

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR

SISTEM REKAP DATA PASIEN KLINIK PRATAMA AVICENA DLINGO
MENGGUNAKAN UBUNTU LAMPP SERVER DAN PROSES
PEMBUATAN DOCKERFILENYA



DISUSUN OLEH:

NAMA ANGGOTA : ADITYA BAYU PRABOWO 123170048
CICI YURIZA 123170055
KELAS : D
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 REKAP DATA PASIEN KLINIK PRATAMA AVICENA DLINGO MENGGUNAKAN UBUNTU LAMPP DAN

Disusun oleh :

Aditya Bayu Prabowo

123170048

Cici Yuriza

123170055

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

Assalamu'alaikum Warrohmatullohi Wabarakatuh

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencerahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul "Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo Menggunakan Ubuntu Lampp Server 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 laboratorium yang selalu membimbing dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun sangat diperlukan.

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

Yogyakarta, 2 April 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR.....	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2 Tujuan Proyek Akhir	2
1.3 Manfaat Proyek Akhir	2
1.4 Tahap Penyelesaian Proyek Akhir.....	2
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.....	11
2.5 Hasil Implementasi	21
2.6 Pengujian Singkat	23
BAB III JADWAL PENGERJAAN DAN PEMBAGIAN TUGAS	28
3.1 Agenda Pengerjaan	28
3.2 Keterangan Pembagian Tugas.....	28
BAB IV KESIMPULAN DAN SARAN.....	29
4.1 Kesimpulan	29
4.2 Saran	29
DAFTAR PUSTAKA	30
LAMPIRAN	31

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Cloud Computing adalah gabungan pemanfaatan teknologi komputer ('komputasi') dan pengembangan berbasis Internet ('awan'). Awan (cloud) adalah meteora dari internet, sebagaimana awan yang sering digambarkan di diagram jaringan komputer, awan (cloud) dalam Cloud Computing juga merupakan abstraksi dari infrastruktur kompleks yang disembunyikannya(Anggeriana, 2011). *Cloud computing* adalah ketersediaan sumber daya sistem komputer sesuai permintaan, terutama penyimpanan data dan daya komputasi, tanpa manajemen aktif langsung oleh pengguna. Istilah ini umumnya digunakan untuk menggambarkan pusat data yang tersedia bagi banyak pengguna melalui Internet. *Cloud computing* sendiri sebenarnya merupakan hasil dari evolusi yang berlangsung secara bertahap. Sebelum *cloud computing* mulai *booming* seperti sekarang ini, terlebih dahulu terjadi beberapa fenomena seperti *Virtualisasi*, *Grid Computing*, *ASP/Application Service Provision* dan juga *Software as a Service* atau yang lebih dikenal pula dengan sebutan *SaaS*.

Cloud computing ini mempunyai kegunaan salah satunya yang kami terapkan pada projek kami yaitu pada *Lampp Ubuntu Server* dan penggunaan *Docker File*. Pada *Lampp Ubuntu Server* sendiri kegunaan yaitu sebagai tempat penyimpanan pemrograman web dengan bahasa pemrograman php yang bersumber pada *database* yang dapat di akses melalui ubuntu server admin. Sedangkan kegunaan *Docker File* yaitu pencatatan file berbasis server yang dapat di akses secara terbuka melalui *ip server cloud*.

Pada judul proyek yang kami ajukan tersebut kami menggunakan teknologi *cloud computing*, karena menurut kami dengan penerapan teknologi tersebut mempermudah dalam dalam pengontrolan dan menghemat akses *ram* pada memori laptop. Kemudian manfaat yang kami dapatkan yaitu kami dapat mengetahui bagaimana cara menggunakannya, seperti pada *Lampp Ubuntu Server* kami dapat belajar bagaimana cara menyimpan suatu pemrograman web menggunakan bahasa php dan penyimpanan data menggunakan *database mysql*. Kemudian pada *Docker File* kami dapat belajar bagaimana cara mendokumentasikan sebuah proyek menggunakan server.

Di sini kami akan sedikit menjelaskan *program* tersebut beserta komponen-komponennya. *Program* tersebut mengimplementasikan bagaimana kodingan pemrograman web tugas kerja praktik di pindahkan ke server melalui *ubuntu lampp server* menggunakan *software* yang bernama *VMWare Workstation* versi 15.5.2. Pada *VMWareWorkstation* di install sebuah *operating system Ubuntu Linux Server* versi 18.04.4 LTS.

Kemudian kami menginstall sebuah *software remote server* dari *VMWare Workstation*. *Putty* ini menurut kami sangat berguna karena server dapat menyesuaikan keadaan terutama ukurannya. Kemudian untuk memindahkan berkas dari laptop ke server kami menginstall *software* yang bernama *winscp*.

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 Rekap Data Pasien Klinik Pratama Avicena Dlingo yang telah dibuat sebelumnya dengan menggunakan Ubuntu 20.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Menjalankan Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo berbasis web di Ubuntu Server (LAMPP).
3. Membuat dockerfile untuk Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo berbasis web agar aplikasi tersebut dapat dibangun, dikemas, dan dijalankan di berbagai lokasi sebagai sebuah *container* yang ringan.

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

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari sistem Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Menentukan konfigurasi yang tepat untuk pengaturan Docker sehingga dapat digunakan sesuai *requirement* yang berupa:

- a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Dapat melakukan migrasi database dengan mudah.
 - c. Menggunakan Ubuntu dengan versi 18.04.4 LTS.
3. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
 4. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Pencatatan Kendaraan ketersediaannya/*availability*-nya maksimal.
 5. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses dan area akses sesuai konfigurasi.
 6. Menghasilkan Sistem yang berbasiskan *cloud computing* sesuai standar ISO 9001.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Pada proyek ini kami menggunakan laptop ASUS X455L dengan spesifikasi processor *Intel Core I-3*, ram jenis DDR-3 10 Gib dan SSD 240 Gib dengan *operating system* Windows 10. Kemudian Program tersebut mengimplementasikan bagaimana kodingan pemrograman web tugas kerja praktik di pindahkan ke server melalui *ubuntu lampp server* menggunakan *software* yang bernama *VMWare Workstation* versi 15.5.2. Pada *VMWare Workstation* di install sebuah *operating system Ubuntu Linux Server* versi 18.04.4 LTS. Kemudian kami menginstall sebuah *software* yang dapat meremote server dari *VMWare Workstation* yang bernama *putty*. *Putty* ini menurut kami sangat berguna karena server dapat menyesuaikan keadaan terutama ukurannya. Kemudian untuk memindahkan berkas dari laptop ke server kami menginstall *software* yang bernama *winscp*. Untuk lebih jelasnya bisa dilihat pada tabel 2.1 berikut :

Tabel 2.1 Spesifikasi VM *cloud computing* untuk proyek pertama

No	Nama Parameter	Nilai	Keterangan
1	VMWare Workstation	15.5.2	Versi terbaru dan <i>support</i> dengan kapasitas laptop yang di gunakan
2	Ubuntu Linux Server	18.04.4 LTS	<i>Support</i> dengan kapasitas laptop yang di gunakan
3	Putty	0.73	
4	Winscp	5.17	
6	IPv4 hypervisor	IP : 192.168.148.131 SM: 255.255.255.0 DNS: 192.168.30.1	IP <i>number</i> untuk akses manajemen <i>hypervisor</i> melalui <i>web based</i> . Pengaturan IP <i>address</i> menggunakan mode statik. Kelas IP/ <i>subnet mask</i> yang digunakan. Alamat IP untuk fungsionalitas DNS.
7	Processor info	Intel Core i-3 4005U, 1.7 GHz	
8	RAM info	10 Gib <i>Memory</i>	

Pada tugas ke-2 proyek ini kami menggunakan laptop ASUS A422U dengan spesifikasi processor *intel Core I-7*, dengan *operating system* Windows 10. Kemudian untuk virtual machine menggunakan VMWare workstation versi 15.5.2 build-15785246, *operating system Ubuntu Linux Server* versi 18.04.4 LTS. Kemudian kami menginstall sebuah *software* yang dapat meremote server dari *VMWare Workstation* yang bernama *putty*. *Putty* ini menurut kami sangat berguna karena server dapat menyesuaikan keadaan terutama ukurannya. Kemudian untuk memindahkan berkas dari laptop ke

server kami menginstall *software* yang bernama *winscp*. Selanjutnya pada Ubuntu dilakukan penginstalan Docker dengan docker-engine version 19.03.8 dan Docker-compose version 1.24.1. Untuk lebih lengkapnya bisa dilihat pada tabel 2.2 berikut :

Tabel 2.2 Spesifikasi VM *cloud computing* dan Ubuntu OS untuk proyek kedua

No	Nama Parameter	Nilai	Keterangan
1	VMWare Workstation	15.5.2	Versi terbaru dan <i>support</i> dengan kapasitas laptop yang di gunakan
2	Ubuntu Linux Server	18.04.4 LTS	<i>Support</i> dengan kapasitas laptop yang di gunakan
3	Putty	0.73	<i>Remote</i> Ubuntu
4	Winscp	5.17	berfungsi untuk transfer file atau copy file antara windows dengan linux
5	IPv4 hypervisor	IP : 192.168.84.128	IP <i>number</i> untuk akses manajemen <i>hypervisor</i> melalui <i>web based</i> . Pengaturan IP <i>address</i> menggunakan mode statik.
		SM: 255.255.255.0	Kelas IP/ <i>subnet mask</i> yang digunakan.
		DNS: 192.168.30.1	Alamat IP untuk fungsionalitas DNS.
6	Processor info	Intel Core i-3 4005U, 1.7 GHz	
7	RAM info	10 Gib <i>Memory</i>	
8	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Dukungan bahasa pemrograman yang digunakan oleh sistem manajemen <i>password</i> .
		MySQL	Dukungan penyimpanan yang digunakan oleh sistem manajemen <i>password</i> .
9	Docker Engine	Docker Server 19.03.8	<i>Docker engine server</i> yang digunakan dalam menjalankan container.
		Docker Client 19.03.8	<i>Docker engine client</i> yang digunakan untuk berkomunikasi dengan <i>docker server</i> .
10	Docker Compose	Docker Compose 1.24.1	Dukungan alat untuk mendefinisikan dan menjalankan <i>multi-container</i> .

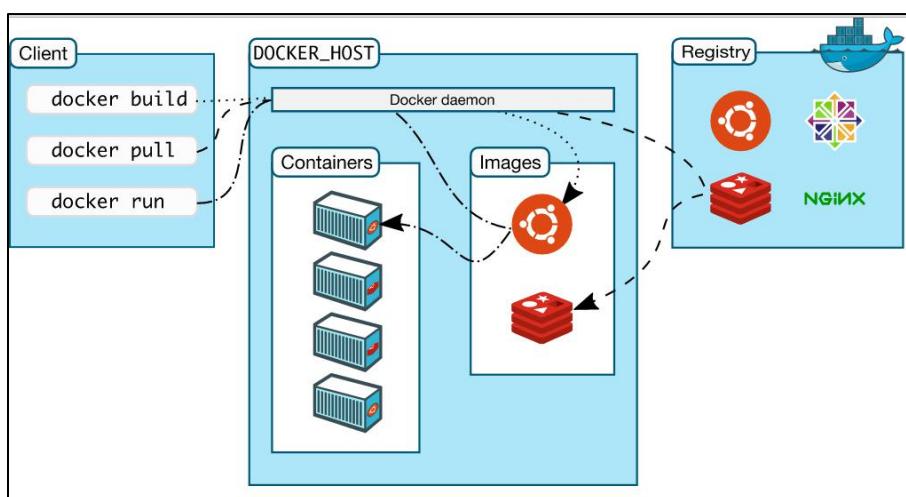
2.2 Rancangan Arsitektur *Cloud Computing*

Pada tugas ke-1 proyek kali ini kami membuat menggunakan arsitektur IaaS dimana hardware laptop ASUS X455L dengan spesifikasi processor *Intel Core I-3*, *ram*

jenis DDR-3 10 Gib dan SSD 240 Gib sebagai layer utama di bagian bawah. Kemudian di atasnya adalah PaaS atau *operating system* Windows 10, *VMware Workstation*, *Putty*, *WinSCP* yang berjalan. Kemudian di atasnya adalah SaaS atau penginstalan *Ubuntu Admin Server*, *Ubuntu Lampp*, *Docker File*. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini.

