

# 577

## PRAKTIKUM TEKNOLOGI CLOUD COMPUTING LAPORAN PROYEK AKHIR



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



#### 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

# F9J=9K98

6 mK U m i '5 'j'Bi [ fc\ c 'U h\$\*. ( , . ' \*ž\$) #&\$&\$

## HALAMAN PENGESAHAN

### SISTEM INFORMASI NILAI SEKOLAH DASAR MENGGUNAKAN UBUNTU LAMP 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 kehadirat 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 Docker **Filenya.**

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

HALAMAN PENGESAHAN .....	i
KATA PENGANTAR .....	ii
DAFTAR ISI .....	iii
BAB I PENDAHULUAN .....	1
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
BAB II ISI DAN PEMBAHASAN .....	3
2.1 Komponen yang Digunakan .....	3
2.2 Rancangan Arsitektur <i>Cloud Computing</i> .....	4
2.3 Parameter dan Konfigurasi .....	4
2.4 Tahap Implementasi .....	6
2.5 Hasil Implementasi .....	12
2.6 Pengujian Singkat .....	14
BAB III JADWAL Pengerjaan dan Pembagian Tugas.....	16
3.1 Agenda Pengerjaan.....	16
3.2 Keterangan Pembagian Tugas .....	16
BAB IV KESIMPULAN DAN SARAN.....	17
4.1 Kesimpulan.....	17
4.2 Saran .....	17
DAFTAR PUSTAKA .....	18
LAMPIRAN .....	19

(gunakan tombol update table (klik kanan daftar isi) lalu format ulang dengan format TNR 12, spacing row single, spacing before after 0 dan indentasi kanan 0, indentasi kiri biarkan)

# 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.

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.

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 ~~Judul~~ ~~berjudul~~ Sistem Informasi Nilai Sekolah Dasar menggunakan Ubuntu LAMP dan Proses Pembuatan Docker Filenya. Menggunakan layanan Linux dengan versi Ubuntu 18.04, LAMP, dan FreeNas sebagai manajemen data. Dalam penerapannya *project* ini akan menghasilkan suatu *website* yang akan menampilkan sistem informasi nilai sekolah dasar, dari *website* tersebut para orangtua akan dengan mudah memantau perkembangan anaknya dan sebagai tenaga pengajar sebagai evaluasi hasil mengajar dari sekolah dasar tersebut.

### 1.3 Manfaat Proyek Akhir

Dengan dibuatnya Sistem Informasi tersebut, maka dapat memudahkan para orang tua dalam melihat hasil pembelajaran anaknya dan juga dengan adanya sistem tersebut, maka pihak sekolah tidak perlu mengundang orang tua murid untuk datang ke sekolah untuk mengambil raport anaknya sehingga orang tua dapat mengakses Sistem Informasi tersebut dimanapun dan kapanpun walaupun mereka sedang bekerja.

### 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 LAMPP
4. Menentukan konfigurasi Dockerfile dan docker-compose yang dibutuhkan untuk proses menjalankan service LAMPP 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 berbasiskan 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 pengoprasiaannya 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 :

Dari ilustrasi diatas digunakan *hostspot* 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.

## 2.3 Parameter dan Konfigurasi

untuk dapat digunakan, maka VMware Workstation perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada **Tabel 2.1** berikut ini:

**Tabel 2.1** 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: 172.20.0.1 (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: ???	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 processor 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 : 172.17.0.1	IP number untuk akses docker

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

**Tabel 2.2** Spesifikasi ubuntu server pada VMware Workstation yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	LAMP	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	<i>Software</i> yang digunakan untuk mengelola server MySQL.

