

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING  
LAPORAN PROYEK AKHIR**

**APLIKASI ASSIGNMENT PRAKTIKUM BERBASIS WEB MENGGUNAKAN  
UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA**



**DISUSUN OLEH:**

**NAMA ANGGOTA : M.ALFA RIZA 123170027**  
**LABIBUL UMAM ALMARBA'I 123170092**  
**KELAS : B**  
**ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.KOM.**  
**MUHAMMAD IMAM ALFATAH**

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

## HALAMAN PENGESAHAN

### APLIKASI ASSIGNMENT PRAKTIKUM BERBASIS WEB MENGGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

M.Alfa Riza

123170027

Labibul Umam Almarba'i

123170092

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

## **KATA PENGANTAR**

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul Aplikasi assignment praktikum berbasis web menggunakan ubuntu lampp dan proses pembuatan dockerfilenya. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten dosen 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, 30 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 .....	3
<b>BAB II ISI DAN PEMBAHASAN .....</b>	<b>4</b>
2.1 Komponen yang Digunakan .....	4
2.1.1 Komponen pada Proyek Pertama .....	4
2.1.2 Komponen pada Proyek Kedua .....	6
2.2 Rancangan Arsitektur <i>Cloud Computing</i> .....	8
2.3 Parameter dan Konfigurasi .....	10
2.4 Tahap Implementasi .....	12
2.5 Hasil Implementasi .....	21
2.5.1 Hasil dan Pengujian Lampp .....	21
2.5.2 Hasil dan Pengujian Docker .....	26
<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

### 1.2 Latar Belakang Proyek Akhir

*Cloud Computing* merupakan istilah yang sering kita dengar beberapa tahun belakangan, karena pemanfaatannya yang sangat luas dan sering kita gunakan dalam kehidupan sehari-hari, *cloud computing* sendiri merupakan pengembangan dari penyimpanan konvensional secara bertahap, sehingga hari ini kita dapat mendengar istilah tersebut. Manfaat dari *cloud computing* diantaranya, dapat diakses dimana saja selama ada internet, hal ini senada dengan pengguna internet yang banyak di dunia.

Beberapa tahun terakhir konsep *Cloud Computing* sudah banyak menarik minat dunia industri dan IT. Solusi berbasis cloud sepertinya menjadi kunci bagi organisasi IT yang mempunyai masalah keterbatasan anggaran (Teng & Magoules, 2010). *Cloud Computing* merupakan paradigma yang baru dalam komputasi terdistribusi menyajikan banyak ide, konsep, teknologi, dan tipe arsitektur yang disajikan secara *service-oriented*. Menurut Foster *Cloud Computing* adalah “Paradigma komputasi terdistribusi dalam skala yang besar yang dilatar belakangi oleh faktor ekonomi, yang mana berisi kumpulan dari virtualisasi abstrak, skalabilitas yang dinamis, pengaturan kekuatan komputasi, tempat penyimpanan, platform, dan layanan yang dapat diakses sesuai dengan kebutuhan oleh pelanggan eksternal melalui media Internet” (Foster et al., 2008).

Mahasiswa saat ini tidak dapat hidup jauh dari Internet. Melalui program seperti *facebook*, *twitter*, *instagram*, dan *gmail*, mahasiswa sudah terbiasa untuk menggunakan layanan teknologi berbasis *cloud computing* (Ercan, 2010). Oleh sebab itu mahasiswa berharap untuk dapat mengakses layanan teknologi digital di lingkungan kampus dimanapun dan kapanpun, termasuk layanan *cloud* yang mendukung media sosial. Sebagai tambahan, ada beberapa riset yang menunjukkan bahwa solusi berbasis *cloud* sangat efektif untuk mendukung pembelajaran yang kooperatif dan kolaboratif (Thorsteinsson, 2010).

Judul projek yang akan kami kerjakan mengenai *1cena assignment* praktikum, berbasis web menggunakan Ubuntu LAMP dan proses pemuatan *dockerfile*nya, di zaman sekarang dalam pengumpulan tugas, terkadang kita perlu bertemu dengan asisten praktikum ataupun dosen untuk mengumpulkan sebuah tugas, namun karena beberapa alasan mengumpulkan tugas secara konvensional tidak efektif, oleh karena itu pengumpulan tugas menggunakan *cloud computing* sangat efektif, karena kita tidak perlu bertemu dengan

asisten praktikum ataupun dosen di tempat, kita dapat mengumpulkan tugas dimana saja, dan kapan saja sesuai batas waktu yang telah ditentukan.

Rancangan proyek akhir kami berupa hosting lokal menggunakan ubuntu Lampp, dimana semua fungsi berjalan dengan lancar. Software yang kami gunakan adalah VMWare Workstation 15, ubuntu, putty. Hardware yang digunakan laptop dengan spesifikasi 4 GB ram, intel i5 gen 8, di bagian bab 2 akan dibahas mengenai instalasi, konfigurasi, testing.

## 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 Aplikasi Assignment Praktikum yang telah dibuat sebelumnya dengan menggunakan Ubuntu 20.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Mengimplementasikan arsitektur *cloud computing* untuk Aplikasi Assignment Praktikum yang telah dibuat sebelumnya dengan menggunakan Ubuntu 20.4 dan Dockerfile (Apache 8, PHP 7.1, MySQL 5.2).

## 1.3 Manfaat Proyek Akhir

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

1. Sistem Monitoring yang telah dirancang untuk penugasan praktikum dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.
2. Pengguna tidak perlu mempermasalahkan *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

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

1. Menganalisis kebutuhan dari aplikasi Assignment Praktikum untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Menentukan konfigurasi yang tepat untuk pengaturan Ubuntu LAMP dan Dockerfile 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.
3. Merancang topologi *cloud computing* untuk masing-masing proyek 3skenario.
4. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga aplikasi Assignment praktikum bisa maksimal.
5. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi.
6. Menghasilkan Aplikasi pengumpulan tugas praktikum 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 “Aplikasi Assignment Praktikum menggunakan 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 7.3 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 pengguna dan data tugas yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaannya ialah seluruh mahasiswa dan asisten praktikum. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet universitas masing-masing . Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang berkepentingan.
4. Sistem tersebut juga harus dapat digunakan sewaktu-waktu dan ditargetkan memiliki nilai *uptime* SLA 99,9% sehingga diperlukan minimal dua buah Ubuntu Server, satu sebagai *primary server* dan salah satunya sebagai *backup server*. Mekanisme untuk peralihan antar *server* membutuhkan *proxy server* dan semuanya menggunakan sistem operasi Ubuntu.

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



**Tabel 2.1** Spesifikasi VM *cloud computing* untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.43.2/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.43.139	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.43.139	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	4GB	Alokasi RAM untuk <i>guest OS</i>
6.	Disk Info	20 GB	Alokasi <i>storage</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

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Dukungan 5cena pemrograman yang digunakan oleh 5cena <i>assignment</i> praktikum.
		MySQL	Dukungan penyimpanan yang digunakan oleh 5cena <i>assignment</i> praktikum.
2.	Putty	PuTTY 0.73	Dukungan file sharing yang digunakan untuk memindahkan proyek <i>assignment</i> praktikum.
4.	SSH	OpenSSH	Dukungan untuk melakukan <i>remote server</i>

### 2.1.2 Komponen pada Proyek Kedua

Untuk membangun “Aplikasi Assignment Praktikum menggunakan Dockerfile” yang berbasiskan konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

2. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.3 dan bahasa HTML dengan versi minimal 4.0.
3. Penyimpanan *database* pada projek ini menggunakan image dari docker, sehingga kita tidak perlu *apache, mysql, phpmyadmin*.
4. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaannya ialah seluruh mahasiswa dan asisten praktikum. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet universitas masing-masing. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang berkepentingan.
5. Sistem tersebut juga harus dapat digunakan sewaktu-waktu dan ditargetkan memiliki nilai *uptime SLA* 99,9% sehingga diperlukan minimal dua buah Ubuntu Server, satu sebagai *primary server* dan salah satunya sebagai *backup server*. Mekanisme untuk peralihan antar *server* membutuhkan *proxy server* dan semuanya menggunakan sistem operasi Ubuntu.

Berdasarkan penjelasan poin-poin tersebut, untuk komponen utama penyusun *cloud computing* yang dibutuhkan dapat disimpulkan dalam bentuk tabel pada **Tabel2.3**:

**Tabel 2.3** Spesifikasi VM *cloud computing* untuk proyek kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.43.2/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.43.139	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.43.139	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	4GB	Alokasi RAM untuk <i>guest OS</i>

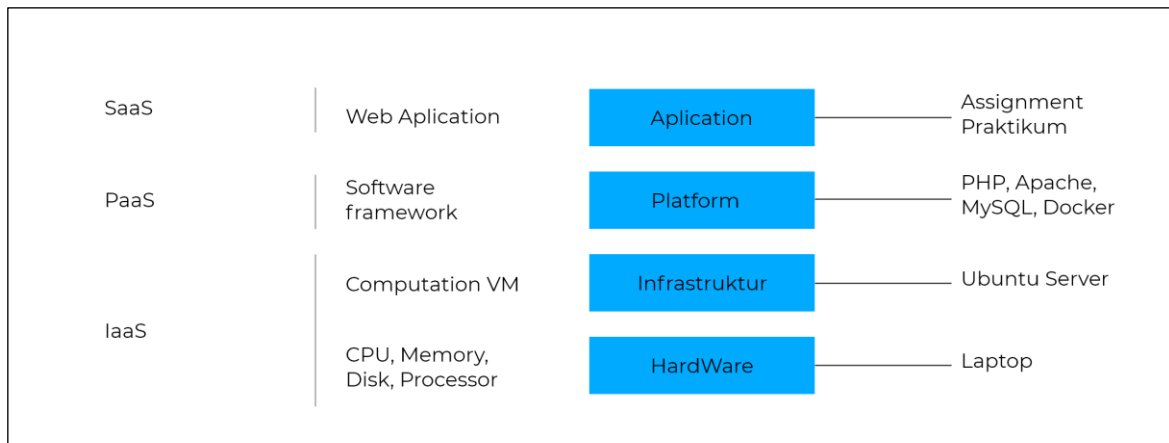
Spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut dan *tools* yang akan digunakan adalah sebagai mana pada **Tabel 2.4**:

**Tabel 2.4** Spesifikasi Ubuntu OS untuk proyek kedua

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Dukungan 8cenar pemrograman yang digunakan oleh 8cenar <i>assignment</i> praktikum.
		MySql	Dukungan penyimpanan yang digunakan oleh 8cenar <i>assignment</i> praktikum.
2.	Putty	PuTTY 0.73	Dukungan file sharing yang digunakan untuk memindahkan proyek <i>assignment</i> praktikum.
3.	Git	Git 2.17.1	Dukungan <i>control version</i> yang digunakan untuk <i>clone</i> proyek dari GitHub <i>repository</i>
4.	SSH	OpenSSH	Dukungan untuk melakukan <i>remote server</i>
5.	Docker Engine	Docker Server 19.03	<i>Docker engine server</i> yang digunakan dalam menjalankan container.
		Docker Client 19.03	<i>Docker engine client</i> yang digunakan untuk berkomunikasi dengan <i>docker server</i> .
6.	Docker Compose	Docker Compose 1.17	Dukungan alat untuk mendefinisikan dan menjalankan <i>multi-container</i>

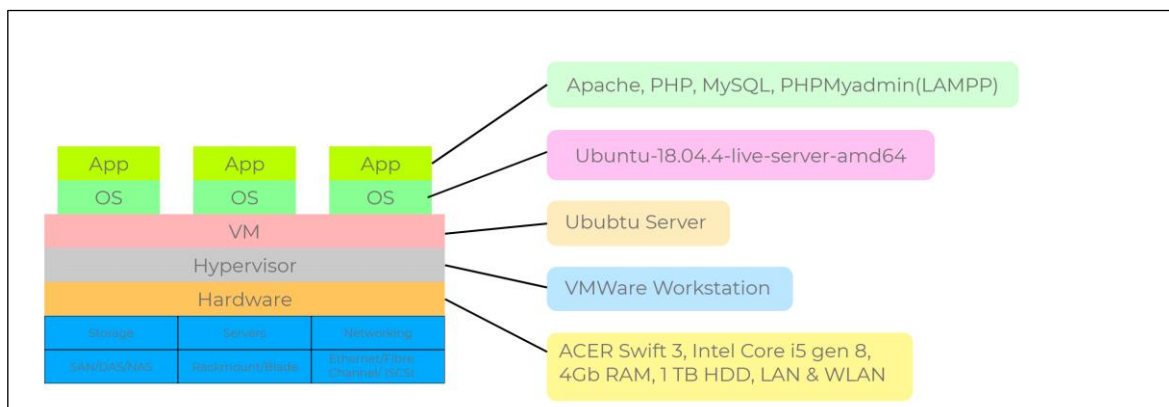
## 2.2 Rancangan Arsitektur Cloud Computing

Pada proyek akhir ini memanfaatkan arsitektur SaaS, PaaS, dan IaaS. Untuk SaaS digunakan untuk menjalankan *assignment* praktikum, untuk PaaS digunakan untuk tempat berjalannya Ubuntu, dan IaaS untuk hardware laptop menjalankan virtualisasi menggunakan VMWare Workstation. Untuk *illustrasi* dapat dilihat pada **Gambar 2.1** berikut :



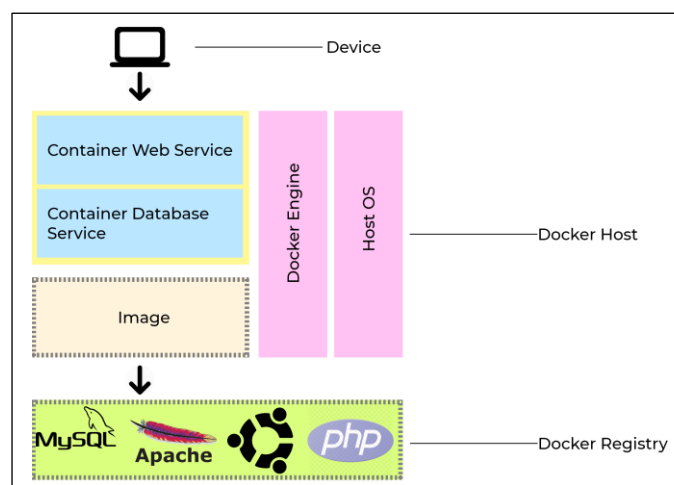
**Gambar 2.1** Arsitektur SaaS, PaaS, dan IaaS

Pengerjaan proyek ini menggunakan layanan hosting local dengan *virtual machine* yang berjalan di laptop. *Virtual machine* sendiri digunakan untuk merepresentasikan kondisi sebenarnya. Arsitektur dapat dilihat pada **Gambar 2.2**



**Gambar 2.2** Arsitektur *virtual machine*

Untuk tugas 2, pembuatan docker kami memanfaatkan *docker* untuk membungkus aplikasi dan *dependency* yang di butuhkan, arsitektur *docker* dapat dilihat pada **Gambar 2.3**.



**Gambar 2.3** Arsitektur docker

## 2.3 Parameter dan Konfigurasi

Ada beberapa konfigurasi yang harus di persiapkan, yang pertama *apache* untuk menjalankan *web server* sehingga dapat berjalan. Parameter untuk apache dapat dilihat pada **Modul 2.1**.

```
$ sudo apt install apache2
```

Sudo = untuk mengeksekusi dengan hak root.  
Apt = package pada ubuntu.  
Install apache2 = untuk menginstal apache

### Modul 2.1 instalasi apache

Untuk mengelola *database* kami menggunakan *mysql* untuk *database management system*. Untuk menginstall *mysql* dapat dilihat **Modul 2.2**.

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

```
$ sudo mysql_secure_install
```

Mysql-server = aplikasi untuk mysql  
mysql\_secure\_install = untuk menjalankan proses instalasi

### Modul 2.2 instalasi mysql

Selain menginstal *apache* dan *mysql* kami juga menggunakan php untuk Bahasa pemrograman, yang kebetulan kami menggunakan framework *codeigniter*. Untuk menginstall php dapat dilihat **Modul 2.3**

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

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

Php, libapache2-mod-php, php-mysql = paket aplikasi  
Nano = membuka nano ubuntu, kemudian ditulis <?php phpinfo() ?>

### Modul 2.3 instalasi php

Kami juga menginstall *phpmyadmin* untuk layanan manajemen *database* MySQL berbasis web gui. Instalasi dapat dilihat **Modul 2.4**

```
$ sudo apt install phpMyAdmin php-mbstring php-gettext
PhpMyAdmin, php-mbstring, php-gettext = paket instalasi
```

### Modul 2.4 instalasi phpMyAdmin

Untuk tugas 1 cukup yang kami sebutkan diatas, untuk tugas 2, kami menginstal *docker* dan *docker-compose*. Sebelum menginstall *docker* kita harus *set up repository*, namun hal ini dilakukan ketika kita pertama kali menginstall *docker*. *Set up repository* dapat dilihat pada **Modul 2.5**.

```
$ sudo apt update

$ sudo apt install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common

update = untuk mengupdate dependency repository pada system

apt-transport-https, ca-certificates, curl, gnupg-agent, software-
properties-common = nama paket aplikasi
```

### Modul 2.5 set up repository

Selanjutnya menambahkan *key* GPG Docker. Instalasi penambahan *key* GPG dapat dilihat pada **Modul 2.6**.

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

Curl = command line tool untuk mentransfer data menggunakan protokol
jaringan.

Apt-key = manajemen daftar key untuk autentikasi package.

Add - = menambahkan key ke daftar trusted key, parameter ini bersifat
opsional.
```

### Modul 2.6 instalasi penambahan key GPG

Selanjutnya mengatur *repository* mana yang akan digunakan, untuk kali ini kami menggunakan *stable* sesuai yang direkomendasikan di dokumentasinya.

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

`add-apt-repository` : menambahkan PPA ke daftar sumber, sehingga Ubuntu tahu untuk mencari pembaruan dari PPA tersebut berasal dari sumber yang resmi.

`lsb_release` : command untuk menampilkan LSB (Linux Standard Base) dan Distribution Information

`stable` : versi repository yang akan diinstall.

### Modul 2.7 mengatur repository

Selanjutnya adalah instalasi *docker engine*, *docker* yang kami gunakan versi *community*, hal ini dilakukan setelah *set up repository* untuk instalasi dapat dilihat pada **Modul 2.8**.

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

Docker-ce = paket aplikasi untuk versi community
Docker-ce-cli = paket aplikasi docker cli
Containerd.io = paket aplikasi untuk container
```

### Modul 2.8 instalasi docker

Selain *docker* kami juga menginstall *docker-compose* untuk memudahkan dalam *build container* dan *image*, banyak *container* secara bersamaan. Instalasi dapat di lihat **Modul 2.9**.

