

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING  
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI RAPORT ONLINE MENGGUNAKAN UBUNTU LAMPP  
DAN PROSES PEMBUATAN DOCKERFILENYA**



**DISUSUN OLEH:**

**NAMA ANGGOTA : RAFLY PRADANA PUTRA 123170028  
SAKTI WICAKSONO 123170031  
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

### SISTEM INFORMASI RAPORT ONLINE MENGGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

Rafly Pradana Putra

123170028

Sakti Wicaksono

123170031

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

Assalamualaikum wr. wb.,

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

Tidak lupa kami ucapkan terima kasih kepada asisten dosen yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun diharapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terima kasih. Semoga laporan ini dapat digunakan sebaik-baiknya.

Wassalamualaikum, wr. wb.,

Yogyakarta, 30 Maret 2020

Penyusun

## DAFTAR ISI

<b>HALAMAN PENGESAHAN .....</b>	<b>ii</b>
<b>KATA PENGANTAR .....</b>	<b>iii</b>
<b>DAFTAR ISI .....</b>	<b>iv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>2</b>
1.1. Latar Belakang Masalah .....	2
1.2 Tujuan Proyek Akhir .....	3
1.3 Manfaat Proyek Akhir .....	3
1.4 Tahap Penyelesaian Proyek Akhir .....	3
<b>BAB II ISI DAN PEMBAHASAN .....</b>	<b>4</b>
2.1 Komponen yang Digunakan .....	4
2.2 Rancangan Arsitektur <i>Cloud Computing</i> .....	6
2.3 Parameter dan Konfigurasi .....	7
2.4 Tahap Implementasi .....	14
2.5 Hasil Implementasi .....	22
2.6 Pengujian Singkat .....	25
<b>BAB III JADWAL Pengerjaan dan Pembagian Tugas .....</b>	<b>28</b>
3.1 Agenda Pengerjaan .....	28
3.2 Keterangan Pembagian Tugas .....	28
<b>BAB IV KESIMPULAN DAN SARAN .....</b>	<b>29</b>
4.1 Kesimpulan .....	29
4.2 Saran .....	29
<b>DAFTAR PUSTAKA .....</b>	<b>30</b>
<b>LAMPIRAN .....</b>	<b>31</b>

## BAB I PENDAHULUAN

### 1.1. Latar Belakang Masalah

*Cloud Computing* merupakan sebuah teknologi yang menggabungkan pemanfaatan teknologi komputer dan pengembangan berbasis 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. Sebenarnya, pada tahun 1960 an pun sudah mulai muncul konsep yang menyatukan beberapa sumber *computing* dengan menggunakan jaringan yang bersifat global, yang biasa disebut *Intergalactic Computer Network*.

Penggunaan *cloud computing* di masyarakat sangat beragam. Salah satu contoh penggunaan *cloud computing* di Indonesia yaitu layanan aplikasi FTP (*File Transfer Protocol*). Melalui proses teknologi *virtualisasi*, 3 server FTP dibangun dalam bentuk virtual didalam satu komputer server *hypervisor* yang dapat diakses melalui jaringan. Untuk ketersediaan layanan FTP yang handal, maka digunakan teknologi *load balancing high availability* dengan aplikasi LVS (*Linux Virtual Server*) dan memakai metode *direct routing*. Sedangkan untuk algoritma yang dijadwalkan yaitu *least connection* (LC), *weighted least connection* (WLC), *source hashing* (SH), *shorted expected delay* (SED), dan *never queue* (NQ) (Lisyadi Oktavianus, 2013).

Terdapat banyak kelebihan dan manfaat yang dirasakan ketika menggunakan *cloud computing*, beberapa diantaranya adalah tidak perlu sibuk untuk maintenance, tidak perlu khawatir kalau *hardware* rusak, terkena virus, dll. Karena itu semua adalah tanggung jawab *vendor* yang menyediakan. Selain itu, *cloud computing* juga merupakan teknologi yang fleksibel dan mudah dikembangkan karena teknologi *cloud computing* ini dapat dibuat sesuai dengan kebutuhan dan kemampuan dari pengguna yang akan memakainya.

Melihat kelebihan dan kekurangan dari *cloud computing*, ada banyak masalah di kehidupan sehari – hari manusia yang dapat diselesaikan serta dipermudah pekerjaannya, seperti misalnya pembuatan dan perhitungan nilai raport dan penyimpanan data dokumentasi sejarah Indonesia. Pembuatan dan perhitungan nilai raport hingga saat ini masih dilakukan secara manual tulis tangan dan menggunakan aplikasi pembantu seperti Excel yang dapat

membantu dalam masalah komputasi nilai raport. Raport yang ditulis tangan atau dicetak memiliki kekurangan seperti boros kertas dan tinta serta mudah rusak.

Komponen penyusun untuk Sistem Informasi Raport *Online* sangat mudah, yaitu dengan menyiapkan *Virtual Private Server* bisa dengan membuat sendiri atau menyewa layanannya yang harganya sekarang sudah terbilang murah. Perangkat yang digunakan bisa menggunakan PC / laptop dan *Smartphone* dengan menggunakan aplikasi berbentuk *website* sehingga bisa diakses di perangkat manapun.

## 1.2 Tujuan Proyek Akhir

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

1. Mengimplementasikan arsitektur *cloud computing* untuk Sistem Informasi Raport Online yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04.3 LTS dan LAMPP (Apache 2.4, PHP 7.4.3, MySQL 8.0.19).
2. Mengimplementasikan arsitektur *cloud computing* untuk membuat *Dockerfile* dari Sistem Informasi Raport *Online*.
3. Mempermudah orang tua siswa dalam memantau nilai tanpa harus menunggu raport jadi.
4. Mempermudah guru dalam manajemen nilai siswa dan mempermudah kolaborasi antar guru pengampu.

## 1.3 Manfaat Proyek Akhir

Manfaat yang didapat dari pembuatan Sistem Informasi Raport *Online* adalah sebagai berikut:

1. Memudahkan para guru untuk manajemen nilai siswa.
2. Memudahkan orang tua siswa dalam memantau nilai anaknya serta kemampuan belajar di sekolah.
3. Data nilai raport tidak mudah rusak dan hilang karena tersimpan secara *online*.
4. Meminimalisir *human error* yang mungkin terjadi jika dikerjakan secara manual.

## 1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari sistem informasi raport *online* untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis PaaS
2. Menentukan konfigurasi yang tepat untuk pengaturan *Dockerfile* sehingga dapat digunakan sesuai *requirement* yang berupa:

- a. Dapat diimplementasikan di sistem manapun dengan Docker.
  - b. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
3. Merancang topologi *cloud computing* untuk mengintegrasikan sistem yang digunakan.
4. Menguji konfigurasi Dockerfile yang dibuat dengan beberapa pengujian yaitu pengimplementasian Dockerfile ke sistem lain dan pengujian fungsionalitas sistem raport *online*.
5. Menghasilkan Sistem Informasi Raport *Online* yang berbasiskan *cloud computing* sesuai standar ISO 9001.

## BAB II

### ISI DAN PEMBAHASAN

#### 2.1 Komponen yang Digunakan

Untuk membangun “Sistem Informasi Raport Online Menggunakan Ubuntu LAMPP dan Proses Pembuatan *Dockerfile*-nya” yang berbasiskan konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.4.3 dan bahasa HTML dengan versi minimal 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 8.0.19 sehingga dapat digunakan untuk menyimpan berbagai data yang dibutuhkan oleh sistem tersebut.
3. Untuk membangun aplikasi, mengemas dan menjalankan aplikasi pada project ini menggunakan Docker dengan versi 19.03.8, *build* afacb8b7f0 dan Docker Compose dengan versi 1.21.2, *build* a133471.
4. VMWare Workstation Pro 15 yang digunakan untuk melakukan instalasi ubuntu server dan LAMPP
5. Virtual Box *Graphical User Interface* versi 6.0.18 r136238 (Qt5.6.2) *virtual machine* ini digunakan untuk melakukan instalasi ubuntu server dan Docker.
6. Apache versi 2.4 sebagai *webserver*.
7. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaannya ialah seluruh guru dan pegawai yang berada di sekolah tersebut dan siswa atau orang tua murid dari sekolah tersebut.

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



**Tabel 2.1** Spesifikasi VM *cloud computing* yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	VirtualBox Graphical User Interface Version 6.0.18 r136238 (Qt5.6.2)	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz (4 CPUs), ~3.6GHz	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: 192.168.8.104/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 1.1.1.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.8.1	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	2GB	Alokasi RAM untuk <i>guest OS</i>
6.	Penyimpanan Data	10GB	Penyimpanan 10 GB digunakan untuk root

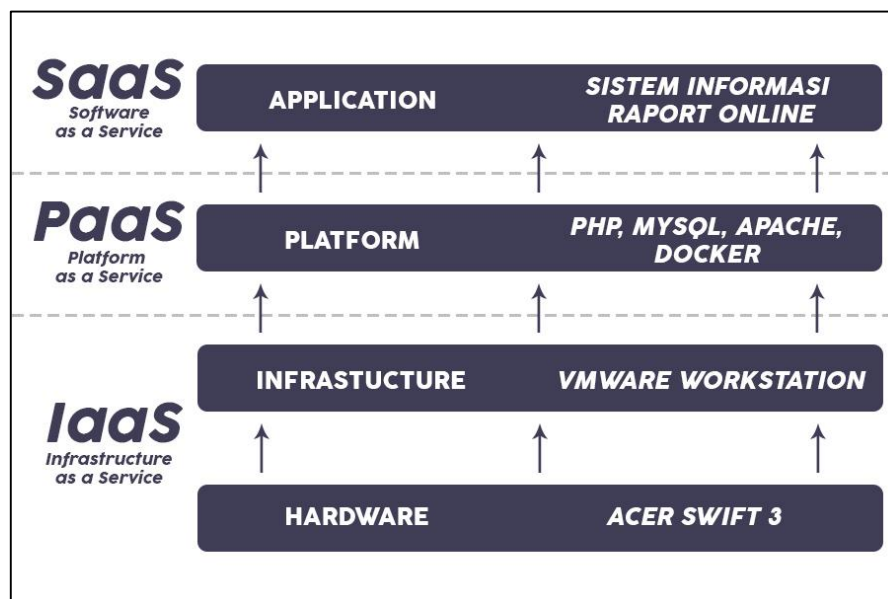
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