Contoh: Parameter yang digunakan untuk instalasi Apache dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ sudo apt install apache2
```

Keterangan:

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

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

Contoh: 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 tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- mysql-server : nama paket aplikasi untuk MySQL

#### **Modul 2.2** Parameter instalasi MySQL

Contoh: 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 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 2.3** Parameter instalasi PHP

Contoh: Parameter yang digunakan untuk instalasi Docker dapat dilihat pada penjelasan **Modul 2.4** berikut ini:

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

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
- dokcer-ce : nama paket aplikasi untuk docker

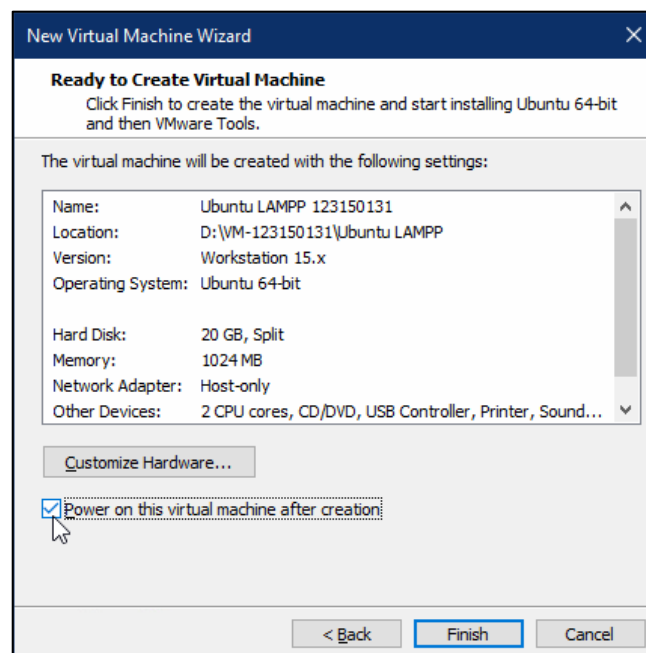
### Modul 2.3 Parameter instalasi Docker

## 2.4 Tahap Implementasi

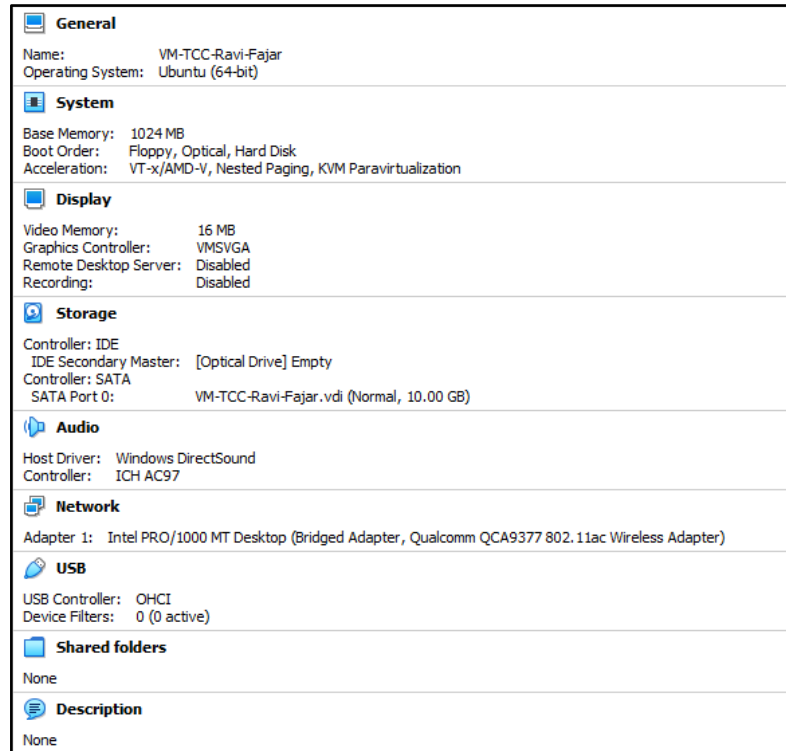
