

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING

LAPORAN PROYEK AKHIR

**APLIKASI PENYEWAAN RUANGAN DAN ALAT MUSEUM SANDI
YOGYAKARTA BERBASIS WEB MENGGUNAKAN UBUNTU LAMP DAN
PROSES PEMBUATAN DOCKERFILENYA**



DISUSUN OLEH:

NAMA ANGGOTA : VERONIKA HANA SELFRIDA S. 123170003
NURZULHIJAH 123170009

KELAS : E

ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.
WAHYU AJI NUGROHO, S.Kom.

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

2020

HALAMAN PENGESAHAN

APLIKASI PENYEWAAN RUANGAN DAN ALAT MUSEUM SANDI YOGYAKARTA BERBASIS WEB MENGGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

Veronika Hana Selfrida Sinaga

123170003

Nurzulhijjah

123170009

Telah diperiksa dan disetujui oleh Asisten Praktikum Teknologi Cloud Computing
pada tanggal :

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Wahyu Aji Nugroho, S.Kom.

Mengetahui,

Ka. Lab. Sistem Digital

Mangaras Yanu Florestivanto, S.T., M.Eng.

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Assalamualaikum wr. wb.,

Puji dan syukur kami panjatkan bagi Tuhan Yang Maha Esa karena berkat rahmat dan hidayah-Nya sehingga kelompok kami dapat menyelesaikan laporan akhir praktikum Teknologi *Cloud Computing* ini. Penulisan laporan akhir ini disusun dengan sistematika penulisan laporan akhir yang sesuai pada buku panduan praktikum.

Laporan akhir ini merupakan tanggung jawab dan tugas kami dalam menjalani kuliah Informatika yang dimana dalam laporan ini berjudul “Aplikasi Penyewaan Ruang dan Alat Museum Sandi Yogyakarta Berbasis Web Menggunakan Ubuntu LAMPP dan Proses Pembuatan Dockerfilenya”.

Laporan ini tentu saja masih jauh dari yang diharapkan untuk memenuhi tugas akhir praktikum Jaringan Komputer, sehingga masukan dari pihak sangat diharapkan untuk menjadikan tugas akhir saya menjadi semakin bagus dan berkualitas.

Kami juga mengucapkan terima kasih kepada semua pihak yang sudah membantu kami dalam proses penyusunan laporan ini terutama kepada saudara Jaluanda Parama, S.Kom. dan saudara Wahyu Aji Nugroho, S.Kom. selaku asisten praktikum.

Demikian laporan ini dibuat agar digunakan sebaik-baiknya dan bermanfaat bagi semua pihak khususnya penulis. Kami mohon maaf jika terdapat kesalahan dalam penulisan, akhir kata kami ucapkan terimakasih.

Wassalamualaikum, wr. Wb.

Yogyakarta, 17 Mei 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Tujuan Proyek Akhir.....	2
1.3 Manfaat Proyek Akhir.....	2
1.4 Tahap Penyelesaian Proyek Akhir	3
BAB II ISI DAN PEMBAHASAN.....	4
2.1 Komponen yang Digunakan	4
2.1.1 Komponen pada Proyek Pertama.....	4
2.1.2 Komponen pada Proyek Kedua	5
2.2 Rancangan Arsitektur Cloud Computing.....	5
2.3 Parameter dan Konfigurasi	7
2.3.1 Parameter dan Konfigurasi pada Proyek Pertama	7
2.3.2 Parameter dan Konfigurasi pada Proyek Kedua	8
2.4 Tahap Implementasi.....	11
2.4.1 Tahap Implementasi pada Proyek Pertama.....	11
2.4.2 Tahap Implementasi pada Proyek Kedua	15
2.5 Hasil Implementasi	18
2.5.1 Hasil Implimentasi pada Proyek Pertama	18
2.6 Pengujian Singkat	19
2.6.1 Pengujian Singkat pada Proyek Pertama	19
2.6.2 Pengujian Singkat pada Proyek Kedua	22
BAB III JADWAL Pengerjaan dan Pembagian Tugas.....	23
3.1 Agenda Pengerjaan	23
3.2 Keterangan Pembagian Tugas.....	23
BAB IV KESIMPULAN DAN SARAN	24
4.1 Kesimpulan	24
4.2 Saran	24
DAFTAR PUSTAKA.....	25
LAMPIRAN.....	26

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

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 60 an pun sudah mulai muncul konsep yang menyatukan beberapa sumber computing dengan menggunakan jaringan yang bersifat global. Pada saat itu, sistem seperti ini disebut dengan “Intergalactic Computer Network”. Sistem ini diciptakan oleh J.C.R. Licklider yang kemudian menjadi penanggung jawab atas pembangunan Advanced Research Projects Agency Network (ARPANET) tepatnya pada tahun 1969.

Cloud computing atau komputasi awan merupakan gabungan pemanfaatan teknologi komputer dan pengembangan berbasis internet. Ia adalah suatu metoda komputasi dimana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan (*as a service*), sehingga pengguna dapat mengaksesnya lewat internet tanpa mengetahui apa yang ada didalamnya, ahli dengannya atau memiliki kendali terhadap infrastruktur teknologi yang membantunya. *Cloud computing* juga merupakan sebuah kombinasi pemanfaatan jaringan internet yang mana berfungsi untuk menyimpan berbagai file dalam satu basis data. Pada teknologi ini, data disimpan di *server* tertentu, begitu pula dengan *software* atau aplikasi lainnya sehingga memungkinkan satu komputer server untuk membagikannya dengan komputer lainnya yang terhubung. Hal ini akan menghemat biaya operasional juga menghemat waktu karena tidak memerlukan hardisk berkapasitas besar untuk menyimpan setiap file *software*. Seperti misalnya data microsoft word yang cukup diinstall satu kali pada server lalu bisa digunakan di

Sistem *cloud computing* bekerja secara online dan terus-menerus tanpa henti melalui jaringan internet. Semua aktifitas akan berpusat di komputer server dan setiap jenis data akan langsung di simpan dan siap pakai kapanpun. Cara kerja selanjutnya ialah memungkinkan *user* menjalankan suatu aplikasi dan setiap hal yang dijalankan pada aplikasi tersebut akan kembali tersimpan pada komputer server. *End user* dan komputer *server* yang disimpan oleh *switch* maupun router untuk meng-*extend* jaringan yang ada. Sehingga kita bisa melihat atau menjalankan kembali aplikasi tersebut dimanapun dan kapanpun.

Contoh dari *cloud computing* adalah Google Apps yang menyediakan aplikasi bisnis umum secara daring yang bisa diakses melalui suatu penjelajah web dengan perangkat lunak

dan data yang tersimpan di server. Contoh pengembangan dari *Cloud Computing* adalah iCloud dan beberapa contoh lain yaitu Google Drive, Windows Azure, Yahoo, Dropbox, dll.

Proyek akhir yang berjudul Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta Berbasis Web Menggunakan Ubuntu LAMPP dan Proses Pembuatan Dockerfilenya ini bertujuan agar memudahkan masyarakat untuk melakukan penyewaan di Museum Sandi Yogyakarta yang dimana penyewa dapat login, membuat akun, memilih daftar yang ingin disewa, serta menunggu konfirmasi dari pihak terkait.

Pada proyek akhir ini, tahap penyelesaian dimulai dari menyiapkan dan menginstall software yang akan digunakan seperti VMWare Workstation, ISO Ubuntu, PuTTY, WinSCP. Setelah itu dilakukan instalasi PHP kemudian Apache dan MySQL pada PuTTY agar proses dapat dikerjakan agar dapat menjalankan database dan program. Kemudian file proyek yang telah dibuat dipindahkan ke dalam LAMPP dan mengecek seluruh bagiannya. Selanjutnya dilanjutkan dengan pembuatan dockerfile agar program lebih mudah untuk diakses yang bertujuan untuk membantu user yang ingin melakukan penyewaan di museum sandi sehingga user tidak perlu lagi datang ke museum sandi untuk melakukan penyewaan. Di dalam sistem sudah tertera informasi data serta tanggal penyewaan yang akan diisi oleh user yang selanjutnya akan dikonfirmasi oleh admin pihak museum sandi. Project akhir ini menggunakan LAMPP dan Docker dalam penyelesaiannya. Selain itu tahapan yang dilakukan yaitu menyiapkan software yang dibutuhkan seperti Vmware Workstation, ubuntu, dan LAMPP yang sudah dilakukan instalasi. Serta membutuhkan web atau sistem penyewaan (PHP/HTML) yang sudah dibuat sebelumnya dari tugas project KP.

1.2 Tujuan Proyek Akhir

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

1. Mengimplementasikan arsitektur *cloud computing* untuk Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Mengintegrasikan Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta yang berada di Ubuntu Server (LAMPP)
3. Membuat Dockerfile Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta berbasis web agar dapat diakses oleh pengguna dimana saja.

1.3 Manfaat Proyek Akhir

Manfaat yang dapat diperoleh dari pembuatan proyek akhir ini adalah sebagai berikut:

1. Sistem Monitoring yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.
2. Pihak Museum Sandi tidak perlu memperlakukan *maintenance*, karena dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.
3. Batasan memori penyimpanan multimedia menjadi tidak terbatas dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing*.
4. Permintaan data secara *realtime* dapat dilakukan secara terpusat maupun secara terdistribusi dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.
5. Pengguna tidak perlu lagi datang ke lokasi museum sandi untuk melakukan penyewaan karena akses ini dapat dilakukan dimana saja.

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta untuk ditransformasikan kedalam arsitektur *cloud computing* berbasis SaaS.
2. Menentukan konfigurasi yang tepat untuk pengaturan Dockerfile sehingga dapat digunakan sesuai requirement yang berupa:
 - a. Dapat diinstal di mana saja tanpa mempengaruhi environment sistem.
 - b. Dapat melakukan migrasi database dengan mudah.
 - c. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
3. Melakukan konfigurasi untuk pembuatan dockerfile agar dapat diakses pada IP tertentu saja.
4. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Penyewaan Alat dan Ruangan Museum Sandi Yogyakarta ketersediaannya/*availability*-nya maksimal.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir ini dibagi dalam dua bagian yaitu yang pertama ialah “Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta Berbasis Web menggunakan Ubuntu LAMPP” dan yang kedua ialah “Pembuatan Dockerfilenya”. Berikut akan dibahas keduanya dalam sub bab berikut.