**Tabel 2.2** Spesifikasi Ubuntu OS yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.4.3	bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum
		MySQL 8.0.19	sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (Database Management System) yang multialur dan multipengguna.
	Docker	Versi 19.03.8, <i>build</i> 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
	Docker Compose	Versi 1.21.2, <i>build</i> a133471	aplikasi yang bersifat open source yang berfungsi untuk mendefinisikan dan menjalankan banyak container dalam aplikasi Docker

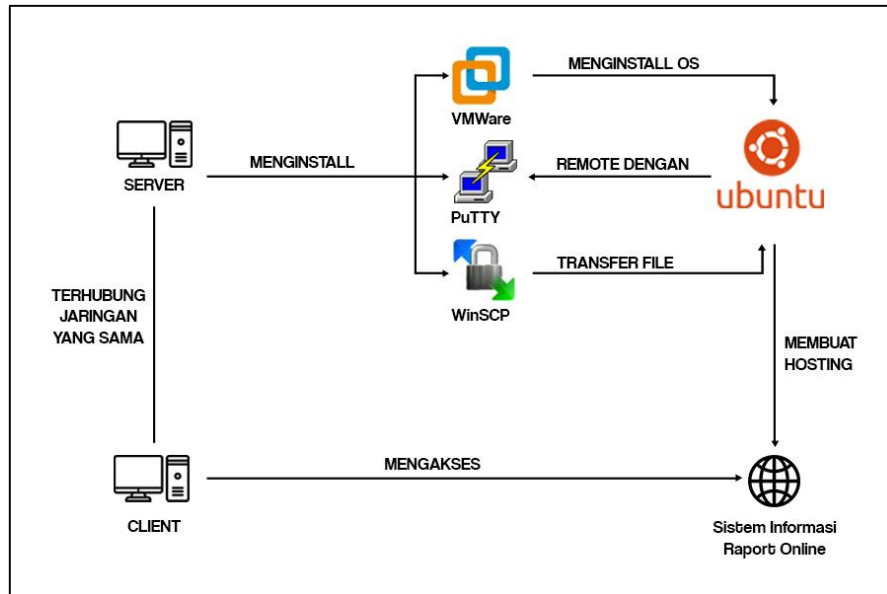
## 2.2 Rancangan Arsitektur Cloud Computing

Pada proyek akhir ini digunakan bentuk rancangan arsitektur *Infrastructure as a Service* di mana *hardware* laptop ACER Swift 3 sebagai layer utama di bagian bawah yang melakukan virtualisasi menggunakan VMWare Workstation, kemudian di atasnya adalah *Platform as a Service* dimana sistem operasi Ubuntu Server yang berjalan. Di atasnya lagi adalah *Software as a Service* dimana Sistem Informasi Report Online berjalan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



**Gambar 2.1** Ilustrasi rancangan arsitektur proyek

Dalam pembuatan proyek untuk membuat *web hosting*, komputer yang berlaku sebagai server memakai sistem operasi Windows 10 di mana diharuskan menginstall *virtual machine*, seperti VMWare Workstation, PuTTY untuk melakukan remote dari VMWare, dan WinSCP untuk membuat transfer berkas yang berisi *source code* agar *website* dapat di-*hosting*. Setelah itu, komputer *client* diharuskan untuk terkoneksi dengan jaringan yang sama agar dapat mengakses *website* yang telah dibuat. Ilustrasi mengenai topologi yang digunakan bisa dilihat pada **Gambar 2.2** berikut:



**Gambar 2.2** Ilustrasi topologi proyek LAMPP

### 2.3 Parameter dan Konfigurasi

Parameter yang digunakan untuk instalasi Apache2 dapat dilihat pada penjelasan

**Modul 2.1** berikut ini:

```
$ sudo apt install apache2
```

Keterangan:

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

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

Keterangan:

Untuk mengatur Firewall agar port 80 dan port 443 diizinkan oleh sistem Ubuntu.

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

Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan

**Modul 2.2** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter pada apt untuk mengeksekusi perintah instalasi paket
- mysql-server : nama paket untuk MySQL

```
$ sudo mysql_secure_installation
```

Keterangan:

Untuk mengatur keamanan pada MySQL contohnya username dan password

### Modul 2.2 Parameter instalasi MySQL

Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan

**Modul 2.3** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket
- php libapache2-mod-php php-mysql : nama paket untuk PHP

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

Keterangan:

Untuk membuat file baru bernama info.php yang berada di direktori var/www/html

### Modul 2.3 Parameter instalasi PHP

Parameter yang digunakan untuk instalasi PHPMyAdmin dapat dilihat pada

penjelasan **Modul 2.4** berikut ini:

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

Keterangan:

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

```
$ sudo mysql -u root
```

Keterangan:

Untuk masuk ke MySQL sebagai user root.

```
mysql> UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('12370028') WHERE User = 'root';
```

Keterangan:

Untuk mengubah password dari user 'root' menjadi '123170028'

```
mysql> FLUSH PRIVILEGES;
```

Keterangan:

Untuk mererefresh akun yang terkoneksi dengan phpmyadmin

```
$ sudo chown rafly /var/www/html
```

Keterangan:

Untuk memberikan akses ke user rafly agar dapat mengubah/menambah/menghapus berkas pada direktori var/www/html

### Modul 2.4 Parameter instalasi PHPMyAdmin

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

```
$ sudo apt update
```

Keterangan:

Untuk mengupdate package yang terinstall di Ubuntu.

```
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Keterangan:

Sebelum install Docker, install package yang diperlukan untuk menginstall dan menggunakan Docker, yaitu package apt-transport-https, ca-certificates, curl, dan software-properties-common.

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

Keterangan:

Untuk menambahkan kunci GPG dari repository Docker ke sistem.

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- curl -fsSL : merupakan perintah untuk mengunduh suatu file berdasarkan link yang diberikan
- apt-key add - : merupakan perintah untuk menambahkan kunci GPG ke sistem

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

Keterangan:

Untuk menambahkan repository Docker ke daftar sumber package APT (package manager Ubuntu).

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- add-apt-repository : merupakan perintah untuk menambahkan repository baru ke daftar sumber package APT (package manager Ubuntu)
- "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" : merupakan sumber package Docker dengan acuan deb adalah jenis package, [arch=amd64] adalah jenis arsitektur, https://download.docker.com/linux/ubuntu adalah link sumber package Docker, bionic adalah versi Ubuntu yang digunakan, dan stable adalah versi rilis yang ingin digunakan.

```
$ sudo apt update
```

Keterangan:

Untuk mengupdate package yang terinstall di Ubuntu.

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

Keterangan:

Untuk mengunduh dan menginstall Docker.

```
$ sudo systemctl status docker
```

Keterangan:

Untuk melihat status dari Docker.

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- systemctl : merupakan system manager pada Ubuntu
- status : parameter tambahan pada systemctl untuk melihat status suatu service
- docker : nama service

### Modul 2.5 Parameter instalasi Docker

Parameter yang digunakan untuk konfigurasi Docker tanpa menggunakan perintah sudo di awal dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

```
$ sudo usermod -aG docker ${USER}
```

Keterangan:

Untuk menambahkan user aktif saat ini ke group docker

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- usermod : merupakan user manager pada Ubuntu
- -aG : parameter tambahan pada usermod untuk menambahkan user ke suatu group
- docker : nama group
- \${USER} : untuk memanggil user yang aktif saat ini

```
$ su - ${USER}
```

Keterangan:

Untuk refresh session user (agar konfigurasi di atas bisa langsung aktif tanpa harus login ulang).

### Modul 2.6 Konfigurasi Docker tanpa perintah sudo

Parameter yang digunakan untuk instalasi Docker Compose dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

```
$ sudo curl -L
https://github.com/docker/compose/releases/download/1.21.2/docker-
compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

Keterangan:

Untuk mengunduh Docker Compose versi 1.21.2.

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- curl -L : merupakan perintah untuk mengunduh suatu file berdasarkan link yang diberikan
- https://github.com/docker/compose/releases/download/1.21.2/docker-compose-`uname -s`-`uname -m` : link Docker Compose versi 1.21.2
- uname -s : perintah untuk mengoutputkan nama kernel
- uname -m : perintah untuk mengoutputkan nama mesin (hardware)
- -o /usr/local/bin/docker-compose : parameter tambahan pada curl untuk mengarahkan file yang diunduh ke direktori yang diinginkan

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

Keterangan:

Untuk mengubah status/permission file docker-compose menjadi executeable(bisa dieksekusi).