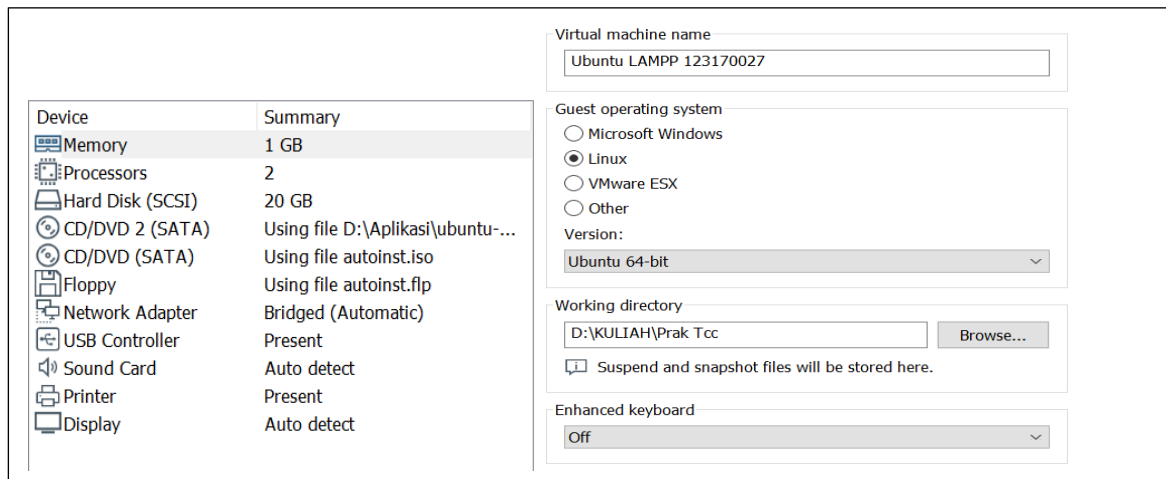
```
$ sudo curl -L \
    "https://github.com/docker/compose/releases/download/1.25.5/dockercompos
    e-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

### Modul 2.9 instalasi docker-compose

## 2.4 Tahap Implementasi

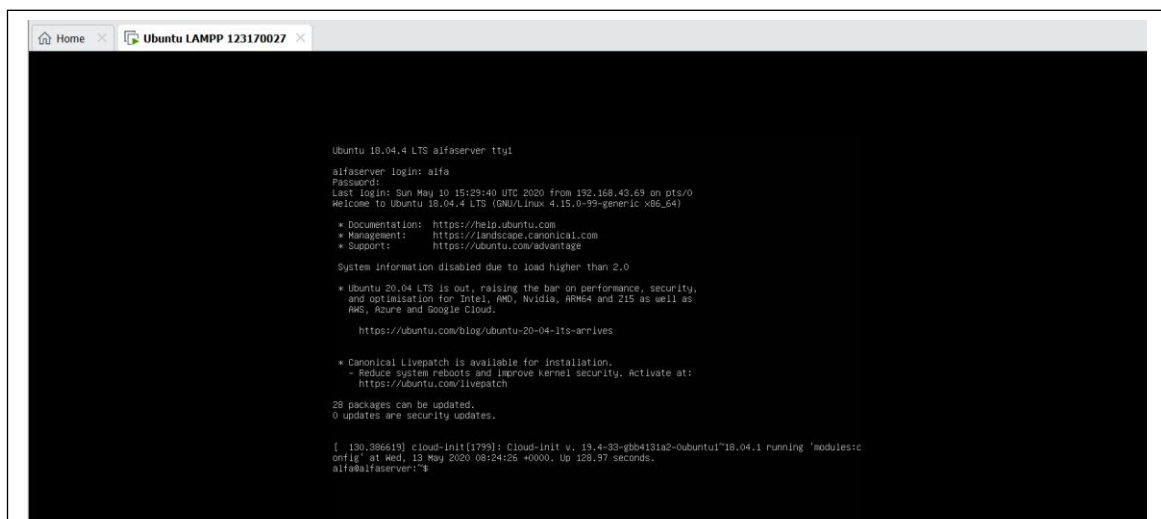
Terdapat beberapa tahapan dalam implementasi hosting local Ubuntu Lampp pada aplikasi *assignment* praktikum. Pertama membuat *virtual machine* dengan VMWare Workstation dengan konfigurasi pada **Gambar 2.4**.





**Gambar 2.4** Konfigurasi *virtual machine*

Mode *bridge* digunakan agar dapat berbagi dalam jaringan yang sama, setelah itu jalankan *virtual machine* dengan terlebih dahulu *login* dengan *aku* yang telah dibuat, maka akan mendapat tampilan seperti pada **Gambar 2.5**.



**Gambar 2.6** Tampilan *virtual machine*

Setelah itu, *remote server* dengan menggunakan PuTTY dibutuhkan beberapa konfigurasi, pertama kita masukkan alamat ip ubuntu *virtual machine* kita, kemudian login, setelah itu maka tampilan akan seperti pada **Gambar 2.7**

```

alfa@alfaserver: ~
Swap usage:          9%
Processes:           180
Users logged in:     1
IP address for ens33: 192.168.43.139
IP address for docker0: 172.17.0.1
IP address for br-c77dac36c6b: 172.18.0.1

* Ubuntu 20.04 LTS is out, raising the bar on performance, security,
and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
AWS, Azure and Google Cloud.

https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

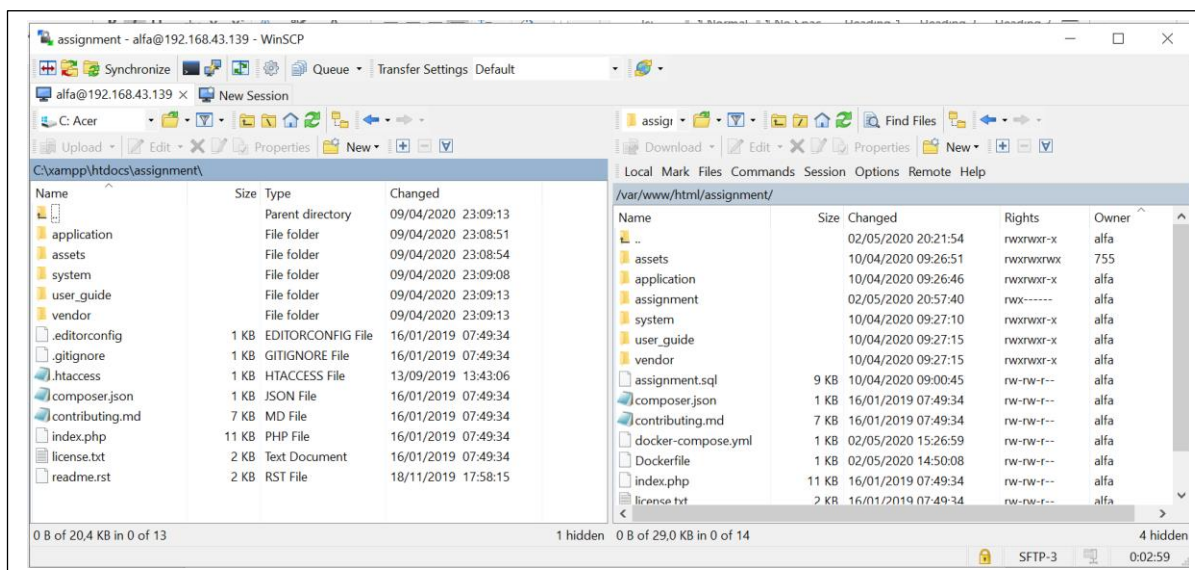
28 packages can be updated.
0 updates are security updates.

