

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR

SISTEM INFORMASI FIST EFFECT BAGIAN TRAINING MENGGUNAKAN
UBUNTU LAMPP DAN DOCKER



DISUSUN OLEH:

NAMA ANGGOTA : M. AZAM FUADI 123170059
AYU NOVIRA S. 123170073
KELAS : A
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.KOM
MUHAMMAD IMAM ALFATAH

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

HALAMAN PENGESAHAN

SISTEM INFORMASI FIST EFFECT BAGIAN TRAINING MENGGUNAKAN UBUNTU LAMPP DAN DOCKER

Disusun oleh :

Muhamad Azam Fuadi

123170059

Ayu Novira Shinta P.

123170073

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan proyek akhir praktikum Teknologi *Cloud computing* yang berjudul Sistem Informasi Fist Effect Bagian Training menggunakan Ubuntu Lampp dan Manajemen Penyimpanan Data Training Fist Effect pada Freenas.

Laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung dan juga proyek pembuatan aplikasi saat melakukan kerja praktik. Proyek akhir ini berupa pembuatan sistem informasi berbasis *web* yang di-*hosting* menggunakan Ubuntu LAMPP.

Tidak lupa ucapan terimakasih kepada asisten dosen yang selalu membimbing dan mengajari saya dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun saya harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, saya ucapkan terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Wassalamualaikum Wr. Wb.

Yogyakarta, 1 April 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	3
1.4 Tahap Penyelesaian Proyek Akhir	3
BAB II ISI DAN PEMBAHASAN	5
2.1 Komponen yang Digunakan	5
2.2 Rancangan Arsitektur <i>Cloud Computing</i>	8
2.3 Parameter dan Konfigurasi	10
2.4 Tahap Implementasi	14
2.5 Hasil Implementasi	36
2.6 Pengujian Singkat	36
BAB III JADWAL Pengerjaan dan Pembagian Tugas	40
3.1 Agenda Pengerjaan	40
3.2 Keterangan Pembagian Tugas	40
BAB IV KESIMPULAN DAN SARAN	41
4.1 Kesimpulan	41
4.2 Saran	41
DAFTAR PUSTAKA	42

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Cloud computing merupakan gabungan pemanfaatan teknologi komputer (komputasi) dalam suatu jaringan dengan pengembangan berbasis internet (awan) yang mempunyai fungsi untuk menjalankan program atau aplikasi melalui komputer – komputer yang terkoneksi pada waktu yang sama, tetapi tak semua yang terkoneksi melalui internet menggunakan cloud computing (Puspitasari, D., 2020).

Kemunculan *cloud computing* ini dimulai dengan adanya gagasan mengenai sebuah jaringan computing yang kemudian akan menjadi infrastuktur untuk *public*. Kemudian sejak saat itulah kemudian *cloud computing* atau sistem komputasi awan ini mulai dikembangkan dan berjalan secara seiringan dengan perkembangan web dan internet. Hingga akhirnya pada saat ini, *cloud computing* telah digunakan dalam berbagai macam bidang dengan banyaknya manfaat yang didapatkan, seperti ruang penyimpanan yang lebih banyak, *worldwide access*, fleksibel dengan dapat diakses dari mana dan kapan saja, dan juga dapat menghemat biaya.

Berdasarkan kelebihan-kelebihan yang didapatkan dari *cloud computing*, maka dibangunlah sebuah sistem informasi berbasis web yang memiliki beberapa bagian pelayanan informasi sesuai dengan jasa yang disediakan oleh Fist Effect. Bagian – bagian tersebut dimuat dalam *sub-menu* masing – masing sehingga dalam pengembangannya tiap bagian dapat dikembangkan oleh *developer* yang berbeda yang nantinya akan digabungkan menjadi satu. Melihat banyaknya bagian pelayanan yang disediakan oleh Fist Effect dan mengingat sedikitnya jumlah karyawan Fist Effect karena statusnya yang masih *start-up* akan sangat sulit untuk mengumpulkan dan melakukan *tracking* terhadap layanan-layanan yang disediakan oleh Fist Effect. Hal ini tentunya akan menyulitkan admin dalam pembuatan laporan dan pengelolaan sumber daya serta akan menyulitkan pimpinan maupun mentor dalam melakukan *tracking*.

Solusi dari permasalahan itu ialah dibuatnya suatu sistem informasi yang memuat segala keperluan Fist Effect mulai dari pusat informasi, registrasi, manajemen layanan, laporan layanan, hingga penggajian ke dalam satu *website*. *Cloud computing* di sini sangat berperan terhadap pembuatan sistem informasi berbasis web tersebut, karena dengan *cloud computing* web tersebut dapat diakses oleh semua orang tentunya dengan *privilegenya* masing – masing. Dengan adanya sistem informasi dan bantuan *cloud computing*, admin

dapat mengelola segala layanan dengan mudah kapan pun dan di mana pun tentunya hanya dengan mengakses *web* tersebut admin sudah dapat mengakses segala layanan yang ada. Pimpinan, tentor, maupun analis dapat melakukan pengawasan dan kontrol dari mana saja dan kapan saja. *Client* dapat mengakses informasi lengkap mengenai Fist Effect dan juga melakukan registrasi tanpa harus datang ke Fist Effect terlebih dahulu. *Cloud computing* tidak hanya membantu setelah sistem jadi saja, tetapi dalam pembuatan dan pengembangan sistem pun *cloud computing* berkontribusi dalam menyediakan *platform* agar tiap *developer* dapat bekerja di tempatnya dan bagiannya masing – masing tanpa mengenal waktu dan tanpa peduli masalah komunikasi, sebagai contoh yaitu penggunaan “git” atau “github” dalam pembangunan dan pengembangan sistem tersebut.

Dalam mencapai pembangunan sistem tersebut, setiap *developer* dapat menyelesaikan bagiannya masing-masing terlebih dahulu dan dipastikan bahwa sistem yang dibangun sudah dapat berjalan dengan baik. Apabila setiap *developer* telah menyelesaikan bagiannya, maka selanjutnya akan dilakukan penggabungan untuk membuat Sistem Informasi Fist Effect secara utuh dan kemudian barulah sistem dapat di-*hosting*.

Untuk dapat melakukan *hosting*, dibutuhkan tempat dimana sistem tersebut akan ditempatkan, untuk itu kami menggunakan server Ubuntu LAMPP dan Docker. LAMPP sendiri merupakan *platform* pengembangan Web *open source* yang menggunakan Linux sebagai sistem operasi, Apache sebagai server Web, MySQL sebagai sistem manajemen basis data relasional dan PHP sebagai bahasa scripting berorientasi objek (Elang N., 2018). Sedangkan Docker yaitu *platform* terbuka bagi pengembang perangkat lunak dan pengelola sistem jaringan untuk membangun, mengirimkan dan menjalankan aplikasi-aplikasi terdistribusi (Adiputra, F., 2015).

Kami juga menggunakan VMware Workstation untuk meng-*install* Ubuntu sebagai *virtual OS* sekaligus untuk menjalankan LAMPP dan Docker tersebut. Selain itu, dalam pembangunan system ini hardware yang kami gunakan adalah sebuah laptop dengan processor Intel i5 dan RAM 8 GB dan MSI GS63 dengan prosessor intel core i-7 dengan RAM sebesar 8GB.

1.2 Tujuan Proyek Akhir

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

1. Mengimplementasikan arsitektur *cloud computing* untuk Sistem Informasi Fist Effect yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04 dan LAMPP (Apache 8, PHP 7.2, MySQL 5.2).
2. Mengintegrasikan Sistem Informasi Fist Effect yang berada di Ubuntu Server (LAMPP) dengan *database online* MYSQL yang disambungkan melalui IP HOST 153.92.10.57.
3. Membuat LAMP menggunakan Docker yang dijalankan pada virtual OS Ubuntu 18.04 untuk dapat mengakses Sistem Informasi Fist Effect.

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. Perusahaan tidak perlu memperlakukan *maintenance*, dikarenakan 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 terpilih dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.
5. Segala fitur layanan perusahaan terutama layanan *training* dapat diakses oleh semua orang di dalam satu *website* saja, walaupun setiap pengguna memiliki *previlegenya* masing – masing.

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari Sistem Informasi Fist Effect terutama bagian *training* untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan LAMPP dan Docker pada *virtual OS* Ubuntu.
2. Menentukan konfigurasi yang tepat untuk pengaturan LAMPP pada Ubuntu sehingga dapat digunakan sesuai *requirement* yang berupa:

- a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Dapat terhubung dengan *database* MYSQL dengan mudah.
 - c. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
3. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda yaitu *web* dan *database* sehingga dapat digunakan secara terintegrasi.
4. Menentukan konfigurasi yang tepat untuk pengaturan LAMP menggunakan Docker pada Ubuntu sehingga dapat digunakan sesuai *requirement* yang berupa:
 - a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Dapat terinstall di mana saja tanpa mempengaruhi sistem.
 - c. Dapat terhubung dengan *database online* MYSQL dengan mudah.
 - d. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
5. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu integrasi *database* dan fungsionalitas dari sistem.
6. Menghasilkan Sistem Informasi First Effect Bagian Training yang berbasis *cloud computing*.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Proyek akhir ini dikerjakan menggunakan laptop MSI GS63 7RD dengan *processor* Intel Core-i7 dan RAM sebesar 8gb dan juga ASUS dengan *processor* Interl Core-i5 dengan RAM sebesar 8gb sebagai hardware yang digunakan. Proyek ini juga menggunakan virtualisasi OS Ubuntu yang dijalankan menggunakan VMWare Workstation versi 15.5.2 yang dapat didownload di <https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html> . Versi ini tersedia dalam versi *free* dan *pro*, untuk menggunakan versi *pro* diperlukan untuk memasukkan beberapa kode yang perlu dibeli atau melakukan *crack* terlebih dahulu, sehingga pada proyek akhir ini menggunakan versi *free* yang mempengaruhi tidak bisanya melakukan setting jaringan dalam pembuatan *virtual OS*.