Pada tahap ini jelaskan mengenai implementasi Anda terhadap penyelesaian soal yang diberikan. Buatlah dalam format yang terstruktur sesuai flow pengerjaan Anda. Beberapa yang perlu Anda tuliskan pada bagian ini di antaranya:

- Tahapan mengenai langkah-langkah kerja
- Tampilan screenshoot hasil dari kerja
- Kodingan atau settingan dari implementasi tahap sebelumnya

Contoh: hasil dari implementasi **Tabel 2.1** mengenai konfigurasi untuk VMware Workstation dapat dilihat pada **Gambar 2.2** berikut ini:



**Gambar 2.2** Gambar disesuaikan agar tidak terlalu besar (tidak makan tempat), maksimal 25% dari halaman untuk satu gambar, pada satu lembar maksimal 3 gambar. Gambar jangan dijejerkan ke samping, beri keterangan satu per satu ke bawah



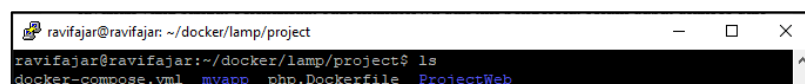
**Gambar 2.2.1** Spesifikasi VM yang digunakan

Pada tugas pertama .....

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.

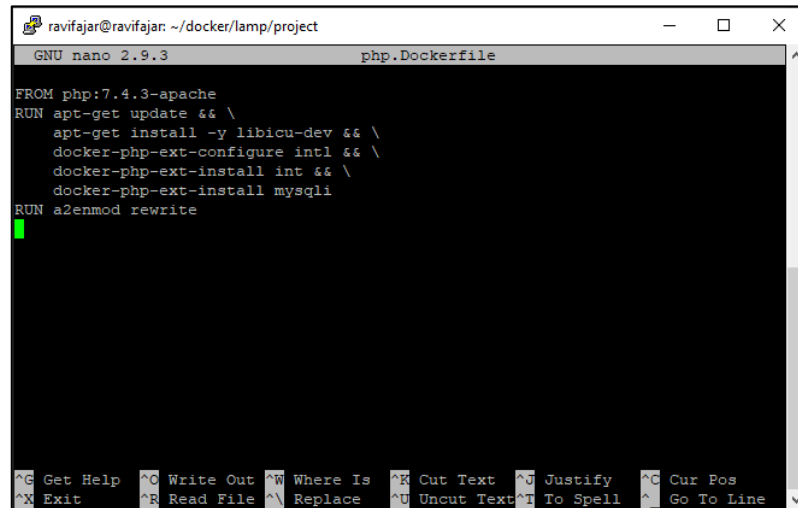
Pada Tugas 2 ini, ada beberapa langkah yang harus dilakukan yaitu:

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 2.2.2** 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:



```
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 2.2.3** Create Image PHP

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

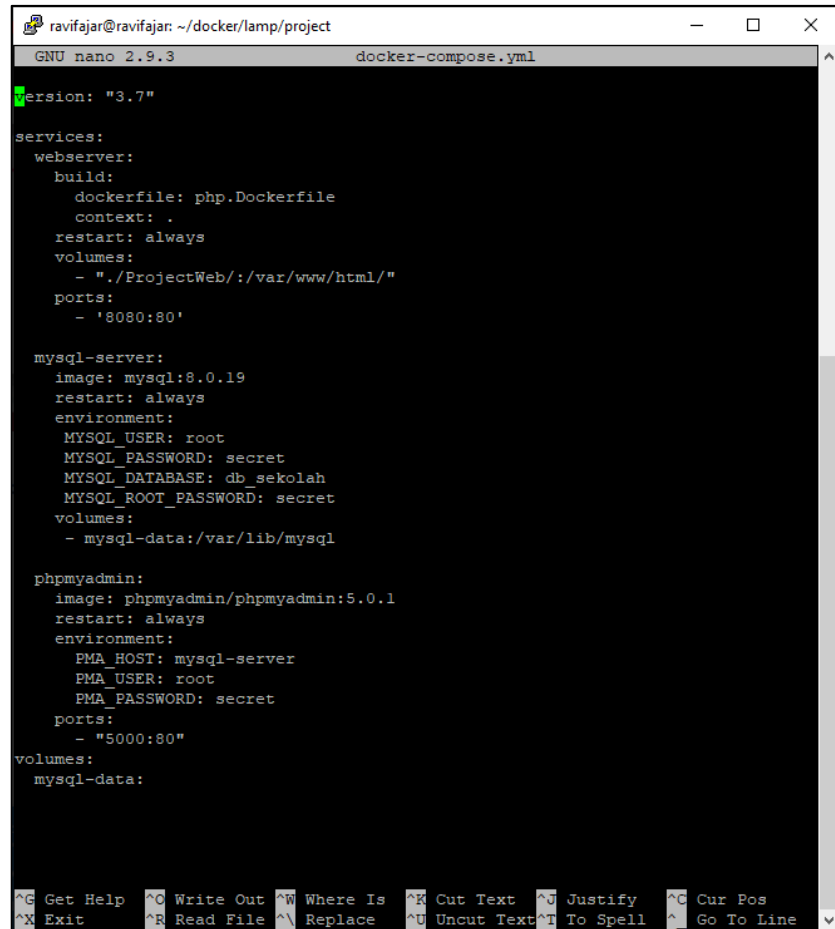
- a. Perintah FROM

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:



```
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 2.2.4** 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 2.2.1** konfigurasi *database*

- b. Menyesuaikan base\_url pada config.php

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

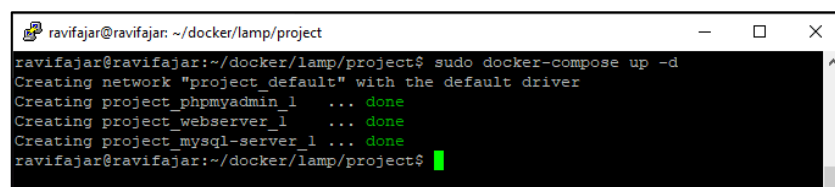
**Source Code 2.2.2** konfigurasi *URL*

- c. Menyesuaikan session pada config.php

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

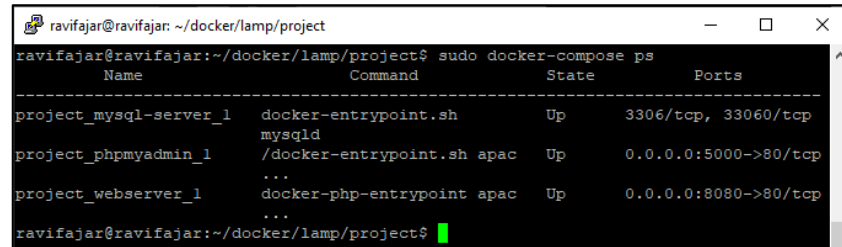
**Source Code 2.2.3** konfigurasi *session*

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 2.2.5** 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

```

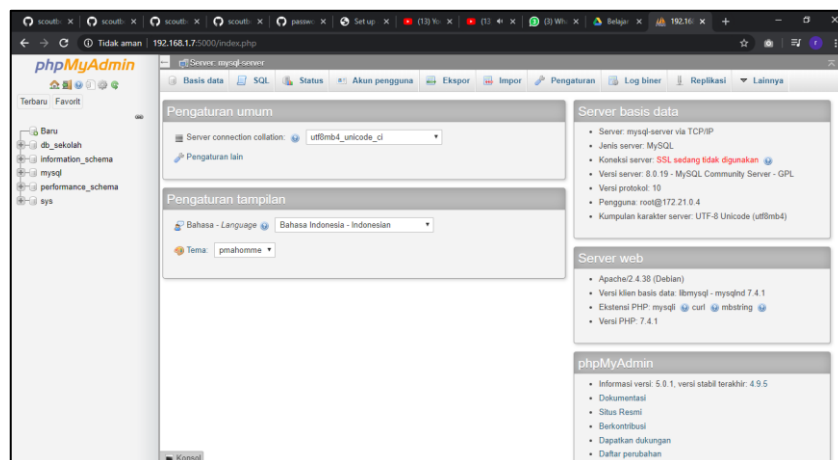
Name	Command	State	Ports
project_mysql-server_1	docker-entrypoint.sh mysqld	Up	3306/tcp, 33060/tcp
project_phpmyadmin_1	/docker-entrypoint.sh apac	Up	0.0.0.0:5000->80/tcp
project_webserver_1	docker-php-entrypoint apac	Up	0.0.0.0:8080->80/tcp