Last login: Wed May 13 08:23:56 2020
alfa@alfaserver:~$

```

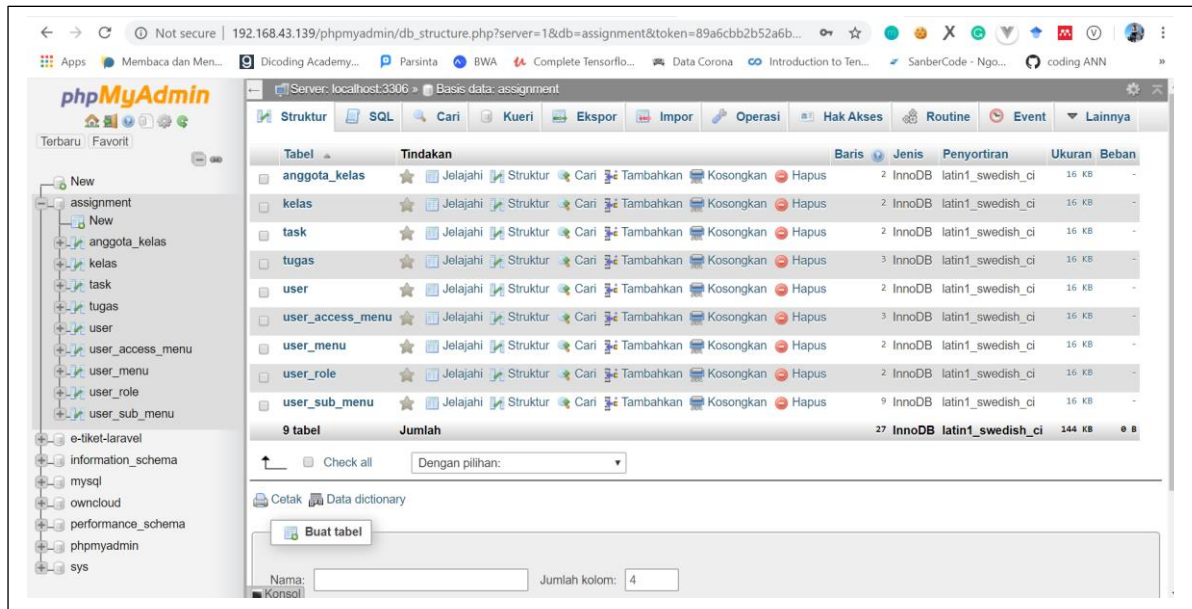
**Gambar 2.7** Tampilan PuTTY

Tahap selanjutnya adalah instalasi apache, php, mysql, phpMyAdmin yang sebelumnya sudah dibahas pada **Modul 2.1, Modul 2.2, Modul 2.3, Modul 2.4.** setelah selesai tahap instalasi, selanjutnya memindahkan folder proyek *assignment* praktikum ke direktori `var/www/html/` pada server ubuntu, untuk memudahkan pemindahan kami menggunakan WinSCP, sebelum memindahkan direktori pada WinSCP harus *login* terlebih dahulu dengan alamat ip dan akun *virtual machine*. Hasil dapat dilihat pada **Gambar 2.8**



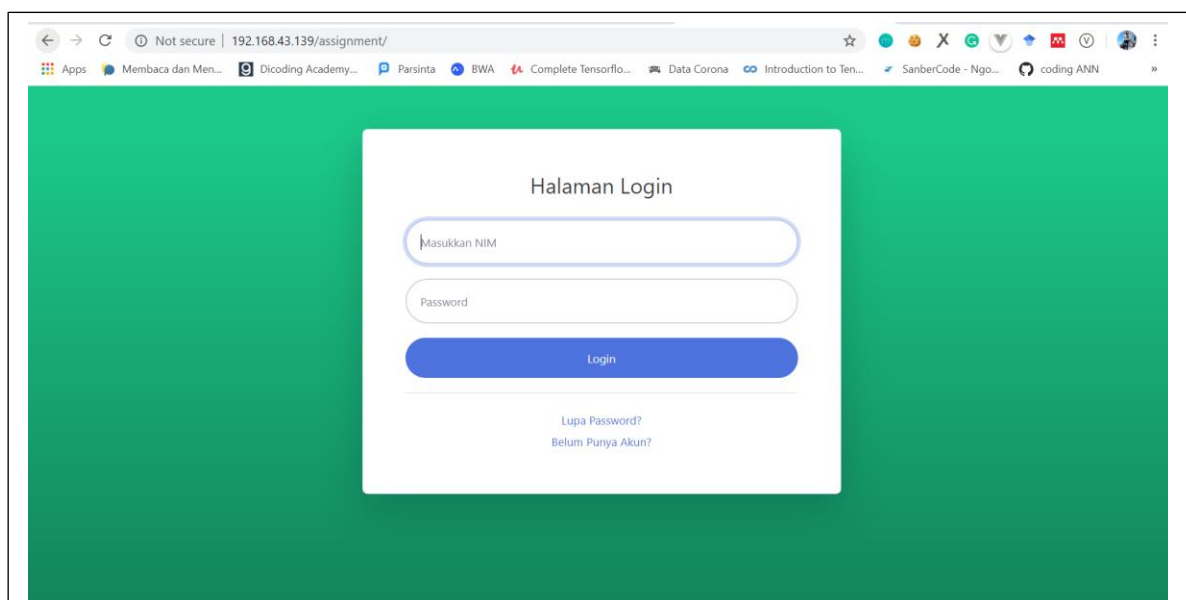
**Gambar 2.8** Tampilan WinSCP

Selanjutnya, kita *import database* agar aplikasi dapat berjalan sebagaimana mestinya, pertama kita siapkan *database* dengan format *.sql*, setelah itu buka *ip/phpmyadmin* lalu login, kemudian pilih *export* lalu pilih *database* yang dituju. Hasil dapat dilihat pada **Gambar 2.9**.



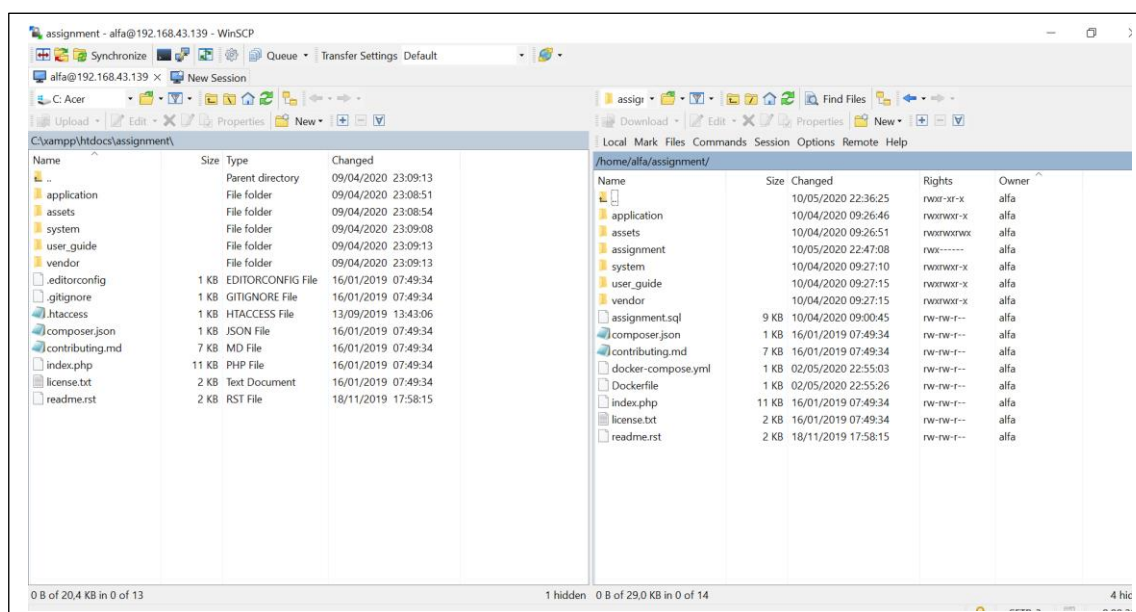
**Gambar 2.9** Tampilan phpmyadmin

Setelah berhasil maka aplikasi dapat dijalankan *ip/nama\_folder*, dalam hal ini berarti *192.168.43.139/assignment*. Hasil dapat dilihat pada **Gambar 2.10**.



**Gambar 2.10** Halaman Login

Selanjutnya pembuatan *dockerfile* untuk tugas 2, pembuatan dilakukan setelah *set up repository* dan instalasi *docker*, *docker-compose* yang dapat dilihat pada **Modul 2.5, Modul 2.6, Modul 2.7, Modul 2.8, Modul 2.9**. Pada implementasi *docker* kami menggunakan PuTTY untuk *remote server* dengan *virtual machine* yang sama, Langkah pertama yang dilakukan adalah memindah folder proyek *assignment* praktikum pada repository */home/alfa/*, tujuan diletakkan pada direktori ini, agar tidak tabrakan dengan tugas 1 dimana



masing-masing tugas ini memiliki *base\_url* yang berbeda, karena dalam *virtual machine* yang sama, setelah berhasil memindahkan folder maka akan menghasilkan seperti **Gambar 2.11**.

**Gambar 2.11** Tampilan winscp

Selanjutnya membuat *dockerfile* pada direktori */home/alfa/* yang berisi *script*, isi *script* dapat dilihat pada **Modul 2.10**.

```

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
//keterangan
FROM php:7.3-apache : merupakan versi image php:7.3-apache yang
digunakan.
upgrade : menginstall versi terbaru dari package yang dimiliki.
RUN : untuk mengeksekusi perintah.
EXPOSE : meng-expose PORT yang digunakan untuk mengakses projek.
a2enmod rewrite : command untuk mengaktifkan mod_rewrite dalam apache
chmod : command untuk memodifikasi akses dari suatu direktori
COPY : command untuk menyalin sutau file atau direktori
service apache2 restart : command untuk merestart apache server

```

### Modul 2.10 Script Dockerfile

Selanjutnya pembuatan *container*, dalam projek ini membutuhkan 2 *container* yang terdiri dari, *service web app* dan *service database*, untuk mempermudah pembuatan *container* kami memanfaatkan docker compose, pertama membuat nano docker-compose.yml, script docker compose dapat dilihat pada **Modul 2.11**.

```

version: '3.3'
services:
  web:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: web-kelompok3
    depends_on:
      - db
    volumes:
      - ./:/var/www/html
    ports:
      - 8000:80
  db:
    container_name: kel3db
    image: mysql:8.0
    command: --default-authentication-
plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: assignment
      MYSQL_USER: alfa
      MYSQL_PASSWORD: 123170027
    ports:
      - 6036:3306

```

//Keterangan

- version = versi dari docker-compose.yml.
- web = nama image yang akan di build.
- build = berisi proyek yang akan di build.
- context = merepresentasikan tempat direktori berada.
- dockerfile = nama file docker
- depends\_on = dependency terhadap container lain.
- volumes = menyimpan data meski container dihapus.
- Ports = port yang akan kita expose.
- db = nama image yang akan di build.
- container\_name = nama container yang akan dibuat.
- Image = image yang akan digunakan.
- Command = perintah yang dijalankan selama container dijalankan.
- Restart = akan merestart jika terjadi kesalahan.
- Environment = variable yang ada pada container

## Modul 2.11 Script docker-compose

Setelah membuat `docker-compose.yml`, selanjutnya adalah mengeksekusinya, perintah untuk mengeksekusi `docker-compose.yml` dapat dilihat pada **Modul 2.12**.

### Modul 2.12 Eksekusi docker-compose

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

//keterangan

Docker-compose up = untuk menjalankan docker-compose.

-d = untuk menjalankan perintah dibelakang layar.

```
assignment$ docker exec -i kel3db mysql -ualfa -p123170027 assignment <
./assignment.sql
```

//keterangan

Exec = untuk menjalankan perintah pada container yang sedang berjalan.

-i = untuk tetap mempertahankan STDIN terbuka meski tidak dilampirkan.

Kel3db = nama container yang dibuat pada `docker-compose.yml`

Mysql = perintah untuk menjalankan mysql

-ualfa = username database pada `docker-compose.yml`

-p123170027 = password database pada `docker-compose.yml`

Assignment < ./assignment = merupakan direktori dan nama database yang akan diekspor.

Setelah itu, jika kita jalankan maka akan *error*, karena *database* belum ada isinya, maka dari itu Langkah selanjutnya adalah *export database*, namun sebelum itu kita harus memastikan terlebih dahulu *assignment.sql* berada pada direktori `/home/alfa/assignment`. Perintah untuk *export database* dapat dilihat pada **Modul 2.13**

### Modul 2.13 Perintah eksport database

Setelah mengeksport *database*, maka Langkah selanjutnya adalah mengubah konfigurasi pada projek kita yang kita sesuaikan dengan *container* yang dijalankan. Pada `database.php` diubah menjadi seperti **Modul 2.14**.

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'db',
    'username' => 'alfa',
    'password' => '123170027',
    'database' => 'assignment',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => TRUE,
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

#### Modul 2.14 Konfigurasi pada database.php

Selanjutnya jangan lupa mengubah *base\_url* pada config.php menjadi 192.168.43.139:8000, karena port tersebut yang kita definisikan pada docker-compose.yml untuk di ekpose. Selanjutnya mengubah konfigurasi pada config.php untuk *path* dapat dilihat pada **Modul 2.15**.