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- chmod +x : perintah untuk menjadikan suatu file menjadi executeable.

```

- /usr/local/bin/docker-compose : lokasi file docker compose

$ docker-compose --version
Keterangan:
Untuk melihat versi docker compose yang terinstal.
- docker-compose : perintah untuk menjalankan docker-compose
- --version : parameter dari docker-compose untuk menampilkan versi
  docker compose yang terpasang

```

### Modul 2.7 Parameter instalasi Docker Compose

Parameter yang digunakan untuk konfigurasi dari php.Dockerfile dapat dilihat pada penjelasan **Modul 2.8** berikut ini:

```

$ nano php.Dockerfile
Keterangan:
Untuk mengedit file (text editor dalam terminal).
- nano : perintah untuk menjalan text editor dalam terminal linux
  (bawaan dari Ubuntu)
- php.Dockerfile : nama file yang diinginkan diedit

# Isi dari php.Dockerfile

FROM php:7.4.3-apache

RUN docker-php-ext-install mysqli pdo pdo_mysql

Keterangan:
Untuk menginstall extensions php yang diperlukan, yaitu mysqli, pdo,
dan pdo_mysql.
- FROM php:7.4.3-apache : lokasi Dockerfile yang akan dituju /
  dieksekusi
- RUN docker-php-ext-install : perintah untuk menjalankan
  instalasi extensions php
- mysqli pdo pdo mysql : nama extensions php

```

### Modul 2.8 Parameter konfigurasi php.Dockerfile

Parameter yang digunakan untuk konfigurasi docker-compose.yaml dapat dilihat pada penjelasan **Modul 2.9** berikut ini:

```

$ nano docker-compose.yaml
Keterangan:
Untuk mengedit file (text editor dalam terminal).
- nano : perintah untuk menjalan text editor dalam terminal linux
  (bawaan dari Ubuntu)
- docker-compose.yaml : nama file yang diinginkan diedit

# Isi dari docker-compose.yaml

version: "3"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "./raport_online:/var/www/html/"
    ports:
      - "8080:80"

```

```

mysql-server:
  image: mysql:8.0.19
  restart: always
  environment:
    MYSQL_DATABASE: raport_online
    MYSQL_USERNAME: root
    MYSQL_PASSWORD: root
    MYSQL_ROOT_PASSWORD: root
  volumes:
    - mysql-data:/var/lib/mysql
    - ./raport_online/db_mysql/raport_online.sql:/docker-
entrypoint-initdb.d/raport_online.sql
  ports:
    - "3306:3306"

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

volumes:
  mysql-data:

```

#### Keterangan:

Untuk mendefinisikan service yang akan diinstall pada container Docker serta konfigurasinya.

- version : versi compose file format yang akan digunakan sesuai dengan docker engine yang terinstall
- services : bagian untuk mendefinisikan service yang ingin diinstall di docker
- web-server, mysql-server, phpmyadmin : nama service yang ingin diinstall
- build : perintah bahwa service akan diinstall sesuai perintah yang ada di dalam build tersebut
- dockerfile : lokasi dockerfile yang ingin digunakan
- context : mengarahkan direktori untuk service ada di folder tersebut (. artinya direktori ada di folder sesuai lokasi docker-compose.yaml berada)
- restart : konfigurasi dari service untuk melakukan restart container ketika sesuatu hal yang tidak diinginkan terjadi
- volumes : perintah dalam service untuk mengarahkan serta mengcopy isi folder sumber ke direktori yang dituju
- ports : perintah dalam service untuk mendefinisikan port yang ingin dibuka/digunakan
- image : konfigurasi untuk memilih image / installer / package dari repository Docker yang ingin digunakan dan diinstall
- environment : perintah pada service untuk mengkonfigurasi service itu sendiri sesuai environment yang berlaku
- MYSQL\_DATABASE : mendefinisikan environment mysql untuk nama database yang akan digunakan
- MYSQL\_USERNAME : mendefinisikan environment mysql untuk nama username yang akan digunakan
- MYSQL\_PASSWORD : mendefinisikan environment mysql untuk password mysql yang akan digunakan
- MYSQL\_ROOT\_PASSWORD : mendefinisikan environment mysql untuk password dari root mysql yang akan digunakan



- PMA\_HOST : mendefinsikan environment phpmyadmin untuk nama host yang akan digunakan
- PMA\_USER : mendefinsikan environment phpmyadmin untuk nama username yang akan digunakan
- PMA\_PASSWORD : mendefinsikan environment phpmyadmin untuk password akun phpmyadmin yang akan digunakan

### **Modul 2.9** Parameter konfigurasi docker-compose.yaml

Parameter yang digunakan untuk konfigurasi dari koneksi.php dapat dilihat pada penjelasan **Modul 2.10** berikut ini:

```
<?php
    $conn = new mysqli('mysql-server','root','root','raport_online');
    if ($conn->connect_error){
        die("Error: " . mysqli_connect_error());
    }
?>
```

Keterangan :  
Untuk menginisiasasi/mendefinisikan koneksi ke database mysql sesuai konfigurasi dari docker-compose.yaml pada Modul 2.9.

### **Modul 2.10** Parameter konfigurasi koneksi.php

Parameter yang digunakan untuk menjalankan Docker Compose dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

```
$ cd docker-project
```

Keterangan :  
Untuk berpindah direktori ke direktori yang terdapat docker-compose.yaml yang akan dijalankan.

- cd : perintah untuk pindah direktori
- docker-project : tujuan direktori

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

Keterangan :  
Untuk membaca isi konfigurasi docker-compose.yaml dan membuat container docker sesuai konfigurasinya.

- docker-compose : perintah untuk menjalankan docker-compose
- up : perintah dari docker-compose untuk membaca docker-compose.yaml dan menjalankan docker
- -d : parameter dari docker-compose untuk menjalankan containers di background

### **Modul 2.11** Parameter konfigurasi menjalankan Docker Compose

Parameter yang digunakan untuk melihat containers Docker yang sudah terpasang sebelumnya pada **Modul 2.11** dapat dilihat pada penjelasan **Modul 2.12** 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 2.12** Parameter melihat containers Docker yang terpasang

Parameter yang digunakan untuk melihat *ip address* dapat dilihat pada penjelasan **Modul 2.13** 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 2.13** Parameter melihat *ip address*

Parameter yang digunakan untuk menghentikan containers Docker yang aktif sebelumnya pada **Modul 2.11** dapat dilihat pada penjelasan **Modul 2.14** berikut ini:

```
$ docker-compose down
```

Keterangan :

Untuk menghentikan containers docker yang aktif.

- docker-compose : perintah untuk menjalankan docker-compose
- up : perintah dari docker-compose untuk menghentikan containers docker yang aktif

#### **Modul 2.14** Parameter menghentikan containers Docker yang aktif

Parameter yang digunakan untuk mengclone repository github dapat dilihat pada penjelasan **Modul 2.15** berikut ini:

```
$ git clone https://github.com/lulu8879/raport_online.git
```

Keterangan :

Untuk menclone repository github.

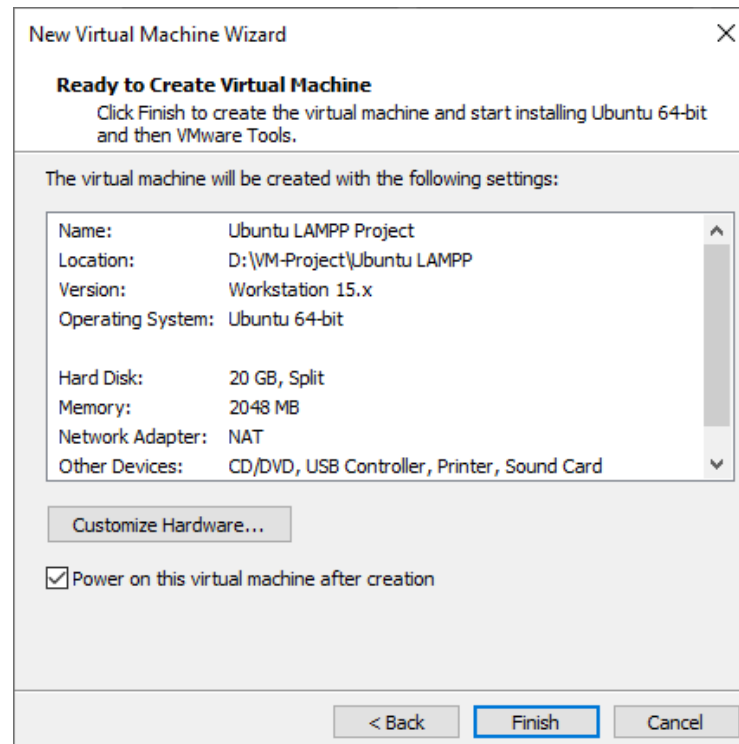
- git : perintah untuk menjalankan git
- clone : perintah dari git untuk mengunduh repository github berdasarkan link yang diberi
- https://github.com/lulu8879/raport\_online.git : link repository github

#### **Modul 2.15** Parameter menghentikan containers Docker yang aktif

## **2.4 Tahap Implementasi**

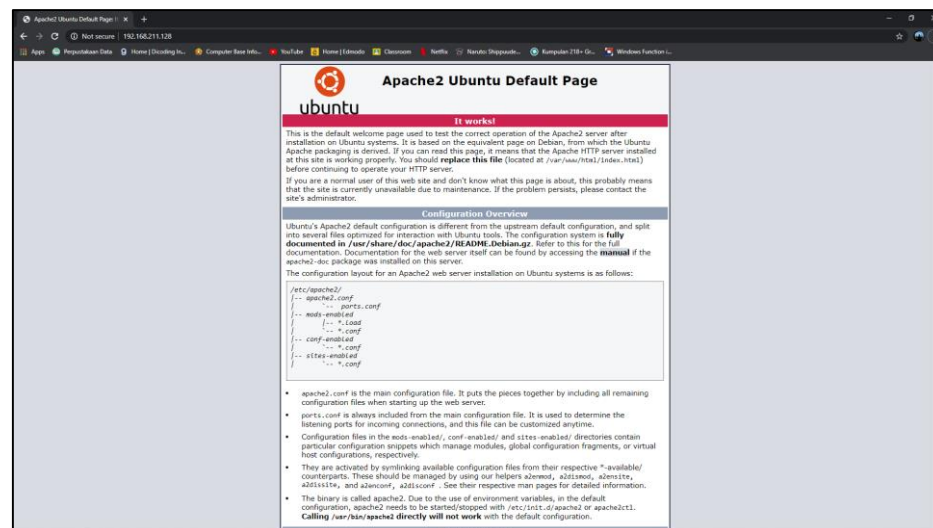
Untuk mengimplementasikan “Sistem Informasi Raport Online Menggunakan Ubuntu LAMPP dan Proses Pembuatan *Dockerfile*-nya” yang berbasiskan konsep *cloud computing*, maka tahapan yang dilakukan adalah sebagai berikut:

- a. Tahapan awal pada *wizard* pembuatan VM pada VMware Workstation digunakan opsi konfigurasi *Custom* seperti pada **Gambar 2.3** berikut ini:



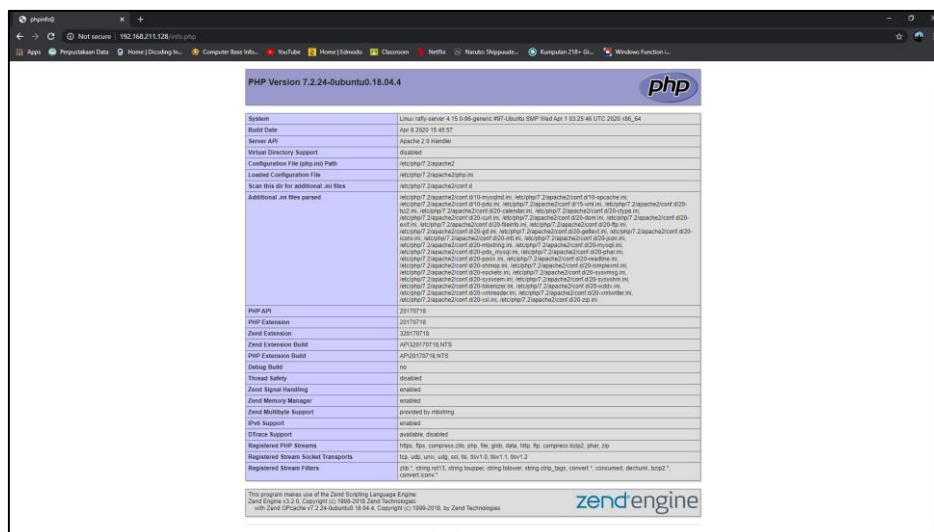
**Gambar 2.3** Tampilan opsi pemilihan *mode wizard* pembuatan VM

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



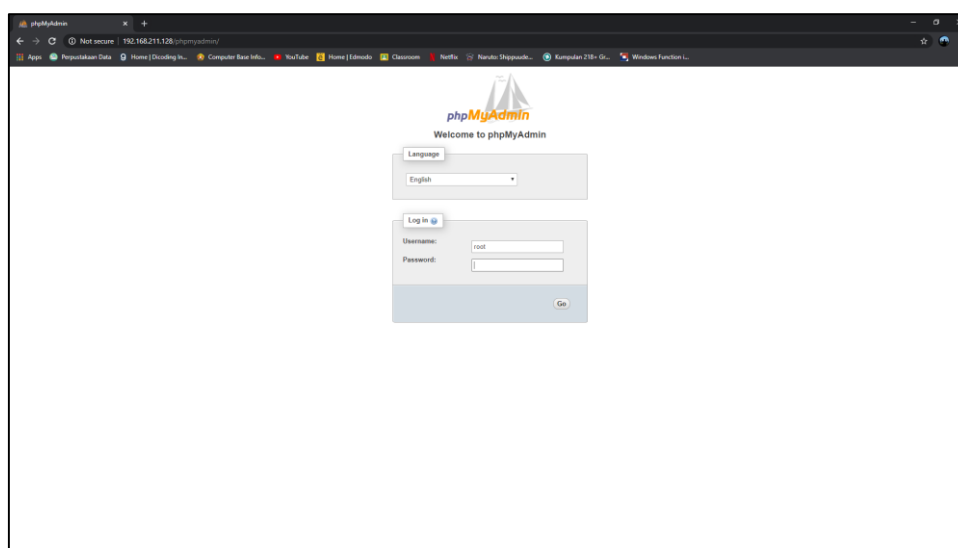
**Gambar 2.4** Tampilan hasil instalasi Apache2

- c. Hasil implementasi dari **Modul 2.3** tentang cara instalasi PHP dapat dilihat pada **Gambar 2.5** berikut ini:



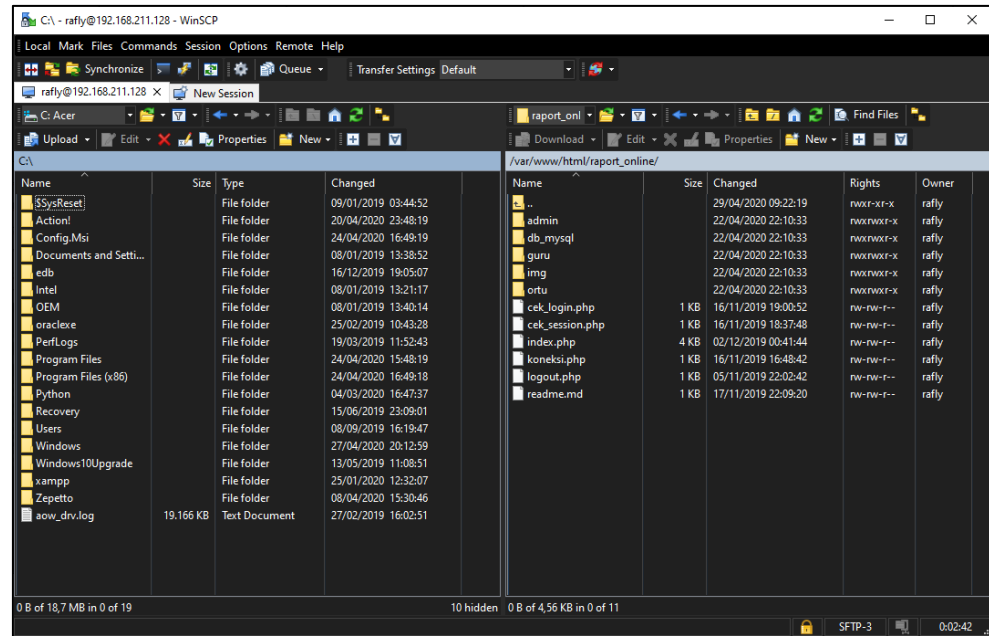
**Gambar 2.5** Tampilan hasil instalasi PHP

- d. Hasil implementasi dari **Modul 2.4** tentang cara instalasi PHPMyAdmin dapat dilihat pada **Gambar 2.6** berikut ini:



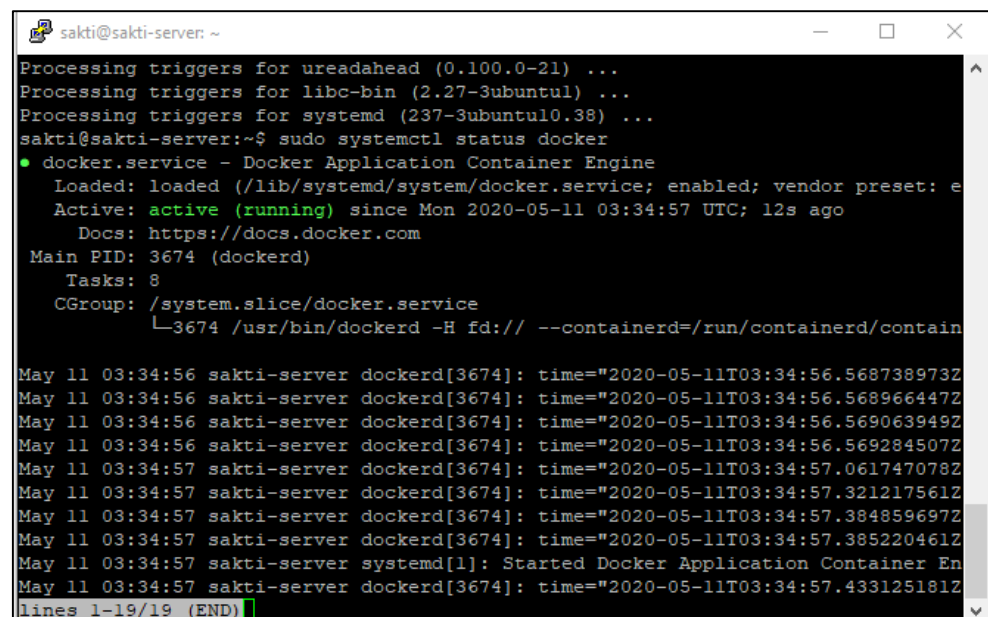
**Gambar 2.6** Tampilan hasil instalasi PHPMyAdmin

- e. Setelah semua proses instalasi LAMPP, selanjutnya adalah memindahkan *source code* ke server dengan menggunakan WinSCP. Pemindahan tersebut dengan cara melakukan *drag and drop* folder yang berisi *source code* ke direktori `/var/www/html/`. Untuk hasil pemindahan *source code* dapat dilihat pada **Gambar 2.7** berikut ini:



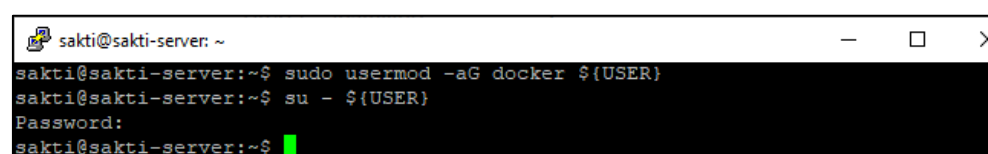
**Gambar 2.7** Hasil pemindahan *source code* ke direktori /var/www/html/

- f. Hasil implementasi dari **Modul 2.5** tentang cara instalasi Docker dapat dilihat pada **Gambar 2.8** berikut ini:



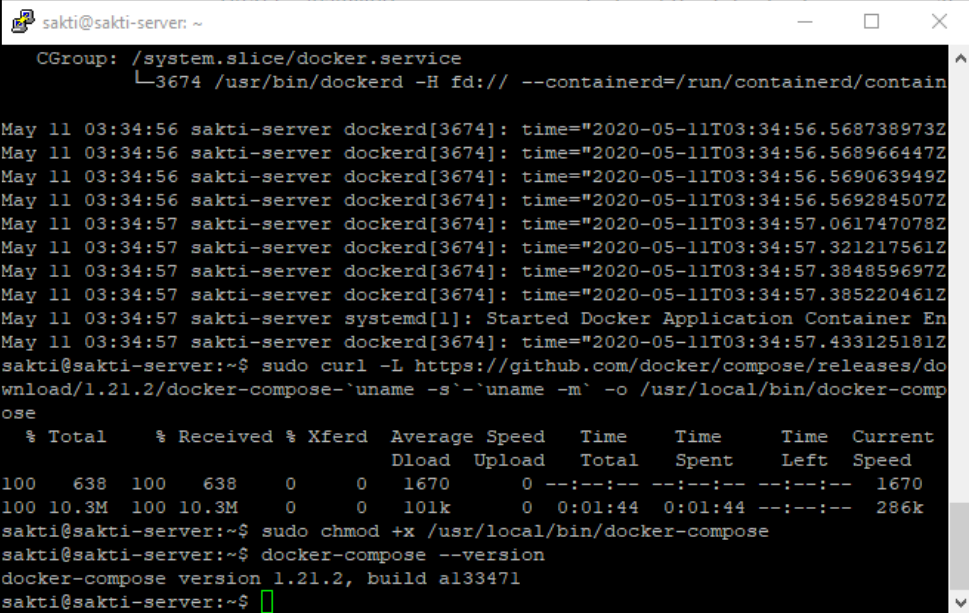
**Gambar 2.8** Tampilan hasil instalasi Docker