Proyek akhir ini menggunakan Putty versi 0.73 yang dapat didownload di <https://putty.org/> , *software* Putty ini digunakan untuk melakukan *remote OS* Ubuntu dengan memasukkan IP dari OS Ubuntu tersebut. Selain Putty, *software* lain yang digunakan untuk melakukan *remote OS* Ubuntu ini adalah WinSCP versi 5.17.4 yang dapat didownload di <https://winscp.net/eng/download.php> . Jika Putty bisa digunakan untuk *remote OS* menggunakan CLI, WinSCP bisa digunakan untuk *remote* file-file yang ada di dalam OS virtual.

2.1.1. Komponen Pada Proyek Pertama

Untuk membangun “Sistem Informasi Fict Effect Menggunakan Ubuntu LAMP” yang berbasis konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang dibangun menggunakan *Virtual OS* Ubuntu versi 18.04. LTS
2. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.3 dan bahasa HTML dengan versi minimal 4.0.
3. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data dari Sistem Informasi Fict Effect Tersebut.

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

Tabel 2.1 Spesifikasi VM *cloud computing* untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Vmware Workstation 15 Player	15.5.2	Keterangan versi yang digunakan.
2.	VMware Workstation 15 Player	15785246	Keterangan <i>build (patch) number</i> .
3.	IPv4 <i>ens33 Ubuntu Hypervisor</i>	IP inet: 192.168.253.129	IP <i>number</i> untuk akses manajemen <i>ubuntu OS</i> melalui <i>remote</i> maupun <i>web base</i> .
		SM: 255.255.255.0	Kelas IP/ <i>subnet mask</i> yang digunakan.
		Broadcast: 192.168.253.255	Alamat IP untuk fungsionalitas broadcast.
4.	<i>Operating System info</i>	Ubuntu 18.04	OS yang digunakan pada virtual OS.
5.	<i>Processor info</i>	12 x Intel (R) Core (TM) i7-8700 CPU @ 4.2GHz	Jenis <i>processor</i> yang digunakan pada <i>Ubuntu</i> .
6.	<i>RAM info</i>	2 GiB Memory	Kapasitas RAM pada <i>Ubuntu</i> .

Virtual OS Ubuntu 18.04 yang dibuat pada VMWare Workstation player *remote* menggunakan Putty dengan parameter konfigurasi pada **Tabel 2.2** berikut ini:

Tabel 2.2 Konfigurasi *Remote* pada Putty yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	Putty	0.73	Keterangan versi yang digunakan.
2.	<i>Host Name / IP Address</i>	192.168.253.129	IP <i>number</i> untuk akses manajemen <i>ubuntu OS</i> melalui <i>remote</i> .
4.	<i>Port</i>	22	Nomor <i>port</i> yang digunakan untuk <i>remote</i> .
5.	<i>Connection Type</i>	SSH	Jenis koneksi yang dijalankan untuk <i>remote</i> virtual OS.

Virtual OS Ubuntu 18.04 yang dibuat pada VMWare Workstation player *remote* menggunakan WinSCP untuk mengakses file-file dengan parameter konfigurasi pada **Tabel 2.3** berikut ini:

Tabel 2.3 Konfigurasi *Remote* pada Putty yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	WinSCP	5.17.4	Keterangan versi yang digunakan.
2.	<i>Host Name / IP Address</i>	192.168.253.129	IP <i>number</i> untuk akses manajemen <i>ubuntu OS</i> melalui <i>remote</i> .
4.	<i>Port Number</i>	22	Nomor <i>port</i> yang digunakan untuk <i>remote</i> .
5.	<i>File Protocol</i>	SFTP	Jenis protocol yang dijalankan untuk akses file.
6.	<i>User Name</i>	Azam	<i>User name</i> dari virtual OS yang akan <i>remote</i>
7.	<i>Password</i>	123170059	<i>Password</i> dari virtual OS yang akan <i>remote</i>

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

Tabel 2.4 Spesifikasi Ubuntu OS untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Bahasa pemrograman yang digunakan untuk membangun web SI Fist Effect
		MYSQL	Basis data yang digunakan untuk menyimpan data SI Fist Effect
		PhpMyAdmin	<i>Software</i> yang digunakan untuk mengelola basis data MYSQL.

2.1.2. Komponen Pada Proyek Kedua

Proyek kedua memiliki komponen dan juga konfigurasi yang hampir sama dengan proyek pertama, yang membedakan adalah pada proyek kedua LAMP akan dijalankan menggunakan docker, untuk membuat proyek kedua ini maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang dibangun menggunakan *Virtual OS* Ubuntu versi 18.04. LTS
2. Sistem menggunakan docker engine versi 19.03.8 untuk menjalankan dockerfile . Selain docker engine, system ini juga menggunakan docker-compose versi 1.25.5.
3. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 8.0 sehingga dapat digunakan untuk menyimpan berbagai data dari Sistem Informasi Fist Effect Tersebut. MySQL yang digunakan dapat diremote online melalui Ip tertentu.

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

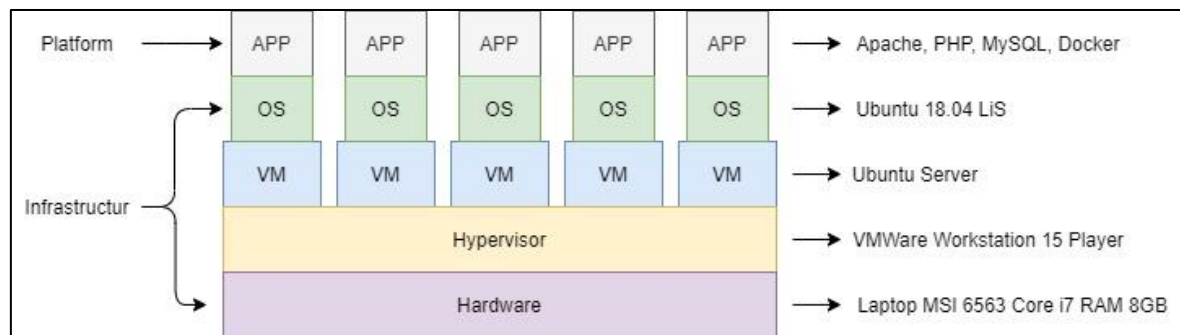
Tabel 2.4 Spesifikasi Ubuntu OS untuk proyek kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Docker Engine	Docker server 19.03.8	Docker engine server yang digunakan untuk menjalankan container
		Docker Client 19.03.8	<i>Docker engine client</i> yang digunakan untuk berinteraksi dengan docker engine server
2.	Docker Compose	Versi 1.25.5	Dukungan alat untuk menjalankan multi-container / images secara bersamaan.

3.	Dbeaver	Versi 7.0.4	Tambahan aplikasi untuk meremote database online
4.	IP Host MySQL	153.92.10.57	IP yang digunakan untuk melakukan remote database

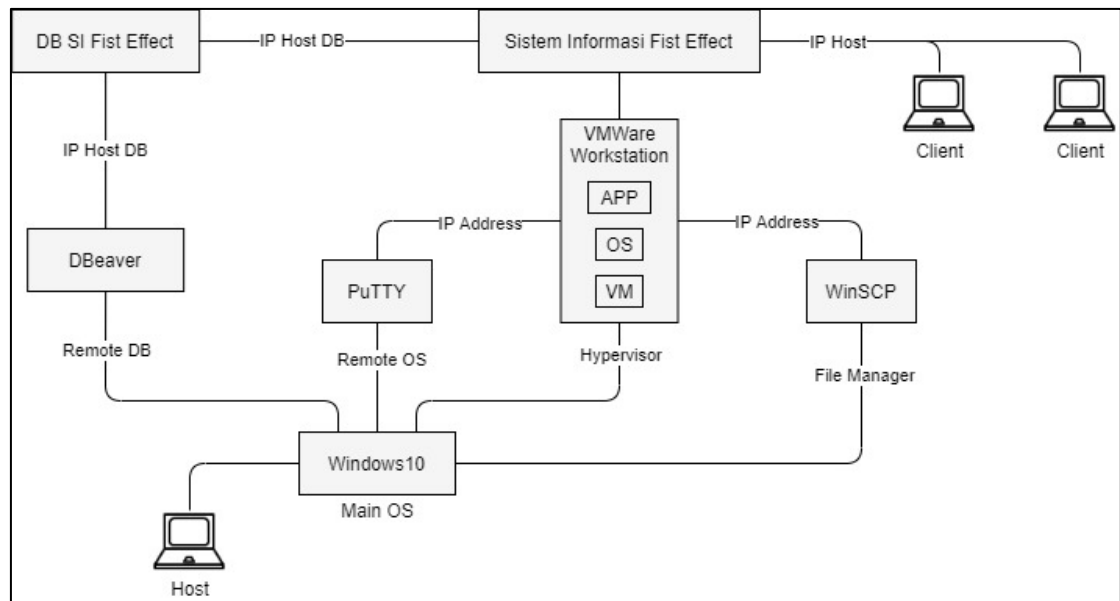
2.2 Rancangan Arsitektur *Cloud Computing*

Proyek akhir ini digunakan bentuk rancangan arsitektur IaaS di mana hardware laptop MSI GS63 sebagai layer utama di bagian bawah, kemudian di atasnya adalah PaaS atau sistem operasi Windows. Hypervisor yang digunakan adalah aplikasi VMware Workstation yang digunakan untuk menjalankan layer di atasnya yaitu *virtual machine* Ubuntu Server dengan virtual OS berupa Ubuntu 18.04 LTS. Virtual OS berupa Ubuntu 18.04 LTS kemudian digunakan untuk menjalankan beberapa aplikasi untuk melakukan *hosting* dengan komponen aplikasi yang digunakan adalah Apache, PHP, MySQL, dan PHPMyAdmin (LAMPP) yang dijalankan pada layer paling atas. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