```

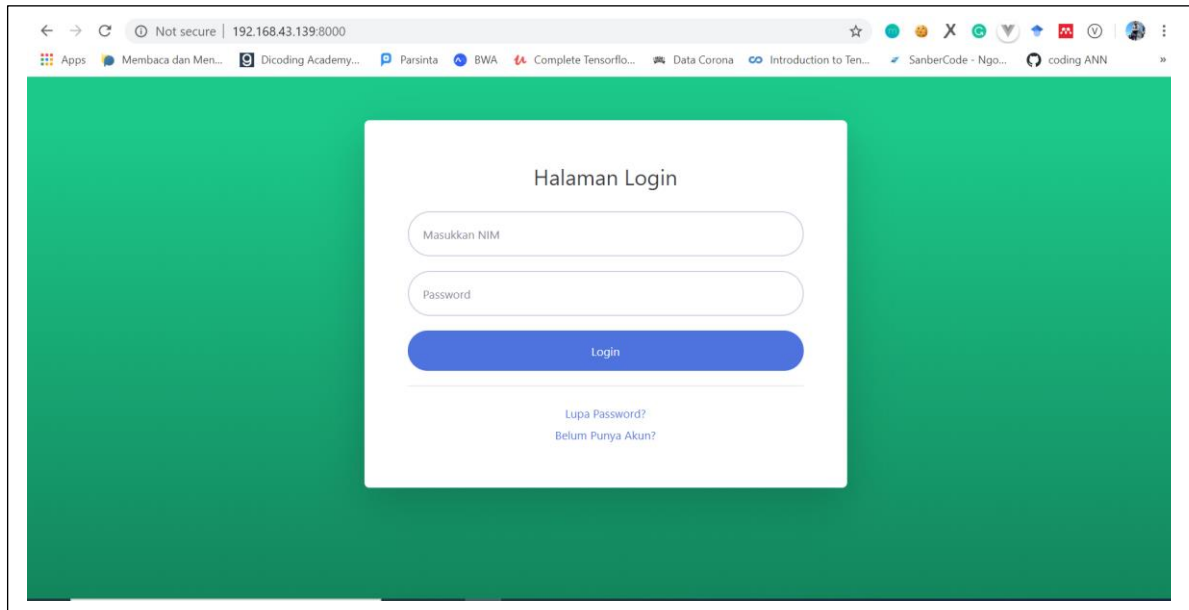
$config['sess_driver'] = 'files';
$config['sess_cookie_name'] = 'ci_session';
$config['sess_expiration'] = 7200;
$config['sess_save_path'] = sys_get_temp_dir();
$config['sess_match_ip'] = FALSE;
$config['sess_time_to_update'] = 300;
$config['sess_regenerate_destroy'] = FALSE;

```

#### Modul 2.15 Konfigurasi pada config.php

Langkah terakhir mencoba mengakses pada browser dengan alamat 192.168.43.139:8000, hasil dapat dilihat pada **Gambar 2.12**.





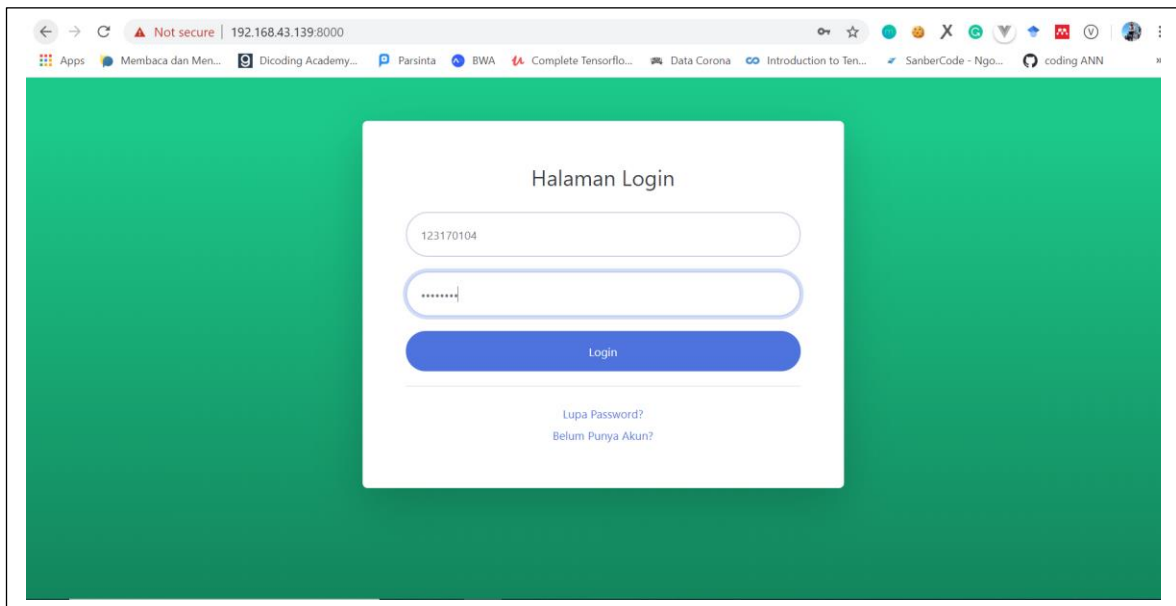
**Gambar 2.12** Tampilan hasil docker

## 2.5 Hasil Implementasi

Pada bagian ini akan kami bagi menjadi 2 bagian, yang pertama implementasi serta pengujian singkat pada Lampp dan yang kedua implementasi serta pengujian singkat pada Docker.

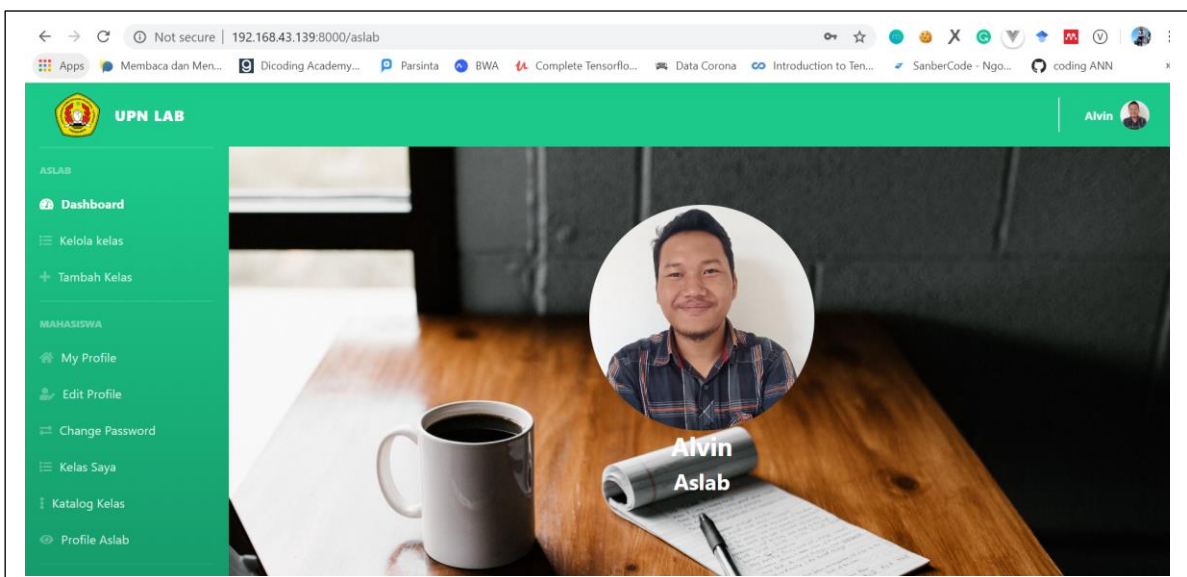
### 2.5.1 Hasil dan Pengujian Lampp

Pengujian pertama pada Lampp, yang perlu kita siapkan adalah *virtual machine* agar dapat diakses, halaman dapat diakses dengan ip/nama\_folder, yang dimana pada proyek ini menjadi 192.168.43.139/assignment, maka akan muncul halaman *login* seperti **Gambar 2.10**, setelah itu akan di uji secara singkat, yaitu *login* menggunakan akun yang sudah ada pada *database*. Akun yang akan digunakan adalah milik aslab dengan nim dan password seperti **Gambar2.13**.



**Gambar 2.13** Skenario 1

Jika berhasil *login*, dalam arti *nim* dan *password* benar maka akan di *redirect* ke *dashboard* aslab, yang hasilnya dapat dilihat pada **Gambar 2.14**.



**Gambar 2.14** Hasil 22cenario 1

Selanjutnya setelah berhasil *login*, ada beberapa fitur pada aslab, yang pertama akan kita coba adalah fitur buat kelas baru, fitur ini berguna untuk membuat kelas baru, dimana aslab menginputkan nama kelas, deskripsi, gambar kelas(optional, jika tidak mengupload

maka akan terisi oleh system secara *default*), dan token kelas(*autogenerate*), hasil pada ini dapat dilihat pada **Gambar 2.15**.

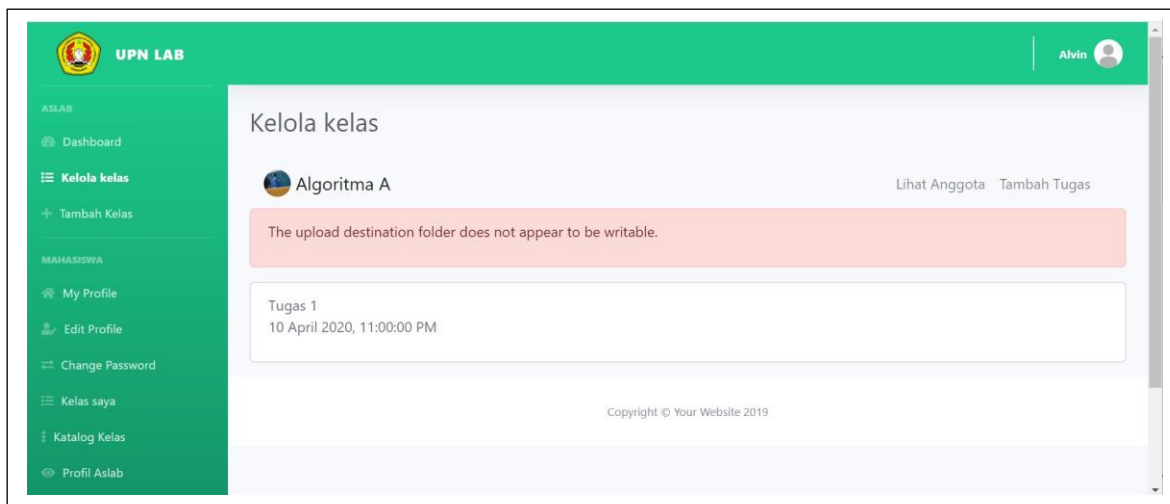
The screenshot shows the 'Tambah Kelas' (Add Class) interface in the UPN LAB system. The form contains the following data:

Field	Value
Nama Kelas	Praktikum Tcc
Deskripsi	Hari Kamis pukul 10.30 wib
Token	kwfd

The interface also features a sidebar with navigation links for ASLAB (Dashboard, Kelola kelas, Tambah Kelas) and MAHASISWA (My Profile, Edit Profile, Change Password, Kelas Saya, Katalog Kelas, Profile Aslab).

**Gambar 2.15** Skenario 2

Maka akan terjadi *error* mengenai hak akses folder, dimana *error* tersebut dapat dilihat pada **Gambar 2.16**.



**Gambar 2.16** Error scenario 2

Cara memperbaikinya yaitu dengan menjalankan perintah pada **Modul 2.16**.

