

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING**  
**LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI NILAI SEKOLAH DASAR MENGGUNAKAN UBUNTU**  
**LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA**



**DISUSUN OLEH:**

**NAMA ANGGOTA** : MUHAMMAD RAVI HABIBILLAH 123170039  
MUHAMMAD FAJAR SIDIQ 123170078  
**KELAS** : D  
**ASISTEN PRAKTIKUM** : JALUANDA PARAMA, S.Kom.  
WAHYU AJI NUGROHO, S.Kom.

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

## HALAMAN PENGESAHAN

### SISTEM INFORMASI NILAI SEKOLAH DASAR MENGGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKER FILENYA

Disusun oleh :

Muhammad Ravi Habibillah

123170039

Muhammad Fajar Sidiq

123170078

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

Alhamdulillah, puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan inayah-Nya sehingga kami dapat menyelesaikan laporan praktikum teknologi cloud computing yang berjudul Sistem Informasi Nilai Sekolah Dasar menggunakan Ubuntu LAMPP dan Proses Pembuatan *Dockerfilenya*.

Terima kasih saya ucapkan kepada asisten laboratorium yang telah membantu kami baik secara moral maupun materi. Terima kasih juga saya ucapkan kepada teman-teman seperjuangan yang telah mendukung kami sehingga kami bisa menyelesaikan tugas ini tepat waktu.

Pada laporan yang berjudul Sistem Informasi Nilai Sekolah Dasar menggunakan *Ubuntu LAMPP* dan dan Proses Pembuatan *Docker Filenya*.. Kami membuat sistem yang mengimplementasikan *cloud computing* tersebut untuk mempermudah menampilkan nilai murid sekolah dasar dan mempermudah manajemen data yang tersedia, dengan menggunakan teknologi *cloud* memudahkan untuk menyimpan data sehingga tidak mudah rusak ataupun hilang.

Kami menyadari, bahwa laporan praktikum teknologi cloud computing yang kami buat ini masih jauh dari kata sempurna baik segi penyusunan, bahasa, maupun penulisannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pembaca guna menjadi acuan agar penulis bisa menjadi lebih baik lagi di masa mendatang.

Semoga laporan praktikum teknologi cloud computing ini bisa menambah wawasan para pembaca dan bisa bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan.

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>1</b>   |
| 1.1. Latar Belakang Masalah .....                          | 1          |
| 1.2. Tujuan Proyek Akhir .....                             | 1          |
| 1.3. Manfaat Proyek Akhir .....                            | 2          |
| 1.4. Tahap Penyelesaian Proyek Akhir .....                 | 2          |
| <b>BAB II ISI DAN PEMBAHASAN .....</b>                     | <b>3</b>   |
| 2.1. Komponen yang Digunakan .....                         | 3          |
| 2.2. Rancangan Arsitektur <i>Cloud Computing</i> .....     | 4          |
| 2.3. Parameter dan Konfigurasi .....                       | 5          |
| 2.4. Tahap Implementasi .....                              | 13         |
| 2.5. Hasil Implementasi .....                              | 21         |
| 2.6. Pengujian Singkat .....                               | 23         |
| <b>BAB III JADWAL Pengerjaan dan Pembagian Tugas .....</b> | <b>31</b>  |
| 3.1. Agenda Pengerjaan .....                               | 31         |
| 3.2. Keterangan Pembagian Tugas .....                      | 31         |
| <b>BAB IV KESIMPULAN DAN SARAN .....</b>                   | <b>32</b>  |
| 4.1. Kesimpulan .....                                      | 32         |
| 4.2. Saran .....   | 32         |
| <b>DAFTAR PUSTAKA .....</b>                                | <b>33</b>  |
| <b>LAMPIRAN .....</b>                                      | <b>34</b>  |

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Teknologi komputasi *cloud* di zaman sekarang banyak dibutuhkan karena semua data berada dan disimpan di *server internet*, begitu juga dengan aplikasi ataupun *software* yang pada umumnya dibutuhkan pengguna, semuanya berada di komputer *server*. Dengan menggunakan *cloud* pengguna tidak perlu ikut merawa penyimpanan, dilihat dari biaya lebih murah, dan paling penting kinerja suatu sistem atau bisnis meningkat (Arif Wicaksono, 2015).

Awalnya *cloud computing* ditemukan oleh J.C.R. Licklider saat membangun *Advanced Research Projects Agency Network* (ARPANET) pada tahun 1969. Kemudian dikembangkan oleh John McCarthy yang memiliki ide untuk menjadikannya infrastruktur *public*. Kemudian pada tahun 1999, *storage.salesforce.com* merupakan pencetus aplikasi perusahaan yang dijalankan menggunakan *internet*. Kemudian berkembang pesat pada tahun 2006 disusul oleh Amazon (Arif Wicaksono, 2015).

Berkaitan dengan sistem informasi yang akan dibuat, harus diselesaikan dengan *cloud computing* agar dapat dilihat secara mudah kapanpun dan dimanapun, data yang tersimpan didalamnya tidak mudah rusak atau hilang. Manfaat utama yang mempermudah akses untuk melihat nilai siswa, kedepannya ingin mengimplementasikannya di sekolah dasar serta menambahkan fitur yang dibutuhkan setelah hasil implementasi tersebut.

Sistem informasi sekolah dasar menggunakan *website* sebagai wadah penyajian informasinya, dalam pengembangannya dipastikan *website* yang akan di *hosting* sudah dapat berjalan dengan baik. Kemudian kami menggunakan VMware untuk menginstal Ubuntu Server versi 18.04 dan menginstal LAMP. Kemudian kami melakukan *hosting* pada *website* tersebut agar dapat diakses secara lokal. Untuk perangkat keras kami menggunakan komputer dengan spesifikasi Core i5-7200U dengan RAM 8 gb.

### 1.2 Tujuan Proyek Akhir

Berdasarkan latar belakang proyek akhir yang telah di jelaskan, tujuan dari pembuatan proyek akhir ini sebagai berikut :

1. Menerapkan arsitektur *cloud computing* untuk membangun Sistem Informasi Nilai Sekolah Dasar menggunakan Ubuntu LAMP dan Proses Pembuatan *Docker Filenya* yang telah dibuat dengan menggunakan Ubuntu *server* 18.04 dan LAMP.

2. Menghasilkan sistem informasi sekolah dasar yang mempermudah orangtua memantau perkembangan anaknya dan untuk tenaga pengajar sebagai evaluasi hasil mengajar dari sekolah dasar tersebut.

### 1.3 Manfaat Proyek Akhir

Manfaat yang didapatkan dari pembuatan sistem informasi sekolah dasar sebagai berikut :

1. Memudahkan orangtua untuk melihat perkembangan putra/ putrinya.
2. Mempermudah para pengajar untuk melakukan manajemen pada nilai siswa.
3. Data siswa dan rapor tidak mudah rusak karena tersimpan di *cloud*.
4. Mengurangi faktor *human error* jika dikerjakan secara manual.
5. Memudahkan akses sistem informasi dimanapun dan kapanpun selama ada jaringan internet.

### 1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari sistem informasi nilai sekolah dasar untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS
2. Penginstalan dan konfigurasi Ubuntu Server.
3. Mengintegrasikan sistem informasi nilai sekolah dasar kedalam Ubuntu Server yang sudah diinstall LAMP
4. Menentukan konfigurasi Dockerfile dan docker-compose yang dibutuhkan untuk proses menjalankan service LAMP serta mengintegrasikan sistem informasi nilai sekolah dasar dengan Dockerfile dan docker-compose tersebut.
5. Merancang topologi *cloud computing* pada tugas 1 dan tugas 2.
6. Menguji keandalan arsitektur cloud computing yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dsb pada Ubuntu Server.
7. Menghasilkan Sistem Informasi Nilai Sekolah Dasar yang berbasis cloud computing sesuai standar ISO 9001.

## BAB II

### ISI DAN PEMBAHASAN

#### 2.1 Komponen yang Digunakan

Untuk menyelesaikan proyek akhir, digunakan beberapa komponen hardware dan software sebagai sarana pengejaan proyek akhir agar dapat diselesaikan. Kelompok kami menggunakan laptop dengan spesifikasi :

- Display : 1366 x 768 (32 bit) (60hz)
- Processor : Intel® Core™ i5-7200U CPU @ 2.50 GHz (4CPUs), ~ 2.7 GHz
- Memory : 8192MB RAM
- Hard Drive : 1TB HDD
- Graphics : Intel(R) HD Graphics 620
- Operating System : Windows 10

Dalam pengoprasiannya kami juga menggunakan software untuk membantu pengerjaan proyek akhir, antara lain :

- VMware Workstation 15 Pro (<https://www.vmware.com/id/products/workstation-pro/workstation-pro-evaluation.html>)
- Ubuntu Server 18.04.4 (<https://releases.ubuntu.com/18.04.4/>)
- Docker 19.03.8
- Putty 0.73 (<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>)
- WinSCP 5.17 (<https://winscp.net/eng/download.php>)

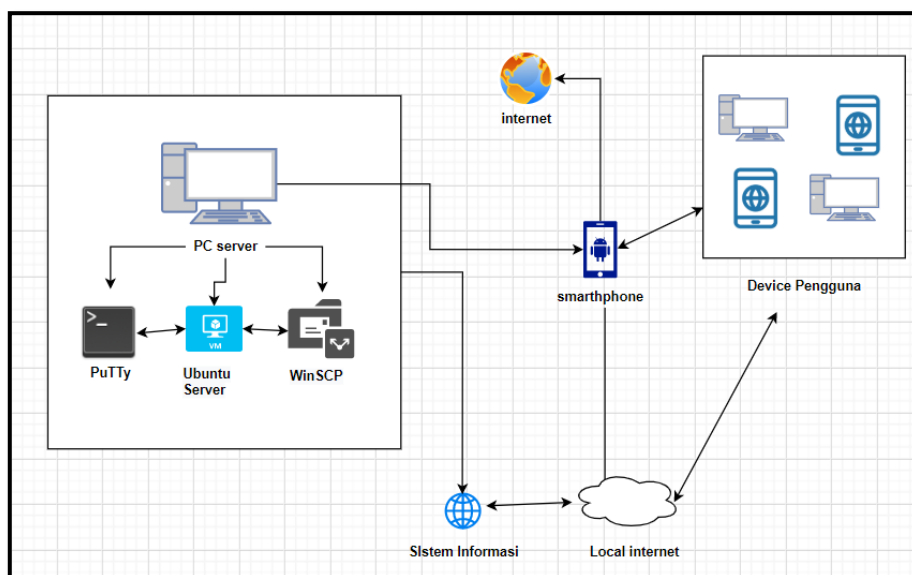
Pada tugas 2 kami menggunakan Ubuntu Server yang sama dengan tugas 1 dan kemudian kami lakukan penginstallan Docker pada Ubuntu Server tersebut. Kami menggunakan beberapa *image* yang memiliki spesifikasi sebagai berikut untuk menjalankan container yang diinginkan:

**Tabel 2.1.1** Spesifikasi *Image* untuk tugas kedua

| No. | Nama Parameter | Nilai            | Keterangan  |
|-----|----------------|------------------|---|
| 1.  | LAMPP          | Apache 2.4       | Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS. |
|     |                | PHP 7.4.3        | Preprosesor Bahasa pemrograman PHP                        |
|     |                | MySQL 8.0.19     | Sebagai tempat penyimpanan data                           |
|     |                | Phpmyadmin 5.0.1 | Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS. |

## 2.2.Rancangan Arsitektur *Cloud Computing*

Pada proyek akhir ini, kami menggunakan laptop sebagai IaaS, sedangkan windows dan Vmware berjalan sebagai PaaS. Koneksi antar device menggunakan *hotspot* Smartphone. Berikut ini ilustrasinya :



**Gambar 2.2.1** Topologi Ubuntu server

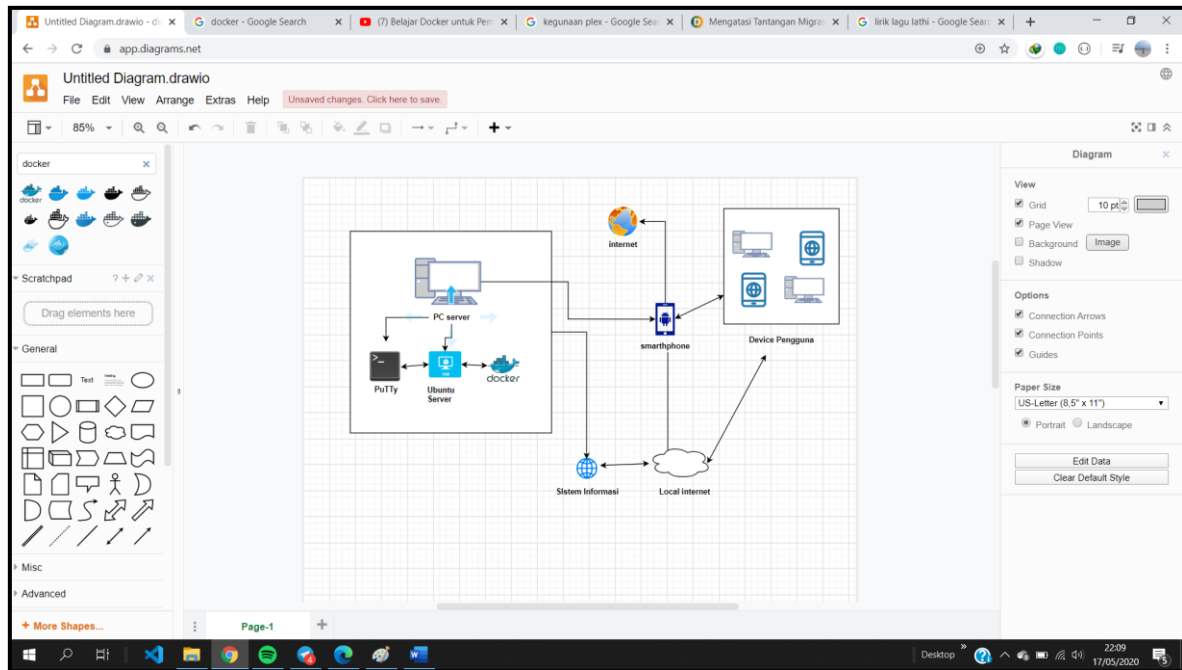
Dari Ilustrasi diatas digunakan *hotspot* digunakan sebagai LAN antar device agar pengguna saling terhubung melalui web yang telah dibuat oleh komputer server. Komputer server membutuhkan penyimpanan *cloud* yang terhubung ke jaringan agar semua orang dapat mengaksesnya. Komputer yang telah terinstal Windows dibutuhkan VMware Workstation, Putty dan WinSCP. Jika semuanya telah siap masuk melalui ip address yang sesuai ubuntu server kemudian data web dimasukkan kedalam WinSCP untuk proses upload ke server sehingga dapat diakses oleh banyak orang.

Kemudian pada proyek ini kami mengatur Docker client untuk mengoperasikan docker container yang berisi aplikasi. Perintah yang dikirim docker *client* akan dikerjakan oleh docker daemon seperti instalasi/ membuat *images* kemudian terbuatlh sebuah *container*, menjalankan atau menghentikan *service container*, menghapus *container*, dan masih banyak lagi. Keseluruhan docker dijalankan oleh docker engine.

Pada proyek kedua kami menggunakan dockerfile dari program yang telah dibuat, digunakan bentuk rancangan arsitektur PaaS dimana infrastruktur berupa laptop sebagai layer utama berada di bagian paling bawah, kemudian di atasnya adalah sistem operasi induk atau bawaan. *Container* tidak memiliki *Guest Operating System*. Container yang berisi Sistem informasi sekolah dasar akan digabung beserta *library*, *file* konfigurasi, dan segala



hal yang dibutuhkannya akan menggunakan sistem operasi induk atau bawaan tanpa mengganggu atau mempengaruhi sistem operasi induk atau bawaan tersebut. Ilustrasi mengenai rancangan topologi jaringan docker sebagai berikut :



Gambar 2.2.2 Topologi Ubuntu server

## 2.3 Parameter dan Konfigurasi

untuk dapat digunakan, maka VMware Workstation perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada Tabel 2.3.1 berikut ini:

Tabel 2.3.1 Konfigurasi pada VMware Workstation yang digunakan

| No. | Nama Parameter              | Nilai  | Keterangan   |
|-----|-----------------------------|--|--|
| 1.  | VMware Workstation 15.5 Pro | 15.5.0                                       | VMware Workstation 15.5 Pro  |
| 2.  | VMware ESXi build           | 13644319                                     | Keterangan <i>build (patch) number</i> .                                   |
| 3.  | IPv4 hypervisor             | IP: 172.20.0.2 (Static)                      | IP number pada Ubuntu  |
|     |                             | SM: 255.255.255.0                            | Kelas IP/subnet mask yang digunakan pada Ubuntu.                           |
|     |                             | DNS: 127.0.0.53                              | Alamat IP untuk fungsionalitas DNS untuk Ubuntu.                           |
|     |                             | GW: 127.20.0.1                               | Alamat untuk gateway atau gerbang menuju akses jaringan luar Untuk Ubuntu. |
| 4.  | Processor info              | 2 x Intel(R) Core(TM) i5-7200 U CPU @ 2.5GHz | Jenis <i>processor</i> yang digunakan pada Ubuntu.                         |
| 5.  | RAM info                    | 2 GB Memory                                  | Kapasitas RAM pada Ubuntu.   |
| 6.  | Harddisk info               | 20 GB Memory                                 | Kapasitas Harddisk pada Ubuntu   |
| 7.  | Username info               | root   | Username pada akun ubuntu  |
| 8.  | Password info               | root   | Password pada akun Ubuntu  |
| 9.  | Ipv4 Docker                 | IP : 172.17.0.2                              | IP number untuk akses docker   |

Berikut ini spesifikasi Ubuntu OS yang telah dibuat pada VMware Workstation 15

Pro :

**Tabel 2.3.2** Spesifikasi ubuntu server pada VMware Workstation yang digunakan

| No. | Nama Parameter | Nilai            | Keterangan  |
|-----|----------------|------------------|---|
| 1.  | LAMPP          | Apache 2.4.29    | Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS. |
|     |                | PHP 7.2.24       | Bahasa pemrograman yang dapat disisipkan ke HTML.         |
|     |                | MySQL 5.7        | DBMS untuk manajemen basis data SQL.                      |
|     |                | phpMyAdmin 4.6.6 | Software yang digunakan untuk mengelola server MySQL.     |

Parameter yang digunakan untuk instalasi Apache dapat dilihat pada penjelasan

**Modul 2.2.1** 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 2.2.1** Parameter instalasi Apache

Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan

**Modul 2.2.2** 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 tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- mysql-server : nama paket aplikasi untuk MySQL

#### **Modul 2.2.2** Parameter instalasi MySQL

Parameter yang digunakan untuk proses pengaturan dasar pengamanan MySQL

dapat dilihat pada penjelasan **Modul 2.2.3** berikut ini:

```
$ sudo mysql_secure_installation
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- mysql\_secure\_installation : perintah untuk proses pengaturan dasar pengamanan MySQL berupa users, kata sandi, hak akses, dan sebagainya.

#### **Modul 2.2.3** Parameter pengaturan dasar pengamanan MySQL

Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan

**Modul 2.2.4** 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 paket aplikasi untuk PHP
- libapache2-mod-php : digunakan untuk menghubungkan apache dan php
- php-mysql : digunakan untuk menghubungkan php dan mysql

#### **Modul 2.2.4** Parameter instalasi PHP

Parameter yang digunakan untuk pengecekan bahwa instalasi PHP berhasil dan PHP dapat berjalan dengan sempurna dapat dilihat pada penjelasan **Modul 2.2.5** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- nano : perintah untuk membuat berkas / teks editor pada command-line linux
- /var/www/html/ : lokasi direktori Web Server Apache
- Info.php : berkas untuk menuliskan kodingan pada modul 2.6

#### **Modul 2.2.5** Parameter pengecekan keberhasilan instalasi PHP

Kodingan yang digunakan untuk mendukung pengecekan keberhasilan pada **Modul 2.2.5** dapat dilihat pada penjelasan **Modul 2.2.6** berikut ini:

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

Keterangan:

- Kodingan tersebut ditulis pada GNU nano, kemudian dilanjutkan dengan proses penyimpanan dengan menekan tombol 'ctrl + o' lalu enter

#### **Modul 2.2.6** Kodingan pada GNU nano

Parameter yang digunakan untuk instalasi phpmyadmin dapat dilihat pada penjelasan **Modul 2.2.7** 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 aplikasi
- phpmyadmin : nama paker aplikasi untuk PHP
- php-mbstring : nama paket untuk mengkonversi string

- php-gettext : nama paket untuk pesan multi-bahasa

### Modul 2.2.7 Parameter instalasi phpmyadmin

Parameter yang digunakan memperbaiki error saat mencoba login pada phpmyadmin dapat dilihat pada penjelasan **Modul 2.2.8** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- mysql : nama server
- root : username phpmyadmin

### Modul 2.2.8 Parameter memperbaiki error phpmyadmin

Sintaks yang digunakan untuk mengatur parameter plugin dari user root pada phpmyadmin dapat dilihat pada penjelasan **Modul 2.2.9** berikut ini:

```
mysql > UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('123170039') WHERE User = 'root';
mysql > FLUSH PRIVILEGES;
```

Keterangan:

- Untuk mengubah password dari user 'root' menjadi '123170039'
- FLUSH PRIVILEGES : sintaks untuk merefresh akun yang terkoneksi dengan phpmyadmin

### Modul 2.2.9 Sintaks parameter plugin

Parameter yang digunakan untuk mengubah hak akses folder pada WinSCP agar dapat dipindahkan ke dalam folder yang dituju dapat dilihat pada penjelasan **Modul 2.2.10** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- chown : mengubah permission berkas atau folder
- ravifajar : user yang diberikan permission untuk akses berkas tersebut
- /var/www/html/ : tujuan folder yang akan diubah hak aksesnya

### Modul 2.2.10 Parameter mengubah hak akses

Parameter yang digunakan untuk instalasi Docker dapat dilihat pada penjelasan **Modul 2.2.11** berikut ini:

```
$sudo apt update
```

Keterangan:

Untuk mengupdate package yang terinstall di Ubuntu.