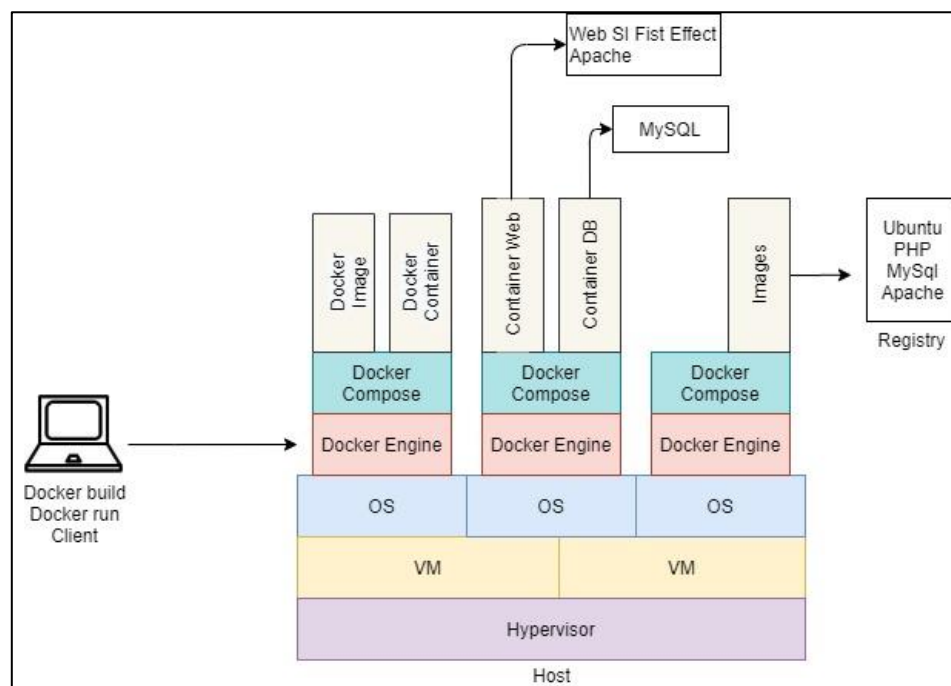
Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya

Arsitektur *cloud computing* di atas akan digunakan untuk menjalankan *web* Sistem Informasi Fist Effect yang dijalankan menggunakan bantuan dari Apache, PHP, dan MySQL. Proyek 1 dan 2 sama-sama dijalankan menggunakan VMWare Workstation 15 Player. Pada proyek 1, Sistem Informasi Fist Effect dijalankan menggunakan *hosting* local dengan bantuan LAMPP. Pada proyek 2, *hosting web* system informasi tersebut menggunakan bantuan docker untuk memuat apa saja yang diperlukan dalam menjalankan *web* tersebut. Ilustrasi mengenai pengerjaan proyek 1 dan 2 seperti yang tergambar pada **Gambar 2.2** berikut :



Gambar 2.2 Topologi Proyek 1 dan Proyek 2

Jika pada proyek 1 menggunakan arsitektur *cloud computing* seperti yang digambarkan oleh **Gambar 2.1**, pada proyek 2 menggunakan arsitektur yang sedikit berbeda karena menggunakan docker. Docker memuat *images* dan *container* yang masing-masing membawa aplikasi atau *registry* yang diperlukan untuk melakukan *hosting web* Sistem Informasi Fist Effect. Beberapa doocker *images* dan *container* dapat dijalankan bersamaan menggunakan *docker-compose*. Ilustrasi mengenai arsitektur docker dapat dilihat pada **Gambar 2.3** berikut :



Gambar 2.3 Topologi Proyek 1 dan Proyek 2

2.3 Parameter dan Konfigurasi

Proyek ini menggunakan virtual OS yang dijalankan menggunakan VMware. Parameter yang digunakan untuk instalasi Apache dan pengaturan *firewall* pada Apache dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ sudo apt install apache2
$ sudo ufw allow in "Apache Full"
```

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
- ufw : parameter untuk masuk ke pengaturan firewall
- allow in : parameter untuk mengizinkan firewall pada suatu paket aplikasi
- "Apache Full" : seluruh konfigurasi apache

Modul 2.1 Parameter instalasi Apache

Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan **Modul 2.2** berikut ini:

```
$ sudo apt install mysql-server
$ sudo 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-server : nama paket aplikasi untuk MySQL server
- mysql_secure_installation : parameter untuk memasuki pengaturan dasar pengamanan MySQL
- Y : untuk opsi "yes" atau "ya"
- N : untuk opsi "No" atau "tidak"

Modul 2.2 Parameter instalasi MySQL Server

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 : nama paket aplikasi untuk mod apache php
- php-mysql : nama paket aplikasi untuk php pada mysql

Modul 2.3 Parameter instalasi PHP

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

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

Keterangan:

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

Modul 2.4 Parameter instalasi PHPMyAdmin

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

```
$ sudo apt-get update
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-
key add -
$ sudo apt-key fingerprint 0EBFCD88
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : parameter tambahan pada apt untuk mengeksekusi perintah mengupdate package yang ada pada OS
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- curl : parameter yang digunakan untuk mengecek konektivitas URL dan sebagai tool transfer data dari URL tersebut
- -fsSL : parameter untuk akses SSL
- Key add - : parameter yang digunakan untuk menambahkan key
- fingerprint : jenis key yang digunakan
- 0EBFCD88 : nilai yang akan digunakan untuk input key fingerprint
- Add-apt-repository : parameter yang digunakan untuk menambahkan repository

Modul 2.5 Parameter instalasi Docker Repository

Parameter yang digunakan untuk instalasi Docker Engine setelah menginstall Docker Repository dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : parameter tambahan pada apt untuk mengeksekusi perintah mengupdate package yang ada pada OS
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- docker-ce : nama paket aplikasi untuk docker engine
- docker-ce-cli : nama paket ekstensi cli pada docker engine
- containerd.io : nama paket aplikasi untuk container docker

Modul 2.6 Parameter instalasi Docker

Setelah berhasil menginstall Docker Engine, hal yang diperlukan selanjutnya adalah Docker Compose. Parameter yang digunakan untuk download Docker Compose setelah menginstall Docker Engine dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- curl : parameter yang digunakan untuk mengecek konektivitas URL dan sebagai tool transfer data dari URL tersebut.
- [https://github.com/docker/compose/releases/download/1.24.1/docker-compose-\\$\(uname -s\)-\\$\(uname -m\)](https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)) : Link untuk download docker compose
- /usr/local/bin/docker-compose : direktori download

Modul 2.7 Parameter Download Docker Compose

Parameter yang digunakan untuk mengeksekusi instalasi Docker Compose setelah mendownloadnya dapat dilihat pada penjelasan **Modul 2.8** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- chmod : merupakan fungsi untuk mengatur hak akses suatu file
- +x : hak akses untuk mengeksekusi file dari suatu direktori
- /usr/local/bin/docker-compose : direktori file yang akan dieksekusi

Modul 2.8 Parameter instalasi Docker Compose

Setelah berhasil install docker compose, selanjutnya adalah mengubah direktori ke folder /var/www/html/fist_ci yang merupakan tempat dimana file web code iginiter sistem informasi fist effect berada . Parameter konfigurasinya dapat dilihat pada penjelasan **Modul 2.9** berikut ini:

```
$ cd /var/www/html/fist_ci
```

Keterangan:

- cd : perintah untuk berpindah direktori

- `/var/www/html/fist_ci` : direktori tujuan tempat file web codeigniter berada

Modul 2.9 Parameter berpindah direktori

Setelah berpindah direktori, yang dilakukan selanjutnya adalah membuat dockerfile yang berisi instalasi php, apache dan hal-hal yang dibutuhkan untuk *hosting web*. Parameter konfigurasi untuk membuat dockerfile dapat dilihat pada penjelasan **Modul 2.10** berikut ini:

```
$ nano Dockerfile
```

Keterangan:

- `nano` : perintah untuk membuka teks editor pada linux
- `Dockerfile` : teks editor yang dibuka merupakan Dockerfile

Modul 2.10 Parameter membuat dockerfile

Parameter konfigurasi untuk membuat file docker-compose.yml dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

```
$ nano docker-compose.yml
```

Keterangan:

- `nano` : perintah untuk membuka teks editor pada linux
- `docker-compose.yml` : teks editor yang dibuka merupakan docker-compose.yml

Modul 2.11 Parameter membuat docker-compose.yml

Parameter konfigurasi untuk mengeksekusi file docker-compose.yml dapat dilihat pada penjelasan **Modul 2.12** berikut ini:

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

Keterangan:

- `docker-compose up -d` : perintah yang digunakan untuk mengeksekusi docker compose yang ada pada direktori tersebut

Modul 2.12 Parameter membuat docker-compose.yml

Parameter konfigurasi untuk menjalankan dan menghentikan file docker-compose dapat dilihat pada penjelasan **Modul 2.13** berikut ini:

```
$ docker-compose start
$ docker-compose stop
```

Keterangan:

- `docker-compose start` : perintah yang digunakan untuk memulai docker compose yang ada pada direktori tersebut
- `docker-compose stop` : perintah yang digunakan untuk menghentikan docker compose yang ada pada direktori tersebut

Modul 2.13 Parameter menjalankan atau menghentikan docker-compose

Parameter konfigurasi untuk membuka .htaccess dapat dilihat pada penjelasan **Modul 2.14** berikut ini:

```
$ nano .htaccess
```

Keterangan:

- `nano` : perintah untuk membuka teks editor pada linux
- `.htaccess` : teks editor untuk .htaccess

Modul 2.14 Parameter membuat docker-compose.yml

2.4 Tahap Implementasi

2.4.1. Tahap Implementasi Proyek 1

Tahap implementasi proyek 1 merupakan tahap dimana melakukan *hosting web* menggunakan LAMPP pada Ubuntu Server versi 18.04 LTS.

2.4.1.1 Instalasi Linux OS

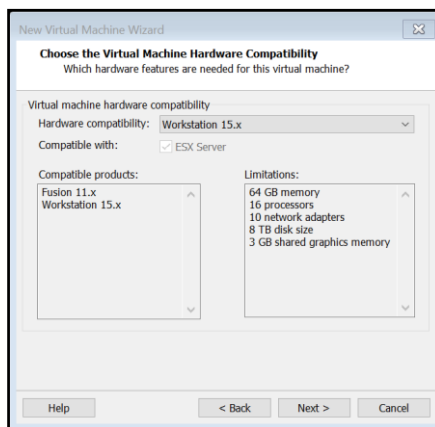
Tahap-tahap untuk instalasi Linux OS adalah sebagai berikut:

1. Siapkan terlebih dahulu ISO yang akan digunakan, yaitu Ubuntu-18.04
2. Jalankan aplikasi VMWare Workstation, lalu pilih mode “Custom”



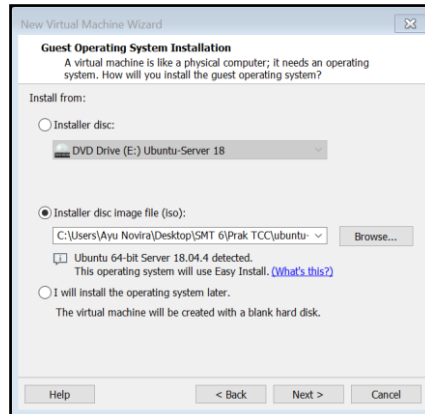
Gambar 2.3 Type of configuration

3. Tentukan komabilitas dari OS yang akan dibuat. Pilihlah kompalitas hardware tertinggi untuk OS VM Modern.



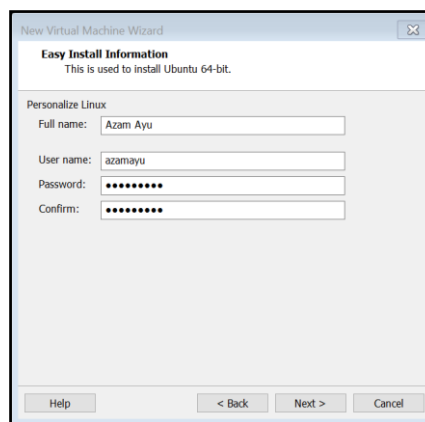
Gambar 2.4 Hardware Compability

4. Pilih berkas ISO untuk VM yang akan dibuat, yaitu berkas ISO yang telah disiapkan sebelumnya.



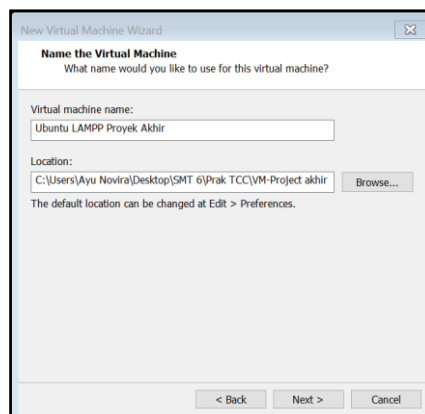
Gambar 2.5 Operating System Installation

5. Lengkapi data “Install Information”.



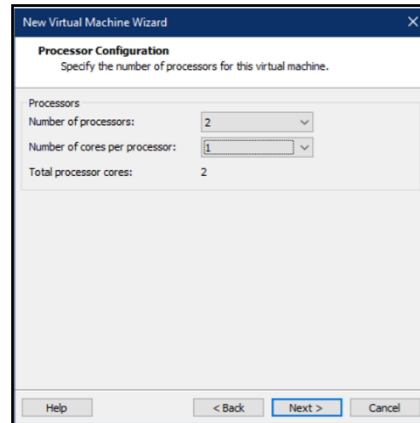
Gambar 2.6 Instal Information

6. Beri nama untuk VM yang akan dibuat. Buatlah folder baru untuk mengelompokkan file VM dalam satu folder yang sama.



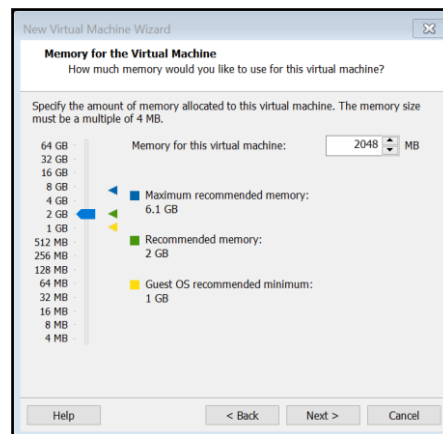
Gambar 2.7 Name of Virtual Machine

7. Gunakan parameter *processors* sebanyak dua (2) dan *cores per processors* tetap satu (1) agar dapat mempercepat kinerja VM tersebut.



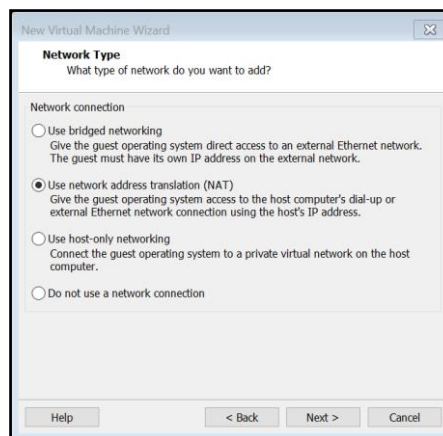
Gambar 2.8 Processor Configuration

8. Gunakan parameter *memory* sebanyak 2GB atau 2048MB.



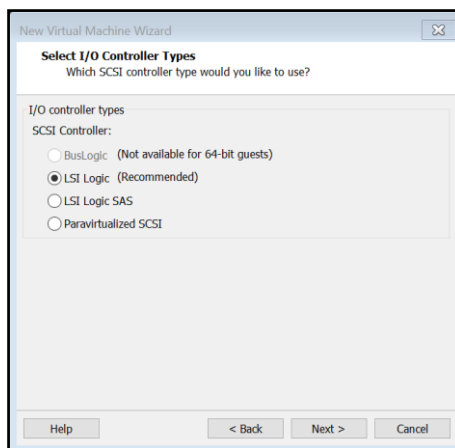
Gambar 2.9 Set Memory for the VM

9. Pilih *network connection* “Network Address Translation (NAT)”.



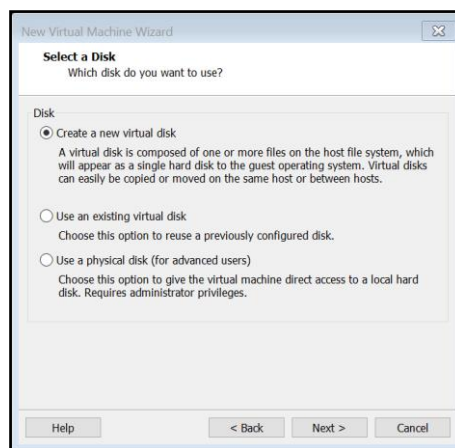
Gambar 2.10 Network Type

10. Pilih *I/O controller type* “LSI Logic (Recommended)”.



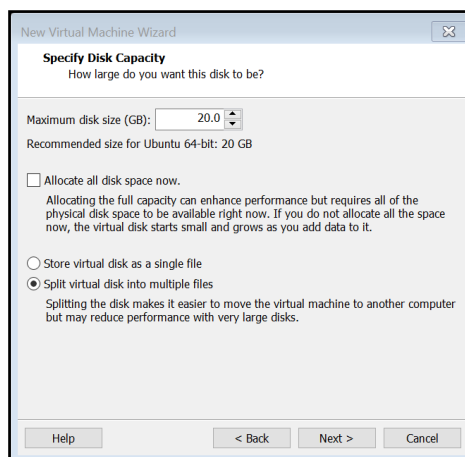
Gambar 2.11 I/O Controller Types

11. Pilih *virtual disk type* “SCSI (Recommended)”.



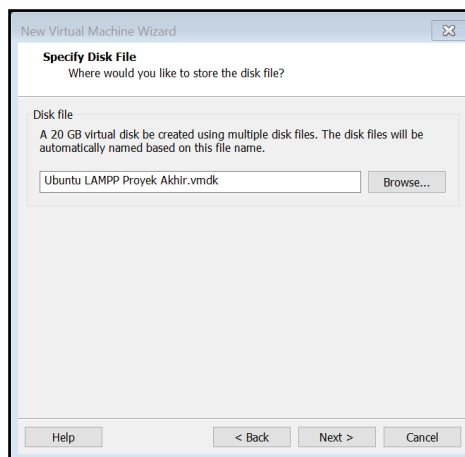
Gambar 2.12 Disk for VM

12. Pada pembuatan VM pertama kali, pilih “Create a new virtual disk”.



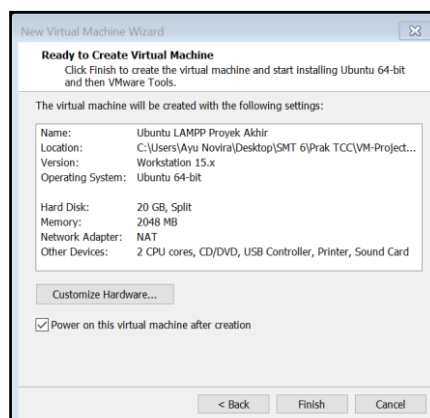
Gambar 2.13 Specify Disk Capacity