```
$ sudo chmod -R 777 /var/www/html/assignment/
```

//keterangan

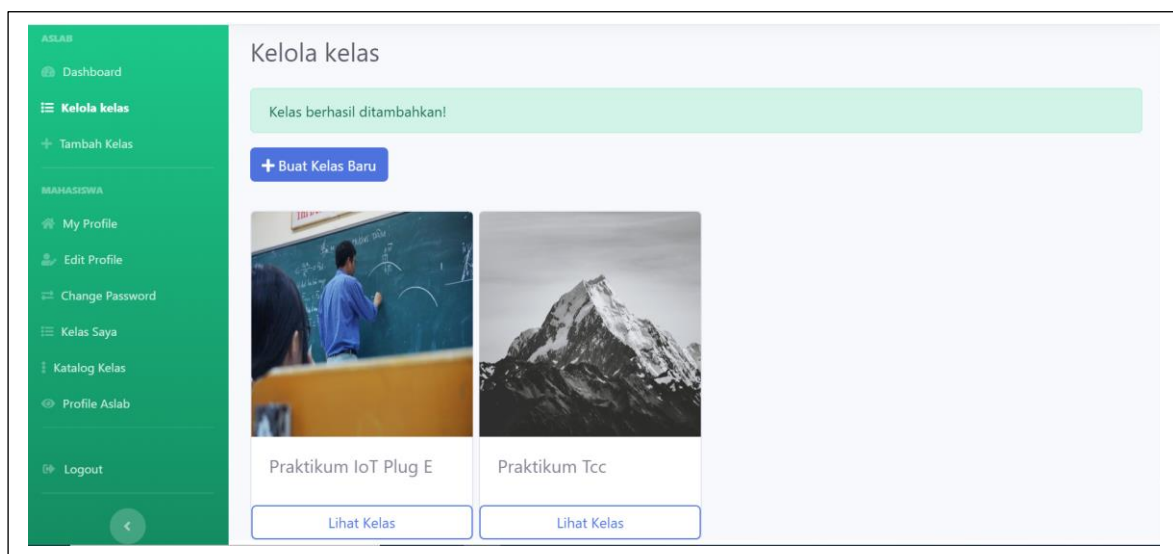
Chmod = untuk mengubah permission

777 = merupakan akses untuk melihat, mengubah.

/var/www/html/assignment/ = direktori tujuan

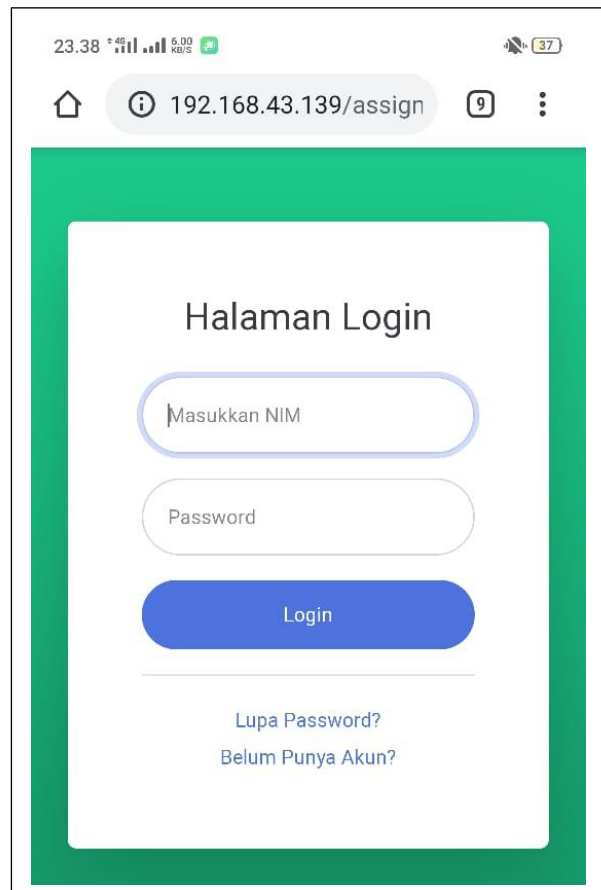
### Modul 2.16 Perintah ubah permission

Maka akan berhasil membuat kelas, dan hasilnya dapat dilihat pada **Gambar 2.17**



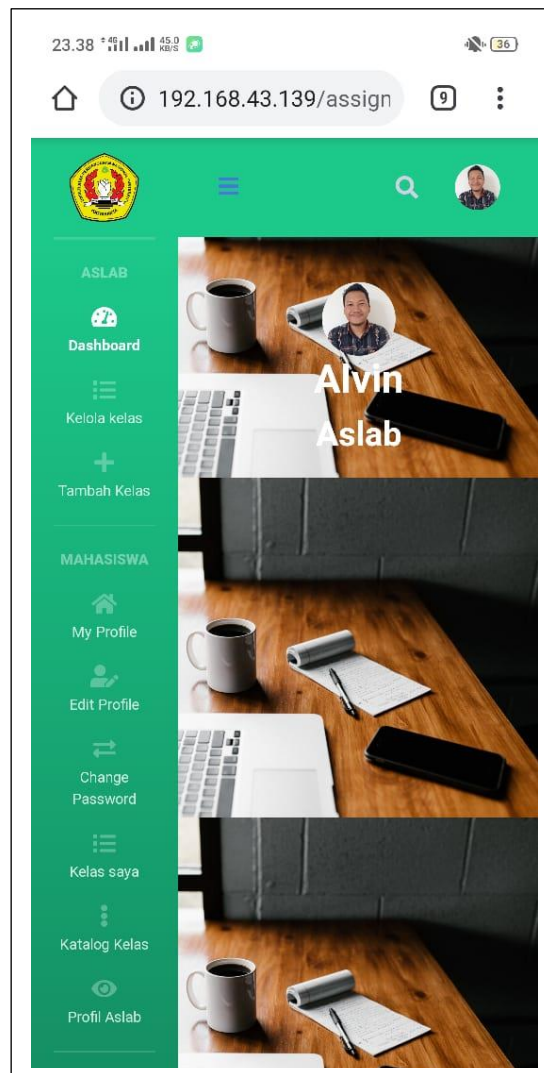
**Gambar 2.17** Hasil 24cenario 2

Untuk fitur lainnya, tidak kami tampilkan karena hampir sama dengan yang kami ujikan, Setelah ujicoba pada laptop, selanjutnya akan di ujicoba pada *smartphone*, hal ini dapat dilakukan karena mode yang digunakan mode *bridge*, dengan terlebih dahulu menyamakan jaringan, disini kami menggunakan *thetering* pada android. Gambar halaman login pada *smartphone* dapat dilihat pada **Gambar 2.21**.



**Gambar 2.21** ujicoba *smartphone*

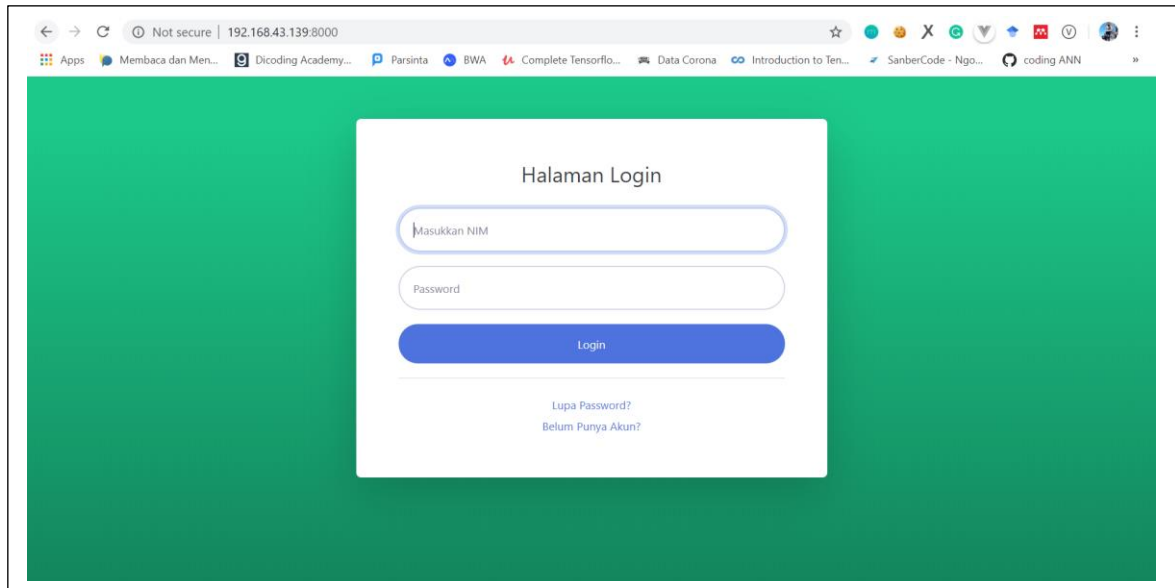
Setelah dapat menampilkan halaman *login*, maka ujicoba selanjutnya mencoba fitur *login*, untuk fitur yang akan di uji hanya *login*, selebihnya hasilnya akan sama pada laptop, karena prinsip berbagi pada jaringan yang sama, hasil setelah *login* dapat dilihat pada **Gambar 2.22**.



**Gambar 2.22** Hasil login

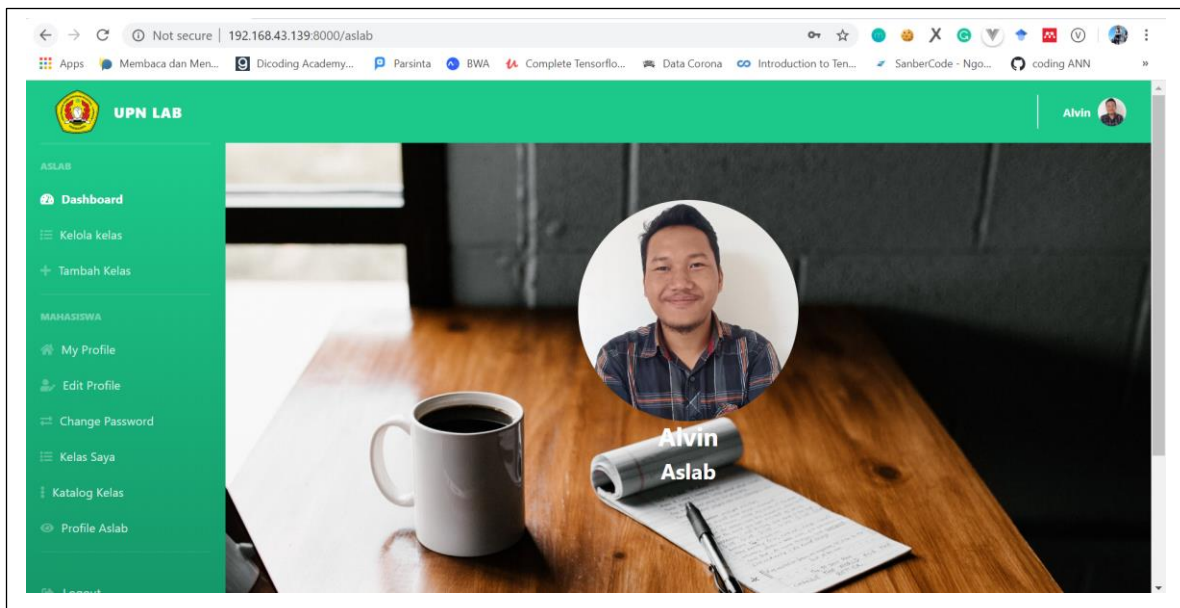
## 2.5.2 Hasil dan Pengujian Docker

Pada pengujian ini, pertama kita harus menjalankan *container* dengan perintah yang dapat dilihat pada **Modul 2.12**. setelah itu, kita uji dengan membuka browser dengan ip:8000, dengan projek kita 192.168.43.139:8000, halaman *login* docker dapat dilihat pada **Gambar 2.23**.



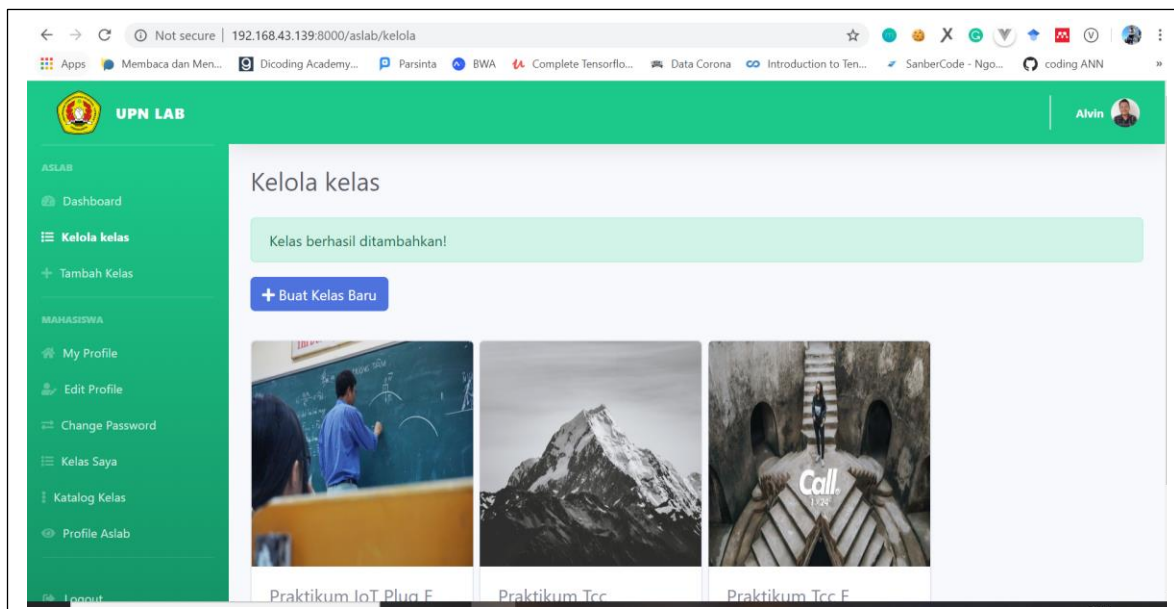
**Gambar 2.23** halaman login port 8000

Setelah berhasil menampilkan halaman *login*, kita cek apakah *login* dapat berjalan dengan baik, hasil pengujian dapat dilihat pada **Gambar 2.24**.



**Gambar 2.24** Dashboard

Selanjutnya, sama seperti ujicoba lamp, kita akan menguji *upload* file pada aplikasi *assignment* praktikum, kita akan menambahkan kelas baru, maka akan menghasilkan error sama seperti **Gambar 2.16**. Solusinya sama yaitu megubah *permission* seperti pada **Modul 2.16**. Maka fitur tambah kelas akan berhasil, dan hasilnya dapat dilihat pada **Gambar 2.25**.