```
$sudo apt install apt-transport-https ca-certificates curl
softwareproperties-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 apt-key fingerprint 0EBFCD88
```

Keterangan :

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt-key : digunakan untuk mengelola daftar kunci yang digunakan oleh apt untuk mengautentikasi paket.
- fingerprint 0EBFCD88 : memasukkan 8 digit terakhir dari kunci 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) 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 \$(lsb\_release -cs) stable" : merupakan sumber package Docker dengan acuan deb adalah jenis package,
- [arch=amd64] : jenis arsitektur
- https://download.docker.com/linux/ubuntu : link sumber package Docker
- \$(lsb\_release -cs) : versi Ubuntu yang digunakan
- stable : versi rilis yang ingin digunakan.

```
$ sudo apt update
```

Keterangan:

Untuk mengupdate package yang terinstall di Ubuntu.

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

Keterangan:

Untuk mengunduh dan menginstall Docker.

```
$ sudo apt-cache madison docker-ce
```

Keterangan:

```

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt-cache madison : digunakan untuk melihat semua versi paket yang terdapat dalam arsip

$ sudo apt-get install docker-ce= docker-cecli= containerd.io

Keterangan : -
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- VERSION_STRING diganti dengan nama versi yang ingin diinstal. Nama versi terdapat di kolom kedua

$ sudo docker run hello-world

Keterangan :
Untuk pengecekan apakah Docker sudah terinstall dengan benar

```

### Modul 2.2.11 Parameter instalasi Docker

Parameter yang digunakan untuk instalasi Docker Compose dapat dilihat pada penjelasan **Modul 2.2.12** berikut ini:

```

$ sudo curl -L
https://github.com/docker/compose/releases/download/1.21.2/dockercompos
e-`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

$ 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 2.2.12 Parameter instalasi Docker Compose

Parameter yang digunakan untuk konfigurasi dari php.Dockerfile dapat dilihat pada penjelasan **Modul 2.2.13** berikut ini:

```

$ nano php.Dockerfile

Keterangan:
Untuk mengedit file (text editor dalam terminal).
- nano : perintah untuk menjalankan 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 apt-get update && \
    apt-get install -y libicu-dev && \
    docker-php-ext-configure intl && \
    docker-php-ext-install int && \
    docker-php-ext-install mysqli
RUN a2enmod rewrite
```

Keterangan:

Untuk menginstall extensions php yang diperlukan dan mererefresh module apache

### Modul 2.2.13 Parameter konfigurasi php.Dockerfile

Parameter yang digunakan untuk konfigurasi docker-compose.yaml dapat dilihat pada penjelasan **Modul 2.2.14** berikut ini:

```
$ nano docker-compose.yaml
```

Keterangan:

Untuk mengedit file (text editor dalam terminal).

- nano : perintah untuk menjalankan text editor dalam terminal linux (bawaan dari Ubuntu)
- docker-compose.yaml : nama file yang diingin diedit

```
# Isi dari docker-compose.yaml
```

```
version: "3.7"
```

```
services:
```

```
  web-server:
```

```
    build:
```

```
      dockerfile: php.Dockerfile
```

```
      context: .
```

```
    restart: always
```

```
    volumes:
```

```
      - "../ProjectWeb/./var/www/html/"
```

```
    ports:
```

```
      - "8080:80"
```

```
  mysql-server:
```

```
    image: mysql:8.0.19
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_USER: root
```

```
      MYSQL_PASSWORD: secret
```

```
      MYSQL_DATABASE: db_sekolah
```

```
      MYSQL_ROOT_PASSWORD: secret
```

```
    volumes:
```

```
      - mysql-data:/var/lib/mysql
```

```
  phpmyadmin:
```

```
    image: phpmyadmin/phpmyadmin:5.0.1
```

```
    restart: always
```

```
    environment:
```

```
      PMA_HOST: mysql-server
```

```
      PMA_USER: root
```

```
      PMA_PASSWORD: secret
```

```
    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 dockercompose.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 : mendefinsikan environment mysql untuk nama database yang akan digunakan
- MYSQL\_USERNAME : mendefinsikan environment mysql untuk nama username yang akan digunakan
- MYSQL\_PASSWORD : mendefinsikan environment mysql untuk password mysql yang akan digunakan
- MYSQL\_ROOT\_PASSWORD : mendefinsikan 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 2.2.14 Parameter konfigurasi docker-compose.yaml

Parameter yang digunakan untuk menjalankan Docker Compose dapat dilihat pada penjelasan **Modul 2.2.15** berikut ini:

```
$ cd docker-project
```

#### Keterangan :

Untuk berpindah direktori ke direktori yang terdapat dockercompose.yaml yang akan dijalankan.

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

```
$ sudo 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 dockercompose.yaml
  dan menjalankan docker
- -d : parameter dari docker-compose untuk menjalankan containers di
  background
```

#### Modul 2.2.15 Parameter konfigurasi menjalankan Docker Compose

Parameter yang digunakan untuk melihat containers Docker yang sudah sudah dijalankan sebelumnya pada **Modul 2.2.15** dapat dilihat pada penjelasan **Modul 2.2.16** berikut ini:

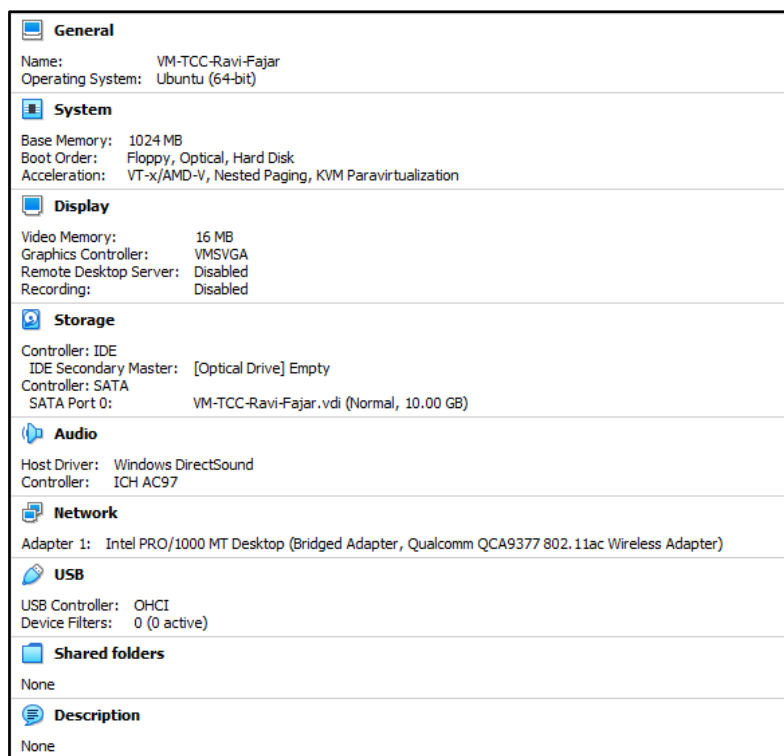
```
$ sudo docker-compose ps

Keterangan :
Untuk melihat containers Docker yang terpasang.
- docker-compose : perintah untuk menjalankan docker-compose
- ps : perintah dari docker untuk menampilkan containers docker
yang aktif
```

#### Modul 2.2.16 Parameter melihat containers Docker yang sedang dijalankan

## 2.4 Tahap Implementasi

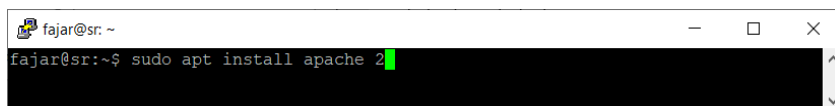
Pada implementasi Sistem Informasi Nilai Sekolah Dasar menggunakan Ubuntu LAMPP dan Proses Pembuatan *Docker Filenya* yang akan menggunakan ubuntu *server* 18.04 yang terinstal, berikut ini merupakan spesifikasi VMware yang kami gunakan :



**Gambar 2.4.1** Spesifikasi VM yang digunakan

Sebelum mengerjakan tugas pertama kami akan melakukan instalasi LAMPP yang terdiri dari Apache versi 2.4.29, MySQL versi 5.7, dan PHP versi 7.2.24. berikut ini langkah-langkah instalasi LAMPP :

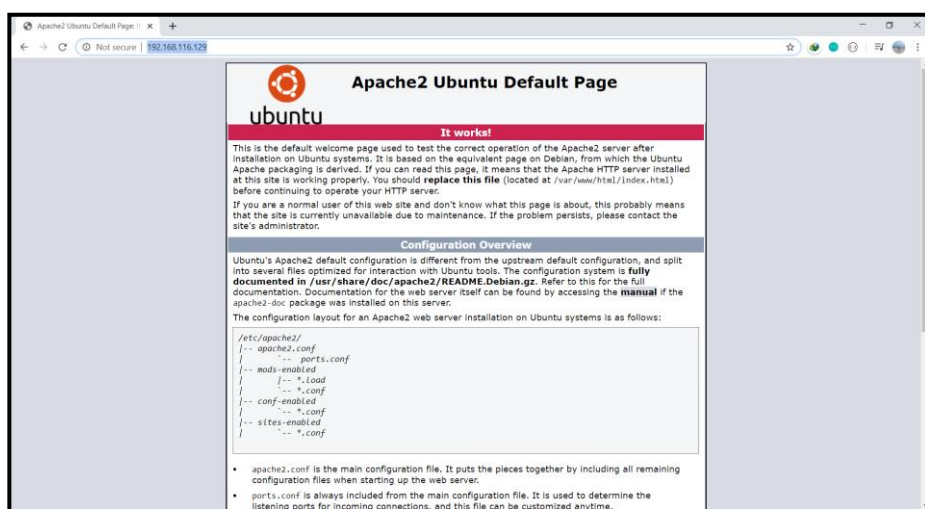
1. Setelah aktifkan VMware dan masuk ke PuTTY menggunakan perintah `sudo apt install apache2`.