13. Gunakan ukuran default yang disarankan oleh Ubuntu yaitu 20GB. Lalu pilih “Split virtual disk into multiple files”.



Gambar 2.14 Specify Disk File

14. Jika sudah sesuai, klik “Next”.
15. Selanjutnya akan ditampilkan konfirmasi terakhir, apabila sudah sesuai dengan yang telah dipilih, beri centang pada “Power on this virtual machine after creation” dan klik “Finish”.

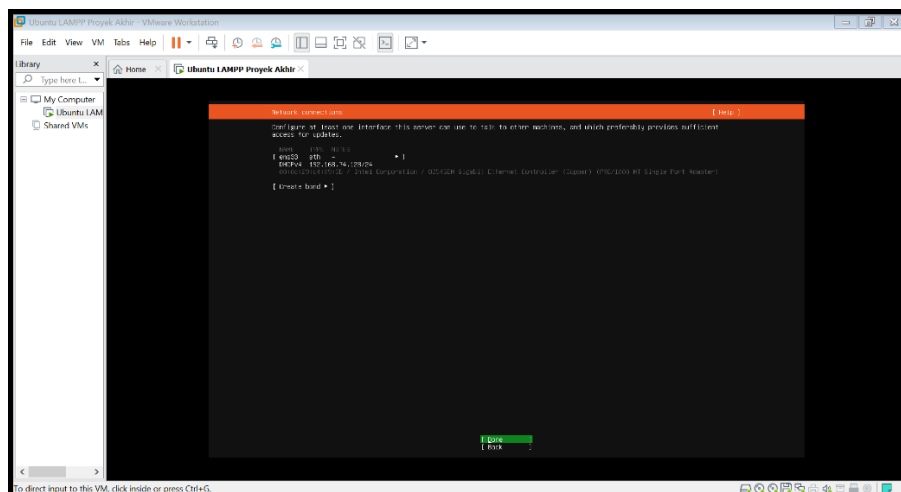


Gambar 2.15 Confirmation to Install

2.4.1.2 Instalasi Ubuntu Server

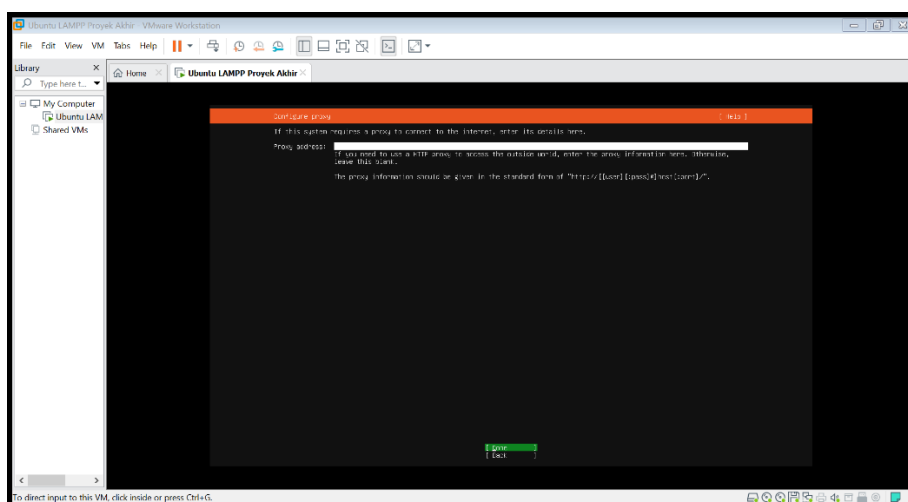
Setelah proses instalisasi Linux OS selesai, selanjutnya akan dijalankan VM yang telah dibuat. Selanjutnya adalah proses instalisasi Ubuntu Server, dimana untuk dapat berinteraksi dengan VM dapat dilakukan dengan meng-*klik window* hitam yang kemudian kursor akan menghilang dan interaksi dijalankan menggunakan keyboard. Sebaliknya, jika ingin menyudahi interaksi tersebut dapat dilakukan dengan mengetikkan “CTRL+Alt”. Dalam proses instalisasi dibutuhkan koneksi jaringan internet, berikut merupakan tahapan-tahapan untuk instalisasi Ubuntu Server :

1. Tentukan Bahasa yang akan digunakan.



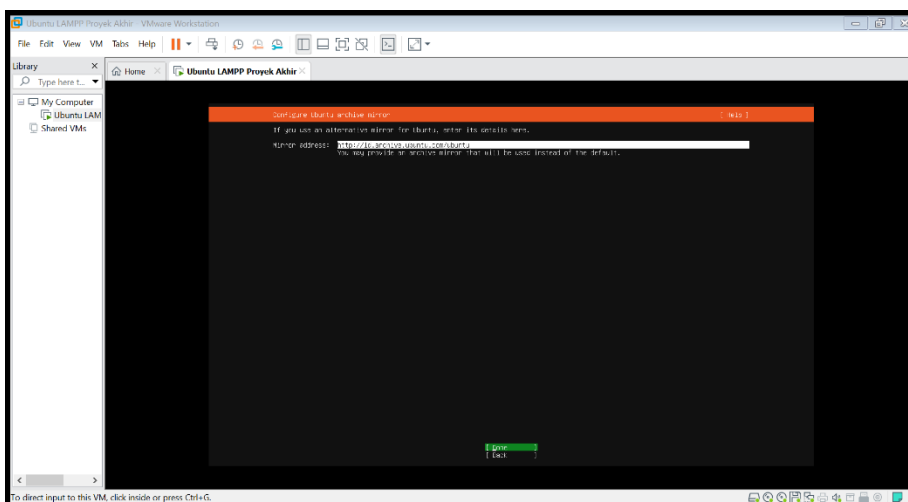
Gambar 2.18 Network Connection

5. Karena *setting* tidak memerlukan *Proxy* maka tahap ini dapat dilewati.



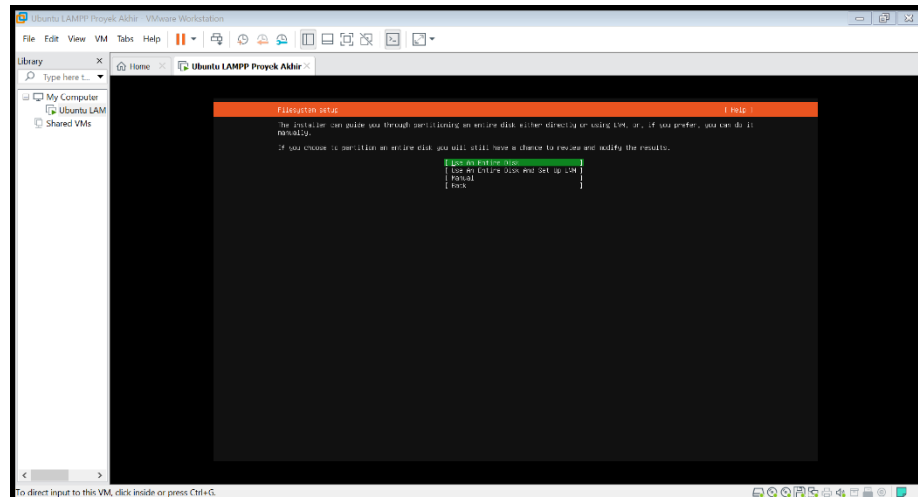
Gambar 2.19 Konfigurasi Proxy

6. Selanjutnya untuk *setting* Mirror address dapat dibiarkan saja dan pilih “Done”.



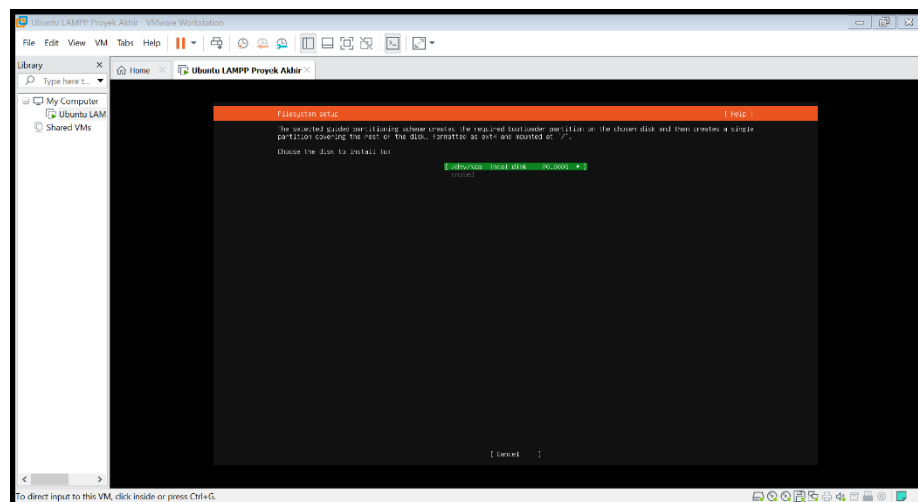
Gambar 2.20 Konfigurasi Ubuntu Archive Mirror

7. Pilih “Use An Entire Disk”.



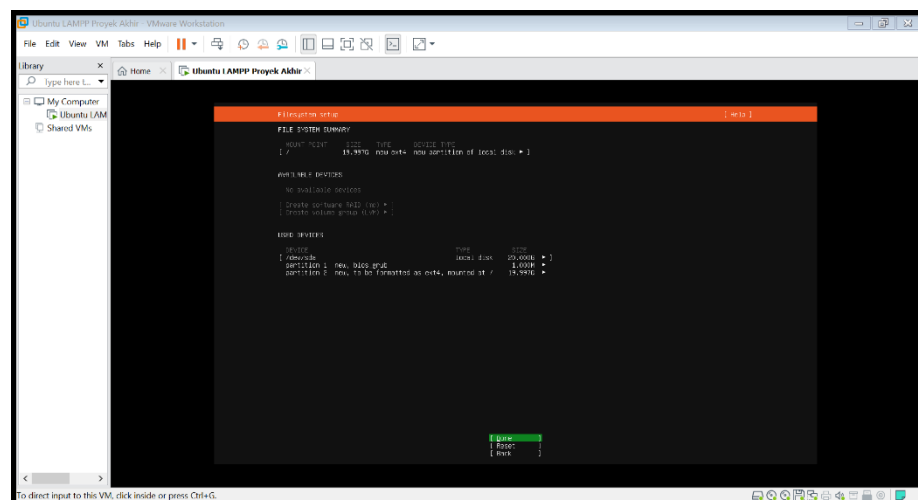
Gambar 2.21 Disk Setup

8. Pilih lokasi hardisk tujuan untuk instalasi Ubuntu Server.



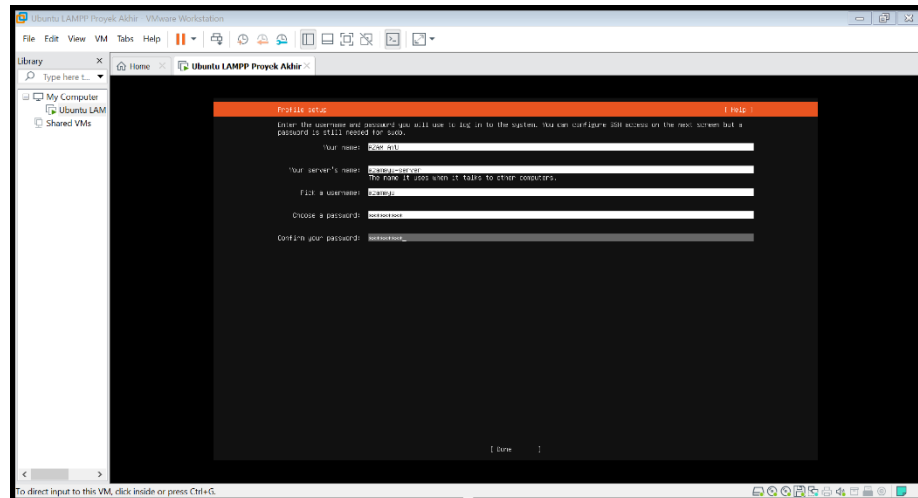
Gambar 2.22 Konfigurasi disk tujuan

9. Selanjutnya akan ditampilkan konfirmasi mengenai *Filesystem setup*. Jika sudah sesuai, pilih “Done”.



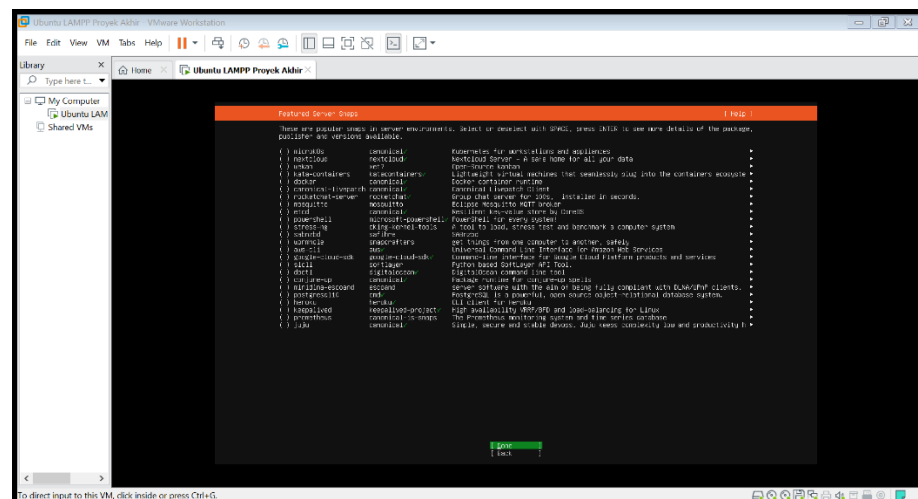
Gambar 2.23 Konfirmasi Filesystem Setup

10. Isi identitas profil yang sesuai, ingatlah *username* dan *password*-nya.



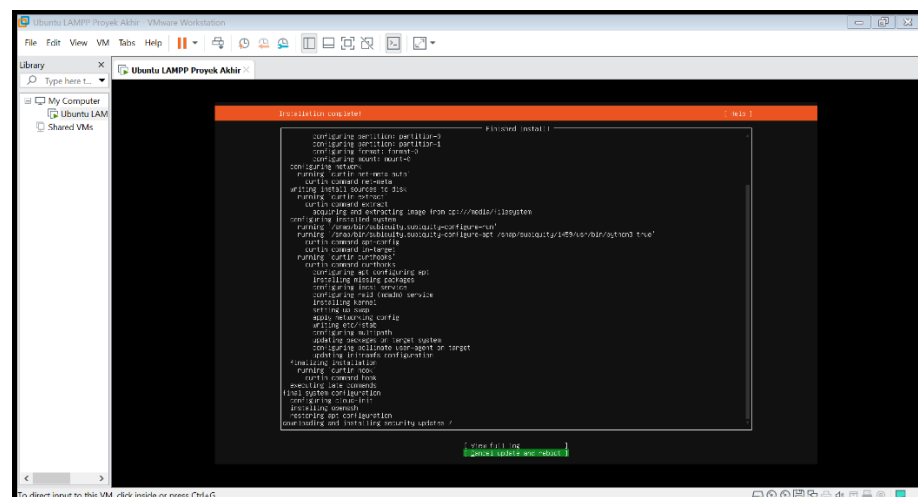
Gambar 2.24 Profile Setup

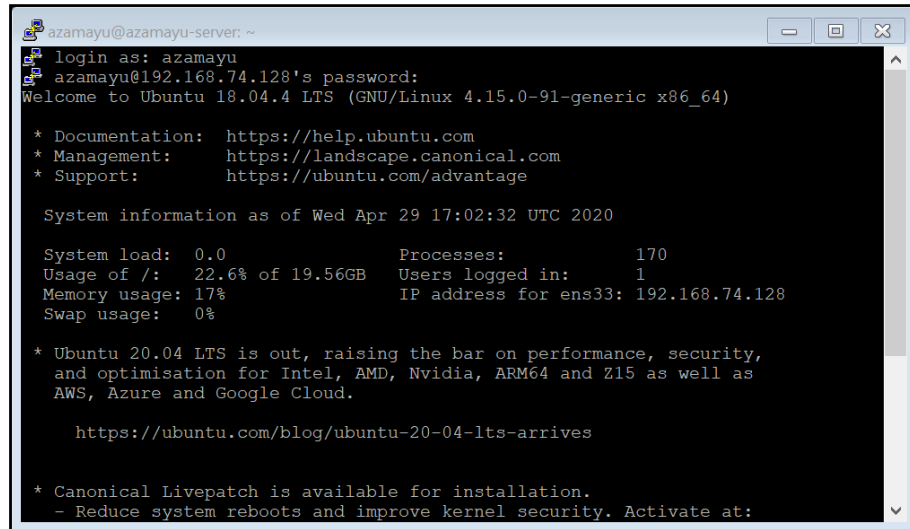
11. Karena tidak memerlukan aplikasi tambahan, maka tahap ini dapat dilewati saja dan pilih “Done”.



Gambar 2.25 Setup aplikasi tambahan

12. Jika sudah selesai terinstall, maka pilih “Reboot”.





