

**PRAKTIKUM TEKNOLOGI CLOUD
COMPUTING LAPORAN PROYEK AKHIR**

**APLIKASI WEB PROFIL STASIUN TELEVISI REPUBLIK
INDONESIA(TVRI) PROVINSI DAERAH ISTIMEWA
YOGYAKARTA OS UBUNTU MENGGUNAKAN LAMPP DAN
KONVERSI MENJADI *DOCKER***



DISUSUN OLEH:

NAMA ANGGOTA : GINA SONIA 123170001

SHINDY INTAN WULANDARI 123170007

KELAS : E

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

APLIKASI WEB PROFIL STASIUN TELEVISI REPUBLIK INDONESIA(TVRI) PROVINSI DAERAH ISTIMEWA YOGYAKARTA OS UBUNTU MENGGUNAKAN LAMPP DAN KONVERSI MENJADI *DOCKER*

Disusun oleh :

Gina Sonia

123170001

Shindy Intan Wulandari

123170007

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 Florestivanto. S.T., M.Eng.

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga kami dapat menyelesaikan Tugas Praktikum Teknologi *Cloud computing*. Laporan proyek akhir ini merupakan salah satu wujud tugas yang disyaratkan oleh asisten praktikum kepada pratikan mata kuliah praktikum ini.

Pada tugas ini kami mengembangkan sebuah Aplikasi Web Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta. Aplikasi ini dikembangkan dengan menggunakan Ubuntu *LAMPP* yang kemudian akan dikonversi menjadi *dockerfile*.

Laporan Praktikum Teknologi *Cloud computing* ini tentunya masih jauh dari kata sempurna, maka dari itu kami mengharapkan adanya kritik dan saran yang membangun agar dapat menjadi bahan evaluasi selanjutnya. Akhir kata, semoga laporan ini dapat bermanfaat bagi semua pihak dan penyusun sendiri pada khususnya. Terima kasih
Wassalamu'alaikum Wr. Wb.

Yogyakarta, 28 Maret 2020

Penyusun

DAFTAR ISI

Halaman Judul	i
Halaman Pengesahan	ii
Kata Pengantar	iii
Daftar Isi	iv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	2
1.2. Tujuan Projek Akhir	2
1.3. Manfaat Projek Akhir	2
1.4. Tahap Penyelesaian Projek Akhir	2
BAB II ISI DAN PEMBAHASAN	4
2.1 Komponen yang Digunakan	4
2.2 Rancangan Arsitektur <i>Cloud computing</i>	7
2.3 Parameter dan Konfigurasi	8
2.4 Tahap Implementasi	14
2.5 Hasil Implementasi	20
2.6 Pengujian Singkat	22
BAB III JADWAL Pengerjaan dan Pembagian Tugas	25
3.1 Agenda Pengerjaan	25
3.2 Keterangan Pembagian Tugas	25
BAB IV KESIMPULAN DAN SARAN	26
4.1. Kesimpulan	26
4.2. Saran	26
DAFTAR PUSTAKA	27
LAMPIRAN	

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Cloud computing adalah suatu model komputasi, dimana sumber daya seperti *processor*, *storage*, *network*, dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan/internet dengan menggunakan pola akses *remote*. Ketersediaan on-demand sesuai kebutuhan, mudah untuk dikontrol, dinamik dan skalabilitas yang hampir tanpa batas adalah beberapa atribut penting dari *cloud computing*. (Johnson dkk, 2010).

Cloud computing menerapkan suatu metode komputasi, yaitu kemampuan yang terkait teknologi informasi yang disajikan sebagai suatu layanan yang diakses melalui internet, tanpa mengetahui infrastruktur didalamnya, tenaga ahli yang merancang sistem tersebut atau memiliki kendali atas infrastruktur yang ada. Arsitektur secara umum terbagi menjadi 3 bagian yaitu infrastruktur, platform dan aplikasi. Setiap layanan yang diakses tidak perlu diinstal pada setiap perangkat end-user, untuk dapat melakukan akses terhadap layanan *cloud computing* hanya dibutuhkan web browser atau interface program (Ernawati, 2013).

Dalam kehidupan nyata, penggunaan *cloud computing* dapat dilihat pada penggunaan Amazon Web Services (AWS). Amazon Web Services memiliki banyak layanan, dan fitur dalam layanannya dibandingkan penyedia *cloud* lainnya, mulai dari teknologi infrastruktur seperti penghitungan, penyimpanan, dan database sehingga teknologi yang berkembang, seperti dan kecerdasan buatan, data lake dan analitik, dan *Internet of Things*. Hal ini membuat seluruh proses komputasi data menjadi lebih cepat, lebih mudah, dan lebih hemat biaya.

Kami membangun sebuah projek yang berjudul “Aplikasi Web Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker*”. Projek ini dibangun untuk membantu pihak Stasiun TVRI Provinsi DIY dalam mengelola profilnya agar lebih menarik bagi pengunjungnya. Adapun sistem ini dibuat berbasis web karena dapat diakses tanpa perlu menginstall software khusus selain web browser. Kemudian sistem ini nantinya akan kami jadikan sebagai *dockerfile* sehingga sistem dapat dipindah ke berbagai perangkat dengan lebih mudah.

1.2 Tujuan Proyek Akhir

Terhadap judul proyek “Aplikasi Web Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker*” maka dengan laporan ini akan dituliskan cara penyelesaian dengan menggunakan :

1. *Ubuntu LAMPP*
2. *Docker*
3. *VMWare Workstation*
4. *Database phpMyAdmin*

Sehingga menghasilkan sebuah Aplikasi Web Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker* yang berbasis web ini bertujuan untuk mempermudah segala kegiatan yang dilakukan dan memperoleh beberapa kelebihan antara lain :

1. Mudah diakses dimana saja dan kapan saja selama terkoneksi internet
2. Media penyimpanan yang berkapasitas besar karena penggunaan *cloud computing*
3. Performa sistem yang lebih baik karena penerapan *Docker*
4. Jaminan keamanan data yang tersimpan tinggi karena disimpan di server terpusat

1.3 Manfaat Proyek Akhir

Dengan dibuatnya proyek akhir ini, maka manfaat yang diperoleh adalah pengguna dapat menggunakannya dengan lebih mudah. Hal ini dikarenakan *dockerfile* sudah *include environment* yang dibutuhkan untuk menjalankan web Aplikasi Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta. Sehingga proses komputasinya menjadi lebih cepat dan tidak perlu mengkonfigurasi banyak hal. Selain itu, biaya dalam perawatan sistem juga tidak mahal, hanya biaya rutin sesuai penggunaan, dan mengurangi pengeluaran perusahaan karena tidak ada lagi biaya tambahan untuk pembelian inventaris seperti infrastruktur, hardisk, dan lain-lain.

1.4 Tahap Penyelesaian Proyek Akhir

Dalam pembuatan proyek akhir Aplikasi Web Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker* ini, beberapa tahapan yang perlu dilakukan

yaitu:

1. Menentukan judul projek akhir dan rancangan sistem projek akhir yang akan dibuat
2. Mempersiapkan komponen software yang dibutuhkan seperti *VMWare Workstation* atau *Virtual Box*, *file ISO Docker*, *ISO Ubuntu*, dan lain-lain.
3. Melakukan Instalasi Virtual Machine yang akan digunakan sebagai Virtual Machine sistem operasi Ubuntu dan software lain yang dibutuhkan seperti *Winscp* dan *Putty*
4. Melakukan instalasi Ubuntu server yang akan digunakan sebagai sistem operasi penunjang *cloud computing*
5. Melakukan installasi *PHP*, *Mysql*, dan *Apache* yang kan digunakan sebagai media installasi dan software pada ubuntu
6. Melakukan konfigurasi *PHP*, *Mysql*, *Apache*, dan lain-lain sesuai dengan kebutuhan
7. Testing program dan membuat topologi

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir Aplikasi Web Profil Stasiun Televisi Republik Indonesia(Tvri) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker* ini dijabarkan sebagai berikut :

1. Spesifikasi laptop yang kami gunakan antara lain ASUS VivoBook S14 A411U CPU intel core i5-8250U, Memory 8GB.
2. Ubuntu 18.04.3 LTS
3. LAMPP
4. Apache
5. *Docker 2.2.0.5 (44384) stable*
6. Vmware Workstation versi 6.7
7. Putty
8. Winscp

Pada point- point yang telah ditulis di atas akan dibahas keduanya masing-masing dalam sub bab terkait, antara lain:

2.1.1 Komponen pada Proyek Pertama

Komponen yang digunakan untuk membangun Aplikasi Web Profil Stasiun (TVRI) Provinsi DIY OS Ubuntu Menggunakan LAMPP adalah sebagai berikut :

1. Laptop

Proyek akhir praktikum Teknologi *Cloud computing* ini kami kerjakan dengan menggunakan Laptop Asus Vivibook S14 A411U dengan spesifikasi sebagai berikut:

- Processor Intel core I5-8250U CPU 3.4 GHz
- RAM 8192 Mb
- OS Windows 10 Pro 64-bit

2. Virtual Machine

Virtual machine merupakan software yang dapat melakukan virtualisasi OS computer sehingga satu computer dapat menjalankan beberapa OS computer sekaligus. Adapun Virtual Machine yang kami gunakan yaitu VMware-workstation dengan versi 15.0.1.