```
fajar@sr: ~
fajar@sr:~$ sudo apt install apache2
```

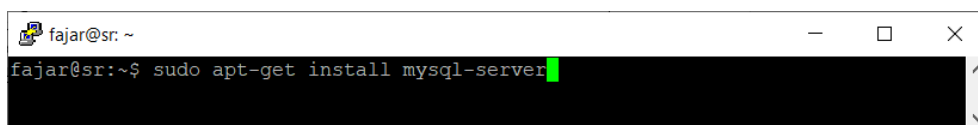
**Gambar 2.4.2** Perintah *install* Apache2

Kemudian setelah selesai dapat cek dengan cara mengetikkan IP ke dalam *web browser*.



**Gambar 2.4.3** Hasil *install* Apache2

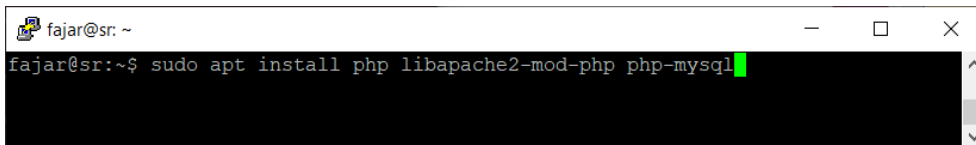
2. Kemudian kita lakukan instalasi MySQL menggunakan perintah `sudo apt install mysql-server`.



```
fajar@sr: ~
fajar@sr:~$ sudo apt-get install mysql-server
```

**Gambar 2.4.4** Perintah *install* MySQL

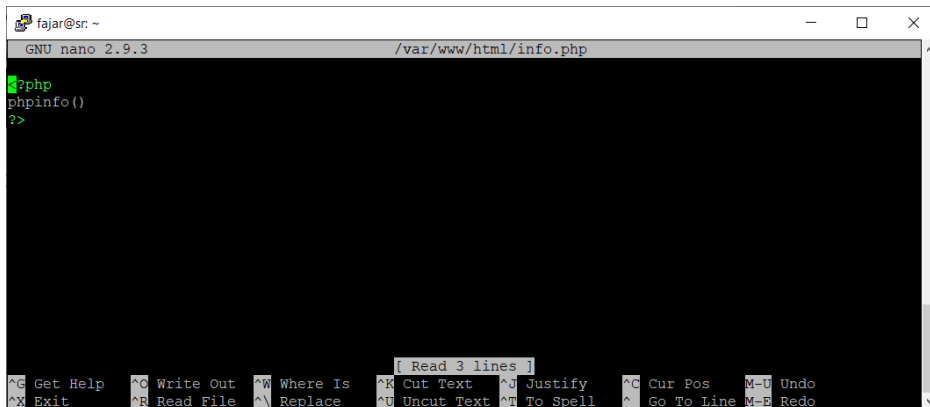
3. Langkah selanjutnya lakukan instalasi PHP dengan perintah `sudo apt install php libapache2-mod-php php-mysql`.



```
fajar@sr: ~
fajar@sr:~$ sudo apt install php libapache2-mod-php php-mysql
```

Gambar 2.4.5 Perintah *install* PHP

Kemudian lakukan perintah `sudo nano /var/www/html/info.php` dan lakukan perubahan isi *file* `info.php` dan lakukan perintah seperti gambar dibawah.

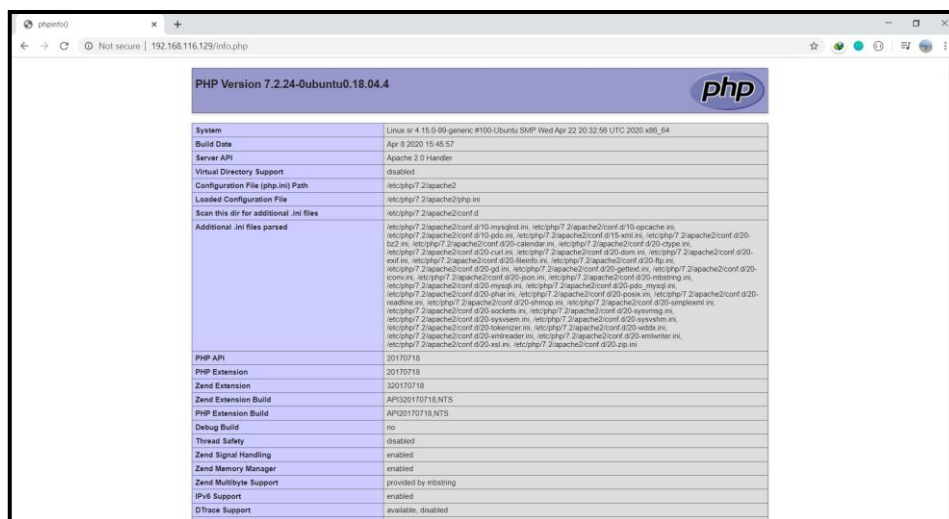


```
fajar@sr: ~
GNU nano 2.9.3 /var/www/html/info.php

<?php
phpinfo()
?>
```

Gambar 2.4.6 Perintah cek versi PHP

Lakukan pengecekan dengan cara `IP_ADDRESS/php.info`.

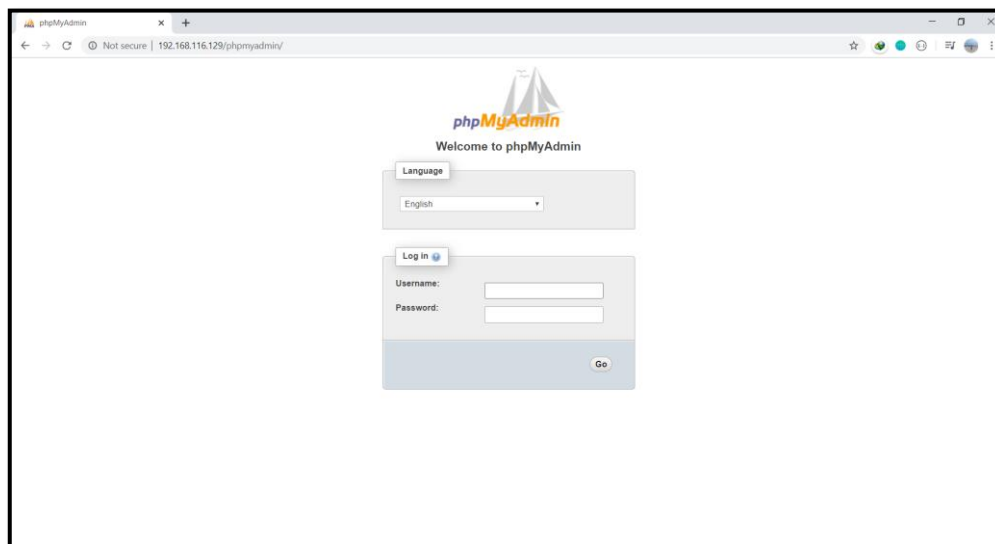
Gambar 2.4.7 Hasil *install* PHP

- Langkah terakhir lakukan perintah `sudo apt install phpmyadmin`.



**Gambar 2.4.8** Perintah *install* phpmyadmin

Dan dapat melihat hasilnya dengan perintah `IP_ADDRESS/phpmyadmin` pada *web browser*.

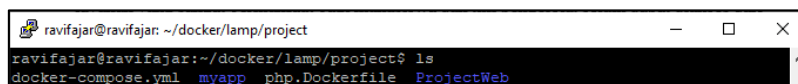


**Gambar 2.4.9** Perintah *install* phpmyadmin

Pada tugas kedua yaitu implementasi docker untuk tugas pertama. Kami menggunakan docker-compose untuk menyimpan dan menghubungkan container yang akan digunakan pada docker tersebut dan kami juga menggunakan Dockerfile yang nantinya akan dieksekusi di docker-compose.

Pada Tugas 2 ini, ada beberapa langkah yang harus dilakukan yaitu:

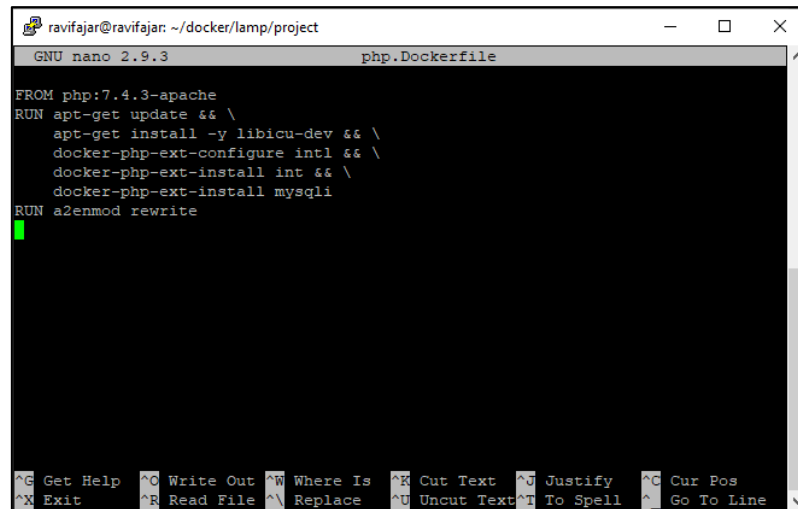
1. Install Docker di Ubuntu LAMPP, langkah instalasinya dapat kita lihat di <https://docs.docker.com/engine/install/ubuntu/>
2. Kemudian buat sebuah folder yang akan menampung aplikasi web kita ,Dockerfile dan *file* docker-compose.yml. Disini kami membuat direktori `~/docker/lamp/project` seperti berikut:



**Gambar 2.4.10** Folder pada docker

3. Setelah itu kita masukkan folder project kita, disini folder project kami adalah folder dengan nama ProjectWeb.

4. Setelah itu kita membuat Dockerfile menggunakan perintah `nano php.Dockerfile` dan mengisi file tersebut seperti berikut:



```

ravifajar@ravifajar: ~/docker/lamp/project
GNU nano 2.9.3      php.Dockerfile

FROM php:7.4.3-apache
RUN apt-get update && \
    apt-get install -y libicu-dev && \
    docker-php-ext-configure intl && \
    docker-php-ext-install intl && \
    docker-php-ext-install mysqli
RUN a2enmod rewrite

```

**Gambar 2.4.11** *Create Image PHP*

Pada file tersebut dapat kita lihat ada beberapa perintah, seperti:

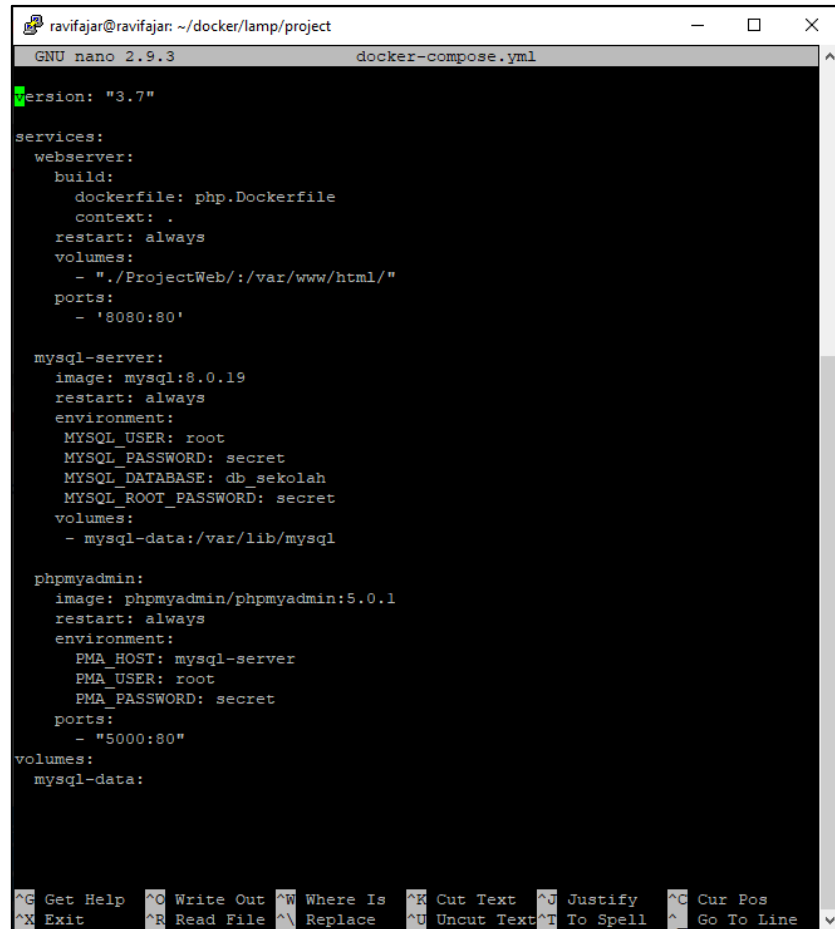
- a. Perintah FROM

Perintah ini digunakan untuk mengambil *image* yang sudah teregistry di docker yang nantinya akan kita jadikan container.

- b. Perintah RUN

Perintah ini digunakan untuk melakukan eksekusi *command* yang akan di digunakan pada *image* yang sebelumnya di *pull*

5. Setelah itu kita buat file `docker-compose.yml` dengan menggunakan perintah `nano docker-compose.yml` dan mengisi file tersebut seperti berikut ini:



```

ravifajar@ravifajar: ~/docker/lamp/project
GNU nano 2.9.3      docker-compose.yml

version: "3.7"

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

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

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

```

**Gambar 2.4.12** Create webserver, mysql-server dan phpMyAdmin

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

Pada *service* **webserver**, akan melakukan *custom-built* Docker *image* seperti yang sudah kita buat di `php.Dockerfile`. Kemudian path project yg ada pada saat ini kita arahkan ke `/var/www/html/`. Setelah itu kita atur *port*-nya yaitu akan tersambung pada *port* `8080:80`.

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

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

*hostname*, *username* dan *password* yang mana nantinya menyambungkan **phpmyadmin** dengan database server MySQL yaitu **mysql-server**. Kemudian kita atur *port*-nya yaitu menggunakan *port* 5000:80.

6. Setelah itu pada aplikasi yang kita gunakan, kita atur sambungan database-nya dan disesuaikan seperti yang ada di docker-compose tadi. Mengubah isi file nya dapat kita lakukan melalui perintah nano seperti diatas tadi, tetapi disini kami menggunakan winSCP agar lebih fleksibel. Aplikasi kami menggunakan framework CodeIgniter, oleh karena itu kami harus mengatur file config.php dan database.php seperti berikut ini:

- a. Menyesuaikan isi database.php

```
'hostname' => 'mysql-server',
'username' => 'root',
'password' => 'secret',
'database' => 'db_sekolah',
```

**Source Code 2.4.1** konfigurasi *database*

- b. Menyesuaikan base\_url pada config.php

```
$config['base_url'] = '/';
```

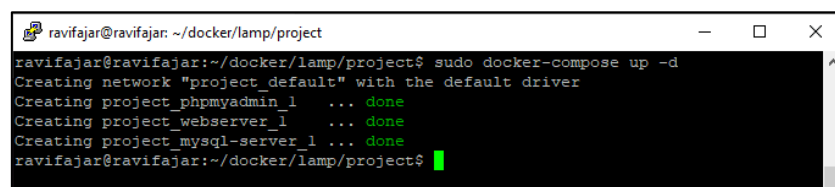
**Source Code 2.4.2** konfigurasi *URL*

- c. Menyesuaikan session pada config.php

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

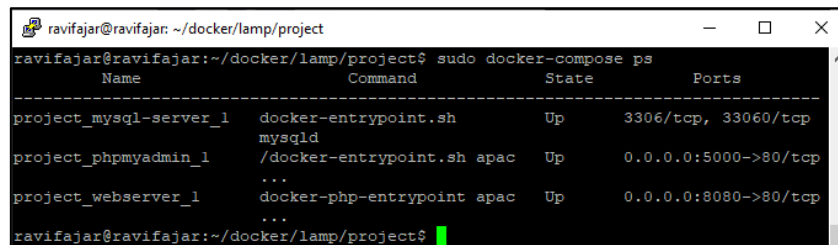
**Source Code 2.4.3** konfigurasi *session*

7. Selanjutnya kita jalankan semua service yang ada di docker-compose tadi dengan mengetikkan perintah `sudo docker-compose up -d`. Hal yang perlu diingat, pada saat akan menjalankan perintah tersebut, kita harus berada di path tempat kita menyimpan file docker-compose tersebut. Perhatikan gambar berikut:



**Gambar 2.4.13** menjalankan *service file* di docker

8. Kemudian kita check apakah container yang kita jalankan sudah dapat berjalan semestinya dengan menggunakan perintah `sudo docker-compose ps` seperti gambar berikut:



```

ravifajar@ravifajar: ~/docker/lamp/project
ravifajar@ravifajar:~/docker/lamp/project$ sudo docker-compose ps

```

| Name                   | Command                        | State | Ports                |
|------------------------|--------------------------------|-------|----------------------|
| project_mysql-server_1 | docker-entrypoint.sh<br>mysqld | Up    | 3306/tcp, 33060/tcp  |
| project_phpmyadmin_1   | /docker-entrypoint.sh apac     | Up    | 0.0.0.0:5000->80/tcp |
| project_webserver_1    | docker-php-entrypoint apac     | Up    | 0.0.0.0:8080->80/tcp |