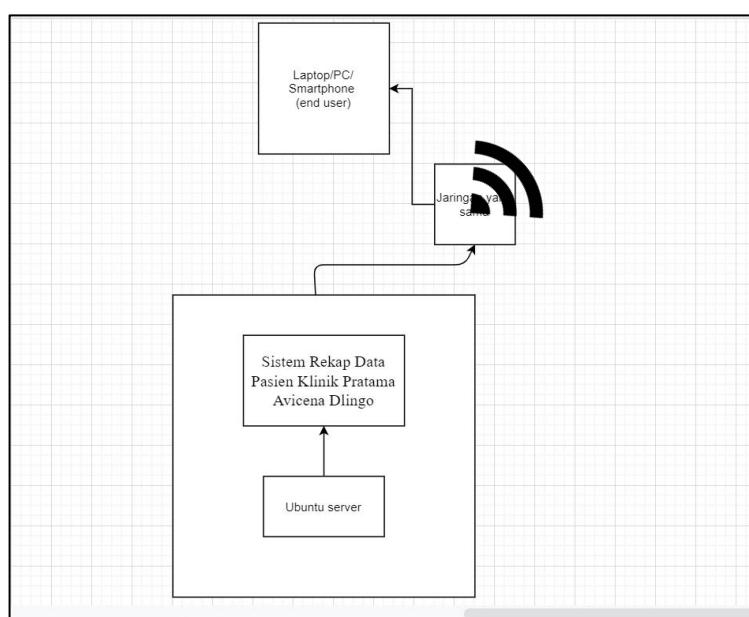
Gambar 2.2 Ilustrasi rancangan arsitektur

Pada tugas ke-2 kami menggunakan laptop ASUS A422U dengan spesifikasi yang telah disebutkan pada komponen yang digunakan di BAB 2. Laptop tersebut digunakan sebagai IaaS, VMware dan windows 10.putty, winSCP, docker, ubuntu digunakan sebagai PaaS. Untuk koneksi jaringan kami menggunakan *hotspot* dari smartphone. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.2** berikut ini.



Gambar 2.2 Ilustrasi Docker

Sistem dapat diakses pada jaringan yang sama baik jaringan melalui *hotspot* ataupun *wifi*. Berikut ilustrasi jaringan dapat dilihat pada **Gambar 2.3** berikut ini:



Gambar 2.3 Ilustrasi Jaringan

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

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
```

```
$ sudo mysql_secure_installation
```

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
- mysql secure instalation : 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
```

```
$ 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
- php libapache2-mod-php php-mysql : nama paket untuk PHP
- nano /var/www/html/info.phpn: 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
$ sudo mysql -u root
```

```
mysql> UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('123170055') WHERE User = 'root';
```

Keterangan: Untuk mengubah password dari user 'root' menjadi '123170055'

```
mysql> FLUSH PRIVILEGES;
```

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
- mysql -u root : masuk ke mysql sebagai user root
- Flush privileges : Untuk merefresh akun yang terkoneksi dengan phpmyadmin

Modul 2.4 Parameter instalasi *PHPMyAdmin*

Langkah pertama dalam install docker adalah meng update list packages.

Update list packages dapat dilihat pada penjelasan **Modul 2.5** berikut ini:

```
$ sudo apt update
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- Update : update list packages

Modul 2.5 Parameter update list packages

Install beberapa package yang diperlukan agar apt bisa diakses HTTPS.

Parameter yang digunakan dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

```
$ sudo apt-get install \
apt-transport-https \
ca-certificates \
Curl \
Gnupg-agent \
software-properties-common
```

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
- apt-transport-https, ca-certificate, curl, gnupg-agent, software- properties-common : nama paket aplikasi

Modul 2.6 Parameter package apt

Dalam install docker adalah menambahkan GPG key dari untuk Docker repository ke sistem kita. Parameter yang digunakan dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo apt-key fingerprint 0EBFCD88
```

Keterangan:

- curl : command line tool untuk mentransfer data menggunakan protokol jaringan.
- apt-key : digunakan untuk memanajemen daftar key yang digunakan oleh apt untuk melakukan autentikasi package.
- add - : parameter tambahan pada key package manager untuk menambahkan key ke daftar trusted key.
-

Modul 2.7 Parameter menambahkan GPG key dari untuk Docker repository ke sistem
Tambahkan Docker repository ke APT. Parameter yang digunakan untuk
menambahkan docker repository ke APT dapat dilihat pada penjelasan **Modul 2.8**
berikut ini:

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

Keterangan:

- 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 2.8 Parameter menambahkan Docker repository ke APT

Instalasi Docker Engine. Parameter instalasi Docker Engine dapat dilihat pada penjelasan **Modul 2.9** berikut ini:

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

Keterangan:

- docker-ce : paket aplikasi untuk docker engine community
- docker-ce-cli : paket aplikasi untuk docker engine command line
- containerd.io : paket aplikasi untuk container

Modul 2.9 Parameter instalasi Docker Engine

Memastikan bahwa docker sudah berjalan. Parameter yang digunakan untuk memastikan bahwa docker sudah berjalan dapat dilihat pada penjelasan **Modul 2.10** berikut ini:

```
$ sudo systemctl status docker
```

Modul 2.10 Parameter untuk memastikan docker sudah berjalan

Setelah docker berjalan, install docker compose agar dapat menjalankan *multi-container*. Parameter yang digunakan untuk *download docker-compose binary file* dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

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

Modul 2.11 Parameter instalasi Docker Compose

Parameter yang digunakan untuk install curl dapat dilihat pada penjelasan **Modul**

2.12 berikut ini:

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

Modul 2.12 Parameter untuk install curl

Parameter yang digunakan untuk install docker compose dapat dilihat pada penjelasan **Modul 2.13** berikut ini:

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

Modul 2.13 Parameter untuk install docker compose

Parameter yang digunakan untuk memastikan docker compose sudah berjalan dengan baik dapat dilihat pada penjelasan **Modul 2.14** berikut ini:

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

Modul 2.14 Parameter untuk memastikan docker compose sudah berjalan

Setelah docker dan docker compose sudah dipastikan berjalan dengan baik, tahapan selanjutnya adalah membuat file docker-compose untuk proyek yang akan dikerjakan.

Buat folder dengan nama docker, folder tersebut digunakan untuk menyimpan file docker-compose dan file Aplikasi. Parameter yang digunakan untuk membuat folder docker dapat dilihat pada penjelasan **Modul 2.15** berikut ini:

```
$ mkdir docker/lamp/projek
```

Modul 2.15 Parameter untuk membuat folder docker

Parameter yang digunakan untuk masuk ke folder diatas dapat dilihat pada penjelasan **Modul 2.16** berikut ini:

```
$ cd docker/lamp/projek/
```

Modul 2.16 Parameter untuk masuk ke folder docker

Parameter yang digunakan untuk membuat Dockerfile dapat dilihat pada penjelasan **Modul 2.17** berikut ini:

```
$ nano php.Dockerfile
```

Modul 2.17 Parameter membuat Dockerfile

Parameter yang digunakan untuk membuat docker-compose dapat dilihat pada penjelasan **Modul 2.18** berikut ini:

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

Modul 2.18 Parameter membuat docker-compose

Parameter yang digunakan menjalankan docker-compose dapat dilihat pada penjelasan **Modul 2.19** berikut ini:

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

Keterangan:

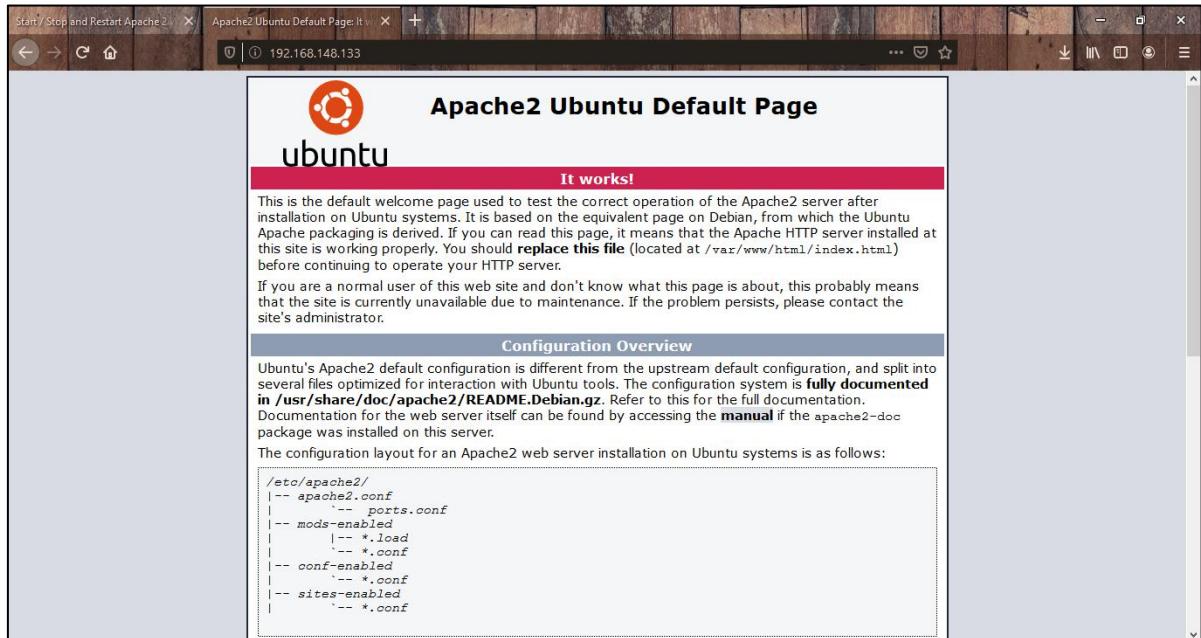
- docker-compose : command untuk mengeksekusi perintah-perintah Docker Compose.
- up : parameter dalam docker-compose untuk menjalankan container.
- -d : parameter dalam docker-compose agar proses berjalan di background.

Modul 2.19 Parameter menjalankan docker-compose

2.4 Tahap Implementasi

Pada bagian ini akan dijelaskan mengenai implementasi yang dilakukan untuk penyelesaian projek ini.

Hasil implementasi instal Apache dapat dilihat pada **Gambar 2.4** berikut ini:



Gambar 2.4 Tampilan Apache setelah Ter-Install

Hasil implementasi nano /var/www/html/info.php dapat dilihat pada **Gambar 2.5** berikut ini:

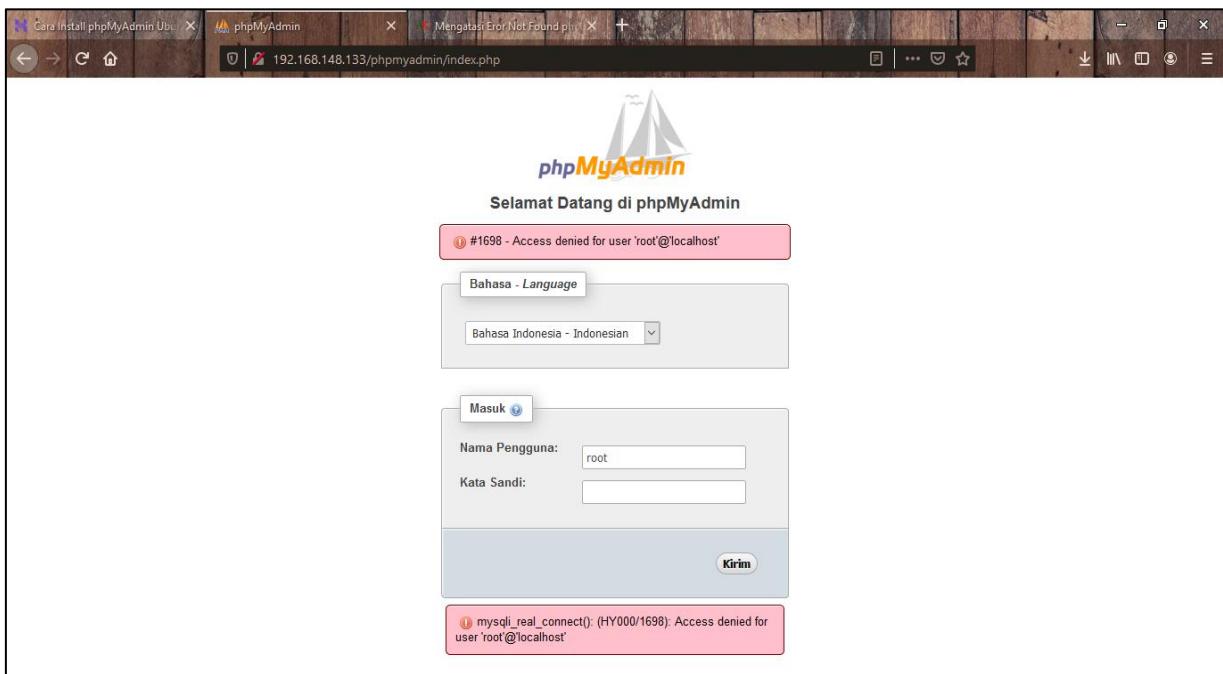
```
aditya0076@aditya: ~
GNU nano 2.9.3          /var/www/html/info.php

?php
phpinfo();
?>

[ Read 3 lines ]
^G Get Help  ^C Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit     ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^  Go To Line
```

Gambar 2.5 Tampilan isi info.php

Hasil implementasi install phpMyAdmin dapat dilihat pada **Gambar 2.6** berikut ini:

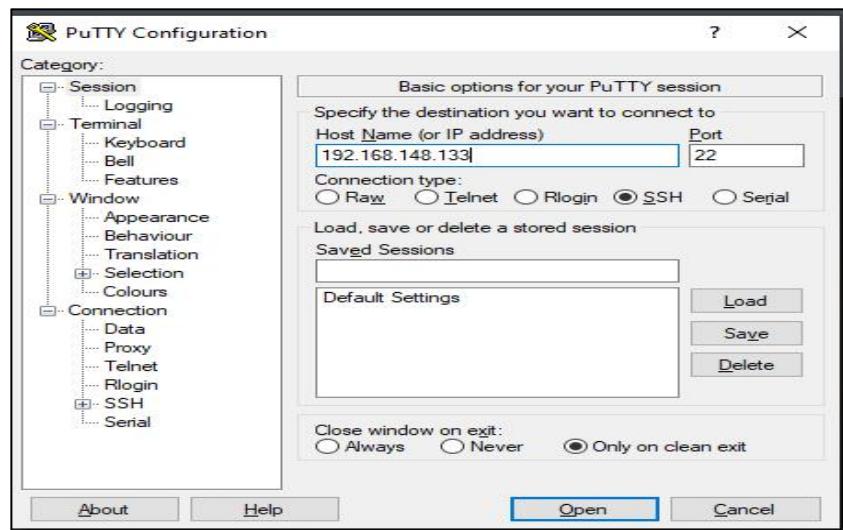


Gambar 2.6 Tampilan install phpMyAdmin

Hasil implementasi setelah *setting password root* dan berhasil login phpMyAdmin dapat dilihat pada **Gambar 2.7** berikut ini:

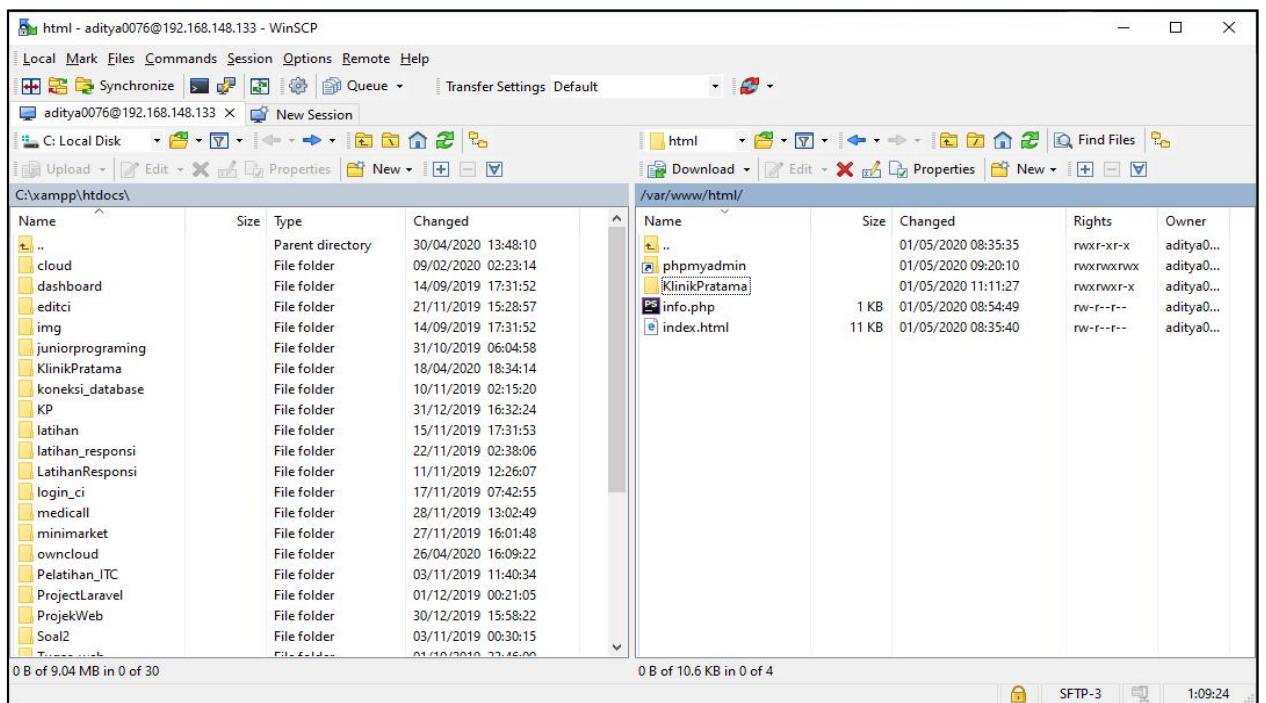
Gambar 2.7 Tampilan berhasil login phpMyAdmin

Remote Ubuntu dengan putty dapat dilihat pada **Gambar 2.8** berikut ini:



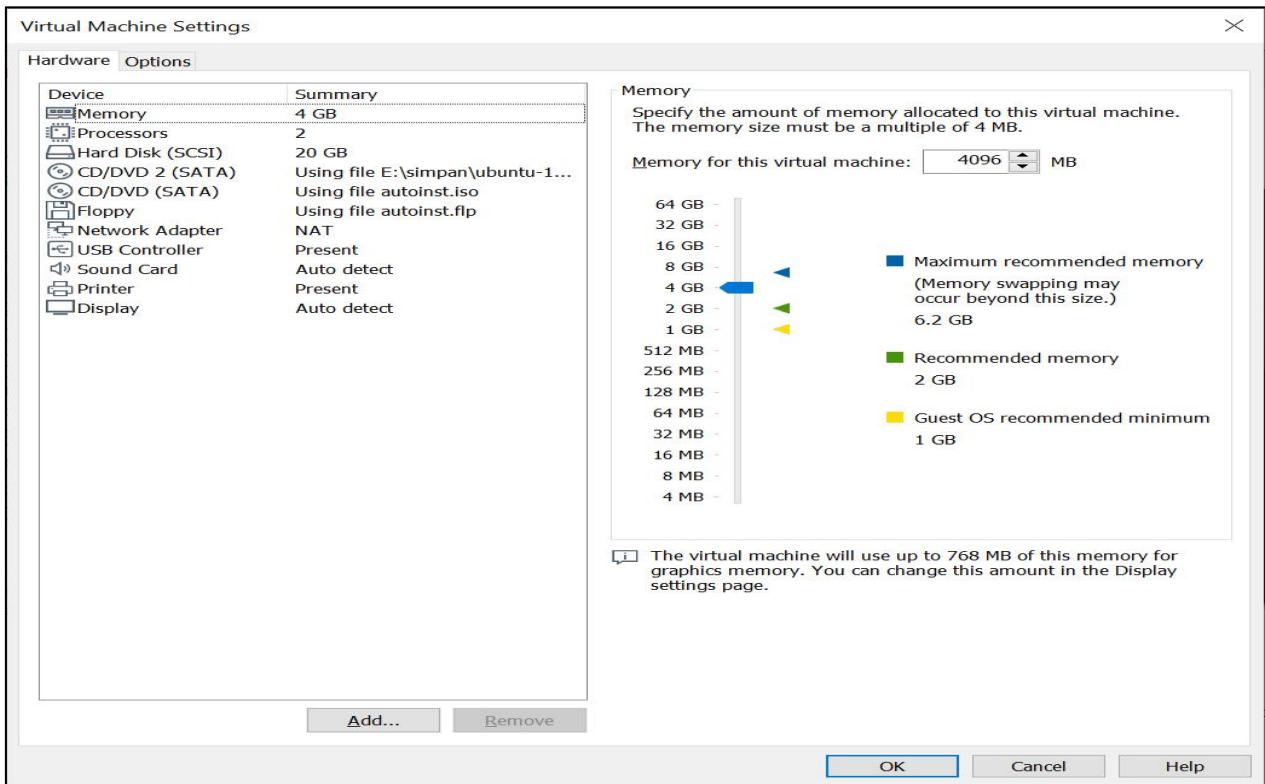
Gambar 2.8 Tampilan putty

Hasil implementasi setelah berhasil *connect* WinSCP dan berhasil drag projek dari local Windows 10 ke Ubuntu folder /var/www/html/ dapat dilihat pada **Gambar 2.9** berikut ini:



Gambar 2.9 Tampilan berhasil drag projek ke folder /var/www/html/

Hasil implementasi Dalam pembuatan *virtual machine* untuk tugas ke-2 dengan VMWare Workstation dilakukan opsi konfigurasi seperti pada **Gambar 2.10** berikut ini:



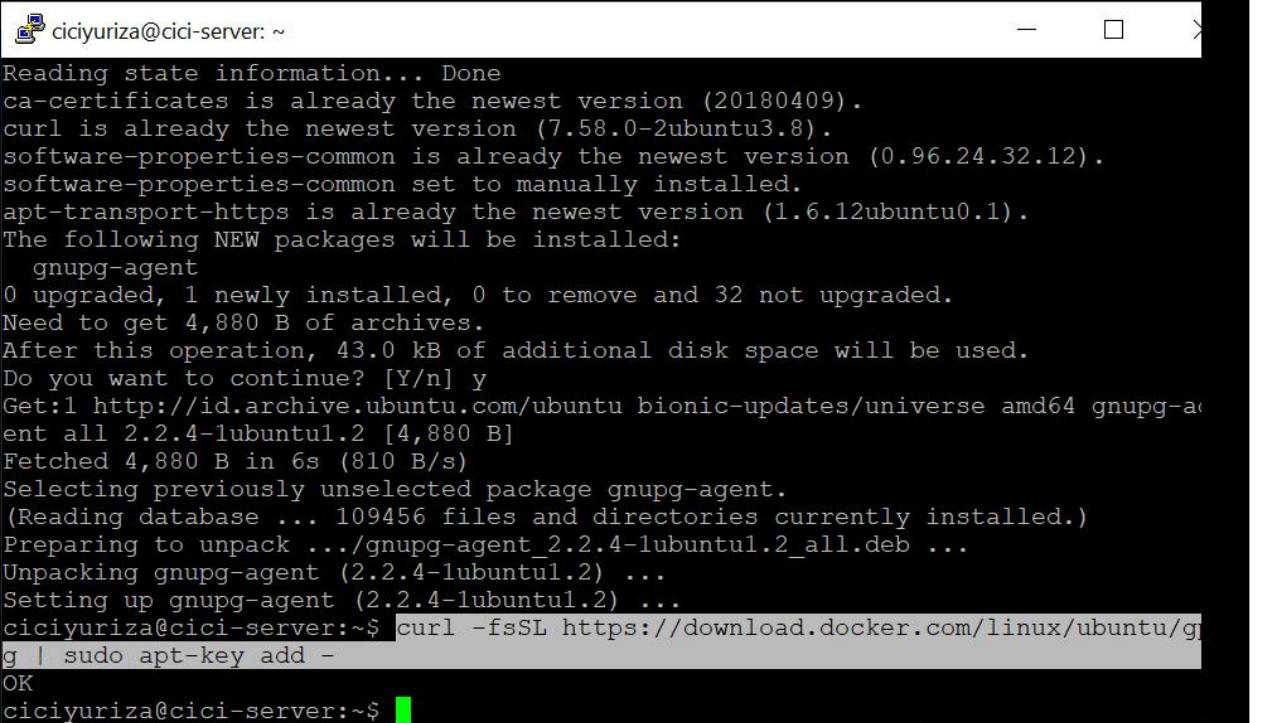
Gambar 2.10 Tampilan konfigurasi VMWare Workstation tugas ke-2

Hasil implementasi **Modul 2.6** dapat dilihat pada **Gambar 2.11** berikut ini:

```
cicyuriza@cici-server: ~
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 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
Hit:3 https://packagecloud.io/dokku/dokku/ubuntu trusty InRelease
Reading package lists... Done
cicyuriza@cici-server:~$ sudo apt-get install \
>     apt-transport-https \
>     ca-certificates \
>     curl \
>     gnupg-agent \
>     software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
curl is already the newest version (7.58.0-2ubuntu3.8).
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
apt-transport-https is already the newest version (1.6.12ubuntu0.1).
The following NEW packages will be installed:
    gnupg-agent
0 upgraded, 1 newly installed, 0 to remove and 32 not upgraded.
```