3. Ubuntu live-server

Ubuntu server adalah sistem operasi server. Ubuntu server dapat melayani situs web, pembagian file seperti cloud storage dan container. Pada proyek kali ini kami menggunakan ubuntu live-server 18.04. Adapun file iso dari Ubuntu live server dapat diunduh pada link berikut:

<http://kambing.ui.ac.id/iso/ubuntu/releases/18.04/ubuntu-18.04-live-server-amd64.iso>.

4. LAMP

LAMP merupakan kombinasi dari *Linux*, *Apache*, *MySQL* dan *PHP*. Kehadiran dari LAMP dapat digunakan untuk membangun web server yang berfungsi sepenuhnya. Pada proyek ini kami menggunakan LAMP untuk membangun Aplikasi Web Profil Stasiun (TVRI) Provinsi DIY. Adapun LAMP yang kami gunakan yaitu Linux Ubuntu live server 18.04, *Apache*, *MySQL* dan *PHP* versi 7.3.

5. Web Service

Web services adalah mekanisme komunikasi dua aplikasi/mesin terlepas dari arsitektur dan teknologi yang digaribawahi. Layanan web memiliki antarmuka, yang menyembunyikan detail implementasi sehingga dapat digunakan secara terpisah dari platform perangkat keras atau perangkat lunak yang digunakannya, dan secara terpisah dari bahasa pemrograman yang digunakan. *Web service* dapat dibangun dengan menggunakan berbagai bahasa seperti *ASP*, *Perl*, *Java*, *Java Script*, *PHP*, *Python*, *Ruby*, dll. Dalam menyelesaikan tugas ini kami membangun aplikasi web menggunakan bahasa pemrograman PHP.

6. Putty

PuTTY adalah sebuah aplikasi *open-source* memanfaatkan protokol jaringan seperti SSH dan Telnet. PuTTY memanfaatkan protokol tersebut untuk mengaktifkan sesi remote pada komputer. Link download Putty sebagai berikut: <https://www.putty.org/>

7. Winscp

WinSCP merupakan aplikasi SSH Client berbasis sistem operasi Windows yang bersifat open source. Fokus utama WinSCP adalah mengirim file ke remote komputer (server) menggunakan port SSH yang lebih aman. Dengan demikian, WinSCP juga merupakan sebuah aplikasi SFTP Client, SCP client

dan *FTP Client*. Link download *Winscp* sebagai berikut:
<https://winscp.net/eng/download.php>

Berikut adalah komponen utama penyusunan *cloud computing* yang dibutuhkan, dapat disimpulkan dalam bentuk tabel berikut:

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	Intel Core i5-8250U CPU @ 3.4 GHz	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: 192.168.1.8/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 127.0.0.53	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.100.1	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	8 GB	Alokasi RAM untuk <i>guest OS</i>

Tabel 2.1 Spesifikasi VM *cloud computing* untuk proyek pertama

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

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Dalam pembuatan dan pengembangan web sistem yang telah dibuat.
		PHPMyadmin	Layanan untuk manajemen database
2.	PHP	Versi 7.4.5	Dukungan bahasa pemrograman yang digunakan oleh Aplikasi Web Profil Stasiun (TVRI) Provinsi DIY
3.	<i>Docker</i>	Versi 19.03.8, build afacb8b7f0	aplikasi yang bersifat open source yang berfungsi sebagai wadah/container untuk mengepak/memasukkan sebuah software secara lengkap beserta semua hal lainnya yang dibutuhkan oleh software tersebut dapat berfungsi
4.	<i>Docker Compose</i>	Versi 1.21.2, build a133471	aplikasi yang bersifat open source yang berfungsi untuk mendefinisikan dan menjalankan banyak container dalam aplikasi <i>Docker</i>

Tabel 2.2 Spesifikasi Ubuntu OS

2.1.2 Komponen pada Proyek Kedua

Komponen yang digunakan untuk membangun Aplikasi Web Profil Stasiun

(TVRI) Provinsi DIY pada *Docker* adalah sebagai berikut :

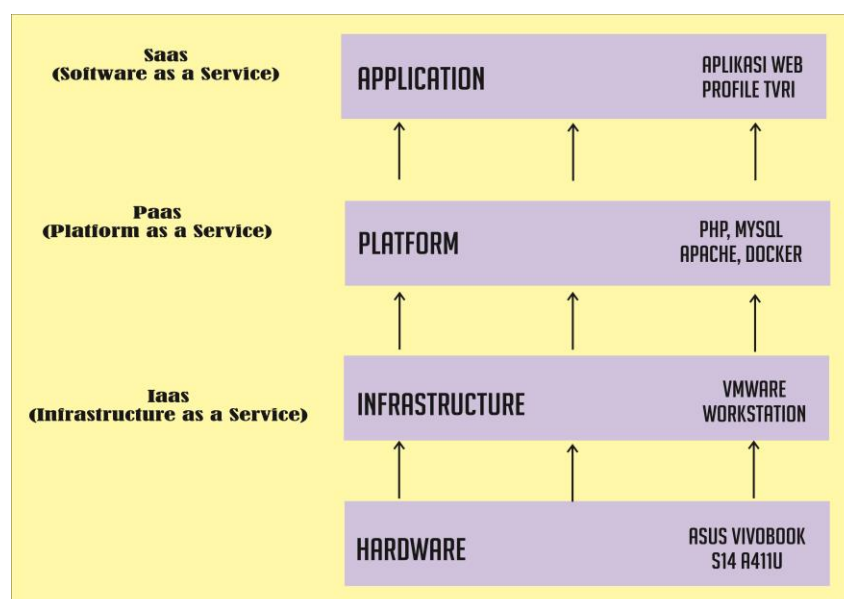
1. *Docker*

Docker adalah alat yang didesain untuk memudahkan kita dalam membuat, mendeploy dan menjalankan aplikasi dengan menggunakan container. Dalam membangun aplikasi biasanya kita perlu menyiapkan environment-nya seperti libraries, beberapa software dan konfigurasi serta dependencies lainnya. Container mengizinkan kita sebagai developer untuk menggabungkan seluruh environment tersebut menjadi satu paket dan mendeploynya.

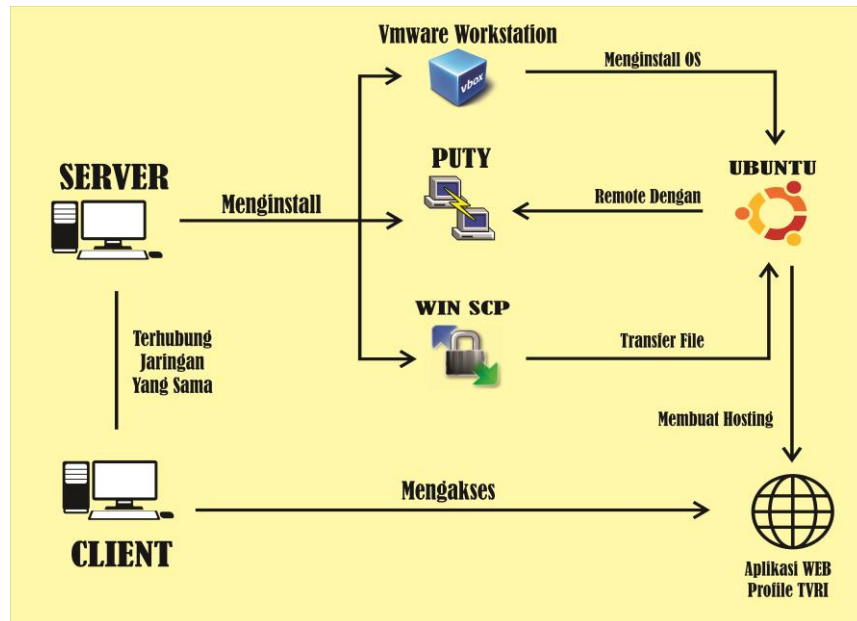
2. Projek Web *service* sebelumnya yakni Aplikasi Web Profil Stasiun (TVRI) Provinsi DIY OS Ubuntu Menggunakan LAMPP yang nantinya akan dikonversi menjadi *docker file*.