- g. Hasil implementasi dari **Modul 2.6** tentang cara konfigurasi Docker tanpa sudo dapat dilihat pada **Gambar 2.9** berikut ini:



**Gambar 2.9** Tampilan hasil konfigurasi Docker tanpa sudo

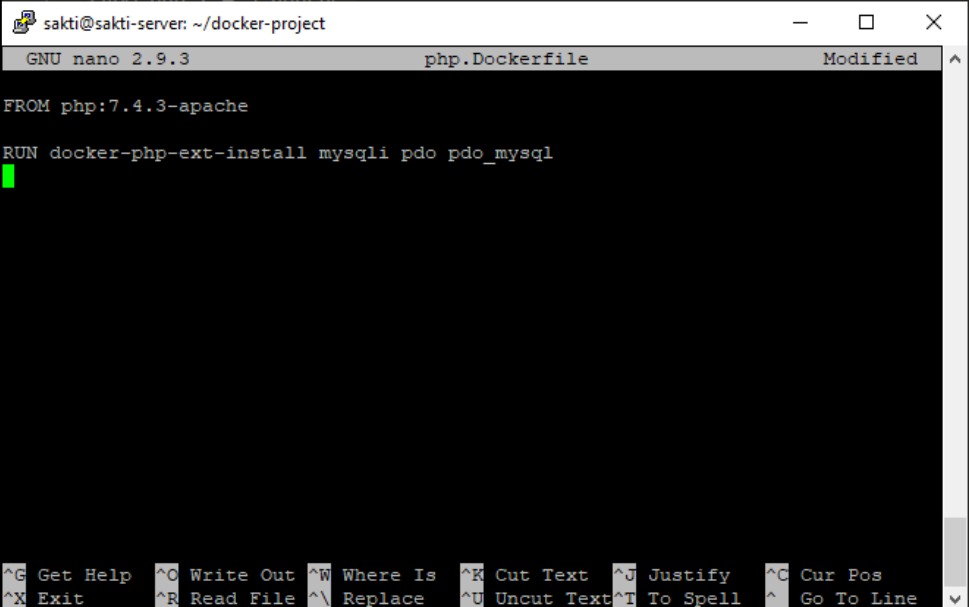
- h. Hasil implementasi dari **Modul 2.7** tentang cara instalasi Docker Compose dapat dilihat pada **Gambar 2.10** berikut ini:



```
sakti@sakti-server: ~
CGroup: /system.slice/docker.service
└─3674 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
May 11 03:34:56 sakti-server dockerd[3674]: time="2020-05-11T03:34:56.568738973Z
May 11 03:34:56 sakti-server dockerd[3674]: time="2020-05-11T03:34:56.568966447Z
May 11 03:34:56 sakti-server dockerd[3674]: time="2020-05-11T03:34:56.569063949Z
May 11 03:34:56 sakti-server dockerd[3674]: time="2020-05-11T03:34:56.569284507Z
May 11 03:34:57 sakti-server dockerd[3674]: time="2020-05-11T03:34:57.061747078Z
May 11 03:34:57 sakti-server dockerd[3674]: time="2020-05-11T03:34:57.321217561Z
May 11 03:34:57 sakti-server dockerd[3674]: time="2020-05-11T03:34:57.384859697Z
May 11 03:34:57 sakti-server dockerd[3674]: time="2020-05-11T03:34:57.385220461Z
May 11 03:34:57 sakti-server systemd[1]: Started Docker Application Container En
May 11 03:34:57 sakti-server dockerd[3674]: time="2020-05-11T03:34:57.433125181Z
sakti@sakti-server:~$ sudo curl -L https://github.com/docker/compose/releases/do
wnload/1.21.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-comp
ose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left    Speed
100  638    100  638    0     0   1670      0  --:--:-- --:--:-- --:--:--   1670
100 10.3M    100 10.3M    0     0  101k      0  0:01:44 0:01:44 --:--:--  286k
sakti@sakti-server:~$ sudo chmod +x /usr/local/bin/docker-compose
sakti@sakti-server:~$ docker-compose --version
docker-compose version 1.21.2, build a133471
sakti@sakti-server:~$
```

**Gambar 2.10** Tampilan hasil instalasi Docker Compose

- i. Hasil implementasi dari **Modul 2.8** tentang cara konfigurasi php.Dockerfile dapat dilihat pada **Gambar 2.11** berikut ini:



```
sakti@sakti-server: ~/docker-project
GNU nano 2.9.3      php.Dockerfile      Modified
FROM php:7.4.3-apache

RUN docker-php-ext-install mysqli pdo pdo_mysql
```

**Gambar 2.11** Tampilan hasil konfigurasi php.Dockerfile

- j. Hasil implementasi dari **Modul 2.9** tentang cara konfigurasi docker-compose.yml dapat dilihat pada **Gambar 2.12** berikut ini:

```

sakti@sakti-server: ~/docker-project
GNU nano 2.9.3

version: "3"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "./raport_online:/var/www/html/"
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_DATABASE: raport_online
      MYSQL_USERNAME: root
      MYSQL_PASSWORD: root
      MYSQL_ROOT_PASSWORD: root
    volumes:
      - mysql-data:/var/lib/mysql
      - ./raport_online/db_mysql/raport_online.sql:/docker-entrypoint-initdb.d/raport_online.sql
    ports:
      - "3306:3306"

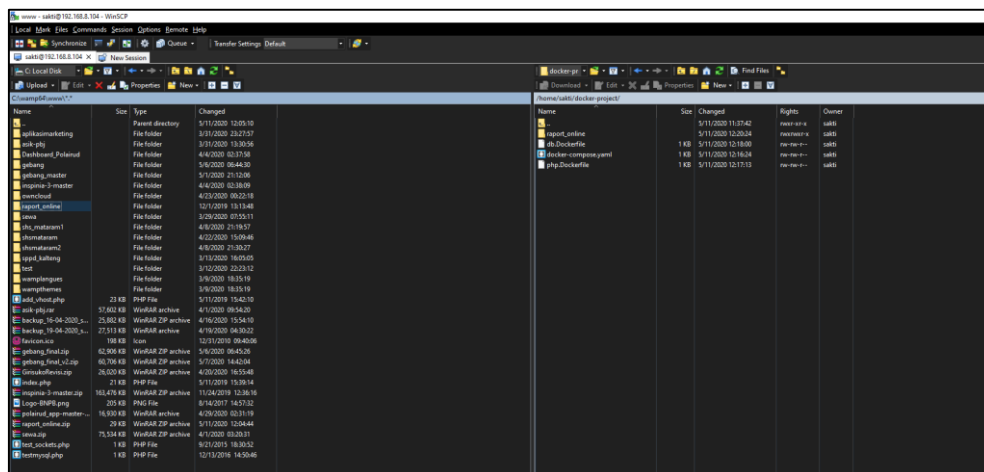
  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: mysql-server
      PMA_USER: root
      PMA_PASSWORD: root
    ports:
      - "5000:80"

volumes:
  mysql-data:

```

**Gambar 2.12** Tampilan hasil konfigurasi docker-compose.yml

- k. Selanjutnya mengcopy file website raport online ke folder docker-project menggunakan WinSCP dapat dilihat pada **Gambar 2.13** berikut ini:



**Gambar 2.13** Tampilan hasil copy file website raport online

1. Hasil implementasi dari **Modul 2.10** tentang cara konfigurasi koneksi.php dapat dilihat pada **Gambar 2.14** berikut ini:

```
<?php
$conn = new mysqli('mysql-server','root','root','raport_online');
if ($conn->connect_error){
    die("Error: " . mysqli_connect_error());
}
?>
```

**Gambar 2.14** Tampilan hasil konfigurasi koneksi.php

- m. Hasil implementasi dari **Modul 2.11** tentang cara menjalankan docker-compose.yaml dengan docker-compose dapat dilihat pada **Gambar 2.15** berikut ini:

```
sakti@sakti-server: ~/docker-project
sakti@sakti-server:~/docker-project$ docker-compose up -d
Pulling phpmyadmin (phpmyadmin/phpmyadmin:5.0.1)...
5.0.1: Pulling from phpmyadmin/phpmyadmin
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
96198bf1ad68: 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 docker-project_web-server_1 ... done
Creating docker-project_phpmyadmin_1 ... done
Creating docker-project_mysql-server_1 ... done
sakti@sakti-server:~/docker-project$
```

**Gambar 2.15** Tampilan hasil menjalankan Docker Compose

- n. Hasil implementasi dari **Modul 2.12** tentang cara melihat containers dalam docker dapat dilihat pada **Gambar 2.16** berikut ini:

```
sakti@sakti-server:~/docker-project
sakti@sakti-server:~/docker-project$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
86f205fca176	docker-project_web-server	"docker-php-entrypoi..."	4 minutes ago	Up 4 minutes	0.0.0.0:8080->80/tcp	docker-project_web-server_1
f1eb5ac5946	mysql:8.0.19	"docker-entrypoint.s..."	4 minutes ago	Up 4 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	docker-project_mysql-server_1
000f4d0d2843	phpmyadmin/phpmyadmin:5.0.1	"/docker-entrypoint..."	4 minutes ago	Up 4 minutes	0.0.0.0:5000->80/tcp	docker-project_phpmyadmin_1