2.1.1 Komponen pada Proyek Pertama

Dalam membangun “Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta Berbasis Web menggunakan Ubuntu LAMPP” maka diperlukan analisis komponen sebagai berikut :

1. Sistem yang telah dibangun menggunakan Bahasa pemrograman yaitu PHP versi 7.3 serta menggunakan Bahasa HTML dengan versi minimal adalah 4.0.
2. MySQL versi 5.2 digunakan sebagai penyimpanan data penyewa alat dan ruangan yang dibutuhkan oleh sistem.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaannya ialah seluruh masyarakat yang ingin menyewa alat dan ruangan museum. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *public*, dapat diakses secara bebas oleh semua orang.
4. Dalam pembuatan Dockerfile diperlukan Docker Engine dan Docker Compose dengan versi 1.24.1.
5. Sistem sewaktu-waktu dapat digunakan sehingga membutuhkan uptime SLA sebesar 99,9%. Dengan demikian dibutuhkan dua buah Ubuntu server sebagai *primary server* dan sebagai *backup server*. Dalam peralihan antar server membutuhkan *proxy server* yang semuanya menggunakan Ubuntu sebagai sistem operasinya.

Berdasarkan poin-poin tersebut, komponen utama dalam Menyusun *cloud computing* yang dibutuhkan akan disimpulkan dalam bentuk table dibawah ini.

Tabel 2.1 Spesifikasi VM cloud computing untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.2Ghz	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: 192.168.181.135	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.110.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.110.1	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	2 GB	Alokasi RAM untuk <i>guest OS</i>

Selain itu, spesifikasi yang telah digunakan dalam Ubuntu OS yang telah dibuat dalam Virtual Machine tersebut adalah sebagai berikut:

Table 2.2 Spesifikasi Ubuntu OS untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preproesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Dalam pembuatan dan pengembangan web system yang telah dibuat.
		PHPMyadmin	Layanan untuk manajemen database
2.	PHP	Versi 7.4.5	Dukungan bahasa pemrograman yang digunakan oleh sistem penyewaan alat dan ruangan
3.	Docker Compose	Versi 1.24.1	Dukungan untuk menjalankan <i>multi container</i> pada Docker
4.	PuTTY	Versi 0.73	Dukungan untuk memindahkan atau pemrosesan proyek.

2.1.2 Komponen pada Proyek Kedua

Dalam membangun Dockerfile maka diperlukan analisis komponen yaitu sistem yang telah dibangun menggunakan instalasi Docker yang dimana fungsinya dapat memungkinkan pihak admin museum dapat menyimpan data pada lokasi tersentralisasi yang dapat diakses oleh penyewa dari beberapa tempat secara bersamaan.

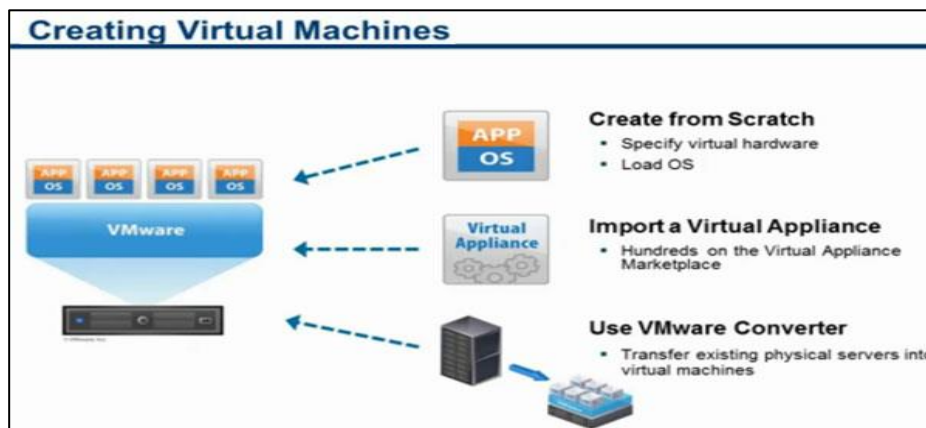
2.2 Rancangan Arsitektur Cloud Computing

Pada project akhir ini digunakan bentuk rancangan arsitektur IaaS di mana hardware laptop ASUS ROG sebagai layer utama di bagian bawah, kemudian di atasnya adalah PaaS atau sistem operasi Windows dan aplikasi VMware Workstation yang berjalan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



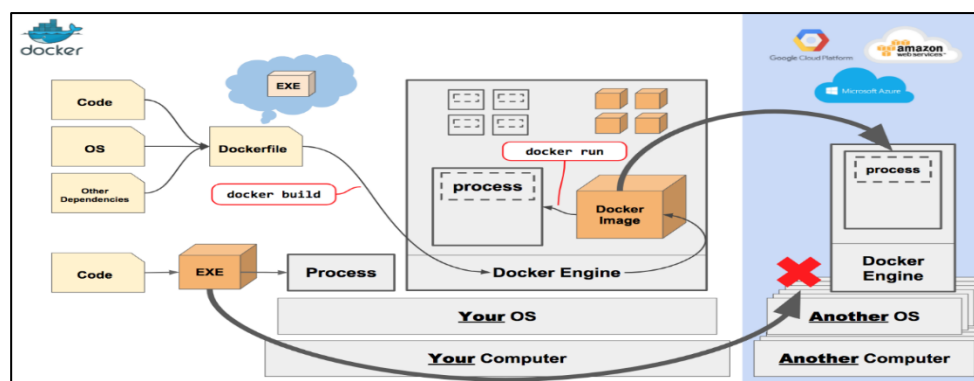
Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya

Proyek akhir ini menggunakan *virtual machine* atau VM yang digunakan sebagai layanan untuk hosting local. VM sendiri bertujuan untuk melakukan virtualisasi, visualisasi software, sistem operasi, device dan aplikasi. Ilustrasi mengenai rancangan arsitektur virtual machine dapat dilihat pada **Gambar 2.2** berikut ini:



Gambar 2.2 Penjelasan penggunaan arsitektur *virtual machine*

Proyek akhir ini juga menggunakan Docker yang digunakan sebagai layanan untuk developer juga administrator admin, supaya pengembang bisa membangun, mengemas, serta menjalankan berbagai aplikasi dalam lokasi manapun sebagai sebuah container yang bersifat sederhana. Ilustrasi mengenai rancangan arsitektur Docker dapat dilihat pada **Gambar 2.3** berikut ini:



Gambar 2.3 Penjelasan arsitektur Docker

2.3 Parameter dan Konfigurasi

2.3.1 Parameter dan Konfigurasi pada Proyek Pertama

Berikut dijelaskan parameter-parameter yang digunakan untuk melakukan instalasi dan konfigurasi layanan hosting dengan LAMPP yang telah dibuat. Parameter yang digunakan untuk pembaruan package dan instalasi Apache dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ sudo apt update
$ sudo apt install apache2
```

Keterangan:

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

Modul 2.1 Parameter instalasi Apache

Parameter yang digunakan untuk pengaturan Firewall dilakukan karena pada Ubuntu dengan versi baru memiliki pengaturan default Firewall yang otomatis aktif, sehingga diperlukan parameter berikut untuk mengatur Firewall sehingga lalu lintas data dari Apache Web Server dengan tipe protocol HTTP (port 80) dan HTTPS (port 443) diijinkan dan dapat dilihat pada penjelasan **Modul 2.2** berikut ini :

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

Keterangan:

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

Modul 2.2 Parameter pengaturan Firewall

Pastikan Web Server menyala (running) dengan cara mengetikkan pada browser <http://IP Server>. Selanjutnya parameter untuk instalasi akan dijelaskan pada **Modul 2.3** dibawah ini:

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

Keterangan:

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

Modul 2.3 Parameter instalasi MySQL

Parameter proses pengaturan dasar pengamanan MySQL yang berupa users, kata sandi, hak akses, dan sebagainya akan dijelaskan pada **Modul 2.4** berikut dibawah ini, serta pilih “Y” untuk setiap pertanyaan sampai proses akhir adalah informasi “All Done!” yang menandakan proses pengaturan MySQL selesai.

```
$ sudo apt install mysql_secure_installation
```

Keterangan:

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

Modul 2.4 Pengaturan dasar pengamanan MySQL

Tahap selanjutnya adalah proses instalasi PHP yang akan dijelaskan pada **Modul 2.5** berikut ini:

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

Keterangan:

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

Modul 2.5 Parameter instalasi PHP

Parameter yang digunakan untuk proses instalasi package dan PHPMyAdmin dapat dilihat pada penjelasan **Modul 2.6** 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 paket yang akan diinstall
- php-mbstring dan php-gettext : fungsi untuk memanggil perintah string dan memperoleh text pada PHPMyAdmin.

Modul 2.6 Parameter instalasi PHPMyAdmin

2.3.2 Parameter dan Konfigurasi pada Proyek Kedua

Untuk melakukan prosese instalasi docker, langkah awal yang perlu dilakukan adalah update list packages terlebih. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.7** berikut ini:

```
$ sudo apt update
```

Keterangan:

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

Modul 2.7 Parameter update list packages

Selanjutnya melakukan instalasi beberapa packages agar dapat diakses oleh HTTPS.

Parameter yang akan digunakan akan dijelaskan pada **Modul 2.8** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- apt-transport-https, ca-certificates, curl, gnupg-agent, software-properties-common : nama paket aplikasi yang akan diinstall

Modul 2.8 Parameter install packages

Setelah packages di update dan diinstall maka langkah selanjutnya adalah melakukan penambahan GPG Key dari Docker repository ke dalam sistem yang akan dibuat. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.9** berikut ini:

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

Keterangan:

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

Modul 2.9 Parameter penambahan GPG key

Langkah selanjutnya adalah melakukan penambahan Docker *repository* dimana *repository* yang akan digunakan adalah *repository* dengan versi *stable*. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.10** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- add-apt-repository : untuk menambahkan repository sehingga Ubuntu tahu untuk mencari pembaruan dari PPA tersebut berasal dari sumber yang resmi.
- lsb_release :
- stable : versi repository yang akan diinstall

Modul 2.9 Parameter penambahan Docker repository

Selanjutnya update package database dan dilanjutkan melakukan instalasi repo ke dalam Ubuntu. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.10** berikut ini:

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

Keterangan:

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

Modul 2.10 Parameter instalasi package database dan Docker repo

Langkah selanjutnya adalah melakukan instalasi Docker agar docker dapat berjalan.

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

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- docker-ce : paket aplikasi untuk docker
- systemctl status: Untuk memastikan status docker telah berjalan

Modul 2.11 Parameter instalasi Docker

Langkah selanjutnya adalah melakukan download docker-compose binary file dan menginstall curl. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.12** berikut ini:

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

Keterangan:

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

Modul 2.12 Parameter download docker-compose binary file dan instalasi curl

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

Parameter yang akan digunakan akan dijelaskan pada **Modul 2.13** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)

- `chmod +x` : untuk melakukan eksekusi data
- `/usr/local/bin/docker-compose` : untuk instalasi docker compose
- `docker-compose version` : untuk mengetahui versi docker compose

Modul 2.13 Parameter instalasi docker-compose

Apabila Docker dan docker compose sudah dipastikan dapat berjalan dengan baik, langkah selanjutnya adalah mengatur docker compose agar proyek dapat dikerjakan dilanjutkan dengan pembuatan Dockerfile. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.14** berikut ini:

```
$ cd /var/www/html/museumsandi/
$ nano Dockerfile
$ nano docker-compose.yml
```

Keterangan:

- `cd` : untuk mungubah file directory
- `/var/www/html/museumsandi/` : diroctery untuk folder museum sandi
- `Nano` : seperti sarana notepad untuk PuTTY
- `Dockefile` : file
- `docker-compose.yml` : file

Modul 2.14 Parameter mengatur folder proyek dan pembuatan Dockerfilenya

Selanjutnya menjalankan fungsi web-server, mysql-server, dan phpmyadmin pada Docker. Parameter yang akan digunakan akan dijelaskan pada **Modul 2.15** berikut ini:

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

Keterangan:

- `docker-compose up` : untuk memanggil semua output container
- `-d` : untuk menjalankan background pada docker compose

Modul 2.15 Parameter menjalankan web-server, mysql-server, dan phpmyadmin

Selanjutnya untuk menjalankan agar proses pengambilan database dari phpMyAdmin dapat dijalankan maka harus memanggil parameter yang akan dijelaskan pada **Modul 2.16** berikut ini:

```
$ docker ps -a
$ docker exec -it peminjaman-db mysql -uvero -pvero museumsandi <
./peminjamnan.sql

$ docker-compose down
```

Keterangan:

- `Docker ps` : untuk menampilkan container pada docker yang aktif
- `-a` : untuk menampilkan seluruh docker container
- `Docker-compose down` : menghentikan container pada docker

Modul 2.16 Parameter pemanggilan database

2.4 Tahap Implementasi

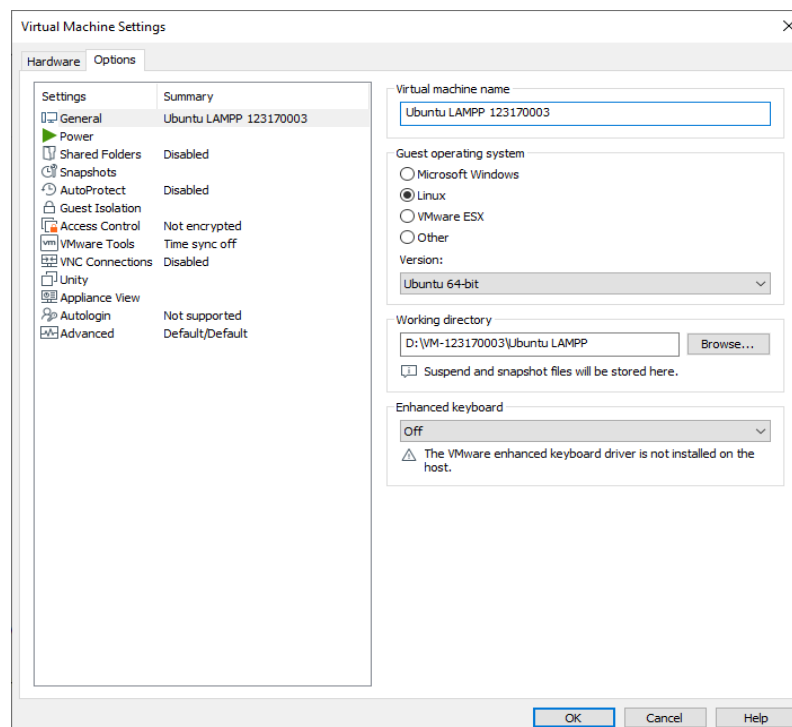
2.4.1 Tahap Implementasi pada Proyek Pertama

Tahapan awal pada wizard pembuatan VM pada VMware Workstation untuk Ubuntu LAMPP digunakan opsi konfigurasi Custom seperti pada **Gambar 2.4** berikut ini:



Gambar 2.4 Proses awal pembuatan VM pada VMware Workstation

Hasil implementasi pembuatan VM pada VMware Workstation untuk Ubuntu LAMPP seperti pada **Gambar 2.5** berikut ini :



Gambar 2.5 Proses akhir pembuatan VM pada VMware Workstation

Hasil implementasi setelah melakukan intallasi pembuatan VM adalah tampilan *login server* ubuntu seperti terlihat pada **Gambar 2.6** berikut ini:


```
[ 144.169411] ccloud-init[1696]: Cloud-init v. 19.4-33-gbb4131a2-0ubuntu1~18.04.1 finished at Mon, 11 May 2020 14:02:59 +0000. DataSource DataSourceNoCloud [seed=/var/lib/cloud/seed/nocloud-net] [dsmode=net]. Up 144.14 seconds

Ubuntu 18.04.4 LTS veronika tty1

veronika login: veronika
Password:
Last login: Mon May 11 13:16:12 UTC 2020 on tty1
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon May 11 14:20:47 UTC 2020

System load:  0.15           Processes:            168
Usage of /:   29.6% of 19.56GB Users logged in:      0
Memory usage: 48%           IP address for ens33: 192.168.181.132
Swap usage:   0%            IP address for docker0: 172.17.0.1

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

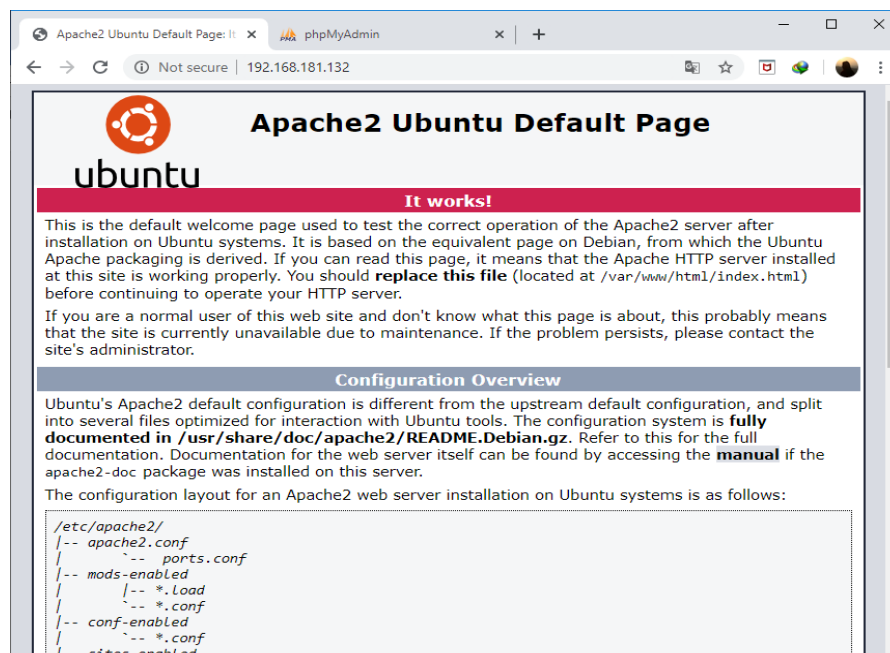
29 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection
or proxy settings