2.2 Rancangan Arsitektur *Cloud computing*

Pada project akhir ini digunakan bentuk rancangan arsitektur Iaas dimana hardware laptop ASUS VivoBook S14 A411U sebagai layer utama di bagian bawah, kemudian di bagian atasnya adalah Paas atau sistem operasi dan aplikasi Vmware Workstation yang berjalan, kemudian di atasnya lagi adalah Saas yakni aplikasi web dengan judul “Aplikasi Web Profil Stasiun Televisi Republik Indonesia(Tvri) Provinsi Daerah Istimewa Yogyakarta” Os Ubuntu Menggunakan Lampp dan juga akan di konversi menjadi *Docker*. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya



Gambar 2.2 Ilustrasi topologi proyek LAMPparameter dan Konfigurasi

2.3 Parameter dan Konfigurasi

Berikut merupakan parameter-parameter yang digunakan untuk melakukan instalasi dan konfigurasi layanan hosting dengan LAMP yang telah dibuat dan kemudian akan dikonversi menjadi *docker*.

Proses instalasi dan konfigurasi layanan hosting pada Lamp dimulai dengan pembaruan package dan 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

Selanjutnya adalah melakukan pengaturan Firewall, Ubuntu dengan versi terbaru memiliki default pengaturan firewall yang aktif otomatis. Oleh karena itu, diperlukan parameter untuk mengaturnya agar lalu lintas data dari Apache Web Server dengan tipe protocol HTTP (port 80) dan HTTPS (port 443) diijinkan. Penjelasan **Modul 2.2** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- ufw : konfigurasi Firewall
- allow : parameter perintah ijin melakukan konfigurasi
- in "Apache Full" : tempat konfigurasi firewall dituju

Modul 2.2 Parameter pengaturan Firewall

Kemudian lakukan pengecekan terhadap Web Server dengan cara http/IP server. Selanjutnya melakukan instalasi mysql-server yang dijelaskan pada **Modul 2.3** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : merupakan perintah untuk melakukan proses pembaruan package
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- mysql-server : proses pembaruan package Apache2

Modul 2.3 Parameter instalasi MySQL

Selanjutnya adalah melakukan pengaturan dasar pengaman MySQL yang berupa users, kata sandi, hak akses, dan lain-lain akan dijelaskan pada **Modul 2.4** berikut dibawah ini, setelah itu pilih perintah "Y" untuk setiap pertanyaan sampai "All Done!" yang menandakan proses pengaturan MySQL selesai.

```
$ sudo apt install mysql_secure_installation
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- mysql_secure_installation : proses pengaturan dasar pengaman MySQL

Modul 2.4 Pengaturan dasar pengaman MySQL

Selanjutnya melakukan instalasi PHP, dijelaskan pada **Modul 2.5** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- php : nama package yang akan diinstall
- libapache2-mod-php : installasi modul Apache PHP
- php-mysql : installasi package PHP dan MySQL

Modul 2.5 Parameter instalasi PHP

Selanjutnya adalah membuat sebuah berkas dengan dengan format .php pada lokasi direktori Web Server Apache yakni di /var/www/html/info.php menggunakan nano yang akan dijelaskan pada **Modul 2.6** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- nano : merupakan aplikasi seperti notepad pada windows namun berbasis CLI
- /var/www/html : lokasi direktori Web Server Apache
- Info.php : nama berkas yang akan dibuat dengan format .php

Modul 2.6 Pembuatan berkar dengan format .php

Selanjutnya adalah mengisi berkas tadi, dengan isi seperti **Modul 2.7** dibawah ini. Kemudian melakukan pengecekan pada <http://ipserver/info.php>.

```
<?php
Phpinfo()
?>
```

Modul 2.7 Listing program info.php

Selanjutnya melakukan instalasi parameter yang digunakan untuk package PHPMyAdmin, dapat dilihat pada penjelasan **Modul 2.8** berikut ini. Kemudian dapat dilakukan pengecekan pada alamat berikut <http://ipserver/phpmyadmin>.

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- phpMyAdmin : nama paket yang akan diinstall
- php-mbstring dan php-gettext : fungsi untuk memanggil perintah string dan memperoleh text pada PHPMyAdmin.

Modul 2.8 Parameter instalasi PHPMyAdmin

Proses instalasi dan konfigurasi parameter pada layanan host LAMPP telah selesai selanjutnya adalah konfugarsi dan instalasi parameter pada *Docker*. Berikut ini merupakan parameter-parameter yang digunakan untuk melakukan konversi dari Ubuntu Lampp menjadi *Docker*. Langkah awal yang perlu dilakukan adalah update list packages terlebih. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.9** berikut ini:

```
$ sudo apt update
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : merupakan perintah untuk melakukan proses pembaruan package

Modul 2.9 Parameter update list packages

Selanjutnya melakukan instalasi beberapa packages agar dapat diakses oleh HTTPS. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.10** berikut ini:

```
$ sudo apt install apt-transport-https ca-certificates curl 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 aplikasi
- apt-transport-https, ca-certificate, curl, gnupg-agent, softwareproperties-common : nama paket aplikasi yang akan diinstall

Modul 2.10 Parameter install packages

Selanjutnya, setelah packages di update dan diinstal adalah melakukan penambahan GPG Key dari *Docker* repository ke dalam sistem yang akan dibuat.

Parameter yang akan digunakan akan dijelaskan pada **Modul 2.11** berikut ini:

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

Keterangan:

- curl : berfungsi untuk mentransfer data menggunakan protokol jaringan
- apt-key : untuk manajemen key yang digunakan oleh apt untuk melakukan autentikasi pada package
- add - : digunakan untuk menambahkan key

Modul 2.11 Parameter penambahan GPG key

Selanjutnya adalah penambahan *Docker repository* dengan versi *stable*.

Parameter yang akan digunakan akan dijelaskan pada **Modul 2.12** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- add-apt-repository : perintah untuk menambahkan repository
- stable : versi repository yang akan diinstall

Modul 2.12 Parameter penambahan *Docker repository*

Selanjutnya melakukan update package database dan dilanjutkan melakukan instalasi repositori ke dalam Ubuntu. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.13** berikut ini:

```
$ sudo apt update
$ apt-cache policy docker-ce
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : merupakan perintah untuk melakukan proses pembaruan

- package
- apt-cache policy : Perintah untuk memanipulasi dan berisi informasi dari package apt's cache
- docker-ce : paket aplikasi untuk *docker*

Modul 2.13 Parameter instalasi package database dan *Docker* repo

Selanjutnya instalasi *Docker*. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.14** berikut ini:

```
$ sudo apt install docker-ce
$ sudo systemctl status docker
```

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
- docker-ce : paket aplikasi untuk *docker*
- systemctl status: Untuk memastikan status *docker* telah berjalan

Modul 2.14 Parameter instalasi *Docker*

Selanjutnya adalah mendownload *docker-compose* binary file dan menginstall *curl*. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.15** 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
$ sudo apt install curl -y
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- curl -L : perintah untuk mentransfer data
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi

Modul 2.15 Parameter download *docker-compose* binary file dan instalasi *curl*

Selanjutnya adalah melakukan instalasi *docker* compose yang sudah di download sebelumnya dan memastikan apakah *docker* compose sudah dapat berjalan.

Parameter yang akan digunakan akan dijelaskan pada **Modul 2.16** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- chmod +x : merupakan perintah untuk melakukan eksekusi
- /usr/local/bin/docker-compose : instalasi *docker* compose pada folder /usr/local/bin/
- docker-compose version : perintah untuk mengetahui versi *docker* compose agar dapat memastikan *docker-compose* telah berjalan

Modul 2.16 Parameter instalasi *docker-compose*

Apabila *Docker* dan *docker* compose sudah dipastikan dapat berjalan dengan baik, langkah selanjutnya adalah mengatur *docker* compose dan dilanjutkan dengan

pembuatan *Dockerfile*, dapat dijelaskan pada **Modul 2.17** berikut ini:

```
$ cd /var/www/html/TVRI/
$ nano php.Dockerfile
$ nano docker-compose.yaml
```

Keterangan:

- `cd` : perintah untuk mengubah directory
- `/var/www/html/TVRI` : merupakan directory
- `nano` : merupakan aplikasi seperti notepad pada windows namun berbasis CLI
- `php.Dockerfile` : sebuah berkas
- `docker-comose.ymail` : sebuah berkas

Modul 2.17 Parameter mengatur folder proyek dan pembuatan *Dockerfile*nya

Selanjutnya menjalankan fungsi web-server, mysql-server, dan phpmyadmin pada *Docker*. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.18** berikut ini:

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

Keterangan:

- `docker-compose up` : perintah untuk mengagregasikan output dari setiap container
- `-d` : parameter dari `docker-compose` untuk menjalankan *containers* di background

Modul 2.18 Parameter menjalankan web-server, mysql-server, dan phpmyadmin

Parameter yang digunakan untuk melihat *containers Docker* yang sudah terpasang sebelumnya pada **Modul 2.18** dapat dilihat pada penjelasan **Modul 2.18** berikut ini:

```
$ docker ps -a
```

Keterangan : Untuk melihat *containers Docker* yang terpasang.