```
sakti@sakti-server:~/docker-project$
```

**Gambar 2.16** Tampilan hasil containers docker

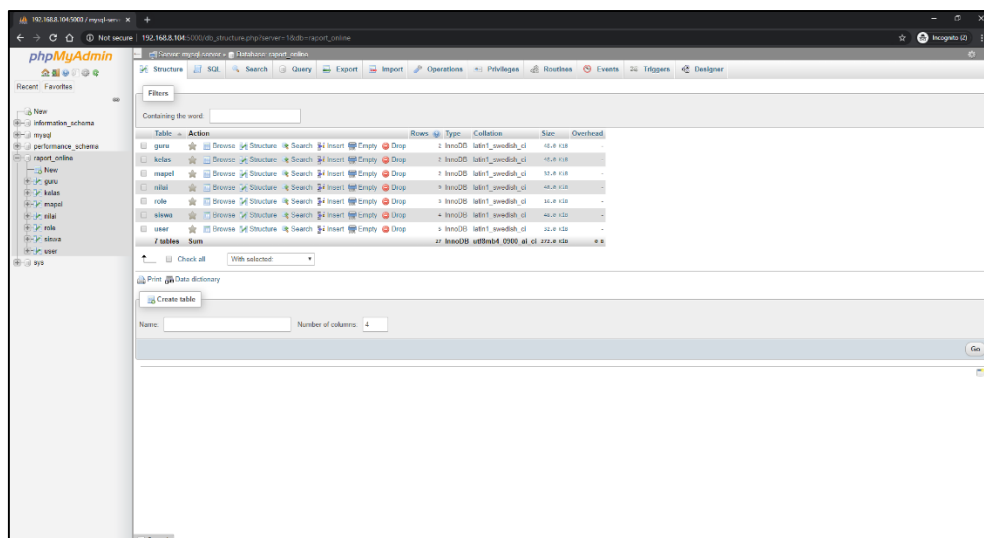
- o. Hasil implementasi dari **Modul 2.13** tentang cara melihat *ip address* dapat dilihat pada **Gambar 2.17** berikut ini:



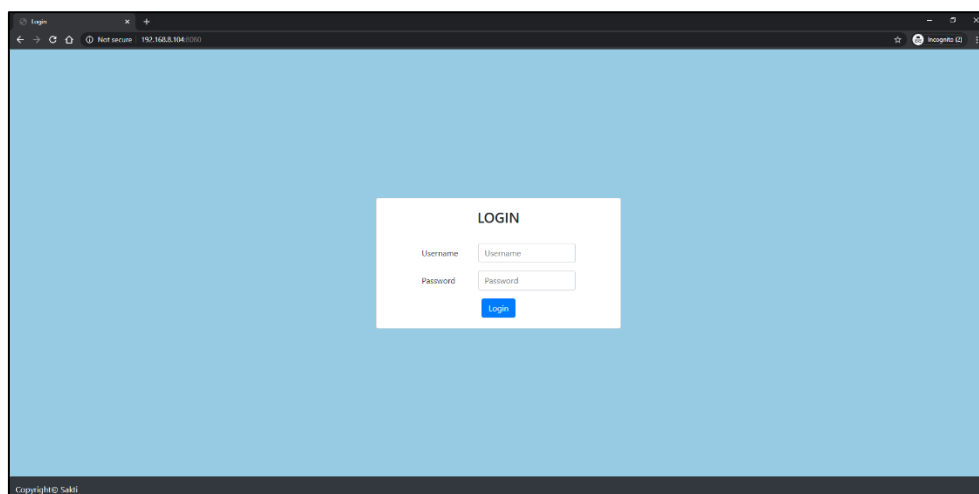
```
sakti@sakti-server:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f8:13:60 brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.105/24 brd 192.168.8.255 scope global dynamic enp0s3
        valid_lft 86372sec preferred_lft 86372sec
    inet6 fe80::a00:27ff:fef8:1360/64 scope link
        valid_lft forever preferred_lft forever
sakti@sakti-server:~$
```

**Gambar 2.17** Tampilan hasil melihat *ip address*

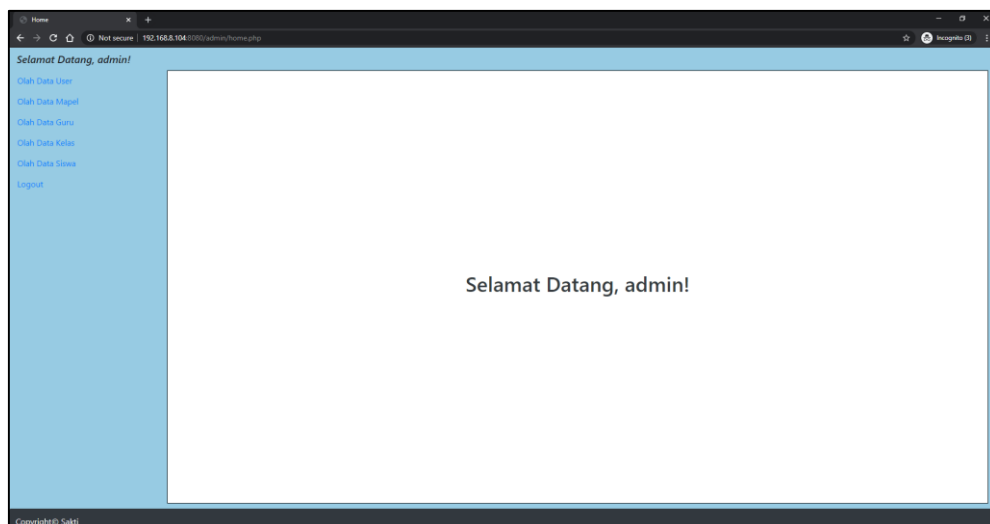
- p. Setelah semua containers aktif, selanjutnya adalah membuka website phpmyadmin dengan mengakses *alamat\_ip:5000* dan website report online *alamat\_ip:8000* dapat dilihat pada **Gambar 2.18**, **Gambar 2.19**, dan **Gambar 2.20** berikut ini:



**Gambar 2.18** Tampilan website phpmyadmin

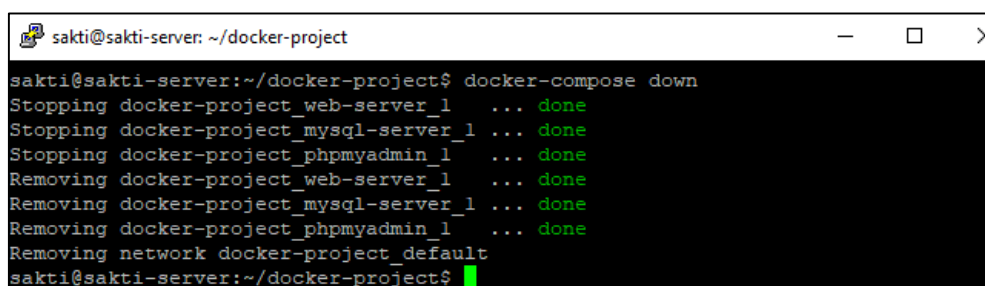


**Gambar 2.19** Tampilan awal website report online



**Gambar 2.20** Tampilan website raport online setelah login

- q. Hasil implementasi dari **Modul 2.14** tentang cara menghentikan containers docker yang aktif dapat dilihat pada **Gambar 2.21** berikut ini:

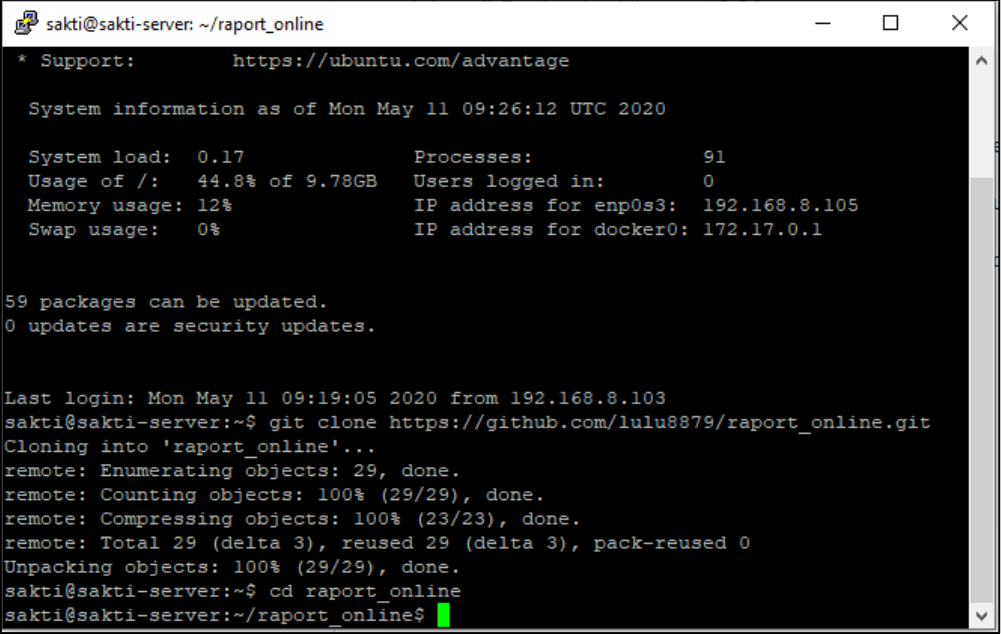


**Gambar 2.21** Tampilan hasil menghentikan containers docker yang aktif

## 2.5 Hasil Implementasi

Hasil dari pembuatan Dockerfile yaitu memudahkan pemasangan aplikasi ke sistem sehingga tidak perlu melakukan konfigurasi lebih lanjut. Sebagai bukti bahwa pembuatan Dockerfile sebelumnya dapat diimplementasikan ke sistem lain, maka dilakukan *testing* pengimplementasian Dockerfile ke sistem baru dengan OS Ubuntu Server 18.04.3 dan website bisa langsung diakses tanpa perlu konfigurasi lebih lanjut. Berikut tahapan pengimplementasian Dockerfile :

- a. Mengclone repository raport online di github sesuai dengan **Modul 2.15**.