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

**Gambar 2.2.6** menjalankan *service file* di docker

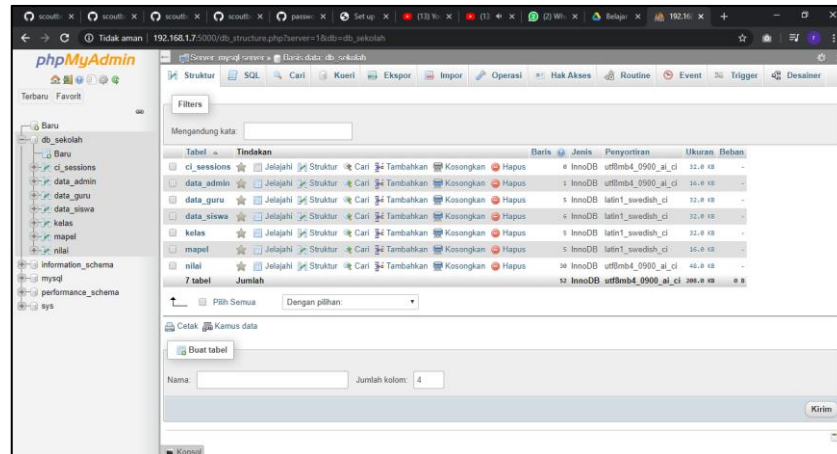
Dapat kita lihat, containernya sudah dapat berjalan dengan semestinya.

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:



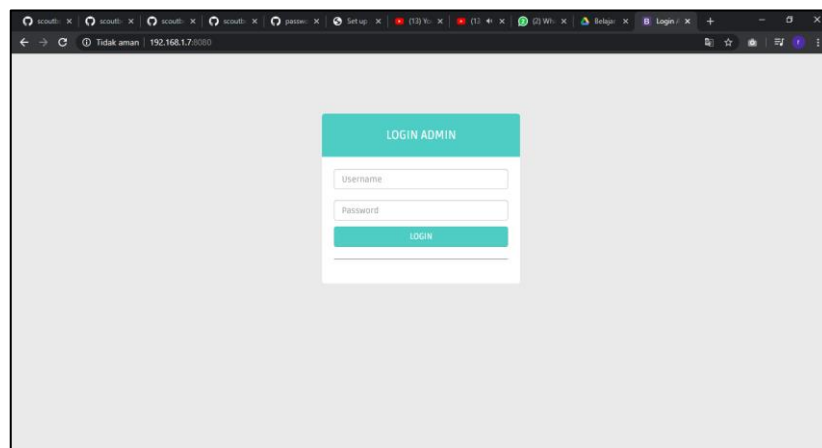
**Gambar 2.2.7** menjalankan *phpMyAdmin*

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 2.2.8** 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 2.2.9** Tampilan web proyek akhir

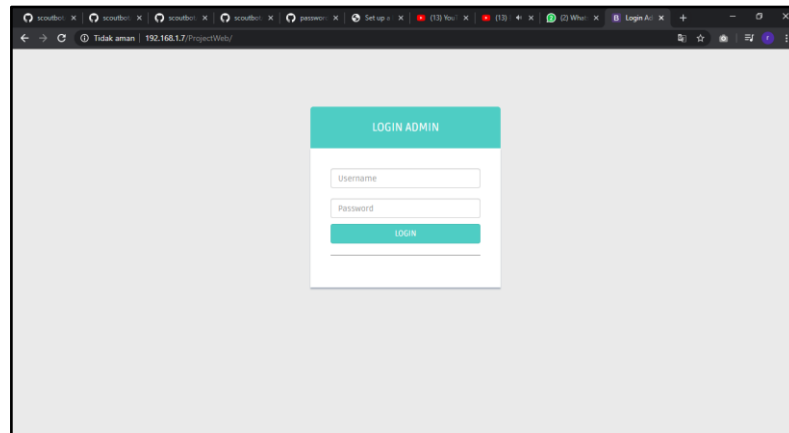
11. Jika sudah tidak ada bagian yang error, berarti kita sudah berhasil menjalankan container docker kita.