**Gambar 2.25** Hasil tambah kelas

Setelah semua fitur berhasil maka Langkah selanjutnya adalah menguji pada *virtual machine* lain, disini kami membuat *virtual machine* baru dengan spesifikasi seperti **Tabel 2.5**.



**Tabel 2.5** Spesifikasi *virtual machine* uji

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 Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.232.2/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.232.139	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.232.139	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	4GB	Alokasi RAM untuk <i>guest OS</i>
6.	Disk Info	20 GB	Alokasi <i>storage</i> untuk <i>guest OS</i>

Selain itu kami juga menyiapkan spesifikasi ubuntu, sama seperti saat *virtual machine* pertama, namun pada kali ini kami tidak menginstall apache, php, phpmyadmin, lamp karena pada ujicoba kali ini akan menggunakan docker image untuk manajemen *database*. Spesifikasi dapat dilihat pada **Tabel 2.6**.

**Tabel 2.6** Spesifikasi Ubuntu

No.	Nama Parameter	Nilai	Keterangan
1.	Putty	PuTTY 0.73	Dukungan file sharing yang digunakan untuk memindahkan proyek <i>assignment</i> praktikum.
2.	Git	Git 2.17.1	Dukungan <i>control version</i> yang digunakan untuk <i>clone</i> proyek dari GitHub <i>repository</i>
3.	SSH	OpenSSH	Dukungan untuk melakukan <i>remote server</i>
4.	Docker Engine	Docker Server 19.03	<i>Docker engine server</i> yang digunakan dalam menjalankan container.
		Docker Client 19.03	<i>Docker engine client</i> yang digunakan untuk berkomunikasi dengan <i>docker server</i> .
5.	Docker Compose	Docker Compose 1.17	Dukungan alat untuk mendefinisikan dan menjalankan <i>multi-container</i>

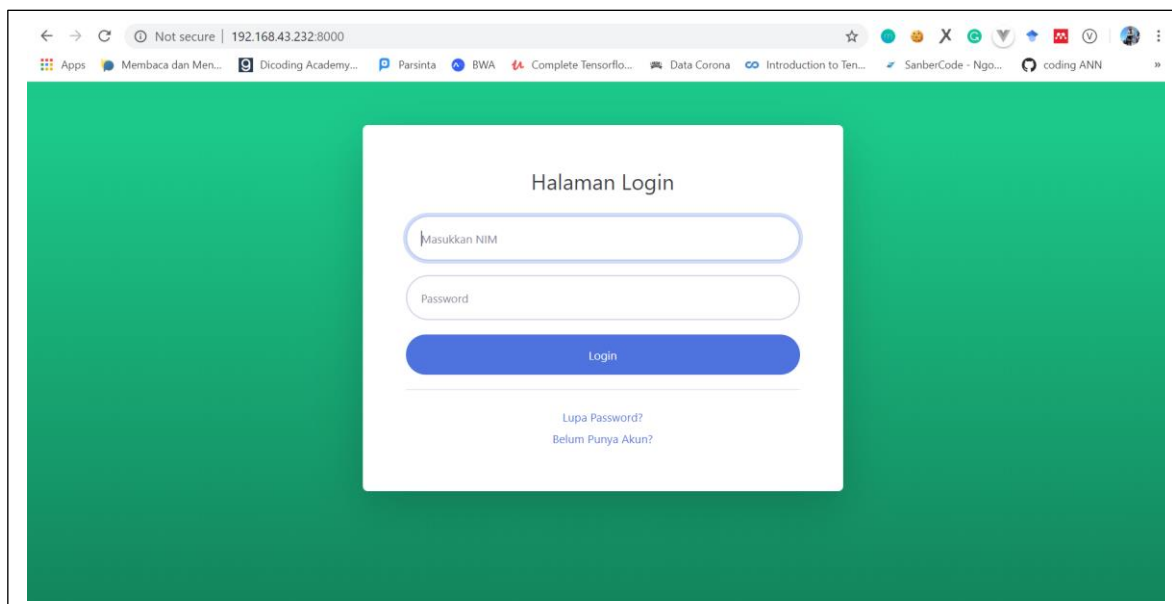
Pertama-tama kami mengupload *docker file* pada *virtual machine* ke github agar memudahkan mendapatkan file proyek, github yang digunakan adalah github AlfaRiza dengan nama repositori *assignment*, perintah untuk *clone* repo pada github dapat dilihat pada **Modul 2.17**.