project TCC
veronika@veronika:~$
```

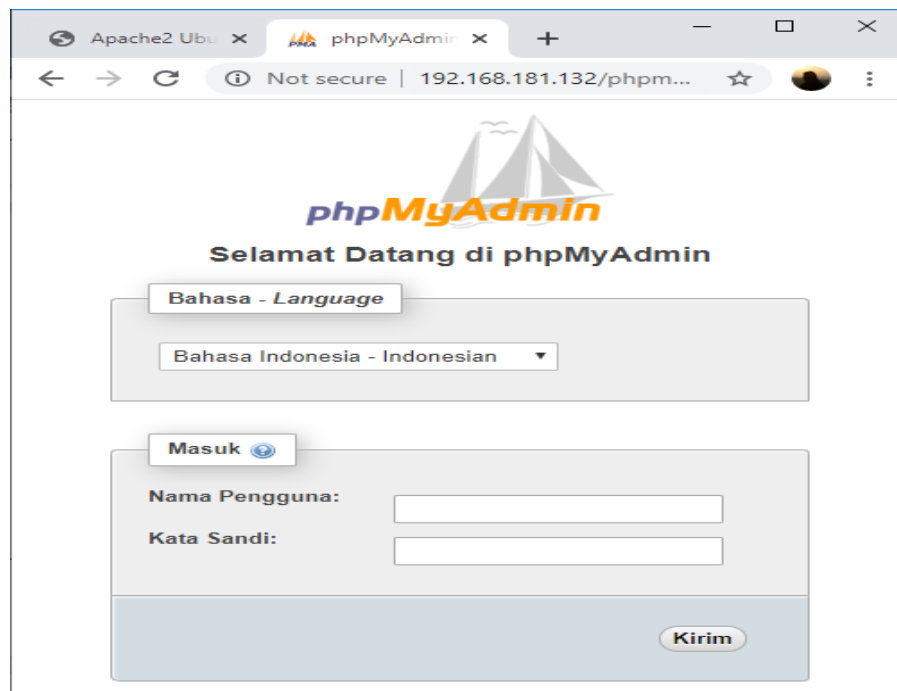
Gambar 2.6 Tampilan *login* ubuntu

Hasil dari implementasi setelah melakukan intallasi Apache dapat dilihat pada **Gambar 2.7** berikut ini:



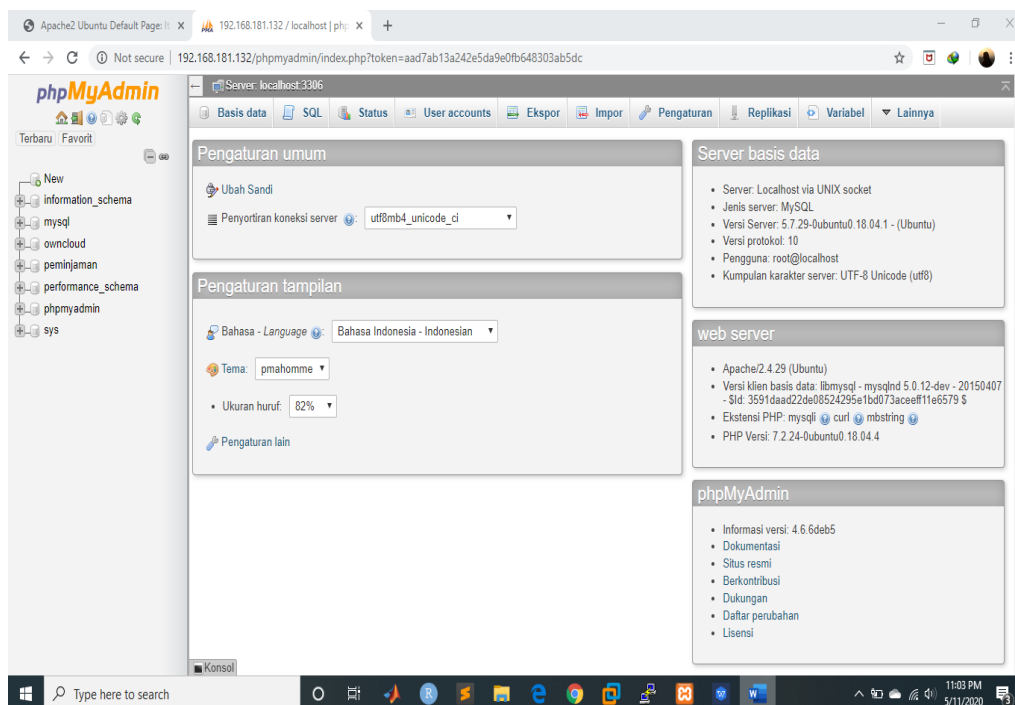
Gambar 2.7 Tampilan setelah installasi *Apache*

Hasil dari implementasi setelah melakukan intallasi phpMyadmin dapat dilihat pada **Gambar 2.8** berikut ini:



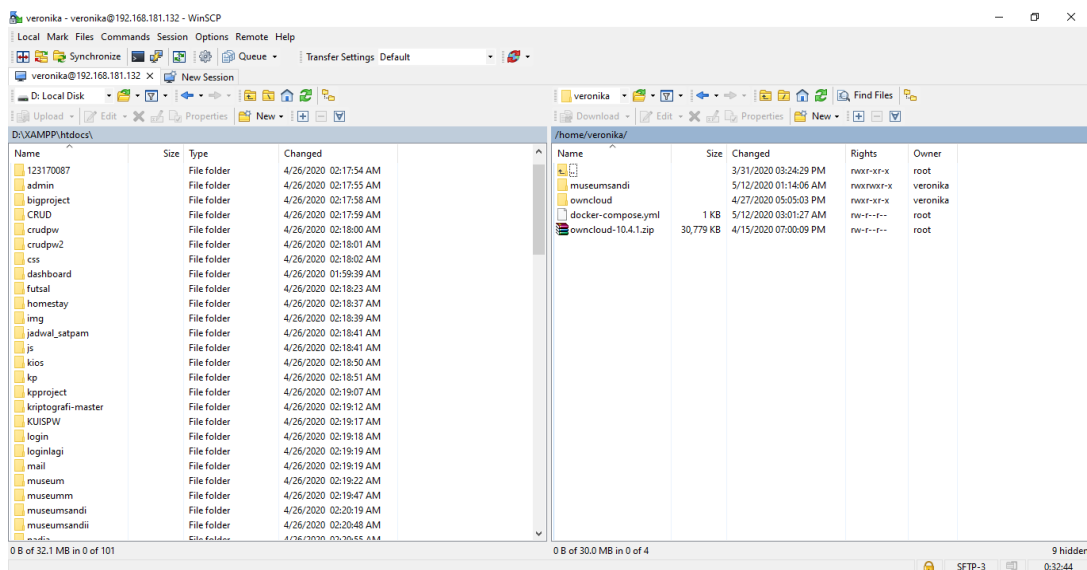
Gambar 2.8 Tampilan *login* phpMyadmin

Hasil dari implementasi saat *login* pada phpMyadmin berhasil akan ditujukan ke halaman localhost yang memuat database yang dapat dilihat pada **Gambar 2.9** berikut ini:



Gambar 2.9 Tampilan *login* phpMyAdmin berhasil

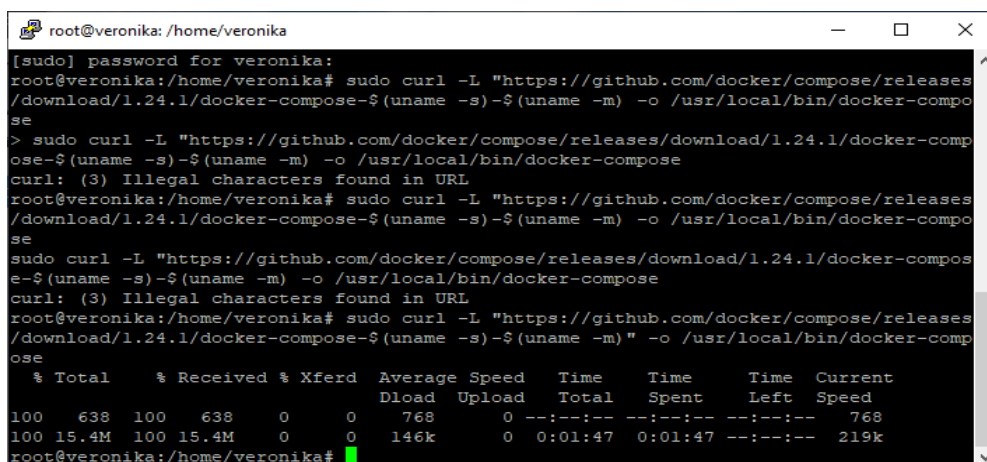
Setelah phpMyAdmin dapat berjalan, kemudian masukkan folder proyek yang akan dikerjakan ke WinSCP serta mengganti setiap username dan password untuk terhubung ke database dengan username dan password yang digunakan untuk login ke phpMyAdmin pada ubuntu. Untuk tampilannya dapat dilihat pada **Gambar 2.10** berikut ini:



Gambar 2.10 Tampilan folder proyek di WinSCP

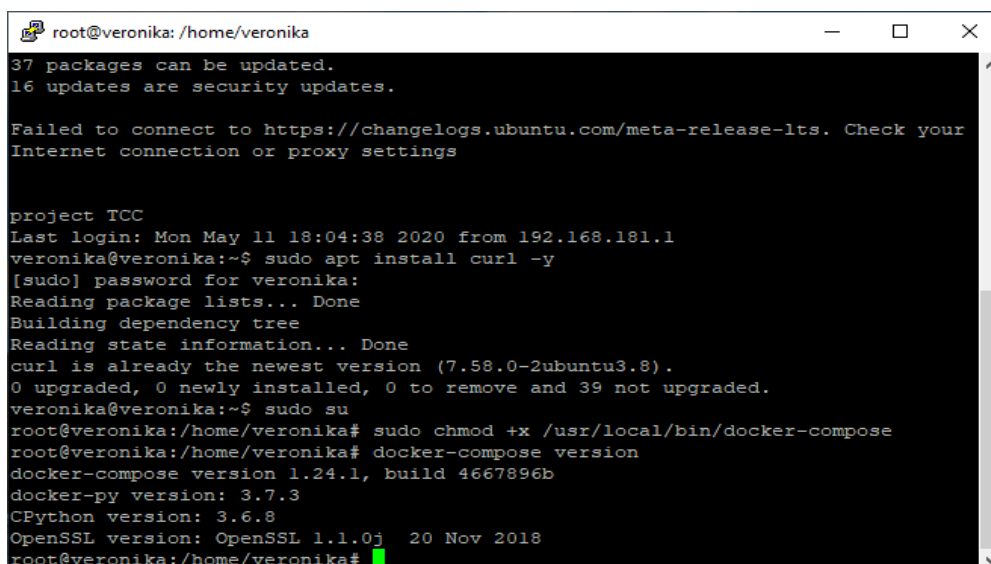
2.4.2 Tahap Implementasi pada Proyek Kedua

Hasil dari implementasi ketika selesai melakukan download docker compese dapat dilihat pada Gambar 2.11 berikut ini:



Gambar 2.11 Tampilan download docker compose berhasil pada PuTTY

Hasil implementasi ketika selesai melakukan instalasi docker-compose dapat dilihat pada Gambar 2.12 berikut ini:



```

root@veronika: /home/veronika
37 packages can be updated.
16 updates are security updates.