## 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

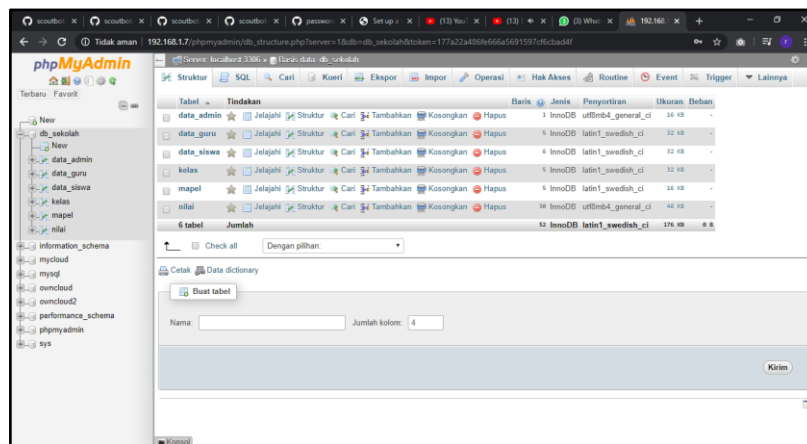


<http://IP Server>NamaFolderTugas> dan untuk tugasnya tersebut kami menggunakan url sebagai berikut: <http://192.168.1.7/ProjectWeb/>.



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

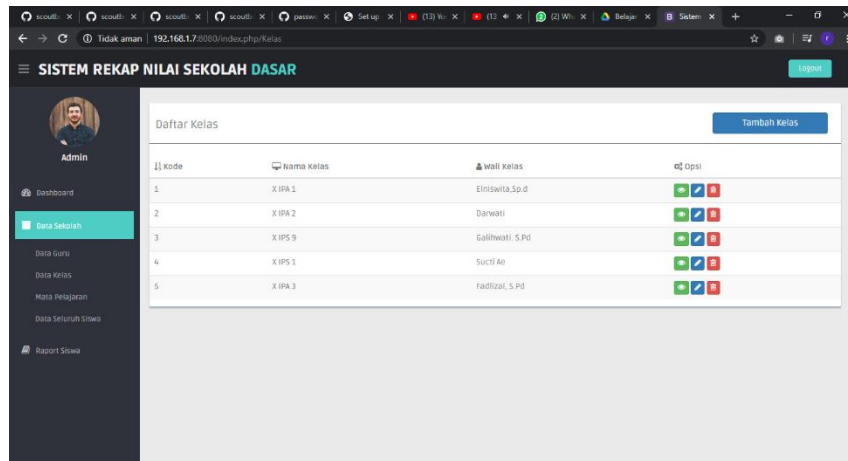
<http://192.168.1.7/ProjectWeb/>



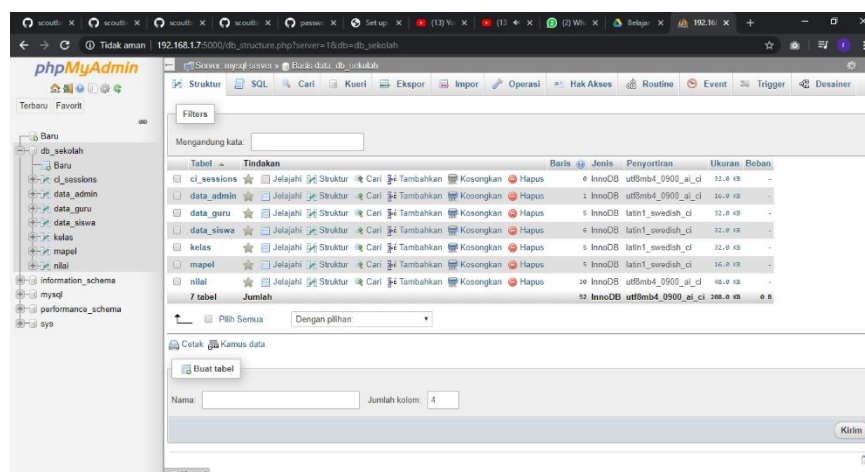
**Gambar 2.5.2** Database yang digunakan untuk tugas pertama

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

Pada tahap ini buktikan bahwa rancangan yang Anda buat telah dapat digunakan terhadap permasalahan acak. Buat sendiri minimal **dua** masalah yang dapat diselesaikan dengan rancangan Anda. Bagian ini akan dibuktikan secara langsung pada saat presentasi proyek akhir. Sehingga tuliskan pada bagian ini dengan format (**dijelaskan beberapa baris kalimat**): masalah dan hasil yang diharapkan, eksekusi penyelesaian masalah, hasil yang

didapat. Bila hasil yang didapat tidak sesuai, maka jelaskan apa kendalanya, tidak diwajibkan hasil harus sesuai dengan harapan.