```
sakti@sakti-server: ~/raport_online
* Support:      https://ubuntu.com/advantage

System information as of Mon May 11 09:26:12 UTC 2020

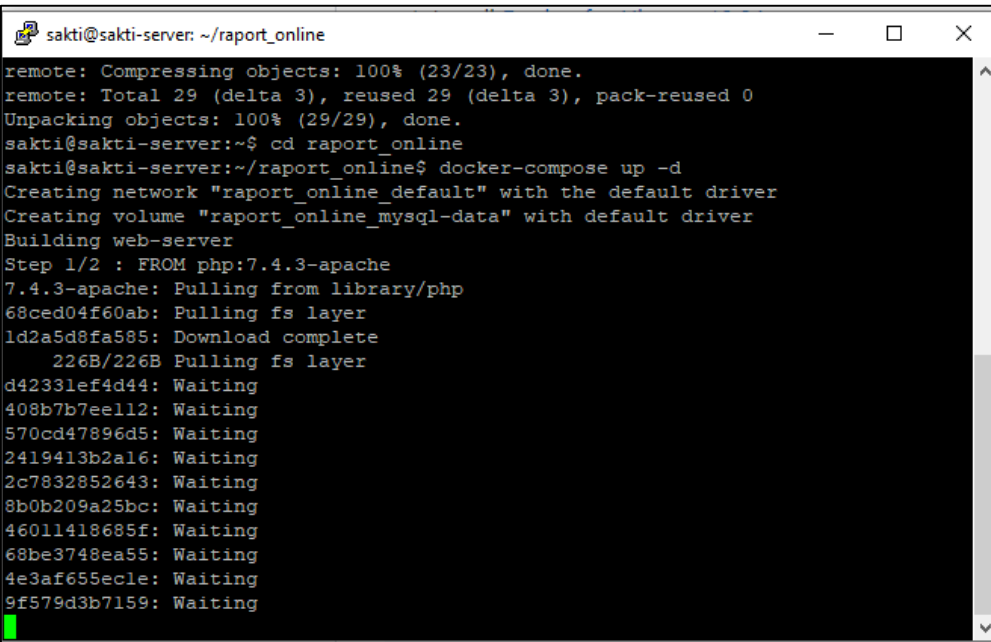
System load:  0.17          Processes:           91
Usage of /:   44.8% of 9.78GB Users logged in:       0
Memory usage: 12%          IP address for enp0s3: 192.168.8.105
Swap usage:   0%           IP address for docker0: 172.17.0.1