```

azamayu@azamayu-server: ~
login as: azamayu
azamayu@192.168.74.128's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

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

System information as of Wed Apr 29 17:02:32 UTC 2020

System load:  0.0               Processes:    170
Usage of /:   22.6% of 19.56GB   Users logged in: 1
Memory usage: 17%              IP address for ens33: 192.168.74.128
Swap usage:   0%

 * Ubuntu 20.04 LTS is out, raising the bar on performance, security,
   and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
   AWS, Azure and Google Cloud.

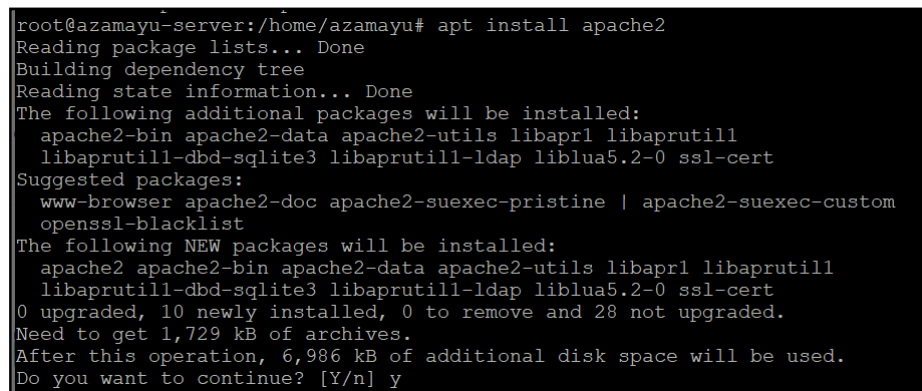
   https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:

```

Gambar 2.29 Login dalam PuTTY

3. Login sebagai “root” dengan mengetikkan *syntax* “`sudo su`” dan masukkan *password*.
4. Install Apache2 dengan mengetikkan *syntax* “`apt install apache2`”. Pada konfirmasi masukkan “Y”. Seperti pada gambar berikut.



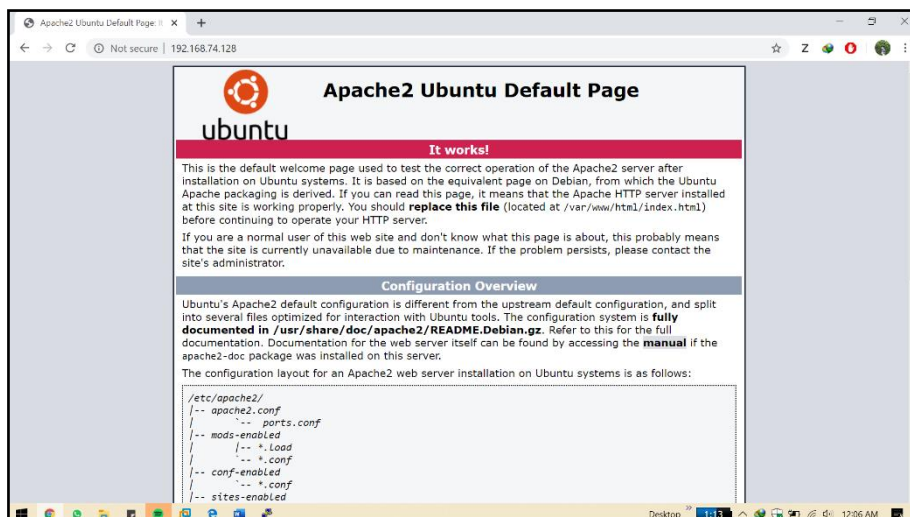
```

root@azamayu-server:/home/azamayu# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 28 not upgraded.
Need to get 1,729 kB of archives.
After this operation, 6,986 kB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