```

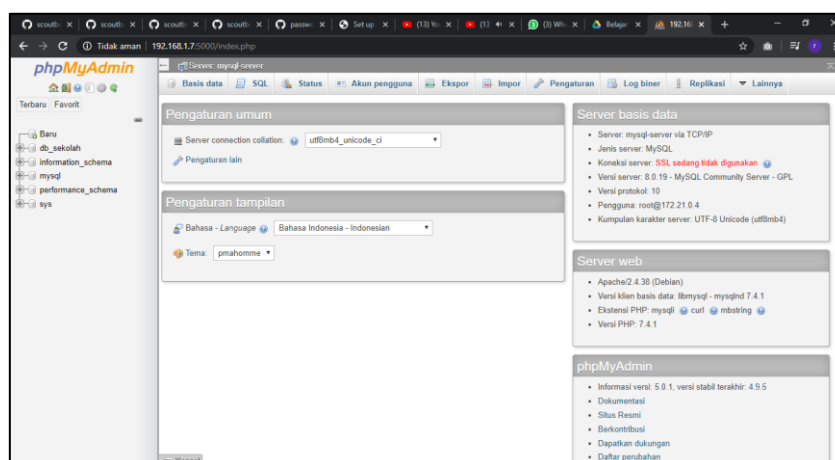
ravifajar@ravifajar:~/docker/lamp/project$

```

**Gambar 2.4.14** menjalankan *service file* di docker

Dapat kita lihat, containernya sudah dapat berjalan dengan semestinya.

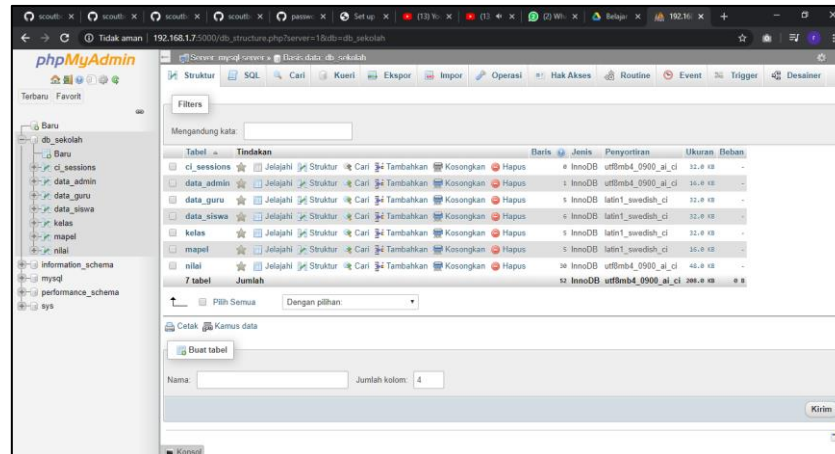
9. Kemudian kita coba check phpmyadmin dengan mengetikkan url [http://IP\\_HOST:5000](http://IP_HOST:5000) . Karena kita menggunakan port 5000 sebelumnya untuk phpmyadmin, maka ketika ingin mengaksesnya kita juga harus mengetikkan portnya. url yang kami akses adalah seperti ini <http://192.168.1.7:5000/> ,maka akan kita dapatkan output seperti berikut ini:



**Gambar 2.4.15** menjalankan *phpMyAdmin*

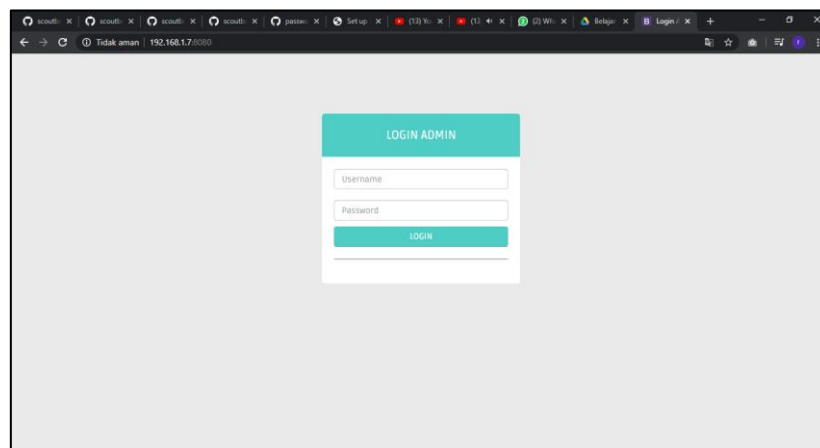
Jangan lupa untuk mengimport database yang akan kita gunakan untuk aplikasi kita dan samakan nama database nya dengan yang sudah diatur di docker-compose dan database.php di aplikasi kita. Disini kami menggunakan database db\_sekolah seperti berikut ini:





**Gambar 2.4.16** database proyek akhir

10. Kemudian kita coba akses aplikasi yang kita build sebelumnya menggunakan url [http://IP\\_SERVER:8080](http://IP_SERVER:8080). Disini kami mengakses menggunakan url <http://192.168.1.7:8080/> maka akan kita dapatkan output sebagai berikut:



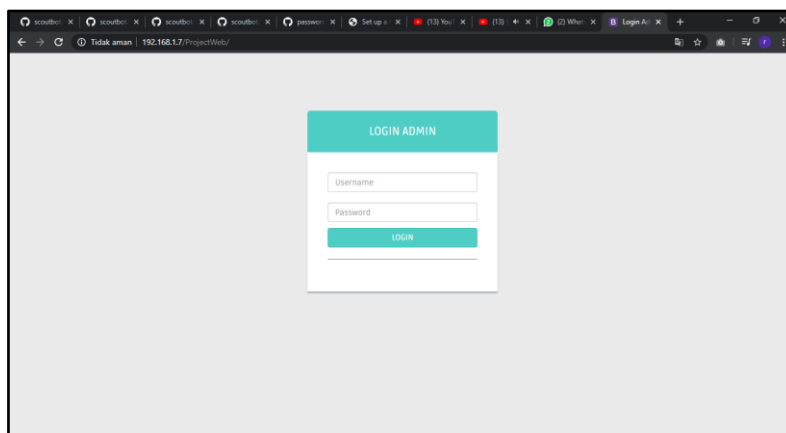
**Gambar 2.4.17** Tampilan web proyek akhir

11. Jika sudah tidak ada bagian yang error, berarti kita sudah berhasil menjalankan container docker kita.

## 2.5 Hasil Implementasi

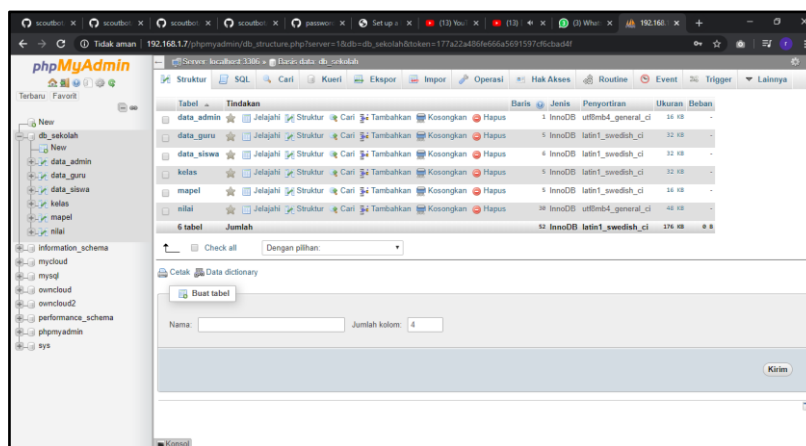
Pada Tugas pertama yaitu menggunakan layanan LAMPP untuk diintegrasikan dengan tugas kuliah pada VM Ubuntu Server dengan judul yang kami gunakan adalah “Sistem Informasi Sekolah Dasar Menggunakan Ubuntu LAMPP” dan tugas satu dari kelompok kami sudah dapat diakses melalui jaringan lokal dengan mengetikkan

[http://IP\\_Server>NamaFolderTugas](http://IP_Server>NamaFolderTugas) dan untuk tugasnya tersebut kami menggunakan url sebagai berikut: <http://192.168.1.7/ProjectWeb/>.



**Gambar 2.5.1** Output yang ditampilkan ketika membuka URL

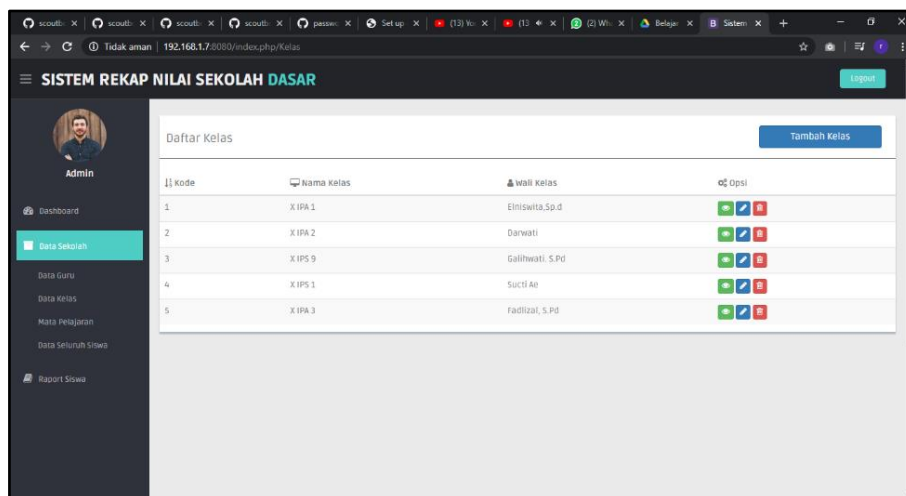
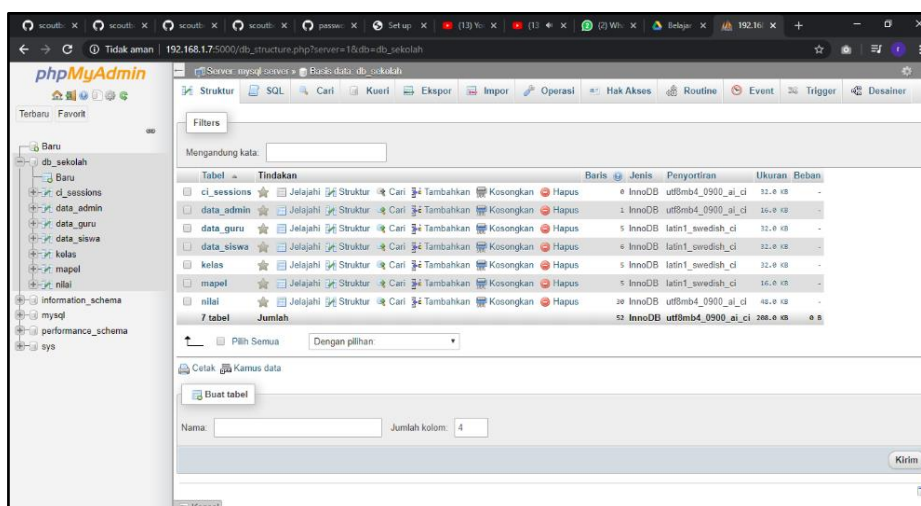
<http://192.168.1.7/ProjectWeb/>



**Gambar 2.5.2** Database yang digunakan untuk tugas pertama

Pada tugas kedua yaitu implementasi docker yang digunakan untuk aplikasi dari tugas pertama, sebelumnya kami memiliki masalah pada saat pengaturan database mysql-server dan phpmyadmin serta pada database yang ada di aplikasi. Selain itu ada masalah juga pada bagian session yang ada di confignya codeigniter, serta path css yang kurang tepat sebelumnya.

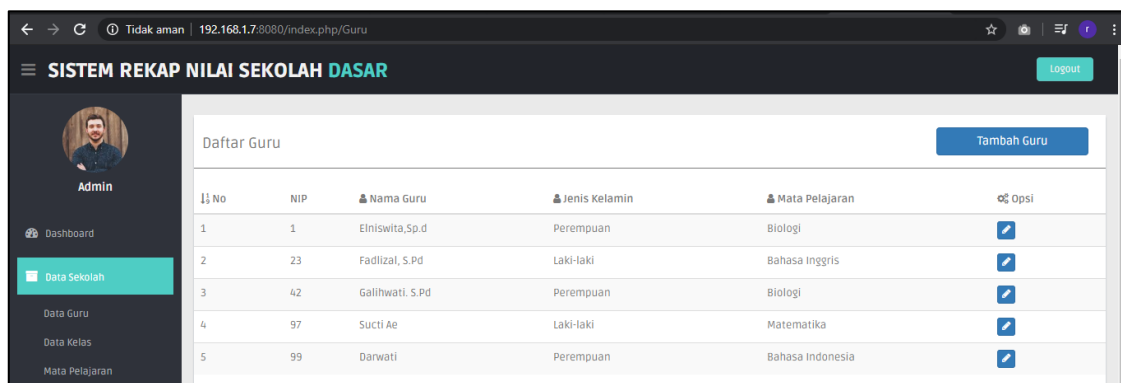
Masalah-masalah diatas sudah dapat kami atasi seperti yang kami tunjukkan pada langkah-langkah menjalankan docker pada pembahasan sebelumnya. Hasil akhir dari implementasi docker tersebut adalah sebagai berikut:

Gambar 2.5.3 Tampilan *Dashboard*Gambar 2.5.4 *Database* proyek akhir

## 2.6 Pengujian Singkat

Masalah pertama yang dihadapi terkait dalam pendataan Guru dan Siswa. Di Sekolah biasanya pendataan untuk guru dan siswa dilakukan secara manual yang mana sebelumnya data individu yang sudah di isi melalui kertas form pendaftaran akan dimasukkan kedalam file excel dan masih perlu disusun, oleh karena itu dengan menggunakan sistem ini kita hanya perlu memasukkan data dari kertas form yang sudah didapat tersebut kedalam form yang sudah disediakan pada sistem ini yang kemudian nantinya akan menyusun seluruh data

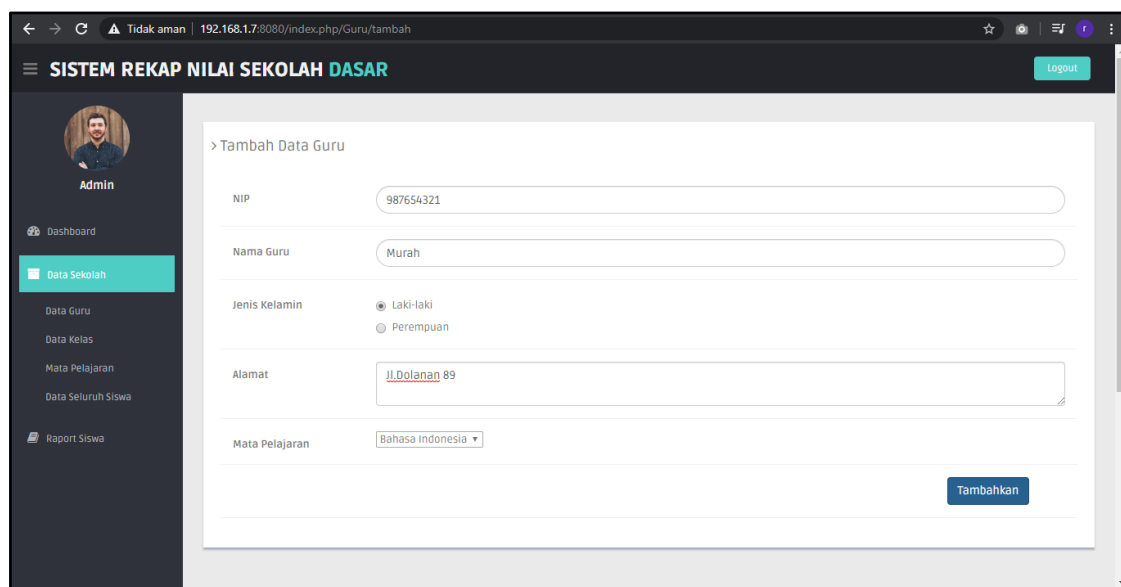
yang ada secara otomatis. Seperti yang dapat kita lihat pada **Gambar 2.6.1** s/d **Gambar 2.6.3** berikut:



| No | NIP | Nama Guru        | Jenis Kelamin | Mata Pelajaran   | Opsi |
|----|-----|------------------|---------------|------------------|------|
| 1  | 1   | Elniswita, Sp.d  | Perempuan     | Biologi          |      |
| 2  | 23  | Fadlilal, S.Pd   | Laki-laki     | Bahasa Inggris   |      |
| 3  | 42  | Gallihwati, S.Pd | Perempuan     | Biologi          |      |
| 4  | 97  | Sucti Ae         | Laki-laki     | Matematika       |      |
| 5  | 99  | Darwati          | Perempuan     | Bahasa Indonesia |      |

**Gambar 2.6.1** Daftar Guru

Pada **Gambar 2.6.1** menampilkan daftar guru yang sebelumnya sudah di data oleh admin.



> Tambah Data Guru

NIP: 987654321

Nama Guru: Murah

Jenis Kelamin: ☒ Laki-laki ☐ Perempuan

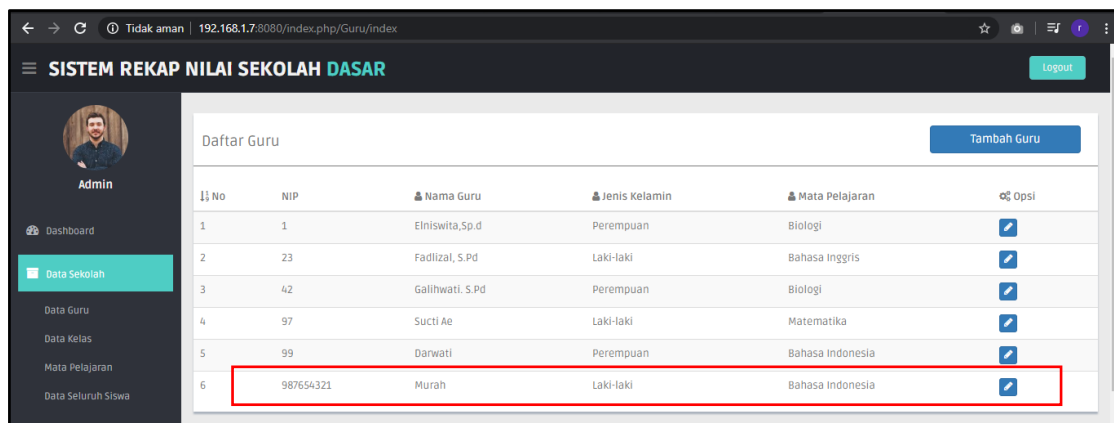
Alamat: Jl Dolanan 89

Mata Pelajaran: Bahasa Indonesia

Tambahkan

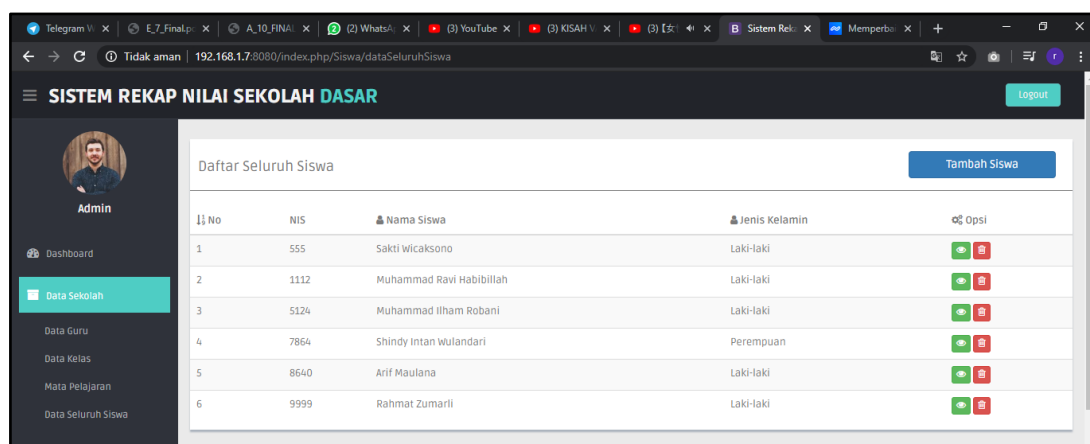
**Gambar 2.6.2** Menambahkan data guru baru

Pada **Gambar 2.6.2** menampilkan form yang perlu di isi untuk memasukkan data guru baru kedalam sistem.