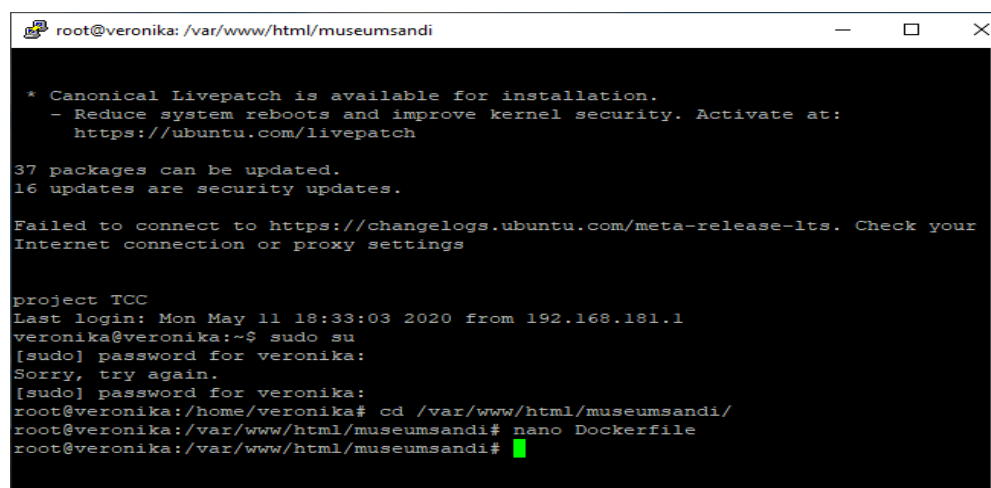
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

project TCC
Last login: Mon May 11 18:04:38 2020 from 192.168.181.1
veronika@veronika:~$ sudo apt install curl -y
[sudo] password for veronika:
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
veronika@veronika:~$ sudo su
root@veronika:/home/veronika# sudo chmod +x /usr/local/bin/docker-compose
root@veronika:/home/veronika# docker-compose version
docker-compose version 1.24.1, build 4667896b
docker-py version: 3.7.3
CPython version: 3.6.8
OpenSSL version: OpenSSL 1.1.0j  20 Nov 2018
root@veronika:/home/veronika#

```

Gambar 2.12 Tampilan instalasi docker-compose berhasil

Hasil implementasi untuk masuk ke dalam folder proyek dan terhubung ke WinSCP serta diikuti proses pembuatan folder Dockerfile dapat dilihat pada **Gambar 2.13** berikut ini:



```

root@veronika: /var/www/html/museumsandi

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

37 packages can be updated.
16 updates are security updates.

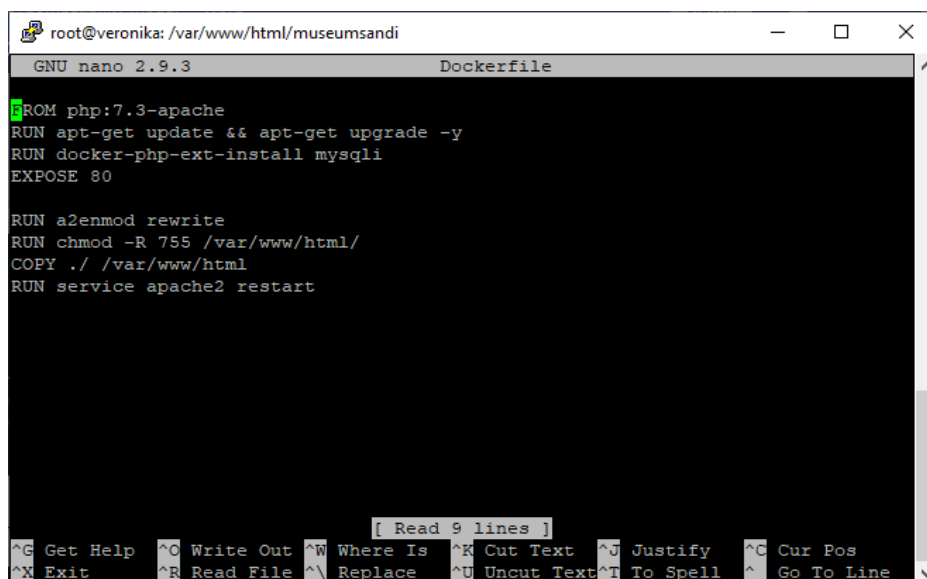
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

project TCC
Last login: Mon May 11 18:33:03 2020 from 192.168.181.1
veronika@veronika:~$ sudo su
[sudo] password for veronika:
Sorry, try again.
[sudo] password for veronika:
root@veronika:/home/veronika# cd /var/www/html/museumsandi/
root@veronika:/var/www/html/museumsandi# nano Dockerfile
root@veronika:/var/www/html/museumsandi#

```

Gambar 2.13 Tampilan instalasi docker-compose dan pembuatan Dockerfile

Setelah folder Dockerfile berhasil dibuat, maka selanjutnya isi perintah folder Dockerfile dapat dilihat pada **Gambar 2.14** berikut ini:



```

root@veronika: /var/www/html/museumsandi
GNU nano 2.9.3 Dockerfile

FROM php:7.3-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli
EXPOSE 80

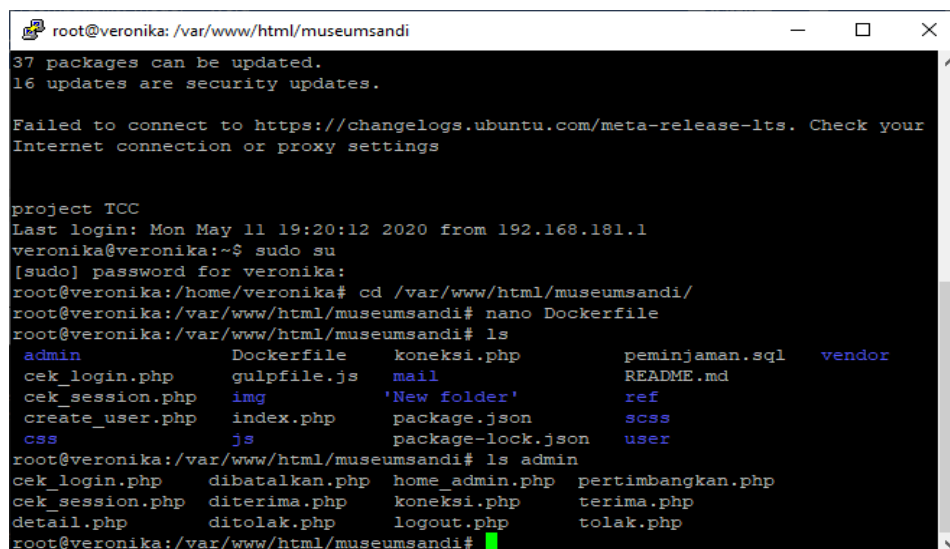
RUN a2enmod rewrite
RUN chmod -R 755 /var/www/html/
COPY ./ /var/www/html
RUN service apache2 restart

[ Read 9 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line

```

Gambar 2.14 Tampilan isi folder Dockerfile

Untuk membuktikan apakah folder Dockerfile telah berhasil dibuat, maka untuk mengeceknya dapat dilakukan dengan cara ketikkan perintah: \$ ls pada PuTTY. Tampilannya dapat dilihat pada **Gambar 2.15** berikut ini:



```

root@veronika: /var/www/html/museumsandi
37 packages can be updated.
16 updates are security updates.

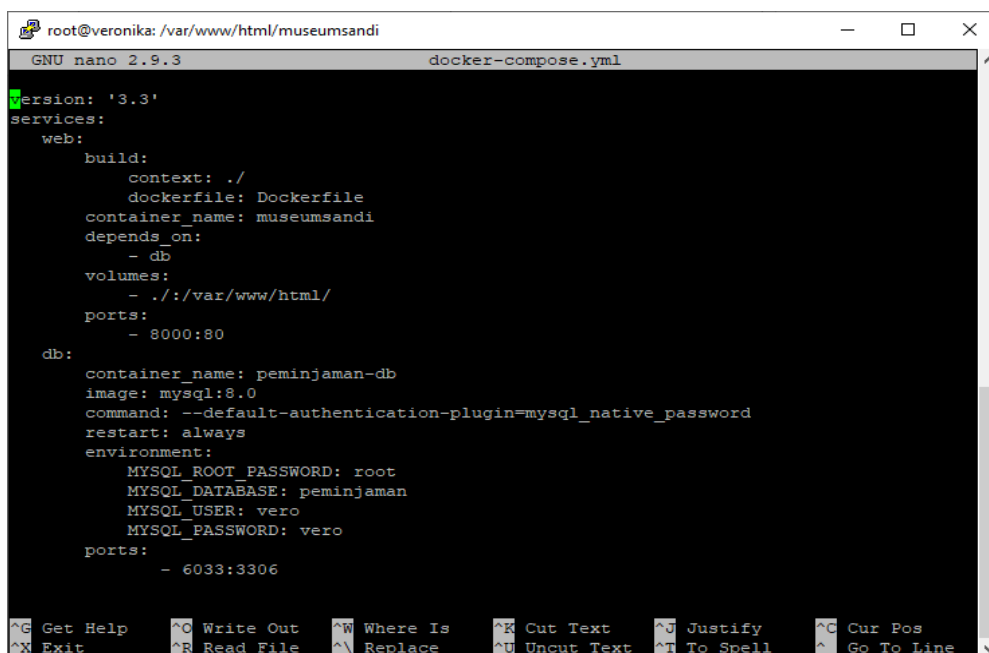
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