Gambar 2.30 Instalasi Apache2

5. Untuk memastikan bahwa Apache2 telah terinstall, dapat dicek dengan memasukkan IP pada browser.



Gambar 2.31 Cek hasil instalasi Apache2

6. Install MySQL-Server dengan mengetikkan *syntax* “`apt install mysql-server`”. Pada konfirmasi masukkan “Y”. Seperti pada gambar berikut.

```
root@azamayu-server:/home/azamayu# apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl
  mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server-5.7
  mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl
  mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server
  mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 28 not upgraded.
Need to get 19.7 MB of archives.
After this operation, 156 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Gambar 2.32 Instalasi MySQL-Server

7. Install PHP dengan mengetikkan *syntax* “`apt install php libapache2-mod-php php-mysql`”. Pada konfirmasi masukkan “Y”. Seperti pada gambar berikut.

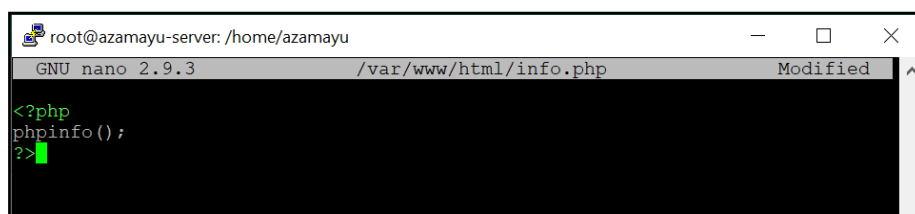
```

root@azamayu-server:/home/azamayu# apt install php libapache2-mod-php php-mys
ql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli
  php7.2-common php7.2-json php7.2-mysql php7.2-openssl php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 libsodium23 php php-common
  php-mysql php7.2 php7.2-cli php7.2-common php7.2-json php7.2-mysql
  php7.2-openssl php7.2-readline
0 upgraded, 13 newly installed, 0 to remove and 28 not upgraded.
Need to get 4,132 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

Gambar 2.33 Instalasi PHP

8. Selanjutnya untuk memastikan bahwa PHP telah terinstall, buatlah file yang berisi phpinfo dengan *syntax* “`sudo nano /var/www/html/info.php`” dan akan masuk ke dalam GNU nano. Pada GNU nano, berikan *syntax* PHP yang dapat menampilkan phpinfo.



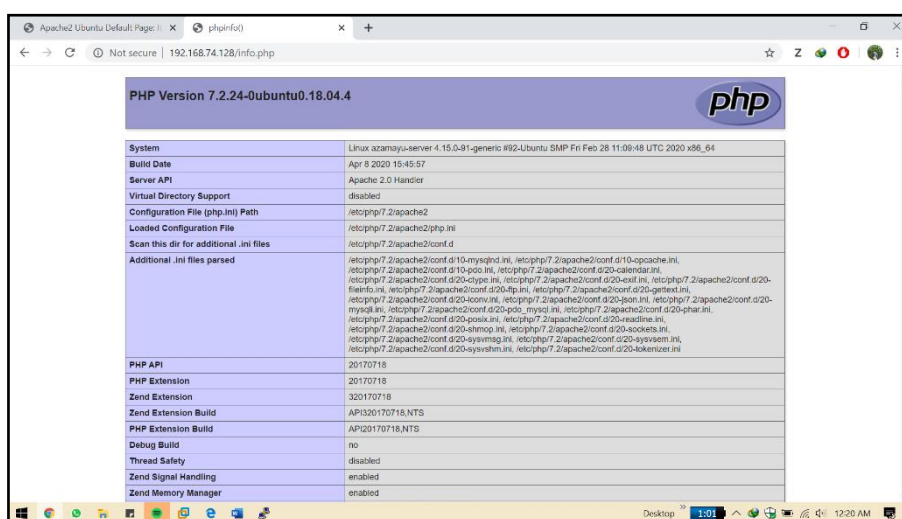
```

GNU nano 2.9.3 /var/www/html/info.php Modified
<?php
phpinfo();
?>

```

Gambar 2.34 Membuat file info.php

Setelah itu, masukkan “IP/info.php” pada browser dan akan ditampilkan phpinfo tersebut.



Gambar 2.35 Cek hasil instalasi PHP

9. Install phpmyadmin dengan mengetikkan *syntax* “`apt install phpMyAdmin php-mbstring php-gettext`”. Pada konfirmasi masukkan “Y”. Seperti pada gambar berikut.

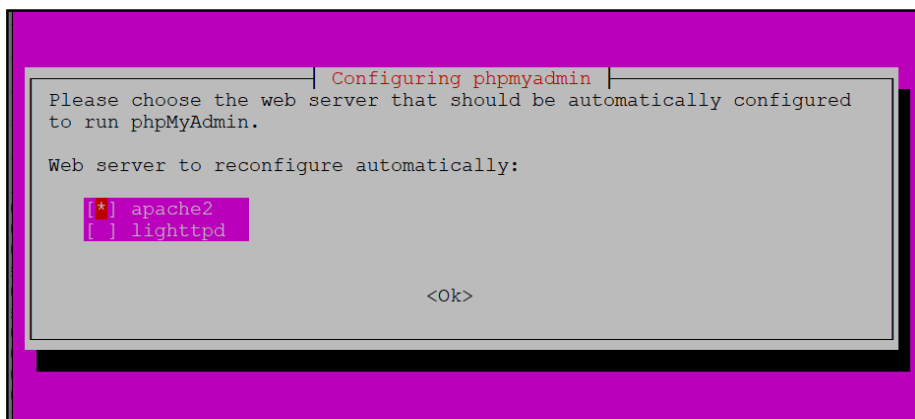
```

root@azamayu-server:/home/azamayu# apt install phpmyadmin php-mbstring php-gd
ttext
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core
  javascript-common libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8
  libjs-jquery libjs-sphinxdoc libjs-underscore libtiff5 libwebp6 libxpm4
  libzip4 php-bz2 php-curl php-gd php-pear php-php-gettext php-phpseclib
  php-tcpdf php-xml php-zip php7.2-bz2 php7.2-curl php7.2-gd
  php7.2-mbstring php7.2-xml php7.2-zip
Suggested packages:
  libgd-tools php-libsodium php-mcrypt php-gmp php-imagick www-browser
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core
  javascript-common libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8
  libjs-jquery libjs-sphinxdoc libjs-underscore libtiff5 libwebp6 libxpm4
  libzip4 php-bz2 php-curl php-gd php-gettext php-mbstring php-pear
  php-php-gettext php-phpseclib php-tcpdf php-xml php-zip php7.2-bz2
  php7.2-curl php7.2-gd php7.2-mbstring php7.2-xml php7.2-zip phpmyadmin
0 upgraded, 35 newly installed, 0 to remove and 28 not upgraded.
Need to get 15.6 MB of archives.
After this operation, 60.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

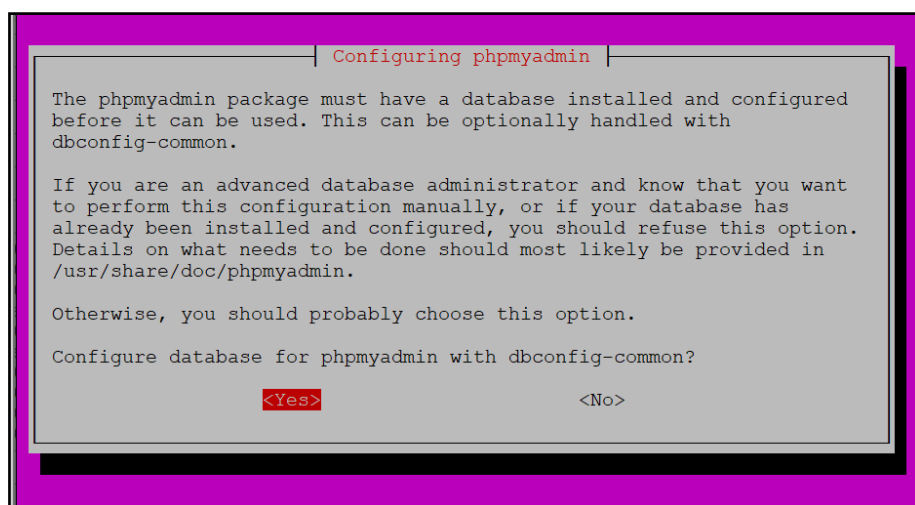
Gambar 2.36 Instalasi phpMyAdmin

10. Selanjutnya pilih “apache2” pada konfigurasi phpMyAdmin.



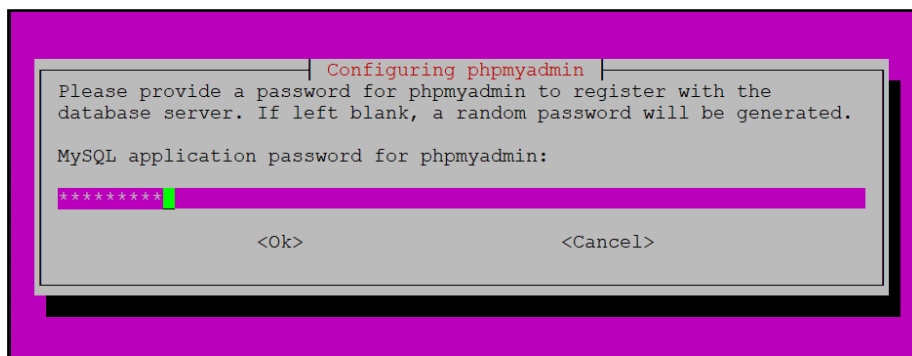
Gambar 2.37 Konfigurasi phpMyAdmin

11. Pilih “Yes” pada konfirmasi *database* untuk phpMyAdmin.



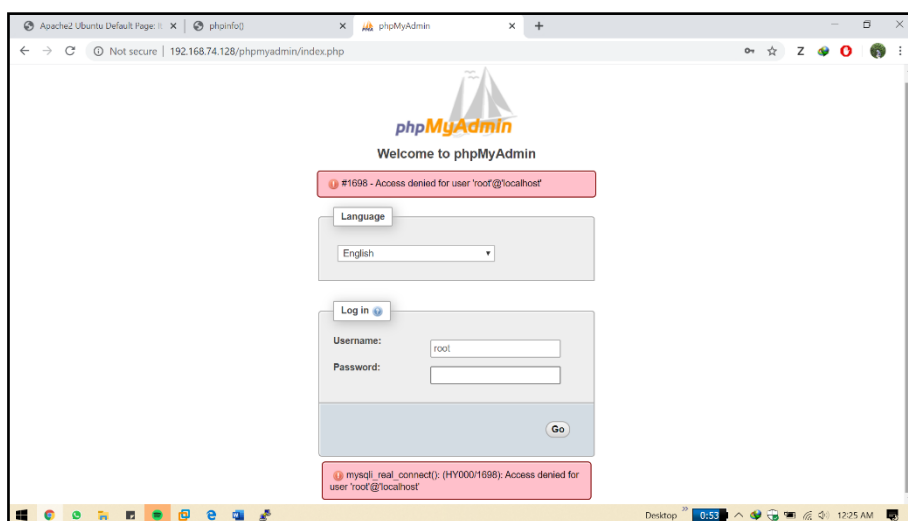
Gambar 2.38 Konfigurasi database untuk phpMyAdmin

12. Buatlah *password* untuk phpMyAdmin.



Gambar 2.39 Buat *password* untuk phpmyadmin

13. Untuk memastikan bahwa phpMyAdmin telah terinstal, ketikkan “IP/phpmyadmin” pada browser. Setelah itu cobalah *login* dengan *username* “root” dan *password* yang telah dibuat.



Gambar 2.40 Login pada phpmyadmin

14. Pada proses login tersebut akan ditemukan kegagalan karena *password* yang tidak sesuai, untuk mengatasinya dapat kembali ke PuTTY dan masuk ke akun root dengan memasukkan *syntax* “`sudo mysql -u root`”. Setelah itu lakukan update terhadap *password* yang telah kita buat dengan memasukkan *syntax* “`UPDATE mysql.user SET plugin = 'mysql_native_password', authentication_string = PASSWORD('password_anda') WHERE User = 'root';`”. Setelah itu masukkan *syntax* “`FLUSH PRIVILEGES;`”, maka *password* telah berhasil diubah dan dapat keluar dari akun root dengan mengetikkan “exit” seperti berikut.