```
$ git clone https://github.com/AlfaRiza/assignment.git
// Keterangan
git = merupakan perintah untuk git
clone link = perintah untuk clone repositori sesuai link
```

#### **Modul 2.17** Git clone

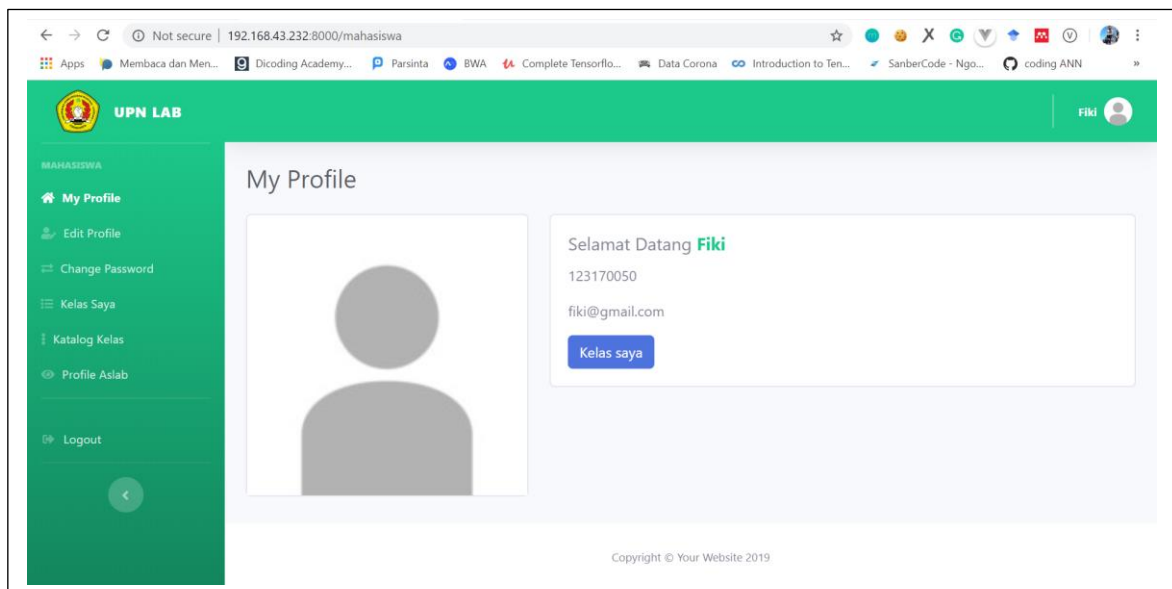
Setelah selesai, maka pada *virtual machine* uji atau kedua pada direktori home/riza/ aka nada folder repositori proyek, setelah itu jalankan perintah seperti pada **Modul 2.12** , kemudian lakukan *import database* seperti pada **Modul 2.13**. Setelah itu kita buka pada

browser dengan ip:8000, dimana kita sesuaikan dengan *virtual machine* kita menjadi 192.168.43.232:8000, hasil dapat dilihat pada **Gambar 2.26**.



**Gambar 2.26** Hasil uji 2

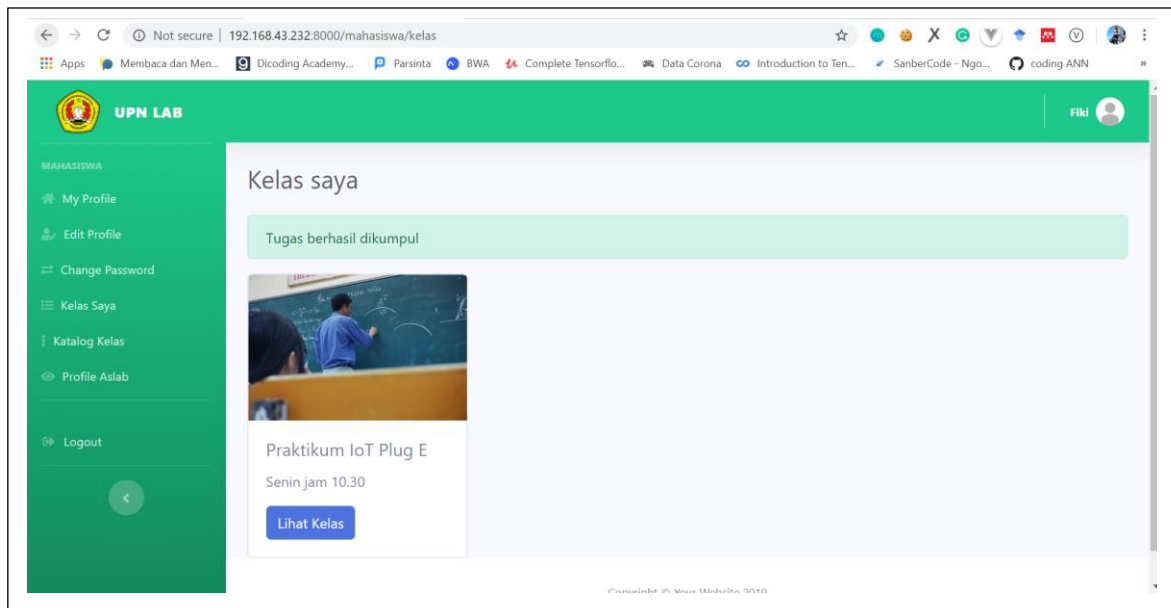
Kemudian kita uji apakah fitur ini dapat berjalan dengan baik, kita *login* dan hasilnya dapat dilihat pada **Gambar 2.27**.



**Gambar 2.27** Hasil login 2

Seperti ujicoba sebelumnya, kita akan mencoba fitur *upload* file dengan mengumpulkan tugas bagi mahasiswa, maka akan *error* seperti **Gambar 2.16**, maka

solusinya sama, yaitu mengganti *permission* dengan perintah seperti **Modul 2.16**, setelah di perbaiki, kita uji lagi dan hasilnya akan tampil pada **Gambar 2.28**.



**Gambar 2.28** Hasil upload file

### 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 sampai Mei tahun 2020:

**Tabel 3.1** Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Pengerjaan Tugas 1 (LAMPP)								
4.	Pembuatan Laporan Submisi								
5.	Pengerjaan Tugas 2 (Docker)								
6.	Melanjutkan Pembuatan Laporan								
7.	Revisi								
8.	Pengujian Singkat								
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	Alfa
2.	Pengujian Singkat	Alfa
3.	Latar Belakang Masalah	Umam
4.	Agenda Pengerjaan Proyek	Alfa
5.	Pengumpulan Progress 1	Alfa
6.	Pengumpulan Progress 2	Umam
7.	Tugas 1 (LAMPP)	Alfa
8.	Tugas 2 (Dockerfile)	Alfa
9.	Pembuatan Laporan	Umam dan Alfa

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berdasarkan masalah yang ada, yaitu membuat *assignment* praktikum diperoleh hasil yang memuaskan. Sistem *Assignment* Praktikum dapat berjalan dengan baik dijalankan melalui *hosting local* maupun melalui *Docker container*. Dari pengujian sistem, semua skenario yang dibuat dapat berjalan dengan baik sesuai dengan yang diharapkan termasuk upload file yang sebelumnya terdapat masalah hak akses.

Pembagian tugas terlaksana dengan baik dan sesuai dengan waktu yang ditentukan, namun terdapat kendala di sinyal internet yang tidak selalu bagus.

#### **4.2 Saran**

Pengerjaan proyek berjalan mulus dan selesai sebelum tenggat waktu yang ditentukan, namun terdapat kendala sinyal internet sehingga terdapat beberapa kendala, seperti pengumpulan laporan yang kurang tepat waktu dan berimbas kepada nilai, dan sinyal internet yang kurang bagus menyebabkan kurangnya komunikasi antara kami berdua.

Ram laptop yang digunakan juga harus cukup besar menurut saya, meskipun saya memiliki RAM 4GB, namun jika di gunakan terlalu lama untuk membuka VMWare Workstation laptop tidak kuat, apalagi membuka chrome untuk pengujian.

## DAFTAR PUSTAKA

Teng, F. and Magoules, F. (2010). "Resource Pricing and Equilibrium Allocation Policy in Cloud Computing". Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology, 2010 of Conference. 1901212: IEEE Computer Society, 195-202.

Foster, I., Y. Zhao, I. Raicu & S. Lu, 2008, "Cloud computing and grid computing 360-degree compared", pada Proceedings of IEEE Grid Computing Environment Workshop. p. 1-10.

Ercan, T. (2010). "Effective use of cloud computing in educational institutions". Procedia Social and Behavioral Sciences, 2, 938-942.

Thorsteinsson G., Page T., Niculescu A., "Using virtual reality for developing design communication". Studies in Informatics and Control, 19 (2), 93-106, (2010).

\_\_\_\_\_, < <https://docs.docker.com/engine/reference/builder/>>, (accessed 03 Mei 2020)

Erick, Jan Solem. 2012. *Programming Computer Vision with Python*

## **LAMPIRAN**