Contoh: terhadap sistem informasi yang telah dirancang, akan dilakukan pengujian terhadap permasalahan yang tertuang pada bab pertama, yaitu bagaimana mempercepat proses pembuatan KTM.

Pertama-tama mahasiswa memasukkan NIM pada dashboard sistem seperti yang terlihat pada **Gambar 2.X** berikut. Blablabla

Hasil dari apa yang telah dilakukan mahasiswa tersebut adalah mendapatkan KTM sama seperti cara konvensional namun dari segi waktu dan tenaga jauh lebih cepat. Hal ini dapat terlihat bahwa waktu yang dibutuhkan hanya 5 menit dan dapat dilakukan dari mana saja selama ada koneksi internet.

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

### BAB III

#### JADWAL Pengerjaan dan Pembagian Tugas

##### 3.1 Agenda Pengerjaan

Berikut pada **Tabel 3.1** merupakan pembagian jenis tugas proyek akhir terhadap alokasi waktu pengerjaan pada bulan Maret dan April tahun 2020:

**Tabel 3.1** Agenda Pengerjaan Proyek

No .	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 8												
9.	Presentasi Proyek Akhir												

##### 3.2 Keterangan Pembagian Tugas

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

**Tabel 3.2** Pembagian Tugas Proyek

No.	Keterangan Tugas	Penanggung Jawab
1.	Perancangan Arsitektur Cloud Computing	Agus
2.	Pengujian Singkat	Budi
3.	Latar Belakang Masalah	Candra
4.	Agenda Pengerjaan Proyek	Dedi
5.	Dsb...	Candra
6.	Tugas 6	Budi
7.	Tugas 7	Dedi
8.	Tugas 8	Candra

(Pembagian tugas proyek akan diverifikasi oleh asisten praktikum, setiap penanggung jawab wajib bertanggungjawab terhadap tugasnya. Penilaian akan berdasarkan nilai individu dan tim.)

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Pada bagian ini, jelaskan mengenai hasil dari proyek yang Anda buat. Anda dapat mencontoh penulisan pada bagian 2.5 dan bagian 2.6. Penulisan pada bagian ini setidaknya memuat:

- a. Berdasarkan masalah, kemudian dikerjakan, maka didapatkan hasil yang seperti apa. Apakah dapat dikerjakan, terdapat kendala, atau hal teknis lainnya.
- b. Berdasarkan pengujian masalah secara acak, apa yang didapatkan.
- c. Berdasarkan pembagian tugas proyek pada bab 3, bagaimana hasilnya.

#### **4.2 Saran**

Pada bagian ini, jelaskan mengenai saran-saran yang mengacu pada bagian 4.1 mengenai kesimpulan proyek. Contohnya: pembagian tugas seharusnya lebih merata, spesifikasi laptop seharusnya dengan ram 10 GB, waktu yang tersedia tidak cukup dikarenakan bla bla bla, dan sebagainya.

## **DAFTAR PUSTAKA**



Tambahkan daftar pustaka dengan format yang digunakan di IF, yakni APA Style. Setidaknya ada 5 daftar pustaka yang Anda gunakan untuk menyelesaikan proyek ini.