project TCC
Last login: Mon May 11 19:20:12 2020 from 192.168.181.1
veronika@veronika:~$ sudo su
[sudo] password for veronika:
root@veronika:/home/veronika# cd /var/www/html/museumsandi/
root@veronika:/var/www/html/museumsandi# nano Dockerfile
root@veronika:/var/www/html/museumsandi# ls
admin          Dockerfile     koneksi.php    peminjaman.sql  vendor
cek_login.php  gulpfile.js    mail           README.md
cek_session.php img            'New folder'   ref
create_user.php index.php      package.json    scss
css            js             package-lock.json user
root@veronika:/var/www/html/museumsandi# ls admin
cek_login.php  dibatalkan.php  home_admin.php  pertimbangkan.php
cek_session.php diterima.php  koneksi.php     terima.php
detail.php     ditolak.php     logout.php      tolak.php
root@veronika:/var/www/html/museumsandi#

```

Gambar 2.15 Tampilan Dockerfile telah masuk ke folder proyek

Hasil dari implementasi pembuatan docker-compose.yml yang telah dibuat, di dalam filenya diisi perintah seperti terlihat pada **Gambar 2.16** dan pembuktian apakah file docker-compose telah masuk ke folder proyek dapat dilihat pada **Gambar 2.17** berikut ini:

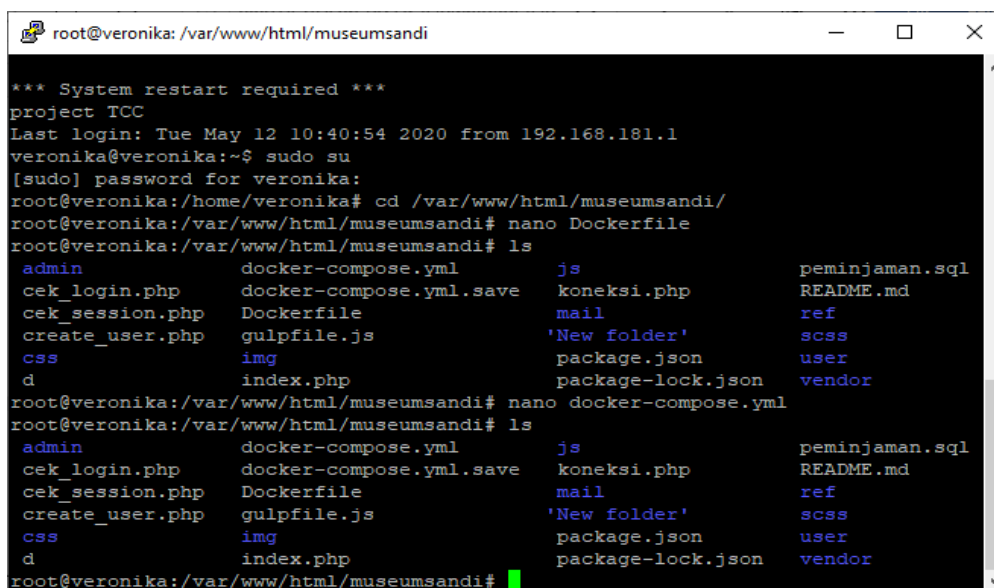


```

root@veronika: /var/www/html/museumsandi
GNU nano 2.9.3 docker-compose.yml
version: '3.3'
services:
  web:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: museumsandi
    depends_on:
      - db
    volumes:
      - ../var/www/html/
    ports:
      - 8000:80
  db:
    container_name: peminjaman-db
    image: mysql:8.0
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: peminjaman
      MYSQL_USER: vero
      MYSQL_PASSWORD: vero
    ports:
      - 6033:3306
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line

```

Gambar 2.16 Tampilan isi docker-compose.yml