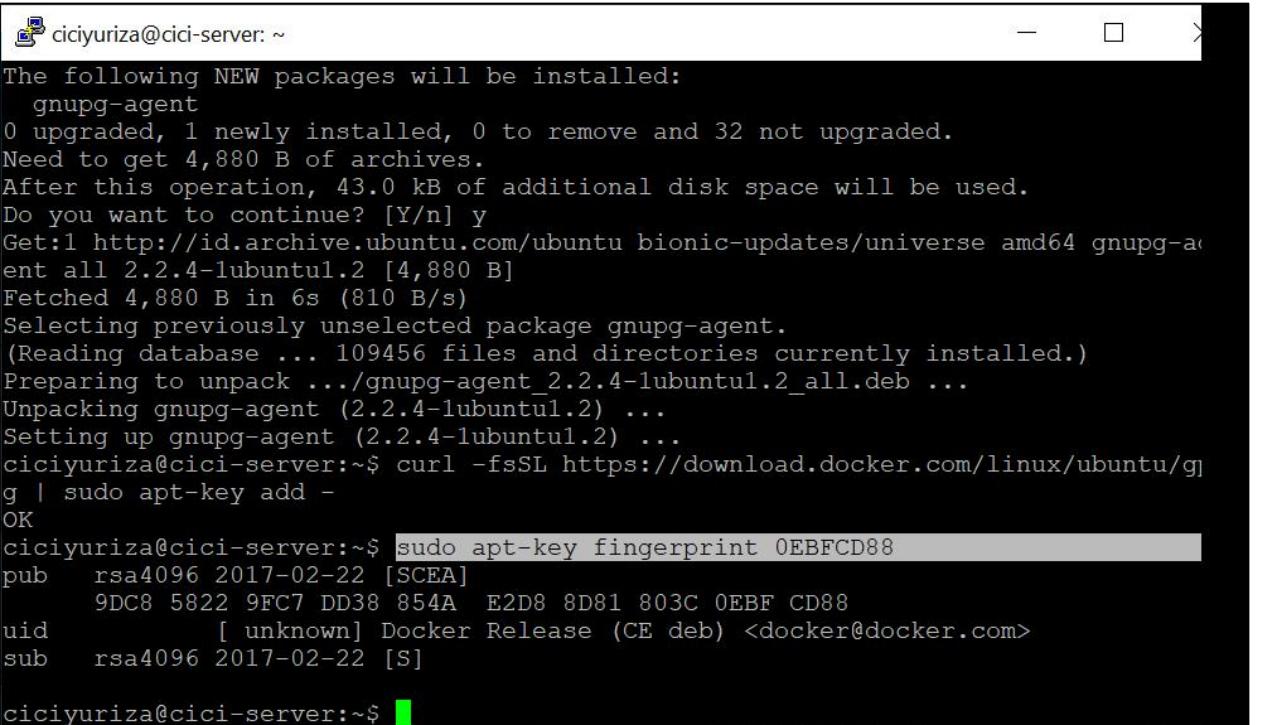
Gambar 2.11 Tampilan implementasi **Modul 2.6**

Hasil implementasi **Modul 2.7** dapat dilihat pada **Gambar 2.12** dan **Gambar 2.13** berikut ini:



```
ciciyuriza@cici-server: ~
Reading state information... Done
ca-certificates is already the newest version (20180409).
curl is already the newest version (7.58.0-2ubuntu3.8).
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
apt-transport-https is already the newest version (1.6.12ubuntu0.1).
The following NEW packages will be installed:
  gnupg-agent
0 upgraded, 1 newly installed, 0 to remove and 32 not upgraded.
Need to get 4,880 B of archives.
After this operation, 43.0 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 gnupg-agent all 2.2.4-1ubuntu1.2 [4,880 B]
Fetched 4,880 B in 6s (810 B/s)
Selecting previously unselected package gnupg-agent.
(Reading database ... 109456 files and directories currently installed.)
Preparing to unpack .../gnupg-agent_2.2.4-1ubuntu1.2_all.deb ...
Unpacking gnupg-agent (2.2.4-1ubuntu1.2) ...
Setting up gnupg-agent (2.2.4-1ubuntu1.2) ...
ciciyuriza@cici-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/g
g | sudo apt-key add -
OK
ciciyuriza@cici-server:~$
```

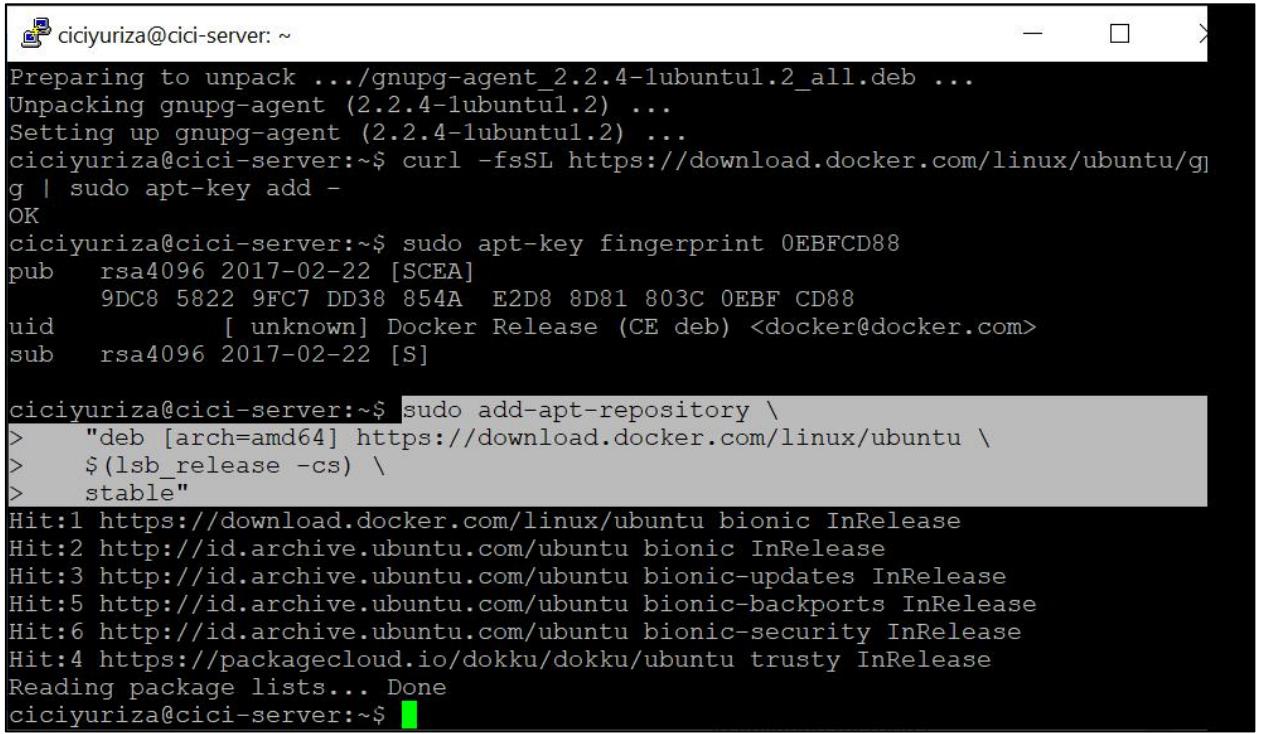
Gambar 2.12 Tampilan implementasi Modul 2.7



```
ciciyuriza@cici-server: ~
The following NEW packages will be installed:
  gnupg-agent
0 upgraded, 1 newly installed, 0 to remove and 32 not upgraded.
Need to get 4,880 B of archives.
After this operation, 43.0 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 gnupg-agent all 2.2.4-1ubuntu1.2 [4,880 B]
Fetched 4,880 B in 6s (810 B/s)
Selecting previously unselected package gnupg-agent.
(Reading database ... 109456 files and directories currently installed.)
Preparing to unpack .../gnupg-agent_2.2.4-1ubuntu1.2_all.deb ...
Unpacking gnupg-agent (2.2.4-1ubuntu1.2) ...
Setting up gnupg-agent (2.2.4-1ubuntu1.2) ...
ciciyuriza@cici-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/g
g | sudo apt-key add -
OK
ciciyuriza@cici-server:~$ 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]
ciciyuriza@cici-server:~$
```

Gambar 2.13 Tampilan implementasi Modul 2.7

Hasil implementasi **Modul 2.8** dapat dilihat pada **Gambar 2.14** berikut ini:



```

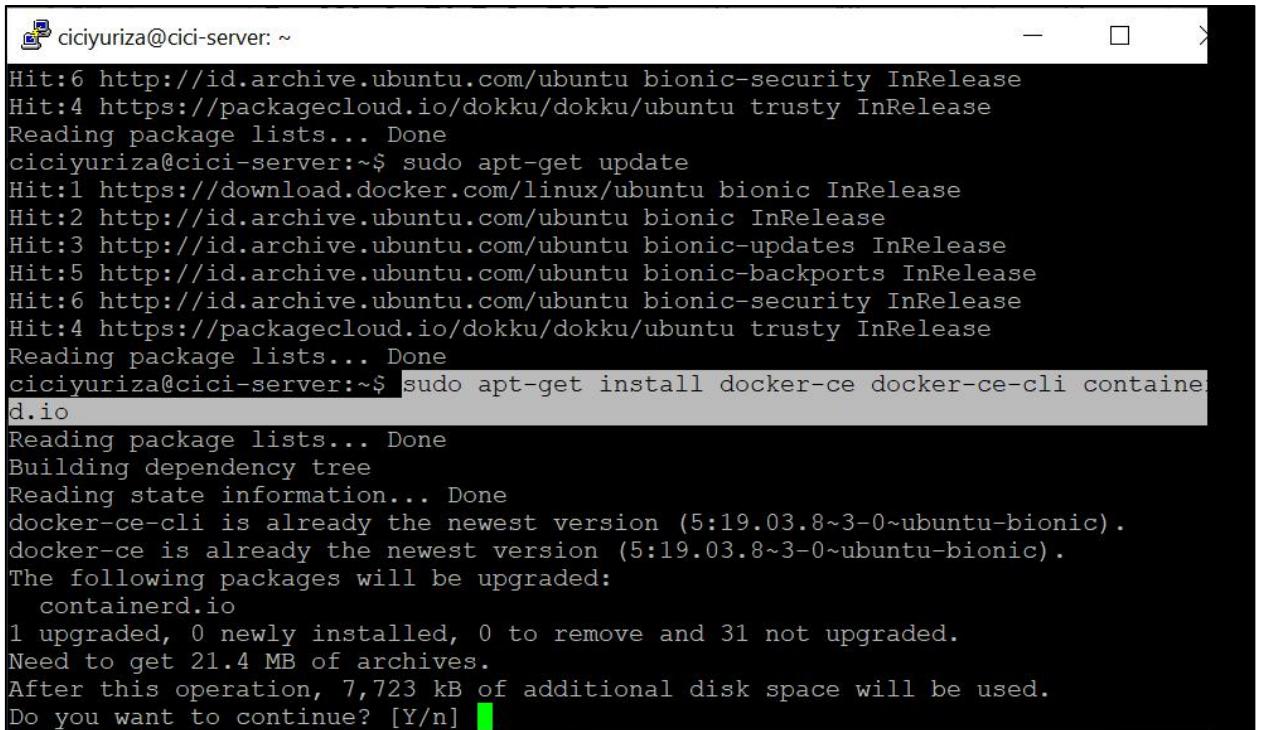
ciciyuriza@cici-server: ~
Preparing to unpack .../gnupg-agent_2.2.4-1ubuntu1.2_all.deb ...
Unpacking gnupg-agent (2.2.4-1ubuntu1.2) ...
Setting up gnupg-agent (2.2.4-1ubuntu1.2) ...
ciciyuriza@cici-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/g
g | sudo apt-key add -
OK
ciciyuriza@cici-server:~$ 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]

ciciyuriza@cici-server:~$ sudo add-apt-repository \
>   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
>   $ (lsb_release -cs) \
>   stable"
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 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
Hit:4 https://packagecloud.io/dokku/dokku/ubuntu trusty InRelease
Reading package lists... Done
ciciyuriza@cici-server:~$ 

```

Gambar 2.14 Tampilan implementasi **Modul 2.8**

Hasil implementasi **Modul 2.9** dapat dilihat pada **Gambar 2.15** berikut ini:



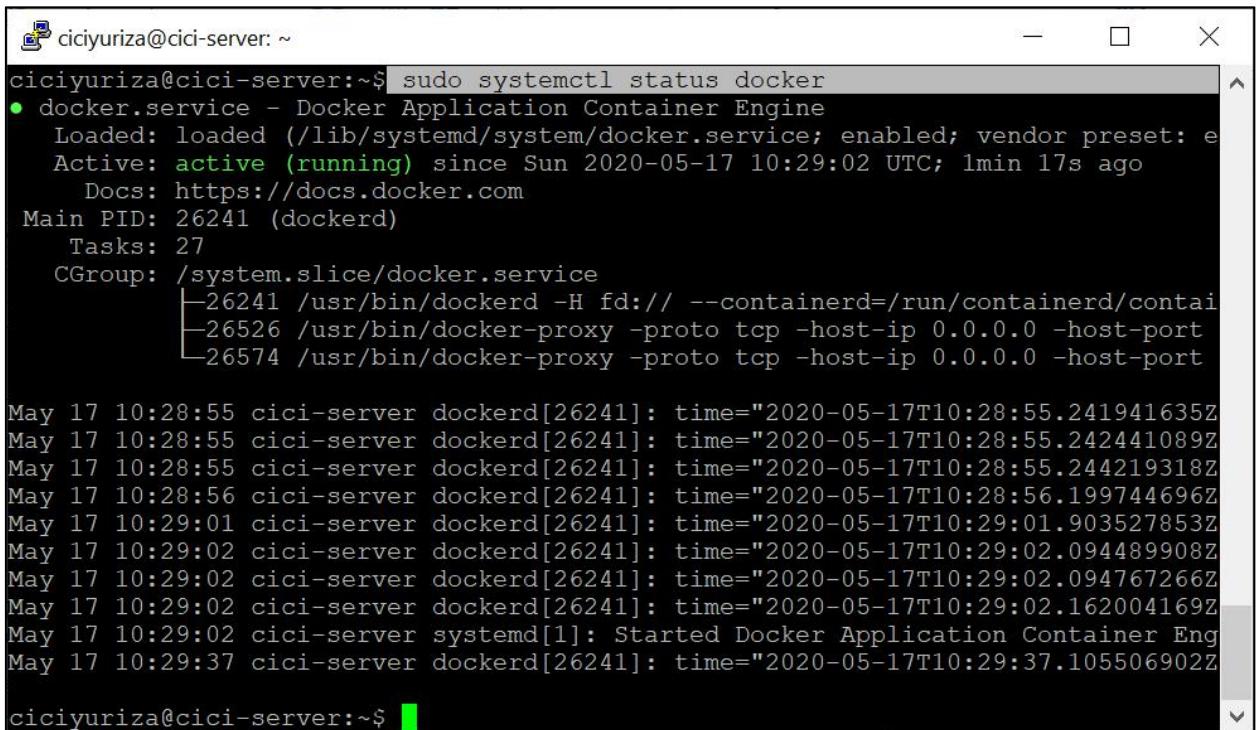
```

ciciyuriza@cici-server: ~
Hit:6 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease
Hit:4 https://packagecloud.io/dokku/dokku/ubuntu trusty InRelease
Reading package lists... Done
ciciyuriza@cici-server:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 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
Hit:4 https://packagecloud.io/dokku/dokku/ubuntu trusty InRelease
Reading package lists... Done
ciciyuriza@cici-server:~$ sudo apt-get install docker-ce docker-ce-cli container
d.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker-ce-cli is already the newest version (5:19.03.8~3-0ubuntu-bionic).
docker-ce is already the newest version (5:19.03.8~3-0ubuntu-bionic).
The following packages will be upgraded:
  containerd.io
1 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
Need to get 21.4 MB of archives.
After this operation, 7,723 kB of additional disk space will be used.
Do you want to continue? [Y/n] 

```

Gambar 2.15 Tampilan implementasi **Modul 2.9**

Hasil implementasi **Modul 2.10** dapat dilihat pada **Gambar 2.16** berikut ini:



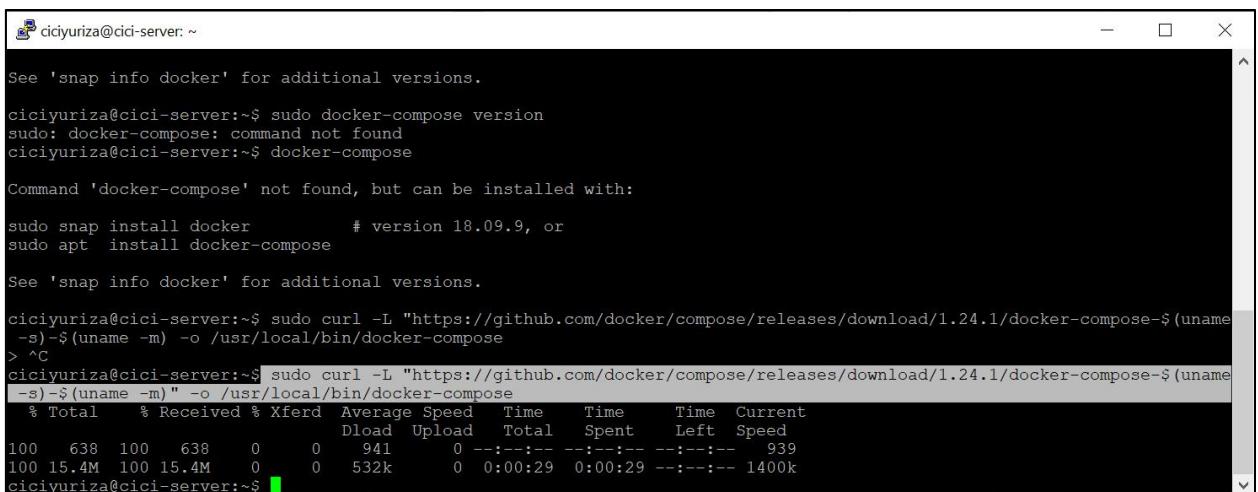
```
ciciyuriza@cici-server:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Sun 2020-05-17 10:29:02 UTC; 1min 17s ago
     Docs: https://docs.docker.com
Main PID: 26241 (dockerd)
   Tasks: 27
      CGrou
      └─26241 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai
         ├─26526 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port
         ├─26574 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port

May 17 10:28:55 cici-server dockerd[26241]: time="2020-05-17T10:28:55.241941635Z"
May 17 10:28:55 cici-server dockerd[26241]: time="2020-05-17T10:28:55.242441089Z"
May 17 10:28:55 cici-server dockerd[26241]: time="2020-05-17T10:28:55.244219318Z"
May 17 10:28:56 cici-server dockerd[26241]: time="2020-05-17T10:28:56.199744696Z"
May 17 10:29:01 cici-server dockerd[26241]: time="2020-05-17T10:29:01.903527853Z"
May 17 10:29:02 cici-server dockerd[26241]: time="2020-05-17T10:29:02.094489908Z"
May 17 10:29:02 cici-server dockerd[26241]: time="2020-05-17T10:29:02.094767266Z"
May 17 10:29:02 cici-server dockerd[26241]: time="2020-05-17T10:29:02.162004169Z"
May 17 10:29:02 cici-server systemd[1]: Started Docker Application Container Eng
May 17 10:29:37 cici-server dockerd[26241]: time="2020-05-17T10:29:37.105506902Z"

ciciyuriza@cici-server:~$
```

Gambar 2.16 Tampilan implementasi **Modul 2.10**

Hasil implementasi **Modul 2.11** dapat dilihat pada **Gambar 2.17** berikut ini:



```
ciciyuriza@cici-server:~$ See 'snap info docker' for additional versions.
ciciyuriza@cici-server:~$ sudo docker-compose version
sudo: docker-compose: command not found
ciciyuriza@cici-server:~$ docker-compose
Command 'docker-compose' not found, but can be installed with:
sudo snap install docker      # version 18.09.9, or
sudo apt install docker-compose
See 'snap info docker' for additional versions.
ciciyuriza@cici-server:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
> ^C
ciciyuriza@cici-server:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total Spent   Left Speed
100  638  100  638    0     0  941      0 --:--:-- 0:00:29 0:00:29 --:--:-- 939
100 15.4M  100 15.4M   0     0  532k      0  0:00:29 0:00:29 --:--:-- 1400k
ciciyuriza@cici-server:~$
```

Gambar 2.17 Tampilan implementasi **Modul 2.11**

Hasil implementasi **Modul 2.12** dapat dilihat pada **Gambar 2.18** berikut ini:

```
ciciyuriza@cici-server:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
ciciyuriza@cici-server:~$
```

Gambar 2.18 Tampilan implementasi **Modul 2.12**

Hasil implementasi **Modul 2.13** dapat dilihat pada **Gambar 2.19** berikut ini:

```
ciciyuriza@cici-server:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
ciciyuriza@cici-server:~$ sudo chmod +x /usr/bin/docker-compose
chmod: cannot access '/usr/bin/docker-compose': No such file or directory
ciciyuriza@cici-server:~$ sudo chmod +x /usr/docker-compose
chmod: cannot access '/usr/docker-compose': No such file or directory
ciciyuriza@cici-server:~$ sudo chmod +x /usr/local/bin/docker-compose
ciciyuriza@cici-server:~$
```

Gambar 2.19 Tampilan implementasi **Modul 2.13**

Hasil implementasi **Modul 2.14** dapat dilihat pada **Gambar 2.20** berikut ini:

```
ciciyuriza@cici-server:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
ciciyuriza@cici-server:~$ sudo chmod +x /usr/bin/docker-compose
chmod: cannot access '/usr/bin/docker-compose': No such file or directory
ciciyuriza@cici-server:~$ sudo chmod +x /usr/docker-compose
chmod: cannot access '/usr/docker-compose': No such file or directory
ciciyuriza@cici-server:~$ sudo chmod +x /usr/local/bin/docker-compose
ciciyuriza@cici-server:~$ docker-compose version
docker-compose version 1.24.1, build 4667896b
docker-py version: 3.7.3
CPython version: 3.6.8
OpenSSL version: OpenSSL 1.1.0j  20 Nov 2018
ciciyuriza@cici-server:~$
```

Gambar 2.20 Tampilan implementasi **Modul 2.14**

Hasil implementasi **Modul 2.17** dapat dilihat pada **Gambar 2.21** berikut ini:

```

ciciyuriza@cici-server: ~/docker/lamp/projek
GNU nano 2.9.3                               docker-compose.yml

version: "3.3"

services:
  web:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes :
      - "./KlinikPratama:/var/www/html/"
    ports:
      - "8080:80"

  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_PASSWORD: 123170055
      MYSQL_USER: root
      MYSQL_ROOT_PASSWORD: 123170055
      MYSQL_DATABASE: klinik_avicena
    volumes:
      - mysql-data:/var/lib/mysql

  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: mysql-server
      PMA_USER: root
      PMA_PASSWORD: 123170055
    ports:
      - "5000:80"
  volumes:
    mysql-data:

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
 ^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^[Go To Line M-E Redo
 M-A Mark Text M-6 Copy Text

Gambar 2.21 Tampilan implementasi **Modul 2.11**

Disini kita membuat tiga buah *service*, yaitu **webserver**, **mysql-server** dan **phpmyadmin**.

Pada *service webserver*, akan melakukan *custom-built* Docker *image* seperti yang sudah kita buat di *php.Dockerfile*. Kemudian path project yg ada pada saat ini kita arahkan ke */var/www/html/*. Setelah itu kita atur *port*-nya yaitu akan tersambung pada *port* 8080.

Pada *service mysql-server*, akan menjalankan *image* *mysql* versi 8.0.19 dari DockerHub didalam containernya. Kemudian kita juga harus mengatur *environment variable* yang dibutuhkan seperti *user*, *password*, *database* yang akan digunakan, *password root* nya. Setelah itu atur path file nya ke */var/lib/mysql*.

Pada *service phpmyadmin*, akan menjalankan *image* *phpmyadmin* versi 5.0.1 dari DockerHub didalam containernya. Kemudian atur *environment variable* yang akan digunakan untuk menyimpan MySQL

hostname, *username* dan *password* yang mana nantinya menyambungkan **phpmyadmin** dengan database server MySQL yaitu **mysql-server**. Kemudian atur *port*-nya yaitu menggunakan *port* 5000.

Ada beberapa perubahan *source code* yang harus dilakukan pada file CI(Codeigniter) yaitu pada *database*, *base_url* dan *session*, apabila tidak dilakukan akan terjadi error. Berikut perubahan yang dilakukan :

'hostname' => 'mysql-server',

```
'username' => 'root',
'password' => '123170055',
'database' => 'klinik_avicena',
```

Source code 2.1 database.php

```
$config['base_url'] = 'http://192.168.84.128:8080/';
```

Source code 2.2 config.php

```
$config['sess_save_path'] = sys_get_temp_dir();
```

Source code 2.3 config.php

Hasil implementasi **Modul 2.18** dapat dilihat pada **Gambar 2.22** berikut ini:

```
ciciyuriza@cici-server: ~/docker/lamp/projek
GNU nano 2.9.3
FROM php:7.4.3-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli
EXPOSE 80