- `docker` : perintah untuk menjalankan *docker*
- `ps` : perintah dari *docker* untuk menampilkan *containers docker* yang aktif
- `-a` : parameter dari *docker* untuk menampilkan seluruh *containers docker* baik yang aktif maupun tidak aktif

Modul 2.18 Parameter melihat *containers Docker* yang terpasang

Parameter yang digunakan untuk melihat ip address dapat dilihat pada penjelasan **Modul 2.19** berikut ini:

```
$ ip addr
```

Keterangan : Untuk melihat ip address.

- `ip` : perintah untuk menjalankan ip yang akan menampilkan seluruh interface jaringan yang ada beserta konfigurasinya seperti ip address
- `addr` : perintah dari ip untuk menampilkan ip address pada setiap interface jaringan yang ada

Modul 2.19 Parameter melihat ip address

Parameter yang digunakan untuk menghentikan *containers Docker* yang aktif sebelumnya, dapat dilihat pada penjelasan **Modul 2.20** berikut ini:

```
$ docker-compose down
```

Keterangan : Untuk menghentikan *containers docker* yang aktif.

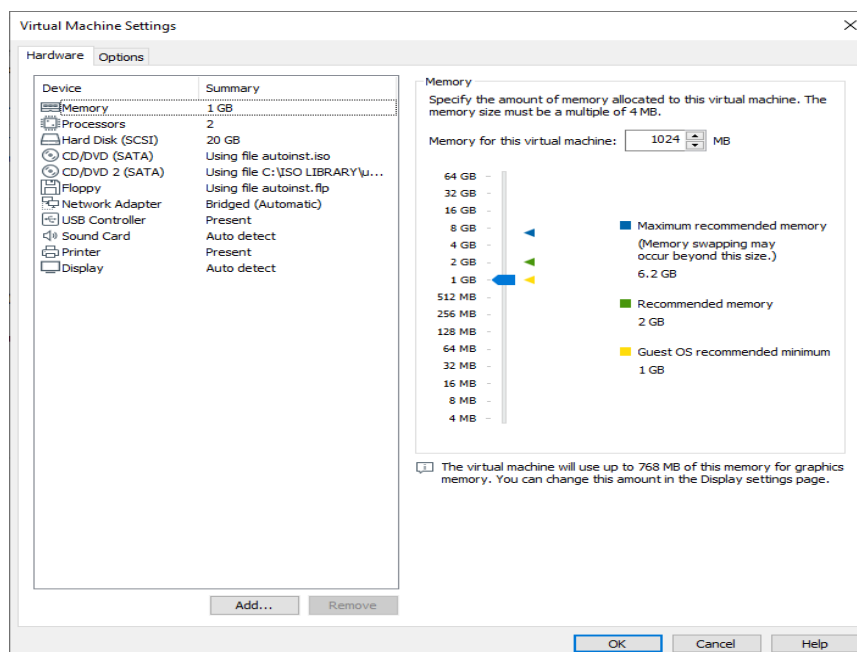
- `docker-compose` : perintah untuk menjalankan *docker-compose*
- `down` : perintah dari *docker-compose* untuk menghentikan

containers docker yang aktif

Modul 2.20 Parameter menghentikan *containers Docker* yang aktif

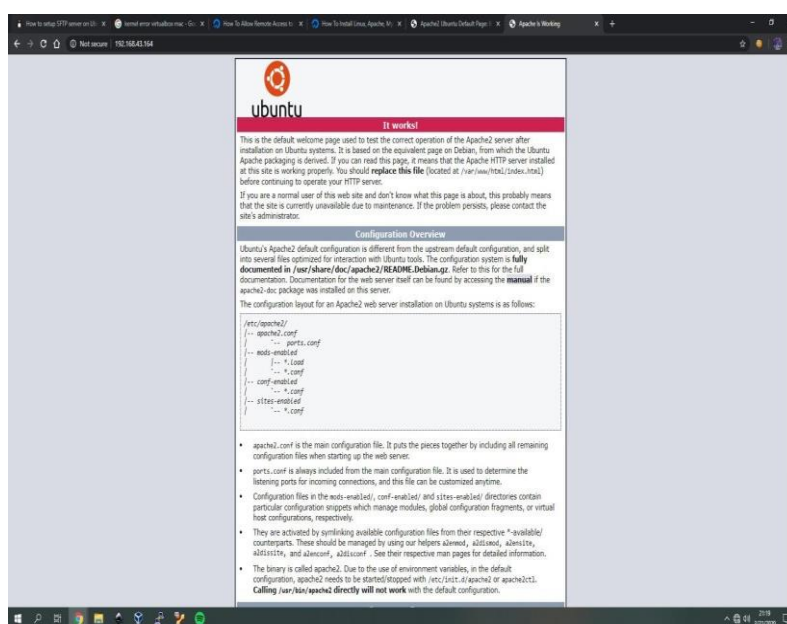
2.4 Tahap Implementasi

Untuk mengimplementasikan “Aplikasi Web Profil Stasiun Televisi Republik Indonesia(Tvri) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker*” berbasiskan konsep *cloud computing*, maka tahapan yang perlu dilakukan adalah sebagai berikut:

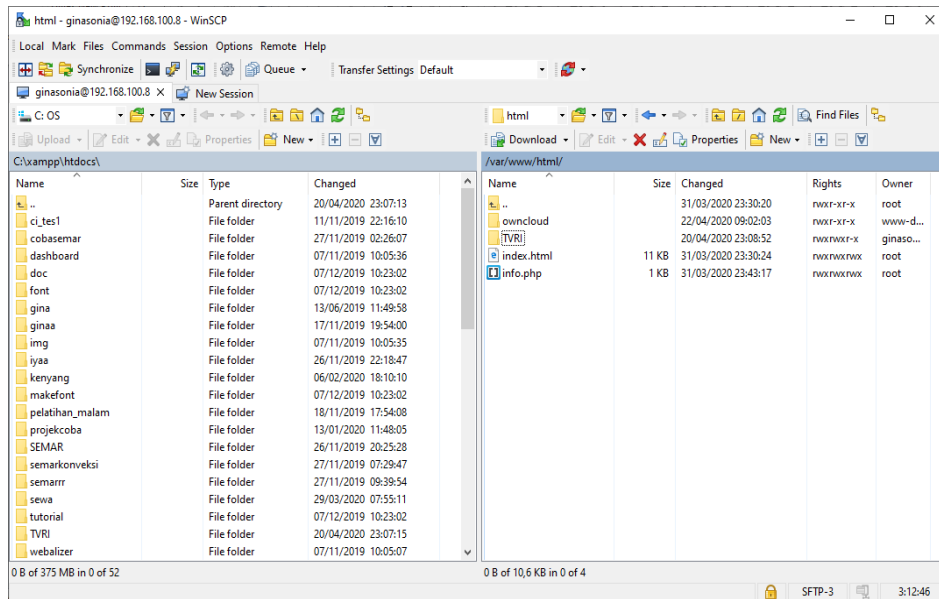


Gambar 2.3 Seting *mode wizard* pembuatan VM

Hasil implementasi dari **Modul 2.1** tentang cara untuk instalasi Apache2 dapat dilihat pada **Gambar 2.4** berikut ini:



Gambar 2.4 Hasil instalasi Apache2



Gambar 2.7 Tampilan hasil pemindahan folder menggunakan Winscp

Hasil implementasi sampai dengan **Modul 2.13** tentang cara instalasi *Docker* dapat dilihat pada **Gambar 2.8** berikut ini:

```

ginasonia@ginasonia: ~
Processing triggers for systemd (237-3ubuntu0.38) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
ginasonia@ginasonia:~$ 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 Fri 2020-05-15 16:25:54 UTC; 42s ago
     Docs: https://docs.docker.com
    Main PID: 13454 (dockerd)
      Tasks: 10
     CGroup: /system.slice/docker.service
             └─13454 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

May 15 16:25:53 ginasonia dockerd[13454]: time="2020-05-15T16:25:53.283316678Z"
May 15 16:25:53 ginasonia dockerd[13454]: time="2020-05-15T16:25:53.283377157Z"
May 15 16:25:53 ginasonia dockerd[13454]: time="2020-05-15T16:25:53.283387356Z"
May 15 16:25:53 ginasonia dockerd[13454]: time="2020-05-15T16:25:53.283591998Z"
May 15 16:25:53 ginasonia dockerd[13454]: time="2020-05-15T16:25:53.949953462Z"
May 15 16:25:54 ginasonia dockerd[13454]: time="2020-05-15T16:25:54.286219375Z"
May 15 16:25:54 ginasonia dockerd[13454]: time="2020-05-15T16:25:54.364110720Z"
May 15 16:25:54 ginasonia dockerd[13454]: time="2020-05-15T16:25:54.364215500Z"
May 15 16:25:54 ginasonia dockerd[13454]: time="2020-05-15T16:25:54.433370451Z"
May 15 16:25:54 ginasonia systemd[1]: Started Docker Application Container Engin
lines 1-19/19 (END)