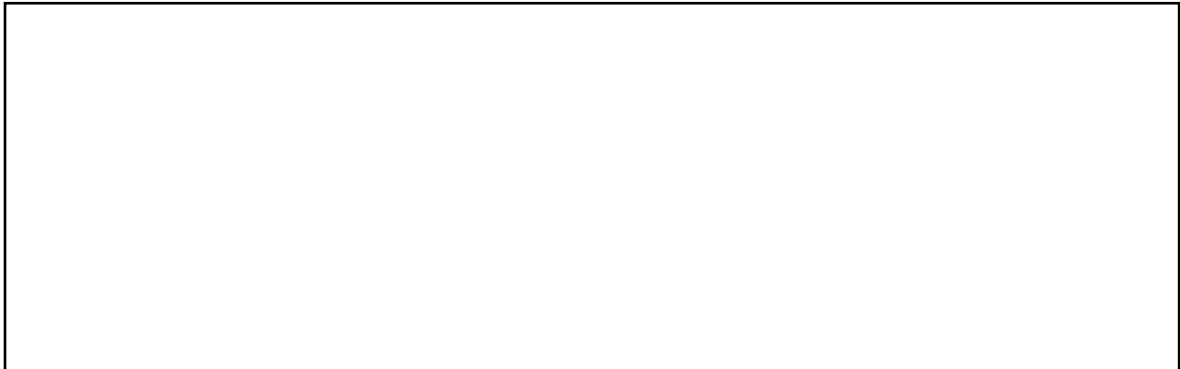
## **LAMPIRAN**

Lampiran pada bagian ini dapat berupa screenshoot, listing program yang terlalu panjang, dan sebagainya, atau dapat juga tugas bilamana diminta oleh asisten praktikum.

## CONTOH FORMAT PENULISAN

(hapus bagian ini)

Beri penjelasan mengenai gambar sebelum mencantumkan gambar. Setiap gambar wajib dengan border hitam tipis. Jangan lupa acu gambar tersebut, yakni pada **Gambar 2.1** berikut:



**Gambar 2.1** Font 11 Center

Contoh penulisan tabel sama seperti gambar, buat dulu penjelasannya kemudian acu pada **Tabel 1.2** berikut ini. Format yakni spacing single dan font size 10:

<i>Field</i>	<b>Tipe Data</b>	<b>Keterangan</b>
Nama Field 1	Tipe Data 1	<i>Jenis Constraint</i>
Nama Field 2	Tipe Data 2	<i>Jenis Constraint</i>
Nama Field 3	Tipe Data 3	
Nama Field 4	Tipe Data 4	
Nama Field 5	Tipe Data 5	

**Tabel 1.2** Tabel Nama Tabel 2

Keterangan dituliskan bila perlu, mengacu pada **Tabel 1.2** yakni:

- Format pengisian pada kolom field yakni nama field
- Keterangan dsb dsb

Untuk penulisan listing program atau kodingan jelaskan terlebih dahulu maksud listing tersebut lalu acu pada **Listing 2.1** berikut:

```
CANTUMKAN QUERY CREATE TABLE 1 DENGAN FONT COURIER SIZE 10 SPACING  
SINGLE TANPA JUSTIFY  
  
public function blabla(){  
    exit();  
}
```

**Listing Program 2.1** Listing Create Table Nama Tabel 1



Bila ingin menyatakan pengetikan parameter semisal `sudo apt install apache2` maka jelaskan terlebih dahulu maksud perintah tersebut lalu gunakan format pada **Perintah Program 2.1** berikut ini:

```
$ sudo apt install apache2
root@server:~# sudo apt install apache2
```

**Perintah Program 2.1** Operasi instalasi paket aplikasi *apache2*

Penjelasan ketentuan penulisan Laporan Proyek Akhir secara umum:

1. Huruf Times New Roman 12
2. Margin Top Left 3, Right Bottom 2.5.
3. Spacing 1,5
4. Penomoran Halaman, pada setiap **awal bab** di **bawah tengah**. Setiap **isi bab**, di **atas kanan**.  
Font TNR 11
5. Jatah nilai untuk format penulisan adalah 25%, bilamana format acak-acakan maka jatah nilai lainnya gugur, sehingga nilai proyek akhir = 0. Laporan merupakan bukti bahwa Anda telah mengerjakan tugas sesuai dengan apa yang Anda presentasikan

*Selamat mengerjakan, semoga sukses skripsi dan lulus cepat dengan nilai memuaskan.*