```

root@veronika: /var/www/html/museumsandi
*** System restart required ***
project TCC
Last login: Tue May 12 10:40:54 2020 from 192.168.181.1
veronika@veronika:~$ sudo su
[sudo] password for veronika:
root@veronika:/home/veronika# cd /var/www/html/museumsandi/
root@veronika:/var/www/html/museumsandi# nano Dockerfile
root@veronika:/var/www/html/museumsandi# ls
admin      docker-compose.yml      js          peminjaman.sql
cek_login.php  docker-compose.yml.save koneksi.php  README.md
cek_session.php Dockerfile              mail        ref
create_user.php gulpfile.js             'New folder' scss
css         img                    package.json user
d           index.php              package-lock.json vendor
root@veronika:/var/www/html/museumsandi# nano docker-compose.yml
root@veronika:/var/www/html/museumsandi# ls
admin      docker-compose.yml      js          peminjaman.sql
cek_login.php  docker-compose.yml.save koneksi.php  README.md
cek_session.php Dockerfile              mail        ref
create_user.php gulpfile.js             'New folder' scss
css         img                    package.json user
d           index.php              package-lock.json vendor
root@veronika:/var/www/html/museumsandi#

```

Gambar 2.17 Tampilan docker-compose.yml telah masuk ke folder proyek

2.5 Hasil Implementasi

2.5.1 Hasil Implimentasi pada Proyek Pertama

Setelah selesai melakukan konfigurasi dengan Ubuntu LAMPP, Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta sudah dapat berjalan. Untuk pembuktian apakah prosesnya sudah berjalan bisa dibuka di browser dengan cara ketikkan IP_VM>Nama_folder/file.php (192.168.181.132/museumsandi/index.php) dan juga dapat dilihat pada **Gambar 2.18** Berikut ini:



Gambar 2.18 Tampilan halaman utama aplikasi menggunakan Ubuntu Lammpp

2.5.2 Hasil Implimentasi pada Proyek kedua

Setelah selesai melakukan konfigurasi Dockerfile, Aplikasi Penyewaan Ruangan dan Alat Museum Sandi Yogyakarta sudah dapat berjalan. Untuk pembuktian apakah prosesnya sudah berjalan bisa dibuka di browser dengan cara ketikkan IP_VM:port (192.168.181.132:8000) dan dapat dilihat pada **Gambar 2.19** Berikut ini:



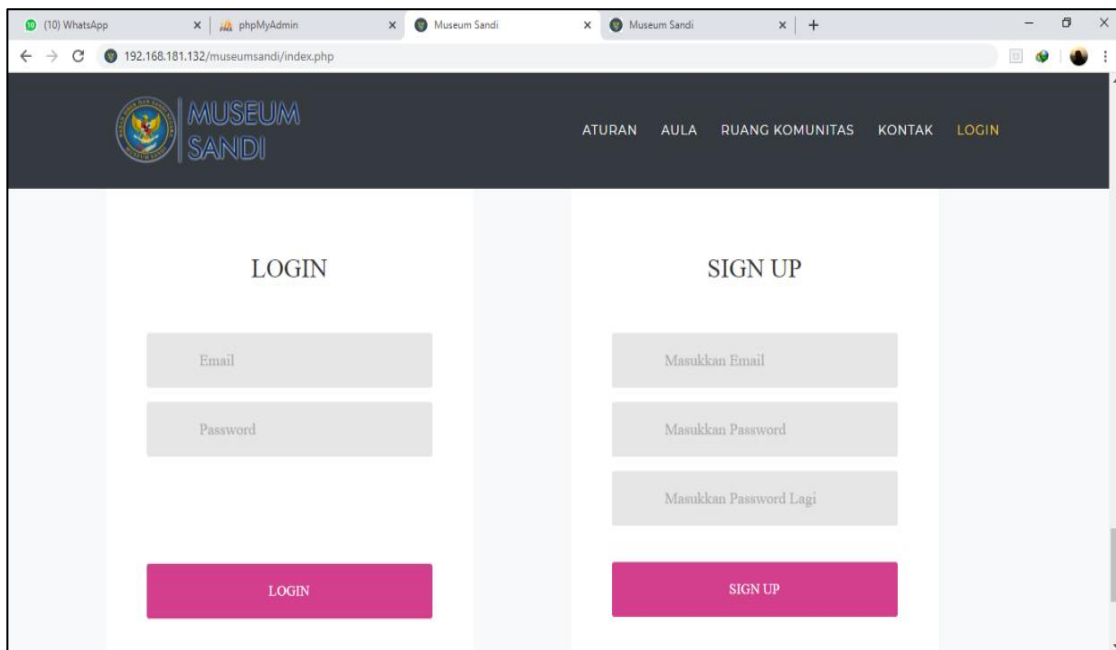
Gambar 2.19 Tampilan halaman utama aplikasi IP_VM:port

2.6 Pengujian Singkat

2.6.1 Pengujian Singkat pada Proyek Pertama

Pada aplikasi penyewaan yang telah dibuat, langkah awalnya adalah *user* atau pengguna melakukan login terlebih dahulu, akan tetapi jika *User* belum mempunyai akun,

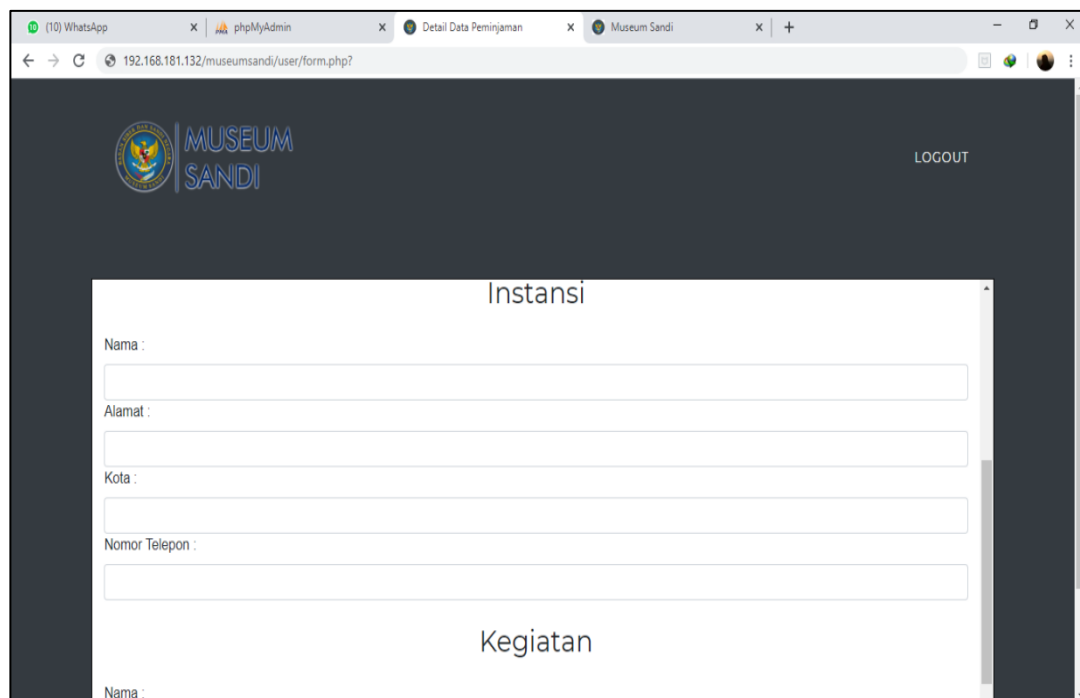
dapat melakukan *Sign Up* terlebih dahulu seperti terlihat pada **Gambar 2.20** berikut ini:



The screenshot shows a web browser window with the URL `192.168.181.132/museumssandi/index.php`. The page features a dark header with the Museum Sandi logo and navigation links: ATURAN, AULA, RUANG KOMUNITAS, KONTAK, and LOGIN. The main content area is divided into two columns. The left column is titled 'LOGIN' and contains two input fields labeled 'Email' and 'Password', followed by a pink 'LOGIN' button. The right column is titled 'SIGN UP' and contains three input fields labeled 'Masukkan Email', 'Masukkan Password', and 'Masukkan Password Lagi', followed by a pink 'SIGN UP' button.

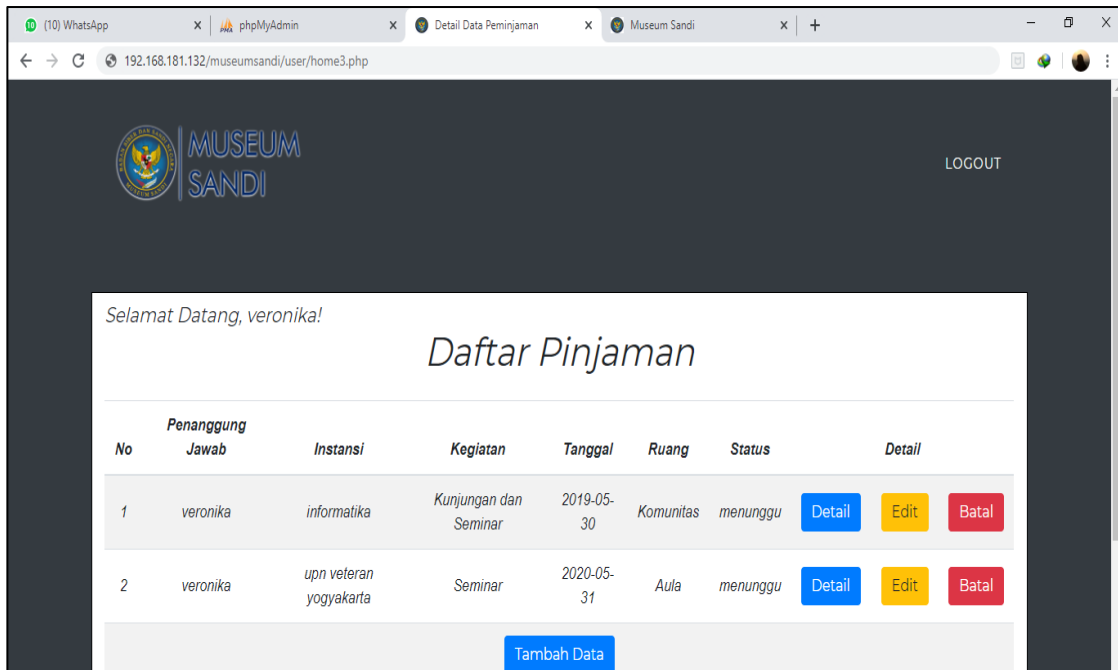
Gambar 2.20 Tampilan Login dan Sign Up

Apabila sudah memiliki akun, *user* bisa login menggunakan email/username dan password yang sudah dibuat serta melakukan proses selanjutnya yaitu melakukan pengisian data penyewaan seperti nama instansi dan jenis kegiatan seperti terlihat pada **Gambar 2.21** dan **Gambar 2.22** berikut ini:



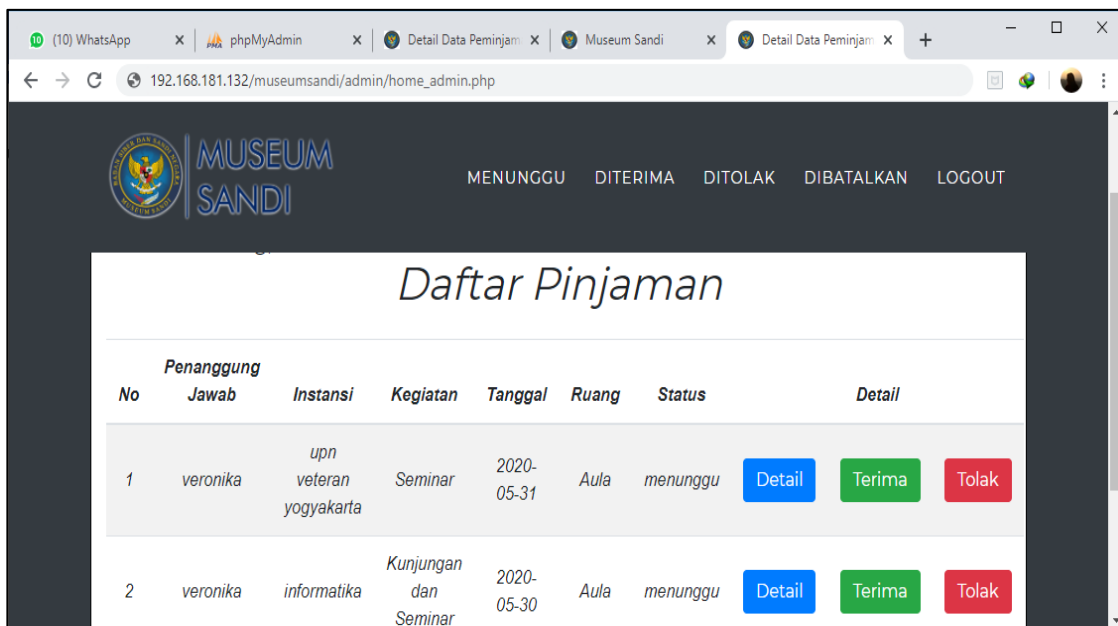
The screenshot shows a web browser window with the URL `192.168.181.132/museumssandi/user/form.php?`. The page features a dark header with the Museum Sandi logo and a 'LOGOUT' link. The main content area is divided into two sections. The top section is titled 'Instansi' and contains four input fields labeled 'Nama :', 'Alamat :', 'Kota :', and 'Nomor Telepon :'. The bottom section is titled 'Kegiatan' and contains one input field labeled 'Nama :'. The form is set against a dark background.

Gambar 2.21 Tampilan pengisian data penyewaan



Gambar 2.22 Tampilan daftar pinjaman/penyewaan

Bagian admin bertugas sebagai pemberi persetujuan apakah kegiatan diterima ataupun ditolak. Untuk tampilannya dapat dilihat pada **Gambar 2.23** berikut ini:



Gambar 2.23 Tampilan daftar pinjaman yang butuh konfirmasi pada bagian admin

Tahap selanjutnya adalah penyewa menunggu konfirmasi pihak museum sandi apakah kegiatan diijinkan atau ditolak. Untuk pemberituannya akan diberikan pada bagian status. Berikut tampilan kegiatan diterima oleh pihak museum dapat dilihat pada **Gambar 2.24** berikut ini:

No	Penanggung Jawab	Instansi	Kegiatan	Tanggal	Ruang	Status	Detail
1	veronika	upn veteran yogyakarta	Seminar	2020-05-31	Aula	diterima	Detail Edit Batal
2	veronika	informatika	Kunjungan dan Seminar	2020-05-30	Aula	menunggu	Detail Edit Batal

Gambar 2.24 Tampilan status kegiatan diterima

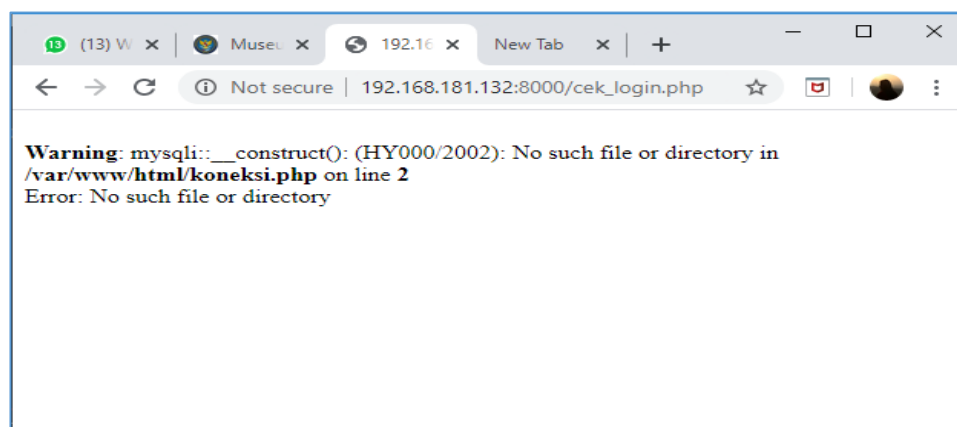
2.6.2 Pengujian Singkat pada Proyek Kedua

Hasil pengujian pada proyek kedua adalah pembuatan Dockerfilenya dapat dilihat pada **Gambar 2.25** berikut ini:



Gambar 2.25 Tampilan halaman utama aplikasi

Pada tampilan halaman selanjutnya, tidak dapat membaca database dari phpmyadmin seperti terlihat pada **Gambar 2.26** berikut ini:



Gambar 2.26 Tampilan halaman Error membaca database

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.	Pernancangan Program								
4.	Pembuatan Database								
5.	Pengerjaan Proyek Pertama (Ubuntu LAMPP)								
6.	Submisi Progress Laporan Pertama								
7.	Pengerjaan Proyek Kedua (Dockerfile)								
8.	Submisi Progress Laporan Kedua dan Revisi								
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	Veronika dan Nur
2.	Pengujian Singkat	Veronika
3.	Latar Belakang Masalah	Nur
4.	Agenda Pengerjaan Proyek	Nur
5.	Pengerjaan Proyek Pertama (Ubuntu LAMPP)	Veronika
6.	Pengerjaan Proyek Kedua (Dockerfile)	Veronika
7.	Pembuatan Laporan	Veronika dan Nur

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan masalah dari pembuatan aplikasi penyewaan yang sudah dibuat didapatkan hasil yang baik karena sistem dapat berjalan dengan lancar dan sangat membantu dari pihak admin maupun penyewa. Dari hasil dari pengujian diketahui bahwa semua sistem dapat berjalan dengan baik sebagaimana mestinya yakni bertujuan membantu penyewa agar lebih mudah dalam melakukan penyewaan.

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

4.2 Saran

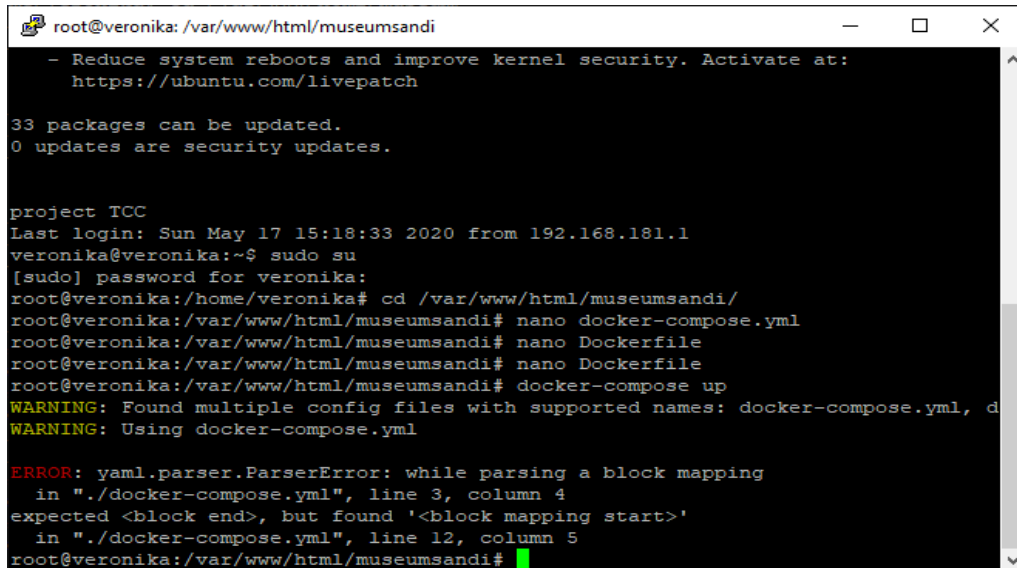
Pada pengerjaan projek ini sebaiknya RAM yang digunakan lebih besar agar proses pengerjaan tidak lama dan sering terjadi *not responding* terutama saat membuka PuTTY, serta pembagian tugas yang tidak merata mengakibatkan proses pengerjaan sedikit memakan waktu lebih lama. Selain itu, masih terdapat kesulitan dalam pengaplikasian Docker karena sedikitnya materi yang didapat dari perkuliahan karena sistem daring dan juga sumber informasi yang kurang lengkap.

DAFTAR PUSTAKA

- _____, < https://id.wikipedia.org/wiki/Komputasi_awan> (accessed, 3 Februari 2016)
- _____, <
<https://blog.lintasarta.net/article/industry-solutions/sejarah-perkembangan-cloud-computing/>>
(accessed, 3 Februari 2016)
- _____, <<https://www.ilmupengetahuan.co.id/linux-sejarah-ubuntu/>> (accessed ,21
Januari 2020)
- _____, <<https://teknobos.com/pengertian-vmware/>> (accessed, 15 November 2017)
- _____,
<[https://idcloudhost.com/mengenal-apa-itu-docker-definisi-fungsi-keunggulan-dan-cara-kerja
nya/](https://idcloudhost.com/mengenal-apa-itu-docker-definisi-fungsi-keunggulan-dan-cara-kerjanya/)> (accessed, 3 Februari 2020)

LAMPIRAN

Terdapat Error pada pengaplikasian docker compose bahkan sudah mencoba pada dua file program yang berbeda seperti terlihat pada **Gambar 2.27** berikut ini:



```
root@veronika: /var/www/html/museumsandi
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

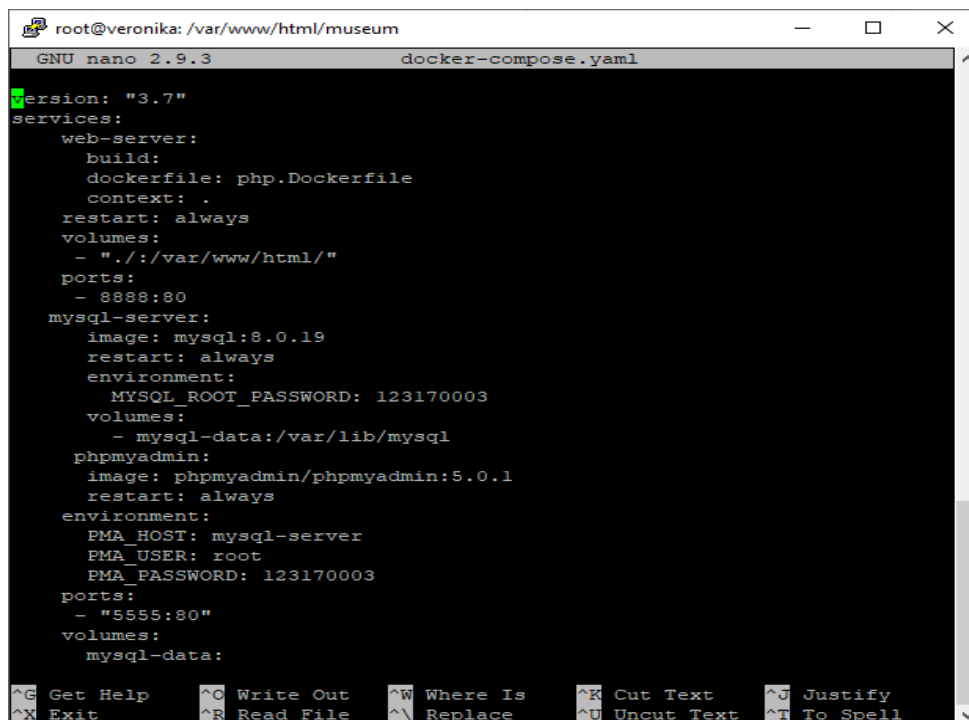
33 packages can be updated.
0 updates are security updates.

project TCC
Last login: Sun May 17 15:18:33 2020 from 192.168.181.1
veronika@veronika:~$ sudo su
[sudo] password for veronika:
root@veronika:/home/veronika# cd /var/www/html/museumsandi/
root@veronika:/var/www/html/museumsandi# nano docker-compose.yml
root@veronika:/var/www/html/museumsandi# nano Dockerfile
root@veronika:/var/www/html/museumsandi# nano Dockerfile
root@veronika:/var/www/html/museumsandi# docker-compose up
WARNING: Found multiple config files with supported names: docker-compose.yml, d
WARNING: Using docker-compose.yml

ERROR: yaml.parser.ParserError: while parsing a block mapping
  in "/docker-compose.yml", line 3, column 4
  expected <block end>, but found '<block mapping start>'
  in "/docker-compose.yml", line 12, column 5
root@veronika:/var/www/html/museumsandi#
```

Gambar 2.27 Tampilan halaman Error pada docker-compose

Pada percobaan selanjutnya agar docker compose dapat berjalan, kami mencoba membuat folder baru dengan nama museum dengan berkas berisi php.Dockerfile dan docker-compose.yml yang isinya berbeda dengan berkas sebelumnya. Namun juga masih terdapat kesalahan atau Error pada docker seperti terlihat pada **Gambar 2.28** dan **Gambar 2.29** berikut ini:



```
GNU nano 2.9.3 docker-compose.yml
version: "3.7"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
    context: .
    restart: always
  volumes:
    - "/./:/var/www/html/"
  ports:
    - 8888:80
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: 123170003
    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: 123170003
    ports:
      - "5555:80"
    volumes:
      mysql-data:

^G Get Help      ^O Write Out    ^W Where Is     ^R Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

Gambar 2.28 Isi berkas docker-compose.yml

```

root@veronika: /var/www/html/museum
cek_session.php      css      home.html      index.html  k ^
root@veronika:/var/www/html/museum# docker-compose up -d
ERROR: yaml.parser.ParserError: while parsing a block mapping
  in "/docker-compose.yaml", line 1, column 1
expected <block end>, but found '<block mapping start>'
  in "/docker-compose.yaml", line 12, column 4
root@veronika:/var/www/html/museum# nano docker-compose.yaml
root@veronika:/var/www/html/museum# docker-compose up -d
ERROR: yaml.parser.ParserError: while parsing a block mapping
  in "/docker-compose.yaml", line 1, column 1
expected <block end>, but found '<block mapping start>'
  in "/docker-compose.yaml", line 12, column 4
root@veronika:/var/www/html/museum# nano docker-compose.yaml
root@veronika:/var/www/html/museum# docker-compose up -d
ERROR: yaml.parser.ParserError: while parsing a block mapping
  in "/docker-compose.yaml", line 1, column 1
expected <block end>, but found '<block mapping start>'
  in "/docker-compose.yaml", line 12, column 4
root@veronika:/var/www/html/museum# nano docker-compose.yaml
Use "fg" to return to nano.

[1]+  Stopped                  nano docker-compose.yaml
root@veronika:/var/www/html/museum# nano docker-compose.yaml
root@veronika:/var/www/html/museum#

```

Gambar 2.29 Tampilan halaman Error pada docker-compose