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

Gambar 2.22 Tampilan implementasi **Modul 2.18**

Penjelasan mengenai parameter untuk php.Dockerfile dapat dilihat pada **Modul 2.20** berikut ini:

```
FROM php:7.4.3-apache
RUN apt-get update && apt-get upgrade -y RUN docker-php-ext-install mysqli
EXPOSE 80

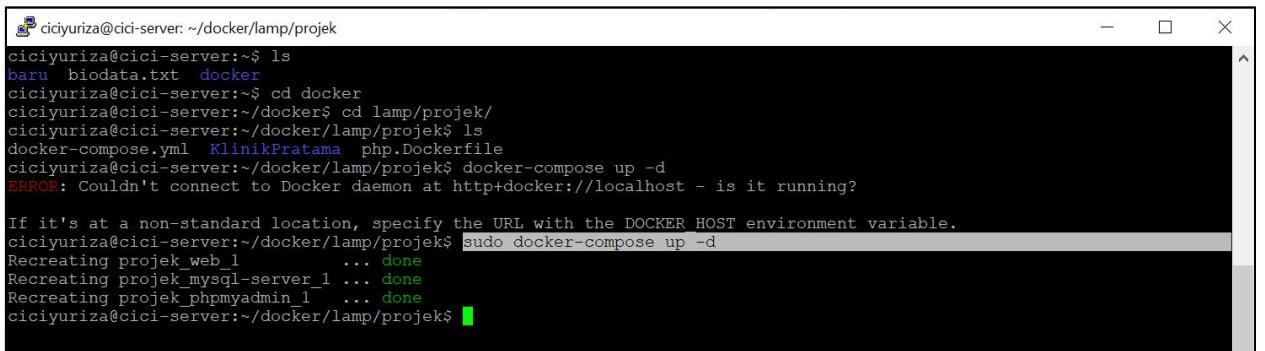
RUN a2enmod rewrite
RUN chmod 755 /var/www/html/ COPY ./ /var/www/html
RUN service apache2 restart
```

Keterangan:

- FROM php:7.4.3-apache : menggunakan image php:7.4.3-apache
- upgrade : menginstall versi terbaru dari package yang dimiliki
- RUN : menjalankan command
- EXPOSE : meng-expose PORT yang digunakan untuk berkomunikasi
- a2enmod rewrite : command untuk mengaktifkan mod_rewrite dalam apache
- chmod : command untuk memodifikasi permission dari suatu direktori
- COPY : command untuk menyalin suta file atau direktori
- service apache2 restart : command untuk merestart apache server

Modul 2.20 parameter dockerfile

Hasil implementasi **Modul 2.19** dapat dilihat pada **Gambar 2.23** berikut ini:

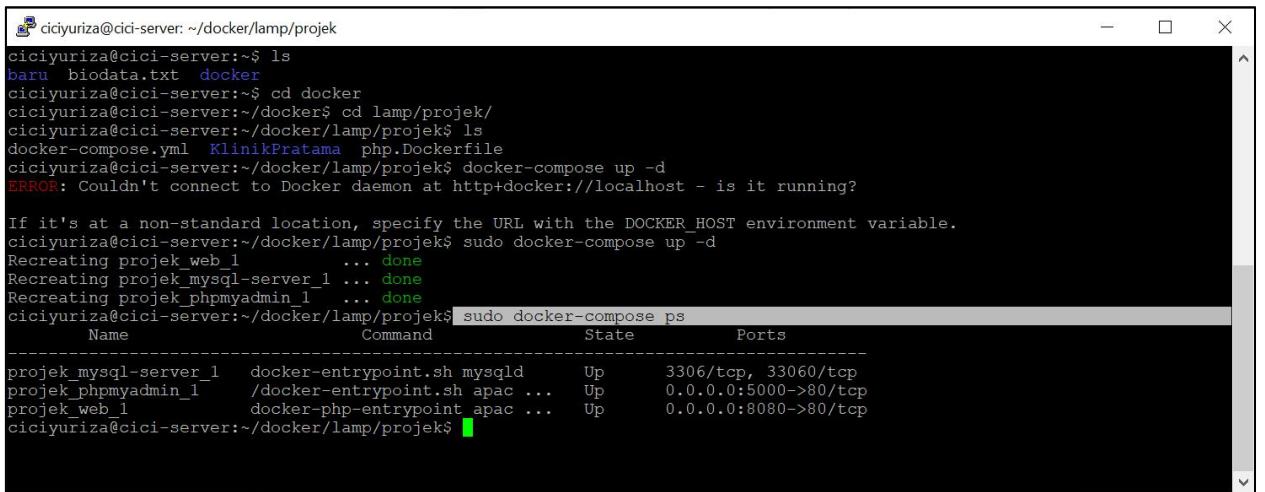


```
ciciyuriza@cici-server:~/docker/lamp/projek
ciciyuriza@cici-server:~$ ls
baru biodata.txt docker
ciciyuriza@cici-server:~$ cd docker
ciciyuriza@cici-server:~/docker$ cd lamp/projek/
ciciyuriza@cici-server:~/docker/lamp/projek$ ls
docker-compose.yml KlinikPratama php.Dockerfile
ciciyuriza@cici-server:~/docker/lamp/projek$ docker-compose up -d
ERROR: Couldn't connect to Docker daemon at http+docker://localhost - is it running?

If it's at a non-standard location, specify the URL with the DOCKER_HOST environment variable.
ciciyuriza@cici-server:~/docker/lamp/projek$ sudo docker-compose up -d
Recreating projek_web_1 ... done
Recreating projek_mysql-server_1 ... done
Recreating projek_phpmyadmin_1 ... done
ciciyuriza@cici-server:~/docker/lamp/projek$
```

Gambar 2.23 Tampilan implementasi **Modul 2.19**

Implementasi melihat container yang berjalan dapat dilihat pada **Gambar 2.24** berikut ini:



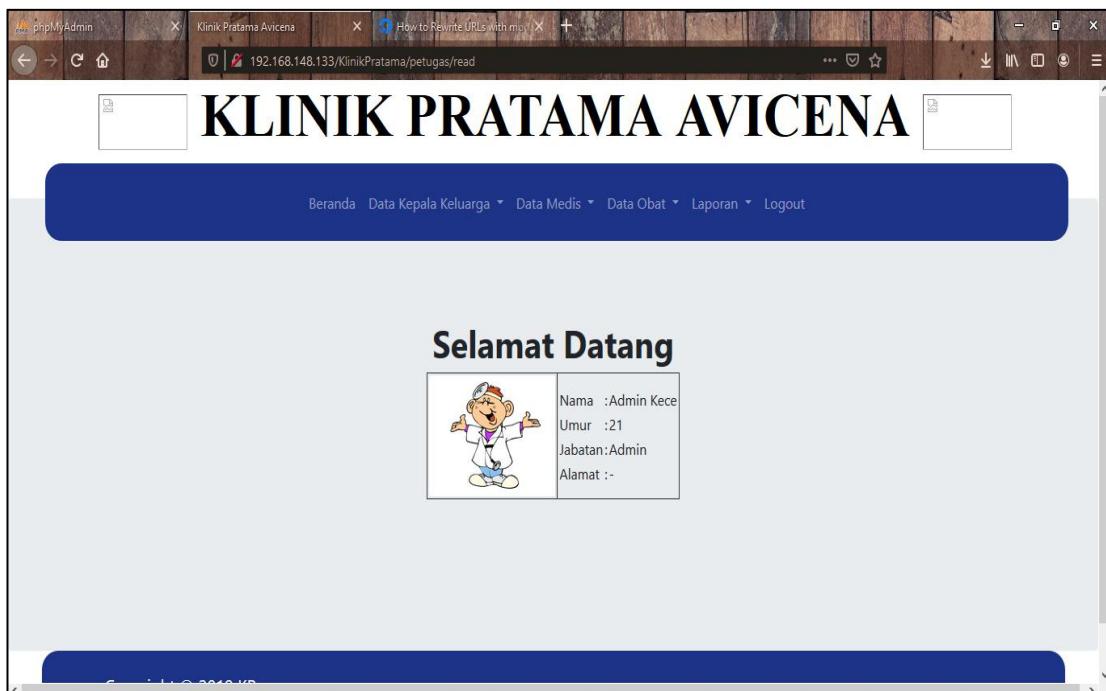
```
ciciyuriza@cici-server:~/docker/lamp/projek
ciciyuriza@cici-server:~$ ls
baru biodata.txt docker
ciciyuriza@cici-server:~$ cd docker
ciciyuriza@cici-server:~/docker$ cd lamp/projek/
ciciyuriza@cici-server:~/docker/lamp/projek$ ls
docker-compose.yml KlinikPratama php.Dockerfile
ciciyuriza@cici-server:~/docker/lamp/projek$ docker-compose up -d
ERROR: Couldn't connect to Docker daemon at http+docker://localhost - is it running?

If it's at a non-standard location, specify the URL with the DOCKER_HOST environment variable.
ciciyuriza@cici-server:~/docker/lamp/projek$ sudo docker-compose up -d
Recreating projek_web_1 ... done
Recreating projek_mysql-server_1 ... done
Recreating projek_phpmyadmin_1 ... done
ciciyuriza@cici-server:~/docker/lamp/projek$ sudo docker-compose ps
      Name           Command    State     Ports
----- 
projek_mysql-server_1   docker-entrypoint.sh mysqld   Up        3306/tcp, 33060/tcp
projek_phpmyadmin_1     /docker-entrypoint.sh apac ... Up        0.0.0.0:5000->80/tcp
projek_web_1            docker-php-entrypoint apac ... Up        0.0.0.0:8080->80/tcp
ciciyuriza@cici-server:~/docker/lamp/projek$
```

Gambar 2.24 Tampilan implementasi melihat container yang berjalan

2.5 Hasil Implementasi

Hasil implementasi tugas ke-1 Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo sudah dapat berjalan di lokal, terlihat pada **Gambar 2.25** berikut ini:

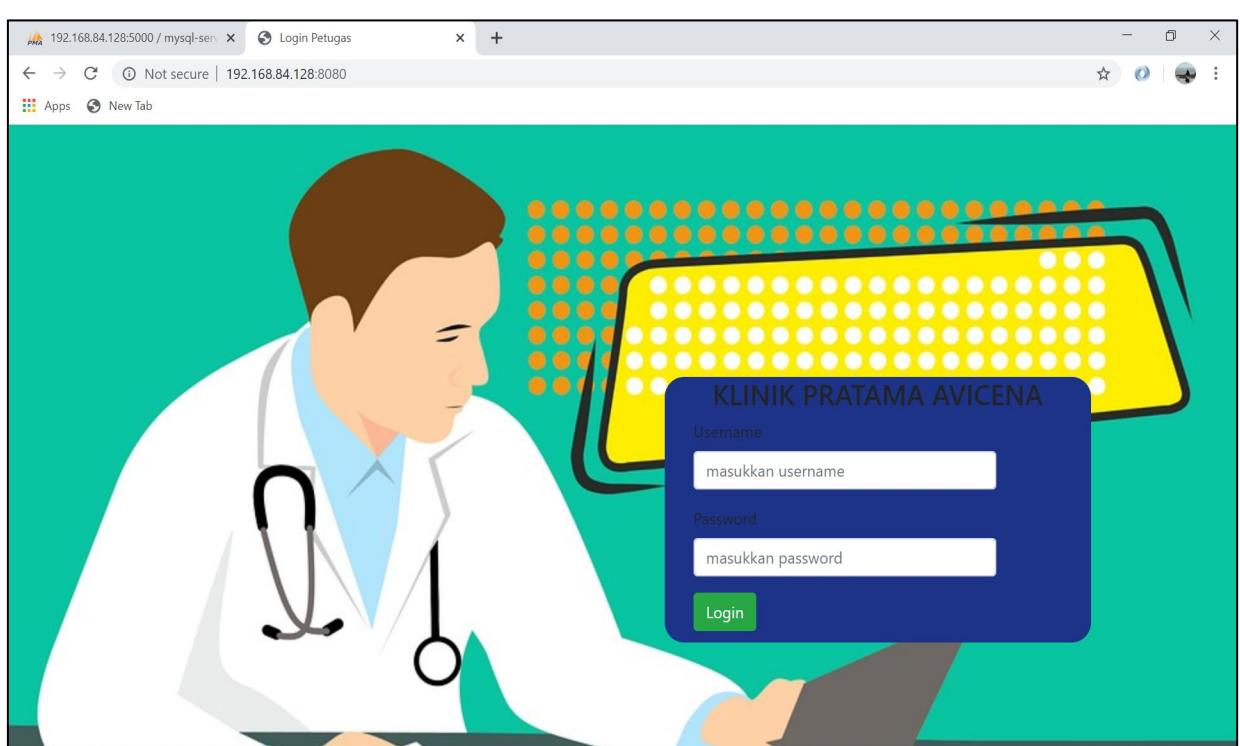


Gambar 2.25 Tampilan berhasil login ke sistem

Hasil implementasi tugas ke-2 Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo phpMyAdmin sudah dapat berjalan dan bisa diakses dengan ip 192.168.84.128 dan port 5000 dan sudah dilakukan import database terlihat pada **Gambar 2.26** berikut ini:

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 Kib	-
belanja	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 Kib	-
data_pasien	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 Kib	-
desa	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 Kib	-
detail_transaksi	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 Kib	-
dusun	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32.0 Kib	-
kepala_keluarga	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	32.0 Kib	-
obat	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 Kib	-
rekam_medis	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 Kib	-
transaksi	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 Kib	-
10 tables	Sum	14	InnoDB	utf8mb4_0900_ai_ci	240.0 Kib	0 B