```

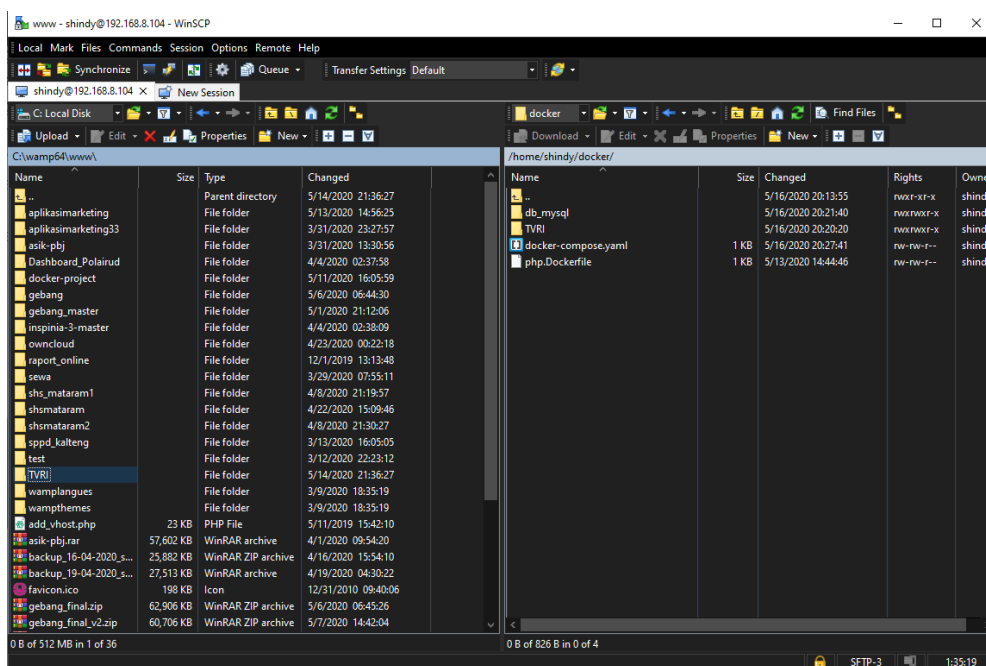
Gambar 2.8 Tampilan hasil instalasi *docker*

Hasil implementasi dari Modul **2.16** tentang cara instalasi *Docker Compose* dapat dilihat pada **Gambar 2.9** berikut ini:


```

1  version: '3.3'
2  services:
3    webserver:
4      build:
5        dockerFile: php.Dockerfile
6        context: .
7      restart: always
8      volumes:
9        - ./TVRI:/var/www/html/"
10     ports:
11       - '8081:80'
12
13    mysql-server:
14      image: mysql:8.0.19
15      restart: always
16      environment:
17        MYSQL_USERNAME: root
18        MYSQL_PASSWORD: root
19        MYSQL_DATABASE: tvrii
20        MYSQL_ROOT_PASSWORD: root
21      volumes:
22        - mysql-data:/var/lib/mysql
23        - ./TVRI/db_mysql/tvrii.sql:/docker-entrypoint-initdb.d/tvrii.sql
24     ports:
25       - "6448:6448"
26
27    phpmyadmin:
28      image: phpmyadmin/phpmyadmin:5.0.1
29      restart: always
30      environment:
31        PMA_HOST: mysql-server
32        PMA_USER: root
33        PMA_PASSWORD: root
34     ports:
35       - "3307:80"
36     volumes:
37       mysql-data:
38

```

Gambar 2.11 Tampilan *docker-compose.yml*Gambar 2.12 Tampilan pada Winscp setelah konfigurasi *docker*

Hasil implementasi *dockercompose.yml* dengan *docker-compose* dapat dilihat pada Gambar 2.13 berikut ini:

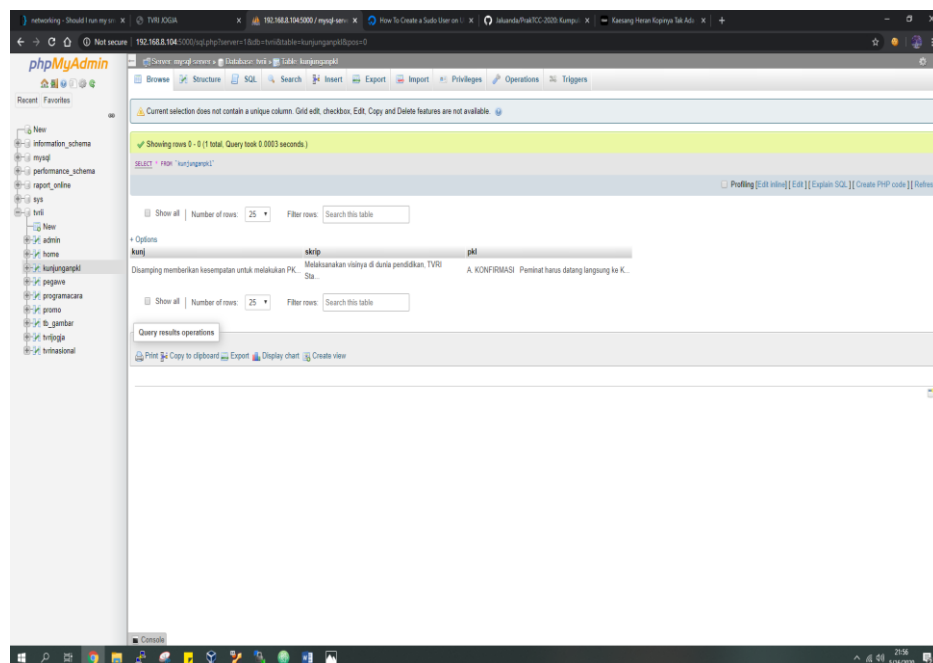

```

ginasonia@ginasonia: /var/www/html
8ec398bc0356: Pull complete
85cf4fc86478: Pull complete
970dadf4ccb6: Pull complete
8c04561117a4: Pull complete
d6b7434b63a2: Pull complete
83d8859e9744: Pull complete
9c3d824d0ad5: Pull complete
0ff2f3c2c8ab: Pull complete
f7a2cdcb0840: Pull complete
fe8c2411b50b: Pull complete
aa0cb4375001: Pull complete
96198bflad68: Pull complete
5fe54d7827f9: Pull complete
f8f818be7009: Pull complete
f91e44aee5b4: Pull complete
775a97d697e6: Pull complete
1bf92c73fccf: Pull complete
28733a694f7e: Pull complete
Digest: sha256:d2191935bda8c825bcf51dlc659efcd4d80a4f5da907ea72c8517c8144136de4
Status: Downloaded newer image for phpmyadmin/phpmyadmin:5.0.1
Creating html_web-server_1 ... done
Creating html_phpmyadmin_1 ... done
Creating html_mysql-server_1 ... done
ginasonia@ginasonia: /var/www/html$

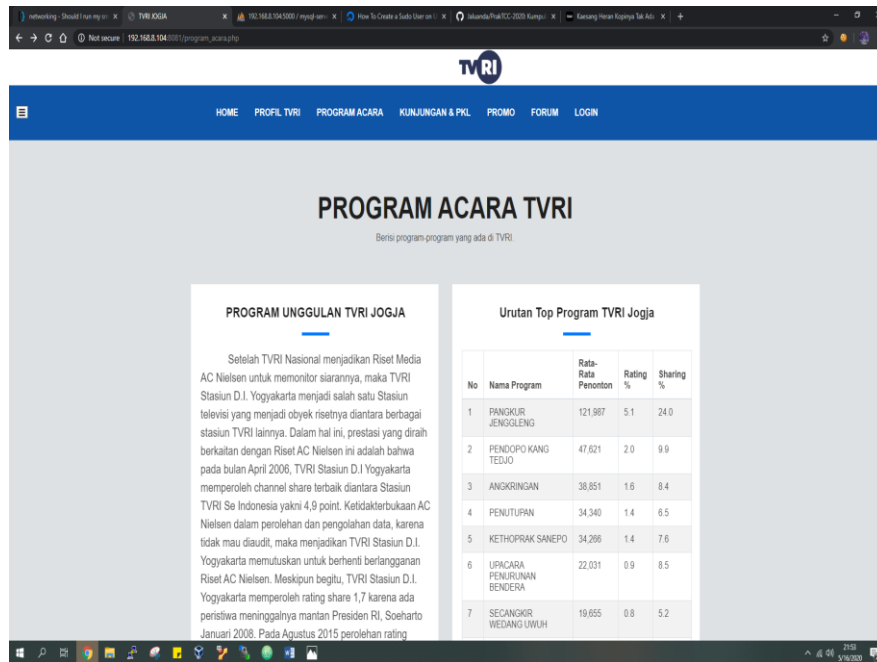
```

