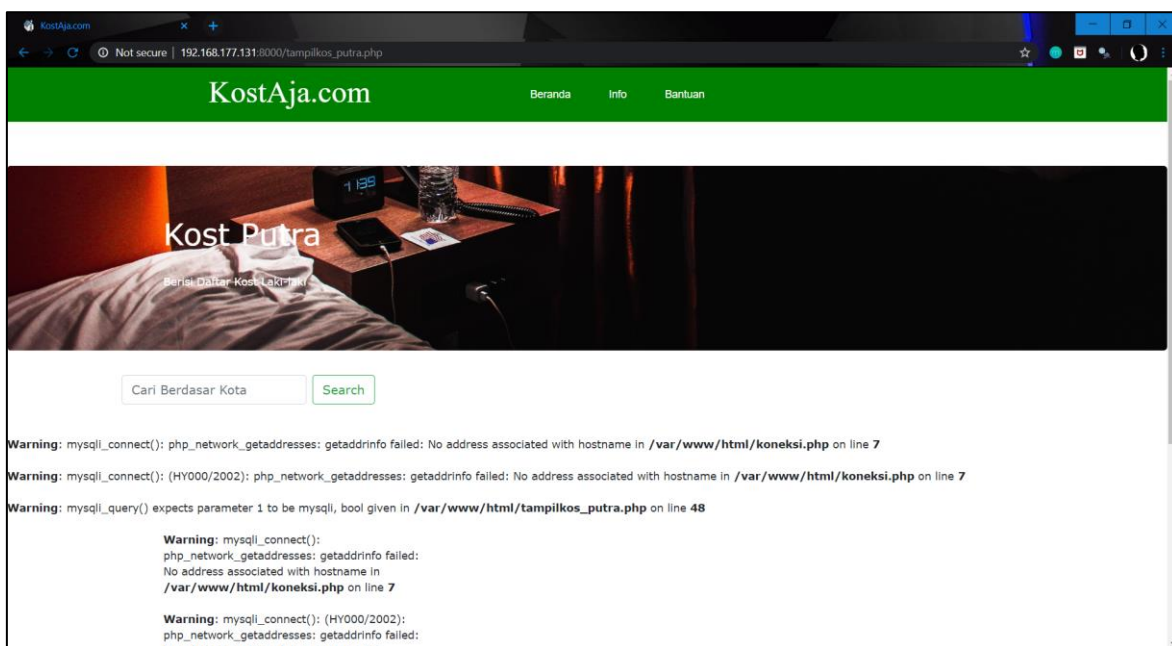
2.6.2 Pengujian Singkat Proyek Kedua

Pada proyek kedua yaitu pembuatan Dockerfile, saat mencoba menjalankan Dockerfile di perangkat yang sama, website dan databasenya berjalan dengan normal. Namun saat kami push Dockerfile ke Dockerhub dan kemudian mencoba melakukan pull menggunakan VM lain website nya bisa tampil namun database nya tidak bisa terkoneksi, sehingga masih ada *error* dan kendala dalam database dari Dockerfile tersebut ketika dijalankan di perangkat lain. Untuk hasilnya dapat di lihat pada **Gambar 2.17** dan **Gambar 2.18** berikut ini:





Gambar 2.17 Hasil running dockerfile di VM lain



Gambar 2.18 Hasil running dockerfile di VM lain

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, April, 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.	Pengerjaan Tugas 1(LAMPP)										
4.	Pembuatan Laporan Submisi										
5.	Pengerjaan Tugas 2 (Docker)										
6.	Melanjutkan pembuatan laporan										
7.	Revisi										
8.	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	Adib
2.	Latar Belakang Masalah	Adib
3.	Agenda Pengerjaan Proyek	Fahmi
4.	Pembuatan Tugas 1 (LAMPP)	Fahmi
5.	Pembuatan Tugas 2 (Docker)	Fahmi
6.	Pengujian Singkat	Adib
7.	Pembuatan Laporan	Adib dan Fahmi

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan permasalahan yang ada yaitu pembuatan sistem informasi kost dengan Ubuntu LAMPP beserta pembuatan dockerfile nya, untuk pembuatan sistem informasi kost dengan Ubuntu LAMPP dapat berjalan dengan baik dan pengujian sistem juga sesuai dengan yang diharapkan. Sementara untuk pembuatan Dockerfile berjalan dengan baik namun saat pengujian dilakukan menggunakan VM/perangkat lain terjadi kendala karena database tidak mau melakukan koneksi, jika pengujian dilakukan dengan perangkat yang sama database bisa terkoneksi. Pembagian tugas terlaksana dengan baik dan sesuai waktu yang ditentukan.

4.2 Saran

Saran untuk pengerjaan proyek ini yaitu dalam pembuatan Dockerfile, untuk pencarian referensi pembuatan sebaiknya dicari dengan teliti referensi yang akan digunakan karena harus disesuaikan juga dengan proyek web yang akan kita buat dockerfile.

DAFTAR PUSTAKA

1. Asrofil, M. S., Yuda, A. E., Widiyanto, H., & Baskara, D. B. (2020). Docker Salah Satu Platform yang Dibangun Berdasarkan Teknologi Container. *Lomba Karya Tulis Ilmiah*, 1(1), 145-153.
2. Sudaryono, S., Aryani, D., & Ningrum, I. T. (1978). Cloud Computing: Teori Dan Implementasinya Dalam Dunia Bisnis Dan Pemasaran. *Creative Communication and Innovative Technology Journal*, 5(2), 145-167.
3. Kavis, M. J., Erl, T., Puttini, R., Mahmood, Z., Dadmehr, M. D., Nait-Ali, A., ... & Sweet, L. H. (2013). Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)
4. denmasyarikin, 2019, *Dasar Penggunaan Docker*, Medium, dilihat 26 April 2020, <<https://medium.com/@denmasyarikin/dasar-penggunaan-docker-f80d9602186c>>
5. docker, 2019, *Install Docker Engine on Ubuntu*, Docker, dilihat 26 April 2020, <<https://docs.docker.com/engine/install/ubuntu/>>