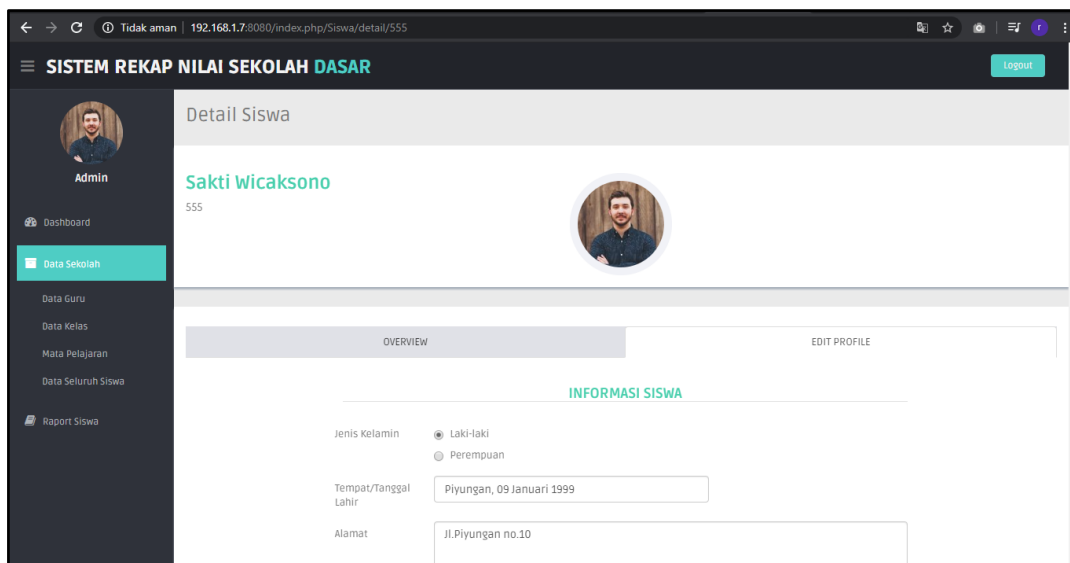


**Gambar 2.6.3** Berhasil menambah data guru baru

Dari yang sudah kita lihat pada **Gambar 2.6.3**, data guru yang sebelumnya sudah di masukkan otomatis disusun, begitu pula dengan data siswa, mata pelajaran dan kelas. Semua data tersebut akan dapat otomatis di sinkronisasi, yang perlu dilakukan admin hanya menginput data-data tersebut dan tidak perlu memikirkan susunannya akan bagaimana nantinya. Berikut contoh tampilan dari data siswa pada **Gambar 2.6.4** dan **Gambar 2.6.5** :

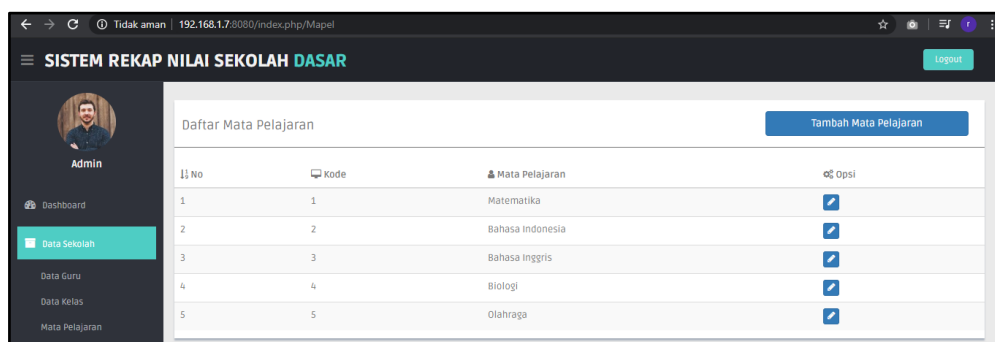


**Gambar 2.6.4** Daftar seluruh siswa



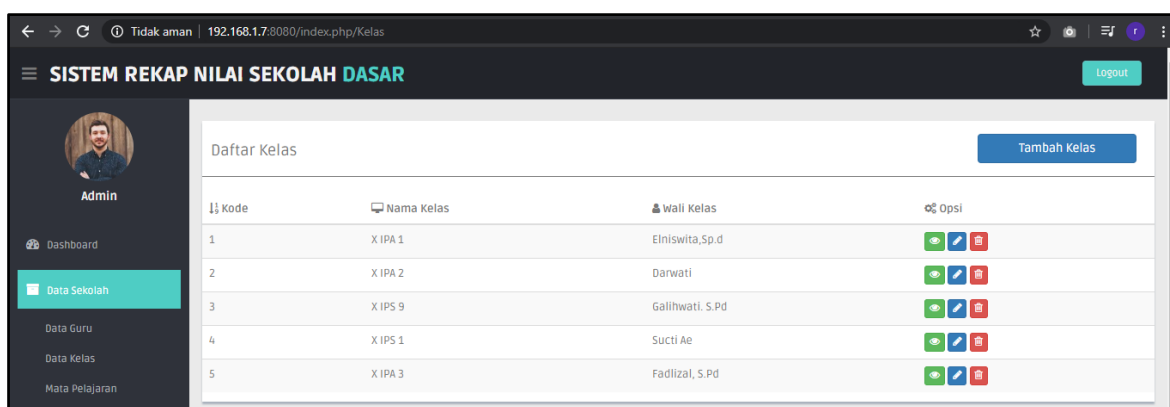
**Gambar 2.6.5** Profil Siswa

Berikut tampilan data mata pelajaran pada **Gambar 2.6.6**:



**Gambar 2.6.6** Daftar mata pelajaran

Berikut tampilan data kelas pada **Gambar 2.6.7** dan **2.6.8**:



**Gambar 2.6.7** Daftar Kelas yang ada

| No Absen | NIS  | Nama Siswa            | Jenis Kelamin | Opsi  |
|----------|------|-----------------------|---------------|---|
| 1        | 555  | Sakti Wicaksono       | Laki-laki     | <input type="checkbox"/> <input type="checkbox"/> |
| 1        | 5124 | Muhammad Itham Robani | Laki-laki     | <input type="checkbox"/> <input type="checkbox"/> |
| 1        | 9999 | Rahmat Zumarli        | Laki-laki     | <input type="checkbox"/> <input type="checkbox"/> |

**Gambar 2.6.8** Daftar Siswa dalam suatu kelas

Masalah kedua yaitu terkait mengenai nilai raport siswa. Biasanya pembuatan raport menggunakan word atau excel dan juga di print. Sehingga akan menghabiskan banyak kertas dan lumayan merepotkan siswa dalam menyimpannya. Oleh karena itu pada sistem ini, nantinya diharapkan siswa dan orang tuanya dapat melihat hasil raport secara online, tetapi pada program ini masih belum menerapkan pembuatan akun khusus untuk para murid, jadi untuk sementara semua nilai raport dapat dilihat melalui akun admin dan dengan menggunakan sistem ini, semua nilai bisa langsung direkap dan data siswa langsung ditampilkan. Misalnya saja kita lakukan langkah-langkah sebagai berikut dalam pembuatan nilai raport:

> Tambah Data Siswa

NIS: 123170039

Nama Siswa: Murah

Jenis Kelamin: ☒ Laki-laki ☐ Perempuan

Alamat: Jl. Murah 123

Tempat/Tanggal lahir: 15 April 1998

Kelas: X IPA 2

Tambahkan

**Gambar 2.6.9** Mengisi data siswa baru

| No | NIS       | Nama Siswa               | Jenis Kelamin | Opsi |
|----|-----------|--------------------------|---------------|------|
| 1  | 555       | Sakti Wicaksono          | Laki-laki     |      |
| 2  | 1112      | Muhammad Ravi Habibillah | Laki-laki     |      |
| 3  | 5124      | Muhammad Ilham Robani    | Laki-laki     |      |
| 4  | 7864      | Shindy Intan Wulandari   | Perempuan     |      |
| 5  | 8640      | Arif Maulana             | Laki-laki     |      |
| 6  | 9999      | Rahmat Zumarli           | Laki-laki     |      |
| 7  | 123170039 | Murah                    | Laki-laki     |      |

**Gambar 2.6.10** Data siswa baru

Pada **Gambar 2.6.9** dan **Gambar 2.6.10**, kita menambahkan data siswa baru terlebih dahulu yang mana nanti masih belum memiliki nilai raport.

| No | NIS       | Nama Siswa               | Jenis Kelamin | Opsi |
|----|-----------|--------------------------|---------------|------|
| 1  | 555       | Sakti Wicaksono          | Laki-laki     |      |
| 2  | 1112      | Muhammad Ravi Habibillah | Laki-laki     |      |
| 3  | 5124      | Muhammad Ilham Robani    | Laki-laki     |      |
| 4  | 7864      | Shindy Intan Wulandari   | Perempuan     |      |
| 5  | 8640      | Arif Maulana             | Laki-laki     |      |
| 6  | 9999      | Rahmat Zumarli           | Laki-laki     |      |
| 7  | 123170039 | Murah                    | Laki-laki     |      |

**Gambar 2.6.11** Daftar seluruh siswa dari menu raport

**Murah**  
NIS : 123170039  
Jenis Kelamin : Laki-laki  
Alamat : Jl. Murah 123

X IPA 2

| NO        | MATA PELAJARAN | NILAI |
|-----------|----------------|-------|
| Total     |                | 0     |
| Rata-rata |                | 0     |

**Gambar 2.6.12** Raport siswa baru



Pada **Gambar 2.6.12** dapat kita lihat bahwa siswa tersebut masih belum memiliki nilai apapun pada raportnya, sehingga kita harus memasukkan nilai terlebih dahulu.

**SISTEM REKAP NILAI SEKOLAH DASAR**

Admin

Dashboard

Data Sekolah

Raport Siswa

Matematika 87

Bahasa Indonesia 85

Bahasa Inggris 88

Biologi 82

Olahraga 90

Tambahkan

Copyrights Dashio. All Rights Reserved  
Created with Dashio template by TemplateMag

**Gambar 2.6.13** Menambahkan nilai ke raport siswa baru

**SISTEM REKAP NILAI SEKOLAH DASAR**

Admin

Dashboard

Data Sekolah