Gambar 2.13 Tampilan hasil menjalankan *docker compose*

Setelah semua *containers* aktif, selanjutnya adalah membuka website phpmyadmin dengan mengakses alamat_ip:5000 dan website raport online alamat_ip:8000 dapat dilihat pada Gambar 2.18, **Gambar 2.14**, dan **Gambar 2.15** berikut ini:



Gambar 2.14 Tampilan akses ip:5000 ke database



Gambar 2.14 Tampilan akses ip:8081 akses ke web profil TVRI

Hasil implementasi dari **Modul 2.20** tentang cara menghentikan *containers docker* yang aktif dapat dilihat pada **Gambar 2.15** berikut ini :

```
shindy@shindy-server:~/docker$ docker-compose up -d
Recreating docker_phpmyadmin_1 ... done
Recreating docker_webserver_1 ... done
Recreating docker_mysql-server_1 ... done
shindy@shindy-server:~/docker$ docker ps -a
```

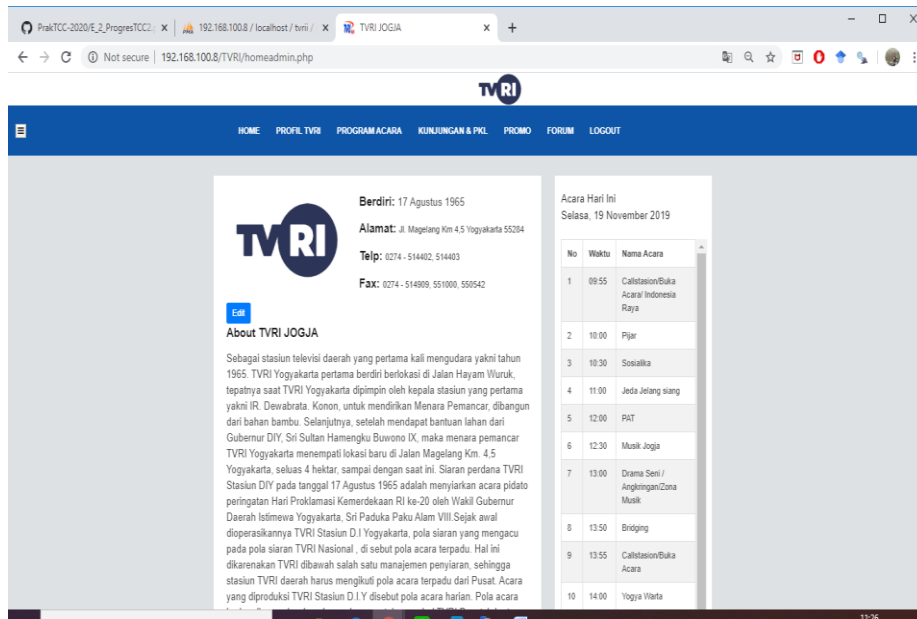
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e6e2b3c3aad3	phpmyadmin/phpmyadmin:5.0.1	"/docker-entrypoint..."	8 seconds ago	Up 5 seconds	0.0.0.0:3307->80/tcp	docker_phpmyadmin_1
cf36fb2b3634b	mysql:8.0.19	"docker-entrypoint.s..."	8 seconds ago	Up 6 seconds	3306/tcp, 33060/tcp, 0.0.0.0:6448->3306/tcp	docker_mysql-server_1
cf6b13e0b63d	docker_webserver	"docker-php-entrypoi..."	8 seconds ago	Up 5 seconds	0.0.0.0:8081->80/tcp	docker_webserver_1
be85fa31f409	raport_online_web-server	"docker-php-entrypoi..."	5 days ago	Up 21 minutes	0.0.0.0:8080->80/tcp	raport_online_web-server_1
74579e214711	phpmyadmin/phpmyadmin:5.0.1	"/docker-entrypoint..."	5 days ago	Up 21 minutes	0.0.0.0:3307->80/tcp	raport_online_phpmyadmin_1
43ee3095b0d9	mysql:8.0.19	"docker-entrypoint.s..."	5 days ago	Up 21 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	raport_online_mysql-server_1

```
shindy@shindy-server:~/docker$
```

Gambar 2.15 Tampilan cara menghentikan *containers docker* yang aktif

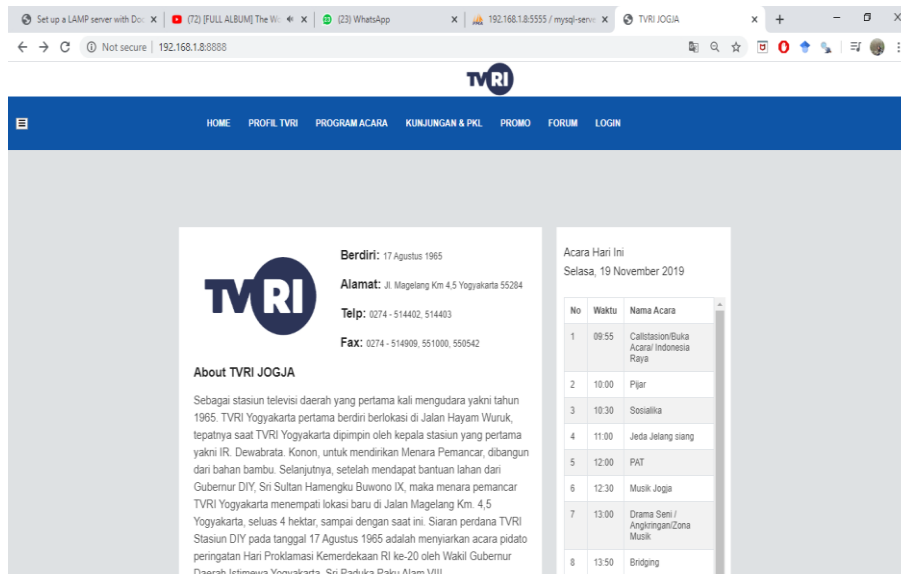
2.5 Hasil Implementasi

Setelah selesai melakukan konfigurasi dengan Ubuntu LAMPP, Pada hasil implementasi proyek Aplikasi Web Profil Stasiun TVRI Provinsi DIY sudah dapat berjalan dengan semestinya. Aplikasi web ini dapat diakses melalui IP server/nama folder/ yakni untuk hasilnya dapat dilihat pada **Gambar 2.16** Berikut ini:



Gambar 2.16 Hasil hosting di LAMPP

Setelah berhasil melakukan hosting di LAMPP, selanjutnya adalah mengkonversinya menjadi *dockerfile*. Aplikasi Web Profil Stasiun TVRI Provinsi DIY sudah dapat berjalan dengan baik. Untuk melihat hasilnya bisa dibuka di browser dengan cara mengetikkan IP_server:port yakni pada (192.168.1.8:8000) dan dapat dilihat pada **Gambar 2.17** Berikut ini:

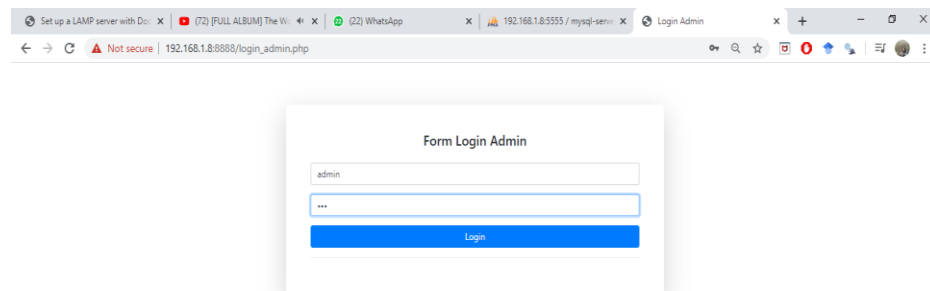


Gambar 2. 17 Hasil konfigurasi *Dockerfile*

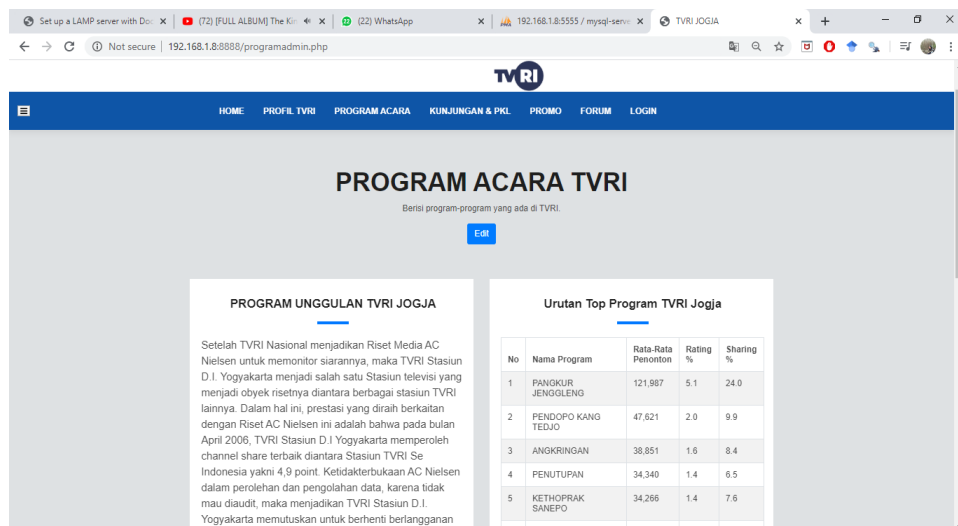
2.6 Pengujian Singkat

Pada Aplikasi Web Profil Stasiun Televisi Republik Indonesia(TVRI) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu dan konversi menjadi *Docker* diharapkan mampu menyelesaikan permasalahan perubahan jam tayang program acara TVRI.

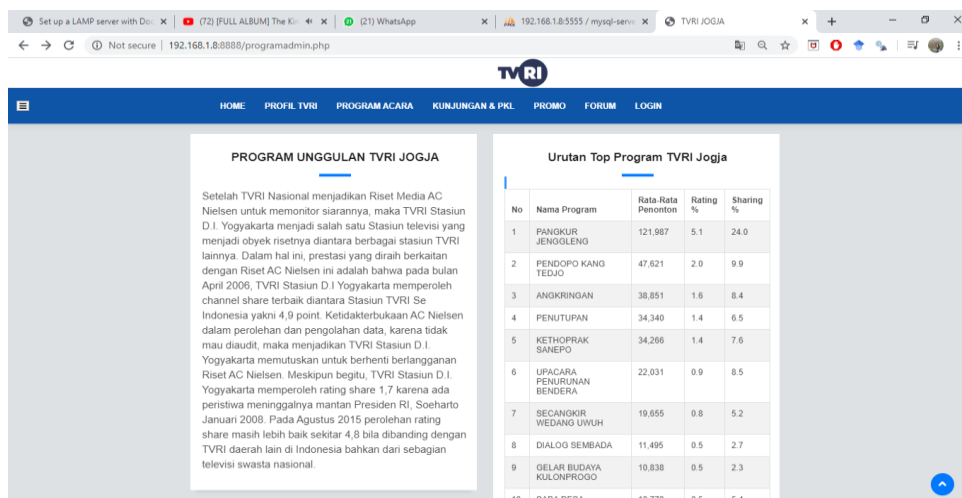
Pertama-tama untuk melakukan perubahan jam tayang adalah harus melakukan login pada baik itu sebagai admin ataupun pegawai, dapat dilihat pada gambar berikut :



Gambar 2.18 Tampilan untuk login admin



Gambar 2.19 Tampilan setelah berhasil login



The screenshot shows the TVRI JOGJA website. The left sidebar contains a section titled 'PROGRAM UNGGULAN TVRI JOGJA' with a paragraph of text. The main content area features a table titled 'Urutan Top Program TVRI Jogja'.

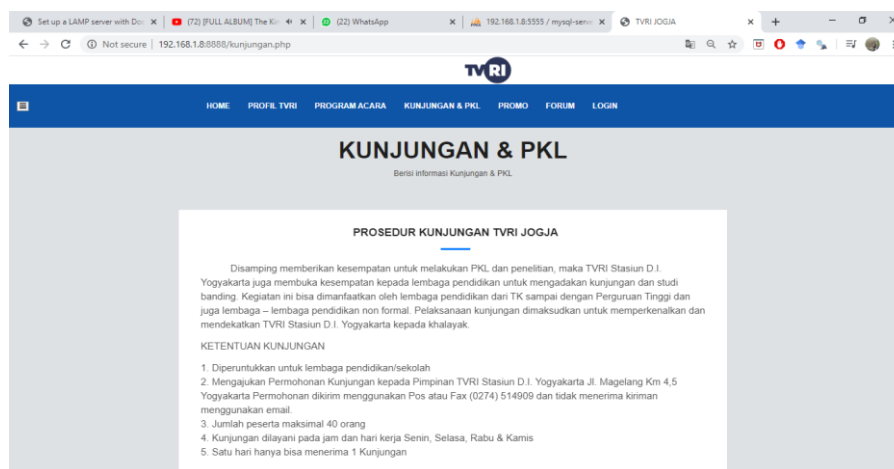
No	Nama Program	Rata-Rata Penonton	Rating %	Sharing %
1	PANGKUR JENGLENG	121.987	5.1	24.0
2	PENDOPO KANG TEDJO	47.621	2.0	9.9
3	ANGKRINGAN	38.851	1.6	8.4
4	PENUTUPAN	34.340	1.4	6.5
5	KETHOPRAK SANEPO	34.266	1.4	7.6
6	UPACARA PENURUNAN BENDERA	22.031	0.9	8.5
7	SECANGKIR WEDANG UWUH	19.655	0.8	5.2
8	DIALOG SEMBADA	11.495	0.5	2.7
9	GELAR BUDAYA KULONPROGO	10.838	0.5	2.3
10	SABA DESA	10.778	0.5	5.4

Gambar 2.20 Tampilan setelah dilakukan proses edit

Hasil dari apa yang telah dilakukan oleh admin tersebut adalah mendapatkan mengubah jam tayang program acara sama seperti cara konvensional namun dari segi waktu dan tenaga jauh lebih cepat. Hal ini dapat terlihat bahwa waktu yang dibutuhkan tidak sampai 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 penonton karena informasi yang diterima lebih cepat dan pihak TVRI, selain dapat diakses dari mana saja, biaya perbaikan berkala jauh lebih murah.

Pada pengujian kedua, adalah Aplikasi Web Profil Stasiun Televisi Republik Indonesia (TVRI) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu dan konversi menjadi *Docker* juga telah dapat memberikan informasi bagi mahasiswa yang ingin melakukan kerja praktek atau program magang di TVRI. Pertama adalah mahasiswa dapat langsung melihat informasi pada bagian menu kunjungan dan pkl, dapat dilihat pada gambar berikut ini:



The screenshot shows the 'KUNJUNGAN & PKL' menu on the TVRI JOGJA website. The page title is 'KUNJUNGAN & PKL' with the subtitle 'Berisi informasi Kunjungan & PKL'. The main content area contains a section titled 'PROSEDUR KUNJUNGAN TVRI JOGJA' with a paragraph of text and a list of 'KETENTUAN KUNJUNGAN'.

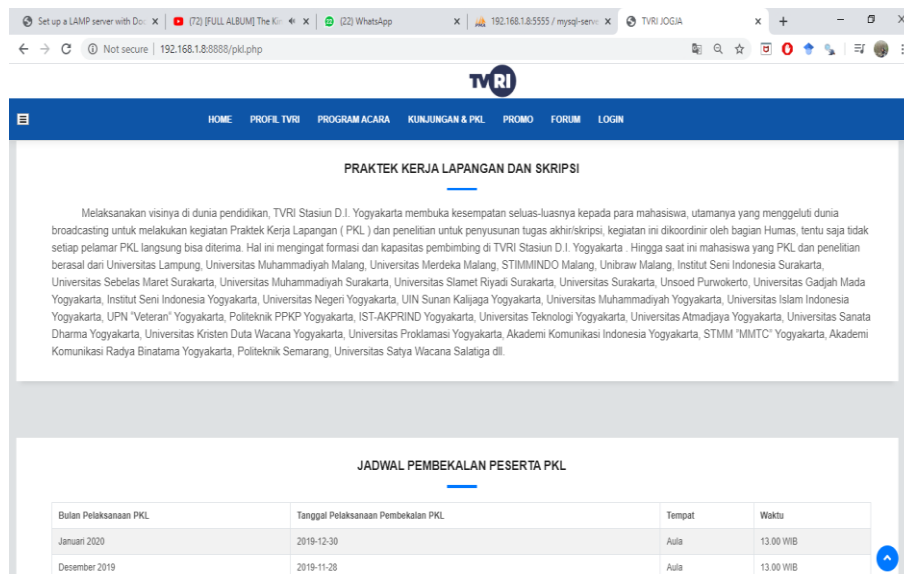
PROSEDUR KUNJUNGAN TVRI JOGJA

Disamping memberikan kesempatan untuk melakukan PKL dan penelitian, maka TVRI Stasiun D.I. Yogyakarta juga membuka kesempatan kepada lembaga pendidikan untuk mengadakan kunjungan dan studi banding. Kegiatan ini bisa dimanfaatkan oleh lembaga pendidikan dari TK sampai dengan Perguruan Tinggi dan juga lembaga – lembaga pendidikan non formal. Pelaksanaan kunjungan dimaksudkan untuk memperkenalkan dan mendekatkan TVRI Stasiun D.I. Yogyakarta kepada khalayak.