Gambar 2.26 Tampilan berhasil akses phpMyAdmin tugas ke-2
Hasil implementasi tugas ke-2 Aplikasi Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo sudah dapat berjalan di lokal dengan ip 192.168.84.128:8080, terlihat pada **Gambar 2.27** berikut ini:

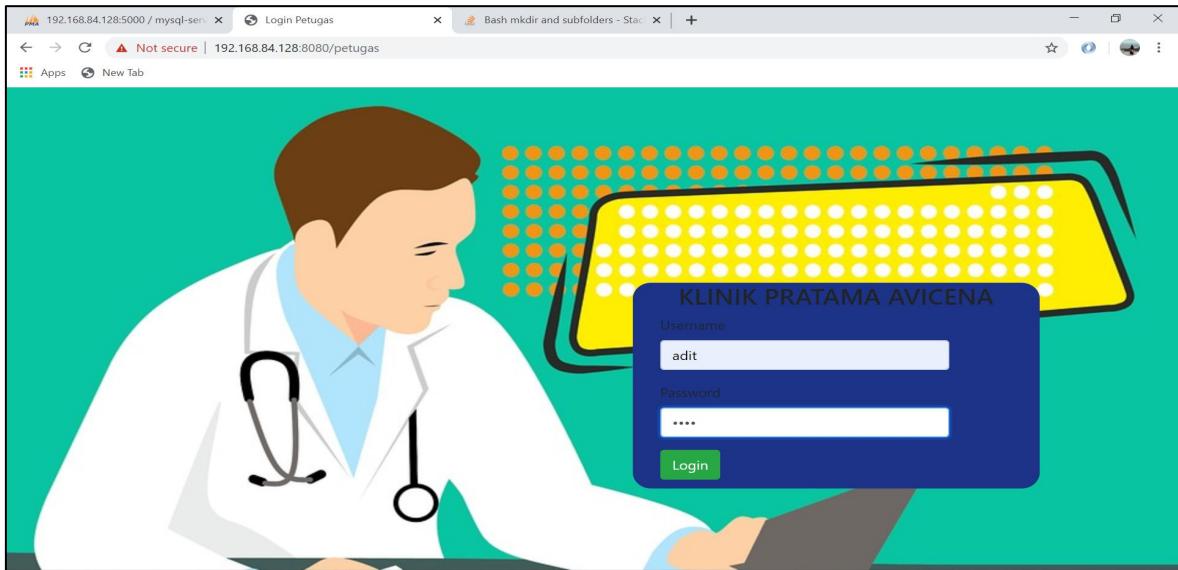


Gambar 2.27 Tampilan berhasil akses Aplikasi pada tugas ke-2

2.6 Pengujian Singkat

Terhadap sistem yang telah dirancang, akan dilakukan pengujian terhadap permasalahan yang tertuang pada bab pertama, yaitu bagaimana meningkatkan efektivitas waktu dan kemudahan pada pengaksesan Sistem Rekap Data Pasien Klinik Pratama Avicena Dlingo. Dalam pengujian singkat ini akan dilakukan percobaan login, entry data, tampilkan data, menghapus data, mencoba menggunakan fitur *search* dan *logout*.

Pengujian login pada sistem seperti yang terlihat pada **Gambar 2.28** berikut ini:



Gambar 2.28 Tampilan percobaan login sistem

Setelah berhasil login, akan tampil beranda seperti yang terlihat pada **Gambar 2.29** berikut ini :



Gambar 2.29 Tampilan Beranda ketika login berhasil

Percobaan berikutnya yaitu menambah data Kepala Keluarga terlihat pada **Gambar 2.30** berikut ini :

Beranda Data Kepala Keluarga Data Medis Data Obat Laporan Logout

Tambah Desa
Tambah Dusun
Tambah Data Kepala Keluarga
Tampil Data Kepala Keluarga

Kode Keluarga	:	A003
Dusun	:	demangan, TEMUWUH
Nama Kepala Keluarga	:	Alif
RT	:	03

Tambah **Batal**

192.168.84.128:8080/kepala_keluarga/create

Gambar 2.30 Tampilan Tambah data Kepala Keluarga
Penambahan data Kepala Keluarga berhasil dilakukan seperti yang terlihat pada **Gambar 2.31** berikut ini:

Beranda Data Kepala Keluarga Data Medis Data Obat Laporan Logout

Data Kepala Keluarga

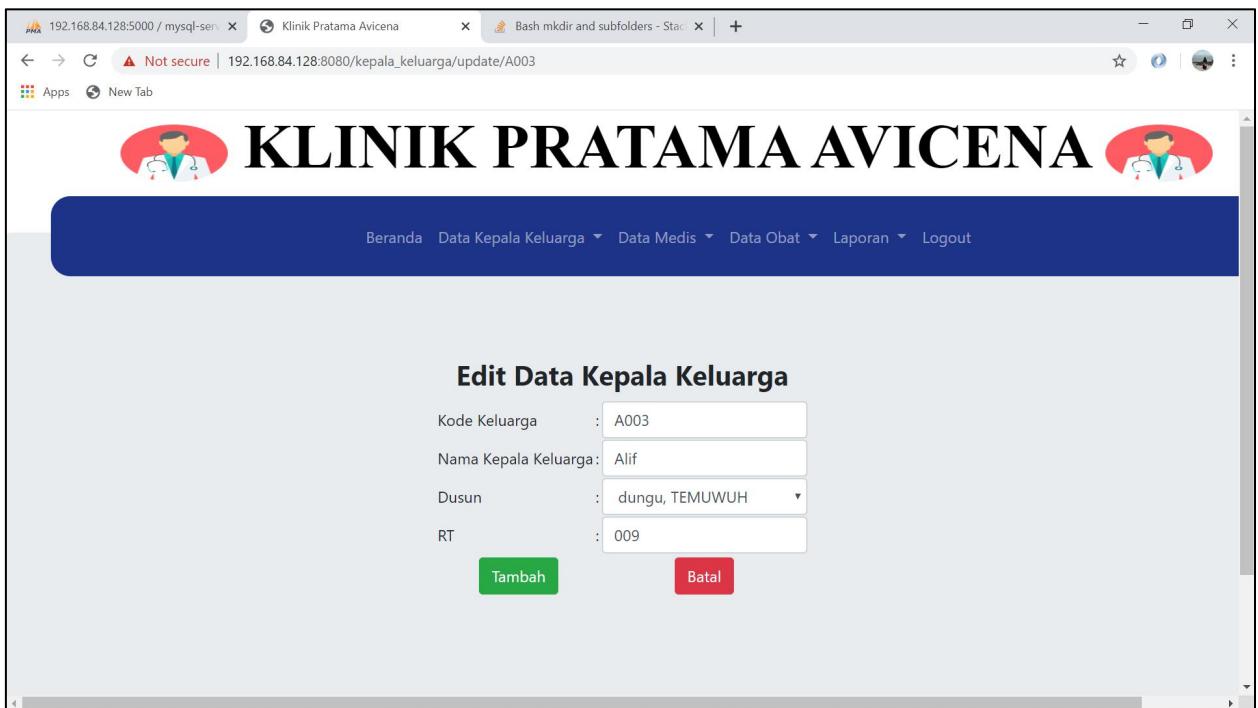
Kode Keluarga	Nama Kepala Keluarga	RT	Pilihan
1231	Rahman	DLINGO 1	01
A001	Aditya Bayu P	TEMUWUH	003
A003	Alif	demangan	03

Tambah Data **Submit**

Data Berhasil Ditambahkan

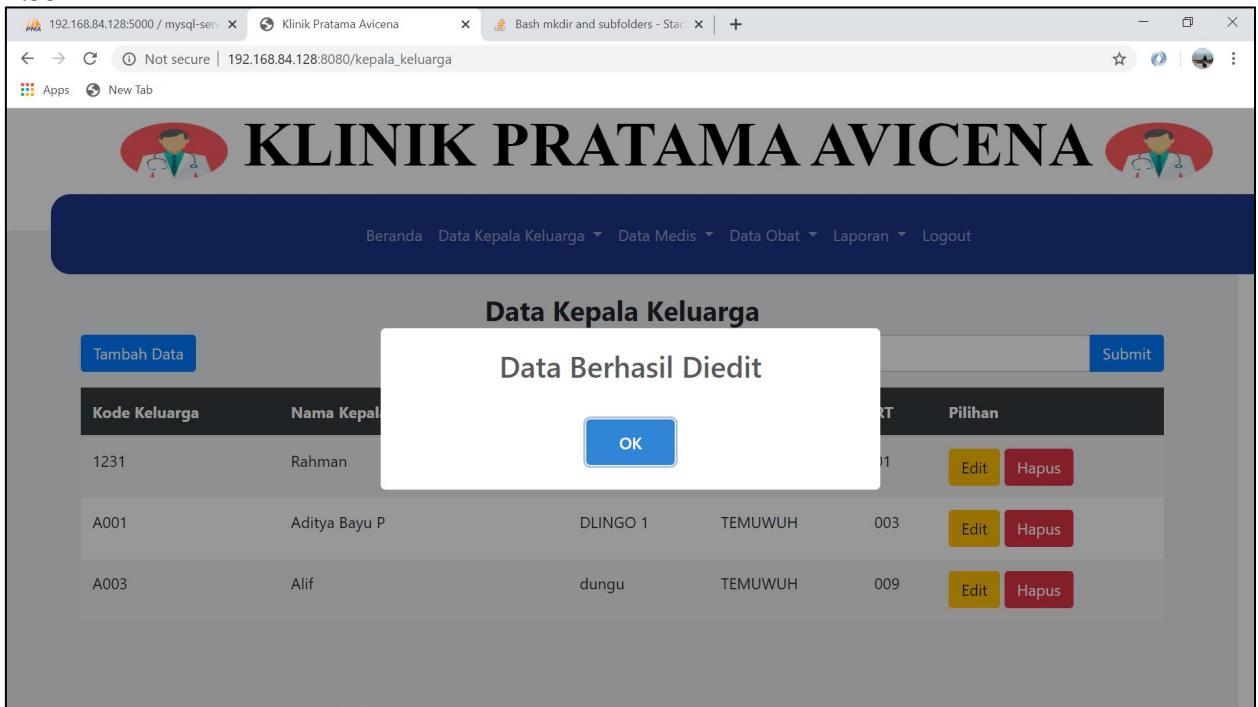
OK

Gambar 2.31 Tampilan berhasil Tambah data Kepala Keluarga
Percobaan berikutnya yaitu edit data Kepala Keluarga terlihat pada **Gambar 2.32** berikut ini :



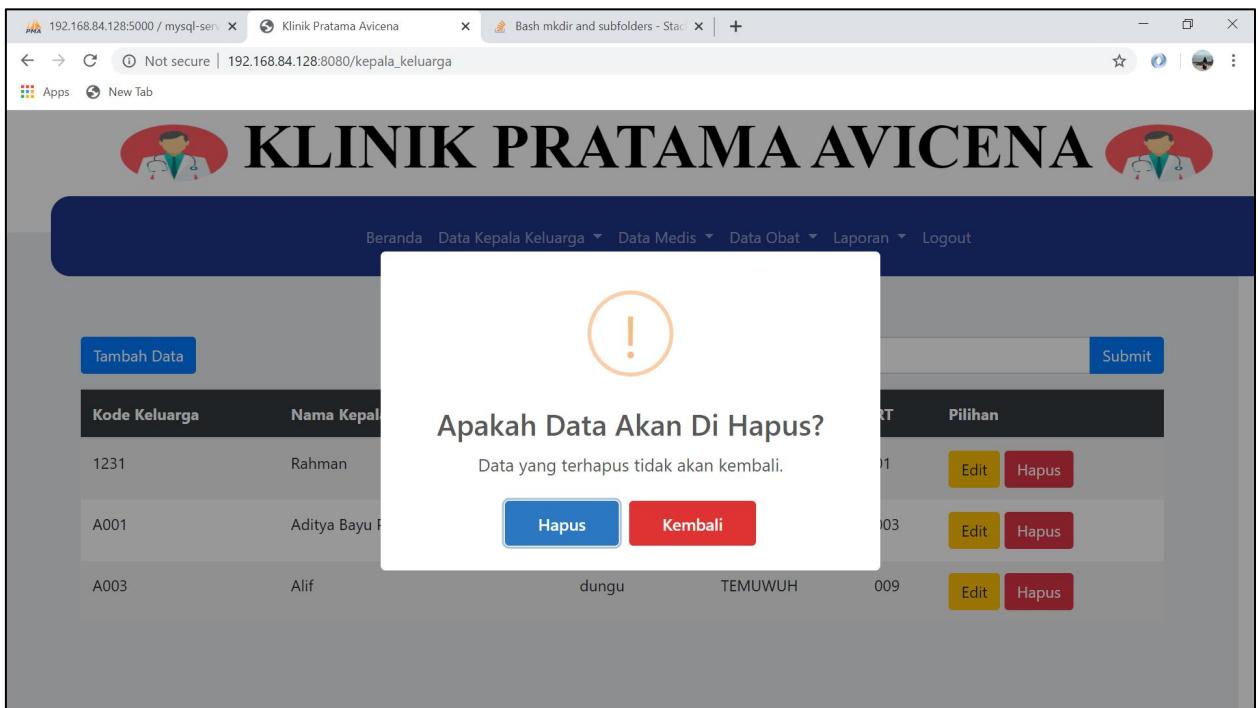
Gambar 2.32 Tampilan edit data Kepala Keluarga