Raport Siswa

Murah

NIS : 123170039

Jenis Kelamin : Laki-laki

Alamat : Jl. Murah 123

X IPA 2

| NO    | MATA PELAJARAN   | NILAI |
|-------|------------------|-------|
| 1     | Matematika       | 87    |
| 2     | Bahasa Indonesia | 85    |
| 3     | Bahasa Inggris   | 88    |
| 4     | Biologi          | 82    |
| 5     | Olahraga         | 90    |
| Total |                  | 432   |

Rata-rata 86.4

**Gambar 2.6.14** Raport siswa

Pada **Gambar 2.6.14** dapat kita lihat nilai yang sebelumnya sudah dimasukkan langsung dikalkulasikan dan ditampilkan.

Hasil dari apa yang telah dilakukan admin tersebut adalah pendataan dan pembuatan raport mungkin hampir mirip seperti cara konvensional namun dari segi efisiensi waktu dan tenaga jauh lebih cepat dan efisien. Hal ini dapat terlihat bahwa waktu yang dibutuhkan hanya lebih kurang 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 siswa dan admin sekolah, selain dapat diakses dari mana saja, biaya alat tulis dan pembuatan raport menjadi berkurang.

### 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<br>. | Jenis Tugas                                 | Waktu Pengerjaan |   |   |   |       |   |   |   |     |   |   |  |
|---------|---|------------------|---|---|---|-------|---|---|---|-----|---|---|--|
|         |   | Maret            |   |   |   | April |   |   |   | Mei |   |   |  |
|         |   | 1                | 2 | 3 | 4 | 1     | 2 | 3 | 4 | 1   | 2 | 3 |  |
| 1.      | Analisa Persoalan                           |                  |   |   |   |       |   |   |   |     |   |   |  |
| 2.      | Pembagian Tugas                             |                  |   |   |   |       |   |   |   |     |   |   |  |
| 3.      | Penginstalan dan Konfigurasi Ubuntu LAMPP   |                  |   |   |   |       |   |   |   |     |   |   |  |
| 4.      | Pemasukkan Project ke Ubuntu LAMPP          |                  |   |   |   |       |   |   |   |     |   |   |  |
| 5.      | Pembuatan Dockerfile dan pemasukkan project |                  |   |   |   |       |   |   |   |     |   |   |  |
| 6.      | Laporan Bab 1                               |                  |   |   |   |       |   |   |   |     |   |   |  |
| 7.      | Laporan Progress 2                          |                  |   |   |   |       |   |   |   |     |   |   |  |
| 8.      | Pengerjaan Docker Lanjutan dan Testing      |                  |   |   |   |       |   |   |   |     |   |   |  |
| 9.      | 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     | Fajar            |
| 2.  | Latar Belakang Masalah                     | Fajar            |
| 3.  | Tujuan Proyek Akhir                        | Fajar            |
| 4.  | Manfaat Proyek Akhir                       | Fajar            |
| 5.  | Tahap Penyelesaian Proyek Akhir            | Ravi             |
| 6.  | BAB II                                     | Fajar dan Ravi   |
| 7.  | BAB III                                    | Fajar dan Ravi   |
| 8.  | BAB IV                                     | Ravi             |
| 9.  | Instalasi LAMPP                            | Fajar            |
| 10. | Intalasi Docker dan konfigurasi dockerfile | Ravi             |
| 11. | Pengujian dockerfile pada sistem lain      | Ravi             |

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berdasarkan dua masalah yang ada, yaitu pendataan guru dan siswa dan pengolahan nilai raport yang dilakukan oleh admin yang sebelumnya menggunakan Word atau Excel, dilakukan pengujian secara acak menggunakan sistem rekap nilai online untuk menyelesaikan masalah tersebut. Hasil yang didapat adalah pertama pendataan guru-guru yang mengajar dan siswa yang mendaftarkan diri dapat direkap secara otomatis oleh sistem dan yang kedua pembuatan raport yang tidak mengeluarkan banyak biaya dan menghemat waktu.

Berdasarkan pembagian tugas proyek, hasil yang didapat adalah sistem dapat dikonfigurasi menggunakan Docker dan dapat diimplementasikan ke sistem lain sehingga sistem rekap nilai 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

- Wicaksono, A. (2015, November 2). Perkembangan Komputasi Awan (Cloud Computing) di Indonesia dari lintasarta blog, LLC:  
<https://blog.lintasarta.net/article/perkembangan-komputasi-awan-cloud-computing-di-indonesia/>
- 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>
- 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/)
- 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-andmysql-with-docker-compose>
- Now, P. Z. (2019, September 5). *Docker untuk Pemula*. Retrieved from Youtube: <https://www.youtube.com/playlist?list=PL-CtdCApEFH-A7jBmdertzbeACuQWvQao>

## **LAMPIRAN**