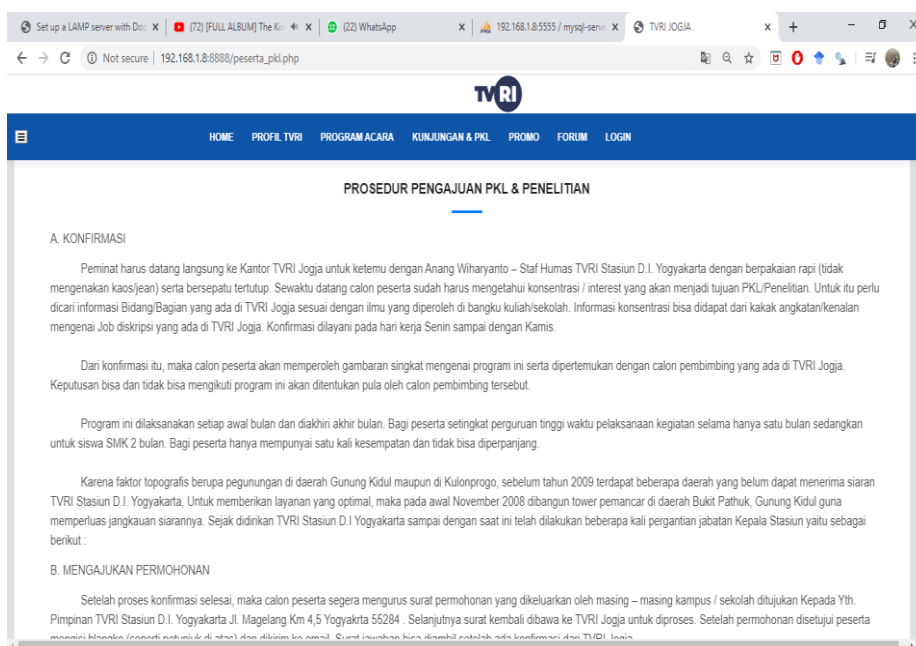
KETENTUAN KUNJUNGAN

1. Diperuntukkan untuk lembaga pendidikan/sekolah
2. Mengajukan Permohonan Kunjungan kepada Pimpinan TVRI Stasiun D.I. Yogyakarta Jl. Magelang Km 4,5 Yogyakarta Permohonan dikirim menggunakan Pos atau Fax (0274) 514909 dan tidak menerima kiriman menggunakan email
3. Jumlah peserta maksimal 40 orang
4. Kunjungan dilayani pada jam dan hari kerja Senin, Selasa, Rabu & Kamis
5. Satu hari hanya bisa menerima 1 Kunjungan

Gambar 2.21 Tampilan menu kunjungan dan pkl



Gambar 2.22 Tampilan prosedur pkl



Gambar 2.23 Tampilan menu pengajuan pkl dan penelitian

Hasil dari apa yang telah diujikan adalah mahasiswa atau masyarakat yang ingin mendapatkan informasi seputar kerja praktek dari segi waktu tidak sampai 5 menit saja sudah dapat memperolehnya, 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 mahasiswa atau siapapun yang ingin melakukan kegiatan magang atau kerja praktek karena sudah tersedia informasi yang dibutuhkan, hanya tinggal melihat laman web ini 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 sampai dengan Mei tahun 2020: **Tabel 3.1** Agenda Pengerjaan Proyek berikut ini:

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Pemilihan judul proyek dan pembuatan rancangan								
4.	Install VMWare, Pembuatan Laporan Bab I								
5.	Install Ubuntu, PHP MyAdmin, PHP, MySQL, Apache2								
6.	Revisi Laporan Bab I								
7.	Install <i>Docker</i>								
8.	Pengerjaan untuk konversi dari file LAMPP ke <i>dockerfile</i>								
9.	Submisi final proyek								

Tabel 3.1 Agenda Pengerjaan Proyek

3.2 Keterangan Pembagian Tugas

Berikut pada **Tabel 3.2** merupakan pembagian tugas-tugas pada proyek akhir terhadap anggota pada tim pembuatan proyek akhir:

No	Keterangan Tugas	Penanggung Jawab
1.	Perancangan Arsitektur <i>Cloud computing</i>	Gina
2.	Pengerjaan proyek pertama (layanan LAMPP)	Gina
4.	Pengagendaan Pengerjaan Proyek Akhir	Gina Shindy
5.	Pengerjaan proyek kedua (<i>Dockerfile</i>)	Gina Shindy
6.	Laporan	Gina Shindy

Tabel 3.2 Pembagian Tugas Proyek

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan masalah dari pembuatan Aplikasi Web Profil Stasiun Televisi Republik Indonesia(Tvri) Provinsi Daerah Istimewa Yogyakarta Os Ubuntu Menggunakan Lampp Dan Konversi Menjadi *Docker* yang sudah dibuat. Didapatkan hasil yang baik karena sistem dapat berjalan dengan lancar dan sangat membantu baik bagi pihak karyawan TVRI maupun pengunjung web.

Dari hasil dari pengujian diketahui bahwa semua sistem dapat berjalan dengan baik sebagaimana mestinya yakni bertujuan membantu pihak TVRI Yogyakarta untuk dapat memberikan informasi sebaik dan secepat mungkin kepada masyarakat.

Pada pembagian tugas sedikit terkendala karena faktor sulitnya koneksi internet di daerah Nur (dikampung halaman), akan tetapi sudah dicoba semaksimal mungkin agar dapat bersama-sama mengerjakan proyek ini secara bersama-sama.

4.2 Saran

Berdasarkan hasil yang kami dapatkan selama pengerjaan proyek ini, Akan lebih baik jika proyek ini dapat dikerjakan secara langsung sehingga mempermudah proses komunikasi antar anggota kelompok. Selain itu, pada bagian pembuatan *Docker* materinya belum sempat tersampaikan secara langsung karena adanya aturan kuliah *online* dan juga tidak ada tutorial pembuatannya yang lebih jelas. Sehingga sedikit menghambat pembuatannya. Apabila ada tutorial slide powerpoint atau media bantu lainnya tentu akan sangat membantu karena dapat menjadi acuan pasti dalam pembuatan *Docker*.

DAFTAR PUSTAKA

- [1] Kurniawan, Hendri (2017). Pengertian dan Manfaat *Cloud computing* yang Wajib Anda Ketahui. Diambil 26 Maret 2020, dari: <https://www.indoworx.com/pengertiandan-manfaat-cloud-computing-yang-wajib-anda-ketahui/amp/>
- [2] Anggi, Fersisilia (2016). Pengertian, Manfaat, Cara Kerja dan Contoh *Cloud computing*. Diambil 26 Maret 2020, dari: <http://pusatteknologi.com/pengertian-manfaat-carakerja-dan-contoh-cloud-computing.html>
- [3] Greiner, Robert.(2014). “Windows Azure IaaS vs. PaaS vs. SaaS”. Diambil dari: <http://robertgreiner.com/2014/03/windows-azure-iaas-paas-saas-overview/>
- [4] Maulana, Halim. (2016, September). Analisis dan Perancangan Sistem Replikasi Database MySQL dengan Menggunakan VMware pada Sistem Operasi *Open Source*, Jurnal Nasional Informatika dan Teknologi Jaringan, No. 1 Vol. I. 60-80.
- [5] Ramadhan, Dwi Rendra (2017). Peran Virtualisasi pada *Cloud computing*.
- [6] Shovan, Shahriar. (2020). Set Up a LAMPP Server with *Docker*. Diambil 28 April 2020, dari https://linuxhint.com/lamp_server_docker/
- [7] Sugianto, M. V. (2016). Virtualisasi Modern berbasis *Docker*. *Teknis Kerja Docker Container untuk Optimalisasi*.

LAMPIRAN