Edit data Kepala Keluarga berhasil dilakukan seperti yang terlihat pada **Gambar 2.33** berikut ini :

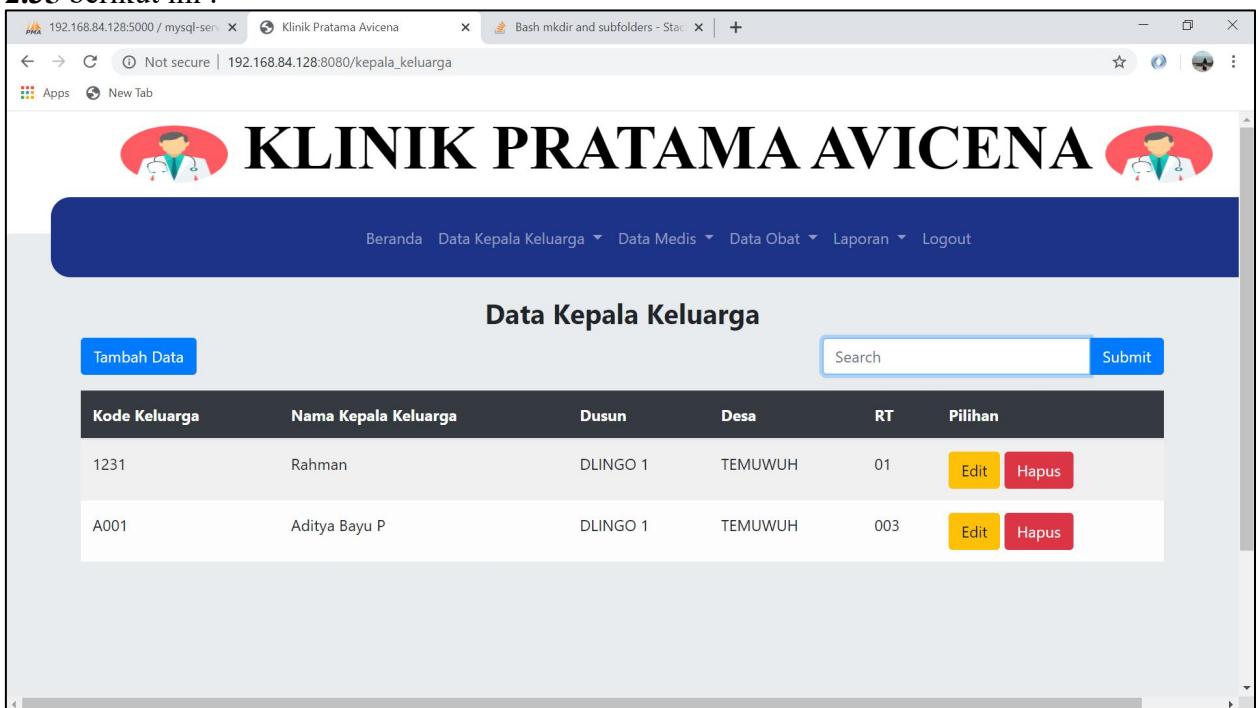


Gambar 2.33 Tampilan berhasil edit data Kepala Keluarga

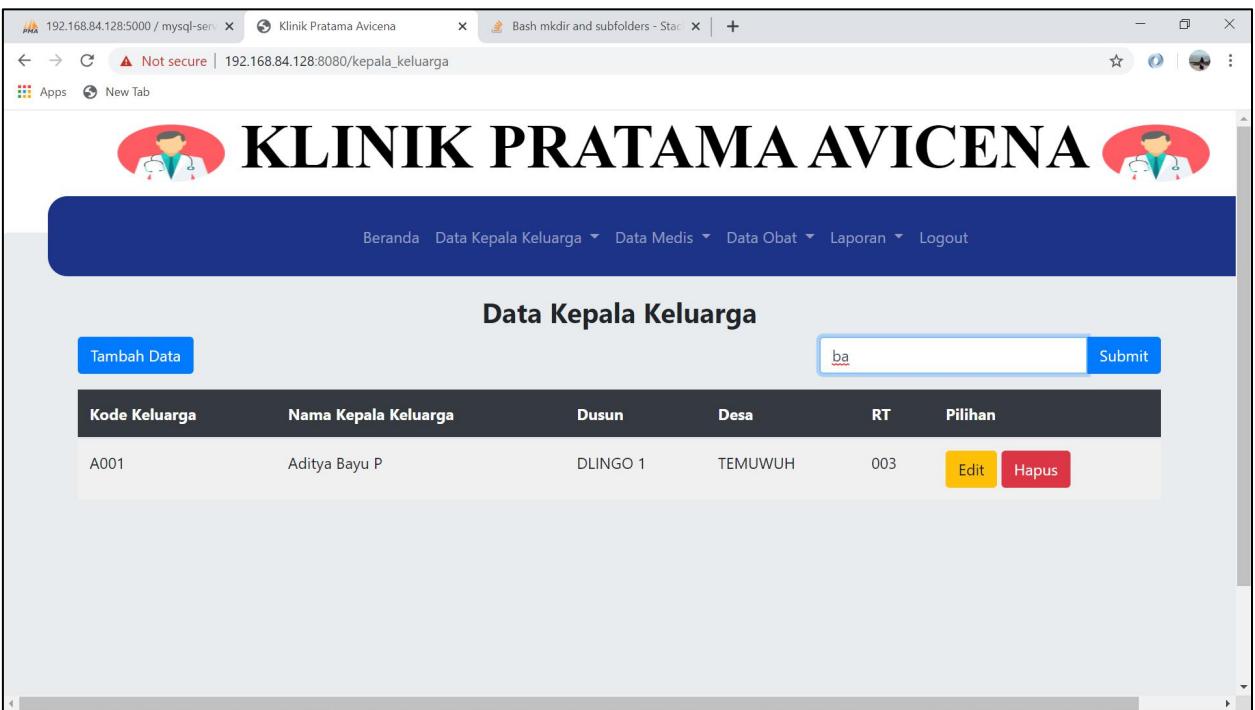
Percobaan berikutnya yaitu menghapus data Kepala Keluarga terlihat pada **Gambar 2.34** berikut ini :



Gambar 2.34 Tampilan konfirmasi hapus data Kepala Keluarga
Hapus data Kepala Keluarga berhasil dilakukan seperti yang terlihat pada **Gambar 2.35** berikut ini :

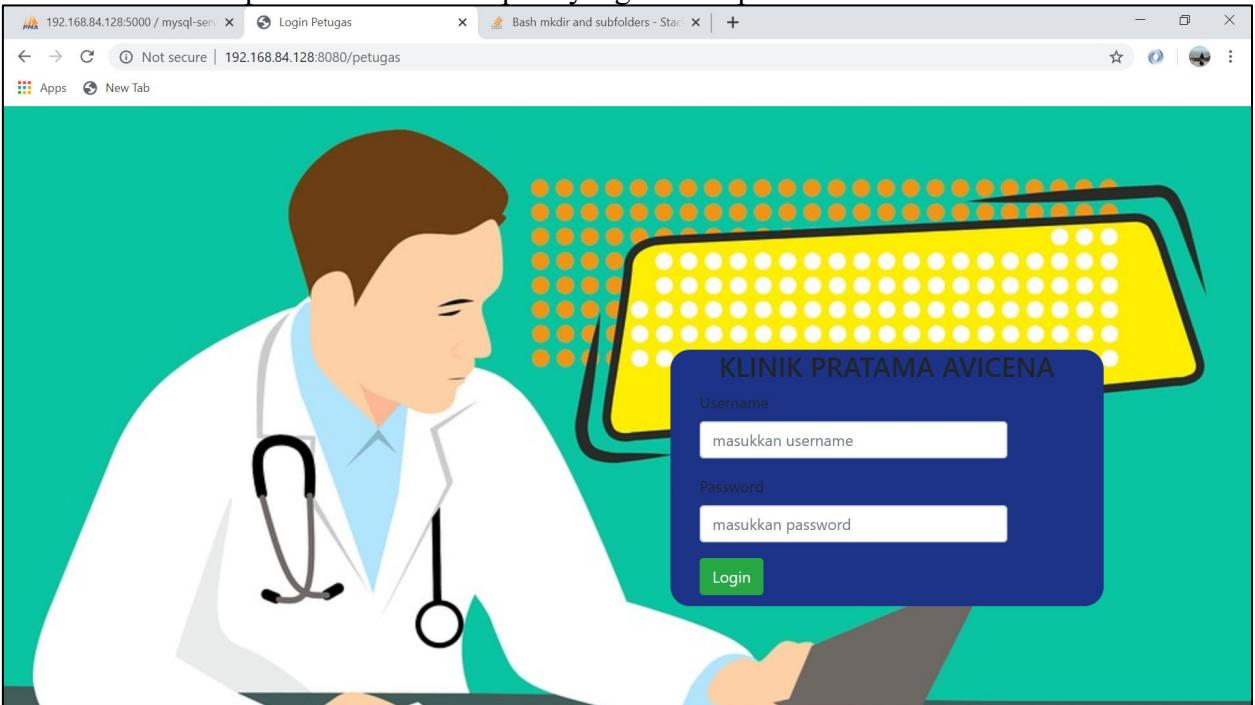


Gambar 2.35 Tampilan berhasil hapus data Kepala Keluarga
Percobaan berikutnya yaitu menggunakan fitur *search* pada data Kepala Keluarga dan fitur tersebut berhasil dilakukan seperti yang terlihat pada **Gambar 2.36** berikut ini :



Gambar 2.36 Tampilan berhasil search data Kepala Keluarga

Percobaan berikutnya yaitu menggunakan fitur *logout* sistem, *logout* berhasil dilakukan dan kembali ketampilan awal sistem seperti yang terlihat pada **Gambar 2.37** berikut ini :



Gambar 2.37 Tampilan berhasil *logout* sistem

Hasil dari apa yang telah dilakukan pengguna tersebut adalah mengolah data Kepala Keluarga. Hal ini dapat terlihat bahwa waktu yang dibutuhkan hanya 5 menit dan dapat dilakukan dari mana saja selama ada koneksi internet.

Kesimpulan dari percobaan tersebut bahwa sistem yang dibuat mampu memberikan layanan yang sangat bermanfaat bagi pengguna sistem, selain dapat diakses dari mana saja, biaya perbaikan berkala jauh lebih murah.

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

N o	Jenis Tugas	Waktu Pengerjaan							
		Maret				April			
		1	2	3	4	1	2	3	4
1	Analisa Persoalan								
2	Pembagian Tugas								
3	Pembuatan VM								
4	Install Ubuntu dan LAMPP								
5	Upload web								
6	Uji coba docker								
7	Pembuatan Laporan								

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

N	Keterangan Tugas	Penanggung
1	Penginstallan Ubuntu Admin Server	Adit
2	Penginstallan Lampp	Adit
3	Latar Belakang Masalah	Adit
4	Pembuatan Laporan	Adit & Cici
5	Agenda Pengerjaan Proyek	Cici
6	Instal docker dan implementasi dockerfile dan docker-	Cici
7	Pengujian Singkat	Cici

BAB IV KESIMPULAN DAN SARAN

4.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan didapat kesimpulan bahwa :

1. Sistem dapat berjalan dengan lancar
2. Berhasil menerapkan *cloud computing* pada sistem
3. Berhasil melakukan *hosting local*
4. Penerapan dockerfile dan docker-compose berhasil dengan baik

4.2 Saran

Pengerjaan seharusnya dapat dilakukan dengan semaksimal mungkin, agar mendapatkan hasil yang maksimal.

DAFTAR PUSTAKA

Anggeriana, H., Kom, S., & Kom, M. (2011). Cloud Computing. *Jurnal Teknik Informatika*, 1.

Mutai, Josphat.(10 November 2019),Install Docker and Docker Compose on Linux Mint 19. Diakses dari <https://computingforgeeks.com/install-docker-and-docker-compose-on-linux-mint-19/>

Install Docker Engine on Ubuntu. Diakses pada 16 Mei 2020, dari <https://docs.docker.com/engine/install/ubuntu/>

Mufrizal, Rizki. (01 Agustus 2016), Belajar Docker. Diakses dari <https://rizzimufrizal.github.io/belajar-docker/>

Shovan, Shariar. Set up a LAMP server with Docker. Diakses pada 16 Mei 2020 dari https://linuxhint.com/lamp_server_docker/

LAMPIRAN