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

Last login: Mon May 11 09:19:05 2020 from 192.168.8.103
sakti@sakti-server:~$ git clone https://github.com/lulu8879/raport_online.git
Cloning into 'raport_online'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 29 (delta 3), reused 29 (delta 3), pack-reused 0
Unpacking objects: 100% (29/29), done.
sakti@sakti-server:~$ cd raport_online
sakti@sakti-server:~/raport_online$
```

**Gambar 2.22** Tampilan hasil *clone repository* github

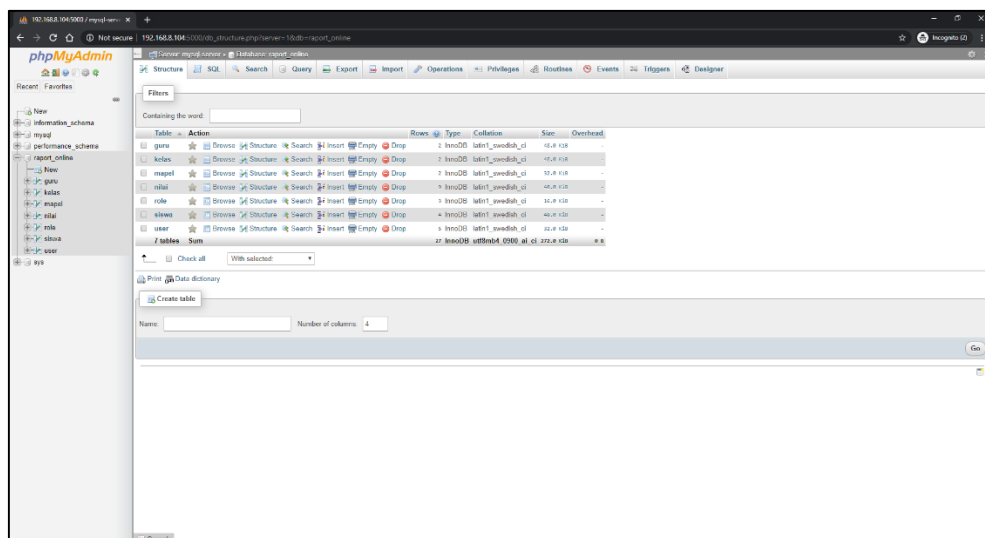
- b. Install docker dan docker-compose sesuai dengan **Modul 2.5**, **Modul 2.6**, dan **Modul 2.7**.
- c. Pindah direktori dan membuild docker sesuai dengan Modul 2.11.



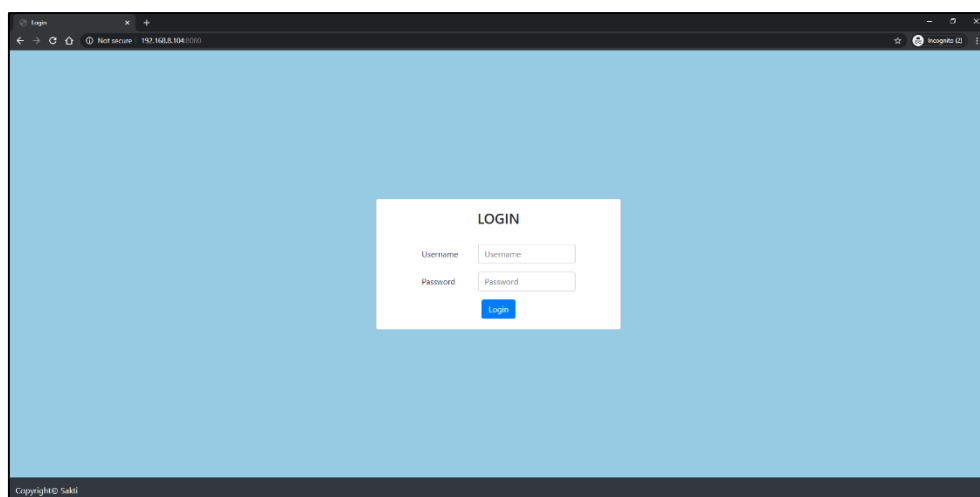
```
sakti@sakti-server: ~/raport_online
remote: Compressing objects: 100% (23/23), done.
remote: Total 29 (delta 3), reused 29 (delta 3), pack-reused 0
Unpacking objects: 100% (29/29), done.
sakti@sakti-server:~$ cd raport_online
sakti@sakti-server:~/raport_online$ docker-compose up -d
Creating network "raport_online default" with the default driver
Creating volume "raport_online_mysql-data" with default driver
Building web-server
Step 1/2 : FROM php:7.4.3-apache
7.4.3-apache: Pulling from library/php
68ced04f60ab: Pulling fs layer
1d2a5d8fa585: Download complete
226B/226B Pulling fs layer
d42331ef4d44: Waiting
408b7b7eell2: Waiting
570cd47896d5: Waiting
2419413b2a16: Waiting
2c7832852643: Waiting
8b0b209a25bc: Waiting
46011418685f: Waiting
68be3748ea55: Waiting
4e3af655ec1e: Waiting
9f579d3b7159: Waiting
```

**Gambar 2.23** Tampilan hasil build docker-compose.yaml

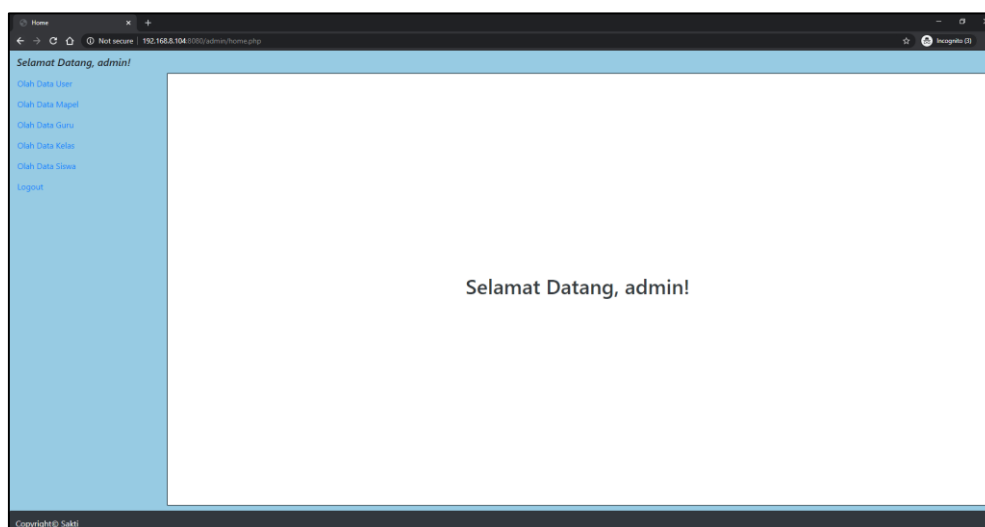
- d. Cek *ip address* sesuai **Modul 2.13** dan buka website phpmyadmin dan website *raport online*



**Gambar 2.24** Tampilan website phpmyadmin



**Gambar 2.25** Tampilan awal website raport online

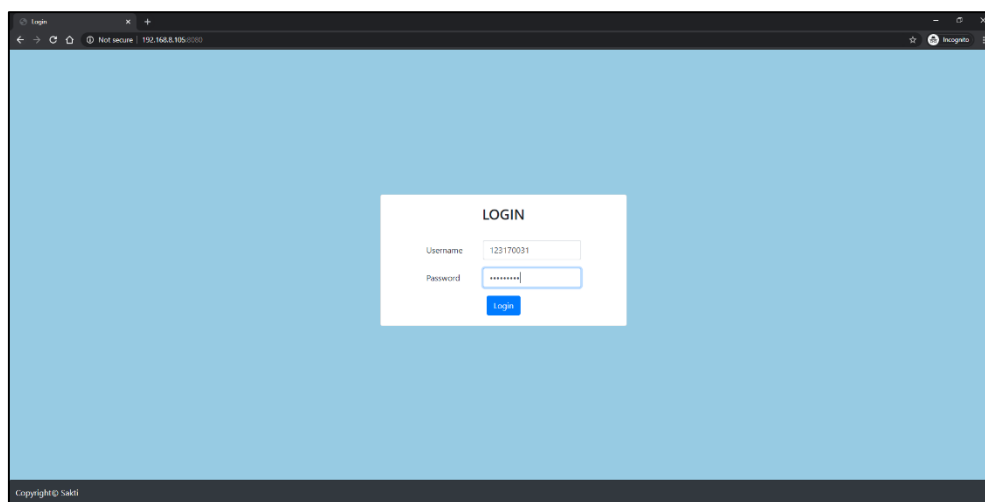


**Gambar 2.26** Tampilan website raport online setelah login

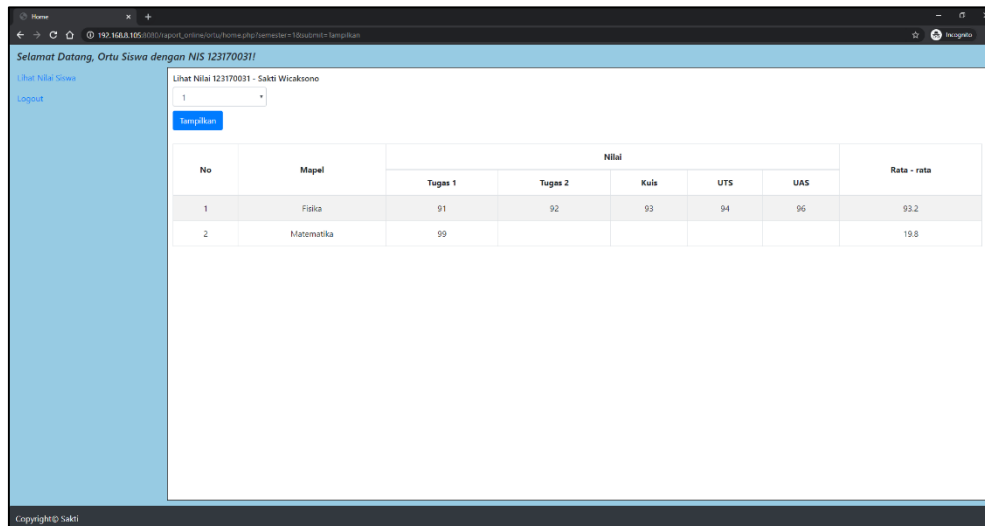
## 2.6 Pengujian Singkat

Masalah yang dihadapi terkait nilai siswa adalah orang tua siswa hanya dapat melihat nilai ketika raport sudah dibagikan, yaitu pada akhir semester. Terhadap sistem informasi yang telah dirancang, dilakukan pengujian terhadap permasalahan tersebut.

Pertama-tama orang tua siswa memasukkan NIM pada dashboard sistem seperti yang terlihat pada **Gambar 2.27** dan **Gambar 2.28** berikut.



**Gambar 2.27** Tampilan awal website raport online dengan akun NIM



**Gambar 2.28** Tampilan nilai siswa

Hasil dari apa yang telah dilakukan orang tua dapat melihat dan memantau nilai anaknya tanpa perlu menunggu raport jadi. Hal ini dapat terlihat informasi nilai siswa dapat diakses kapanpun dan dari mana saja selama ada koneksi internet.

Masalah berikutnya adalah penggabungan nilai antar guru yang harus dilakukan secara manual, yaitu dengan nilai yang tulis tangan ke Excel atau dari data nilai Excel guru A ke guru B yang kemudian harus diolah lagi oleh wali kelas. Adanya sistem ini kemudian

dilakukan pengujian terhadap kolaborasi antar guru mapel, sehingga wali kelas tidak perlu khawatir terhadap nilai mapel lain yang diinputkan oleh guru pengampu karena sudah diolah oleh sistem. Terhadap sistem informasi yang telah dirancang, dilakukan pengujian terhadap permasalahan tersebut.

Pertama-tama, guru A login dan menginputkan nilai mapel yang diampunya, misal mapel Fisika seperti pada **Gambar 2.29** berikut.

No	NIS	Nama	Nilai					Option
			Tugas 1	Tugas 2	Kuis	UTS	UAS	
1	123170031	Sakti Wicaksono	91	92	93	94	96	<a href="#">Edit</a>
2	123170032	Riky	81	82				<a href="#">Edit</a>

**Gambar 2.29** Tampilan daftar nilai fisika tiap siswa

Kedua, guru B login dan menginputkan nilai mapel yang diampunya, misal mapel Matematika seperti pada **Gambar 2.30** berikut.

No	NIS	Nama	Nilai					Option
			Tugas 1	Tugas 2	Kuis	UTS	UAS	
1	123170031	Sakti Wicaksono	99					<a href="#">Edit</a>
2	123170032	Riky		96				<a href="#">Edit</a>

**Gambar 2.30** Tampilan daftar nilai matematika tiap siswa

Terakhir, orang tua siswa dengan NIM 123170031 login dan kemudian orang tua siswa dapat melihat nilai Fisika dan Matematika yang telah diinputkan oleh guru A dan guru B seperti pada **Gambar 2.31** berikut.

No	Mapel	Nilai					Rata-rata
		Tugas 1	Tugas 2	Kuis	UTS	UAS	
1	Fisika	91	92	93	94	96	93.2
2	Matematika	99					19.8

**Gambar 2.31** Tampilan nilai siswa dengan NIM 123170031

Hasil dari apa yang telah dilakukan nilai dari guru A dan guru B sudah saling terhubung dan wali kelas tidak perlu merekap ulang nilai mapel dari beberapa guru ke tiap siswa. Hal ini dapat terlihat informasi nilai siswa dapat diolah oleh guru manapun dan dari mana saja selama ada koneksi internet.

Kesimpulan dari percobaan tersebut bahwa sistem yang dibuat mampu memberikan layanan yang sangat bermanfaat bagi guru, siswa, dan orang tua siswa, selain dapat diakses dari mana saja, biaya untuk pembuatan raport menjadi terpengkas 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

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Pengerjaan LAMPP								
4.	Pengerjaan Laporan Progress Project 1								
5.	Pengerjaan Docker								
6.	Pengerjaan Laporan Progress Project 2								
7.	Pengerjaan Docker Lanjutan dan Testing								
8.	Pengerjaan Laporan Akhir								

##### 3.2 Keterangan Pembagian Tugas

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

**Tabel 3.2** Pembagian Tugas Proyek

No.	Keterangan Tugas	Penanggung Jawab
1.	Perancangan Arsitektur Cloud Computing	Rafly
2.	Latar Belakang Masalah	Rafly
3.	Tujuan Proyek Akhir	Rafly
4.	Manfaat Proyek Akhir	Rafly
5.	Tahap Penyelesaian Proyek Akhir	Sakti
6.	BAB II	Rafly dan Sakti
7.	BAB III	Sakti
8.	BAB IV	Sakti
9.	Instalasi LAMPP	Rafly
10.	Instalasi Docker dan konfigurasi dockerfile	Sakti
11.	Pengujian dockerfile pada sistem lain	Sakti



## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berdasarkan dua masalah yang ada, yaitu orang tua siswa hanya bisa melihat nilai anaknya ketika raport sudah jadi dan pengolahan data nilai dari banyak guru dilakukan oleh wali kelas menggunakan Excel, dilakukan pengujian secara acak menggunakan website raport *online* untuk menyelesaikan masalah tersebut. Hasil yang didapat adalah pertama orang tua siswa dapat melihat nilai anaknya kapanpun tanpa harus menunggu raport jadi sehingga mudah untuk memantau anaknya. Kedua wali kelas tidak perlu menggabungkan dan mengolah data nilai dari tiap guru pengampu karena sudah tersinkronkan oleh sistem.

Berdasarkan pembagian tugas proyek, hasil yang didapat adalah sistem dapat dikonfigurasi menggunakan Docker dan dapat diimplementasikan ke sistem lain sehingga website raport *online* bisa langsung digunakan tanpa konfigurasi lanjut.

#### **4.2 Saran**

Berdasarkan hasil pengerjaan proyek, akan lebih baik jika ada tutorial baik menggunakan slide powerpoint atau video tutorial *youtube* mengenai materi Docker yang sebelumnya tidak tersampaikan pada perkuliahan karena masalah pandemik yang terjadi. Literasi dari internet memang banyak, namun tidak ada acuan pasti dalam penggunaan Docker.

## DAFTAR PUSTAKA

- Anderson, M. (2018, Juli 9). *How To Install Docker Compose on Ubuntu 18.04*. Diambil kembali dari DigitalOcean, LLC: <https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-18-04>
- Bashir, F. (2020, Maret 9). *How To Set Up Laravel, Nginx, and MySQL with Docker Compose*. Diambil kembali dari DigitalOcean, LLC: <https://www.digitalocean.com/community/tutorials/how-to-set-up-laravel-nginx-and-mysql-with-docker-compose>
- Hogan, B. (2018, Juli 5). *How To Install and Use Docker on Ubuntu 18.04*. Diambil kembali dari DigitalOcean, LLC: <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>
- Import data.sql MySQL Docker Container*. (2017, Mei 9). Diambil kembali dari Stack Exchange Inc: <https://stackoverflow.com/questions/43880026/import-data-sql-mysql-docker-container>
- Shovon, S. (2020, Maret). *Set up a LAMP server with Docker*. Diambil kembali dari Linux Hint LLC: [https://linuxhint.com/lamp\\_server\\_docker/](https://linuxhint.com/lamp_server_docker/)

## **LAMPIRAN**