```
mysql> UPDATE mysql.user SET plugin = 'mysql_native_password', authentication_string = PASSWORD('kasihamas') WHERE User = 'root';
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 1

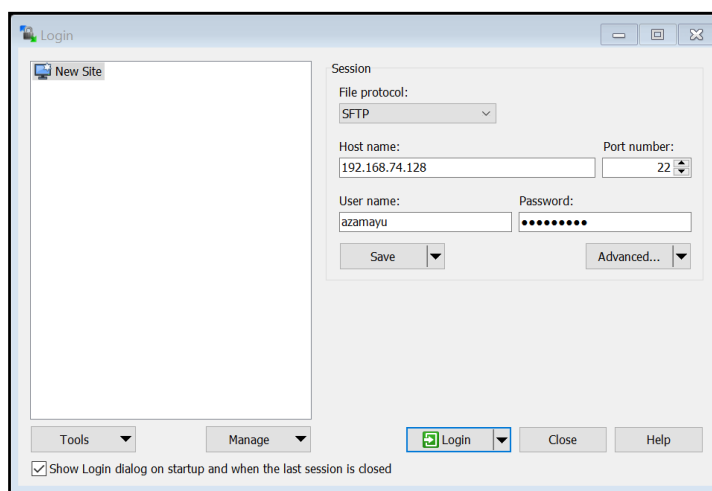
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
root@azamayu-server:/home/azamayu#
```

Gambar 2.41 Update *password* untuk phpmyadmin

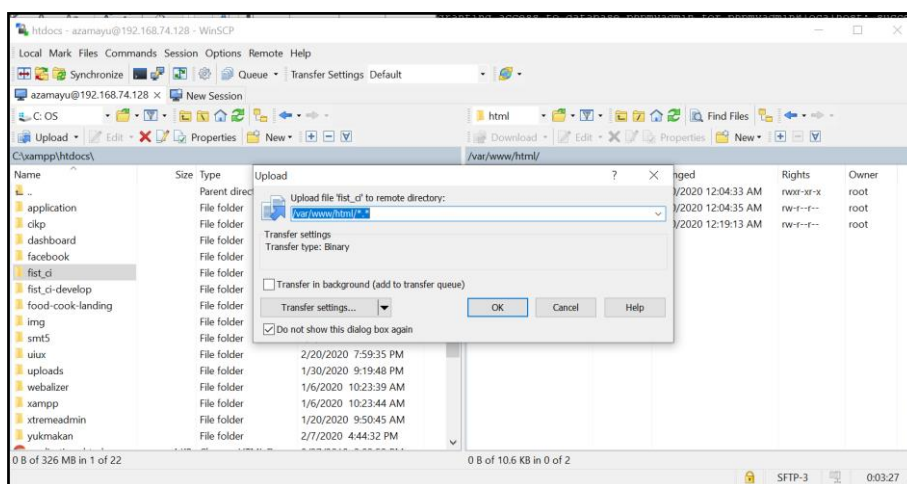
Selanjutnya dapat dicek dengan *login* kembali pada phpMyAdmin dengan menggunakan *password* tersebut.

15. Untuk dapat mengunduh file, jalankan aplikasi WinSCP dan masukkan *Host name*, *username*, dan *password* VM.



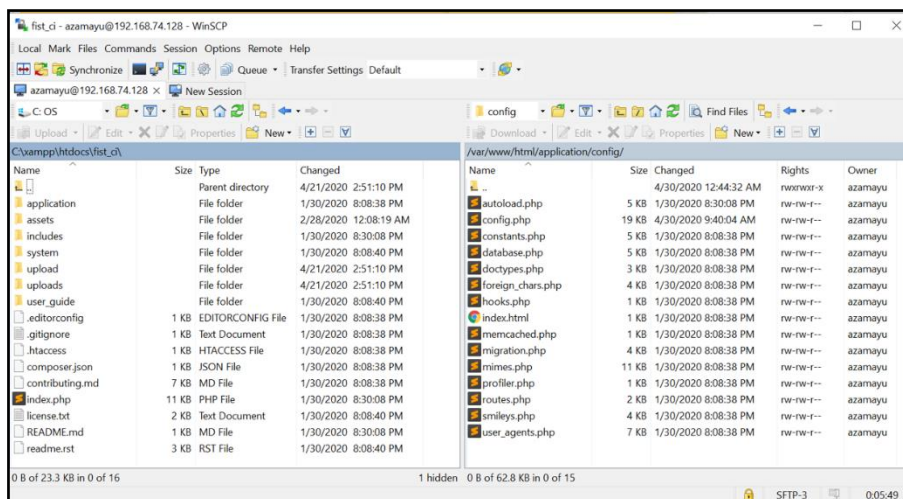
Gambar 2.42 Login pada WinSCP

16. Unduh file yang akan di-*hosting* dengan cara *men-drag and drop* file tersebut dari *window* sebelah kiri WinSCP ke *window* sebelah kanan.



Gambar 2.43 Mengunduh file yang akan di-*hosting*

17. File berhasil diunduh dan dapat diakses melalui browser.



Gambar 2.44 File berhasil diunduh

2.4.2. Tahap Implementasi Proyek 2

Pada proyek 2, *hosting web* menggunakan LAMPP yang menggunakan docker dimulai dengan instalasi docker engine terlebih dahulu. Untuk menginstall Docker engine, diperlukan beberapa *package* pada linux yang nantinya akan membantu proses instalasi Docker Engine. Parameter instalasi secara detail dapat dilihat pada **Modul 2.5** yang telah dijelaskan sebelumnya, kemudian jika ada pertanyaan maka jawab dengan parameter “Y”. Selain menginstall package yang dibutuhkan, pada modul tersebut juga dijelaskan mengenai parameter yang digunakan untuk mengakses link download docker engine beserta *key* untuk mengaksesnya. Hasil dari parameter yang dimasukkan pada **Modul 2.5** dapat dilihat pada **Gambar 2.45** hingga **Gambar 2.47** berikut :

```

azam@aza-server: ~
Package 'docker.io' is not installed, so not removed
Package 'runc' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 36 not upgraded.
azam@aza-server:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
0 upgraded, 2 newly installed, 0 to remove and 36 not upgraded.
Need to get 6,572 B of archives.
After this operation, 196 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

Gambar 2.45 Instalasi package yang dibutuhkan untuk install docker engine

```

azam@aza-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt
-key add -
OK
azam@aza-server:~$ sudo apt-key fingerprint 0EBFCD88
pub  rsa4096 2017-02-22 [SCEA]
    9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid      [ unknown] Docker Release (CE deb) <docker@docker.com>
sub  rsa4096 2017-02-22 [S]
azam@aza-server:~$

```

Gambar 2.46 Konfigurasi key yang dibutuhkan untuk akses repository docker engine

```

azam@aza-server:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"

```

Gambar 2.47 Penambahan repository docker engine untuk instalasi

Setelah dapat mengakses repository dari docker engine yang akan diinstal, langkah selanjutnya adalah melakukan instalasi docker-engine yang dapat dilakukan menggunakan parameter dijelaskan pada **Modul 2.6** sebelumnya. Hasil dari parameter tersebut dapat dilihat pada **Gambar 2.48** yang kemudian pada pilihan *continue* masukkan parameter “Y” untuk lanjut seperti gambar berikut :

```

azam@aza-server:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount libltdl7 pigz
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli libltdl7 pigz
0 upgraded, 7 newly installed, 0 to remove and 36 not upgraded.
Need to get 85.8 MB of archives.
After this operation, 385 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

Gambar 2.48 Proses instalasi docker engine

Setelah docker engine berhasil diinstall, Langkah selanjutnya adalah menginstall docker compose yang akan digunakan untuk menjalankan beberapa images maupun container maupun dockerfile secara bersamaan. Parameter yang digunakan untuk instalasi docker compose dapat dilihat pada **Modul 2.7** hingga **Modul 2.8** yang hasilnya akan terlihat pada **Gambar 2.49** hingga **Gambar 2.50** berikut :

```

azam@aza-server:~$ sudo curl -L "https://github.com/docker/compose/releases/down
load/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-com
pose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 638      100 638    0     0  1052      0  --:--:-- --:--:-- --:--:--  1051
100 16.7M  100 16.7M    0     0  91617      0  0:03:11 0:03:11 --:--:-- 84571
azam@aza-server:~$

```

Gambar 2.49 Proses mengakses link instalasi docker compose

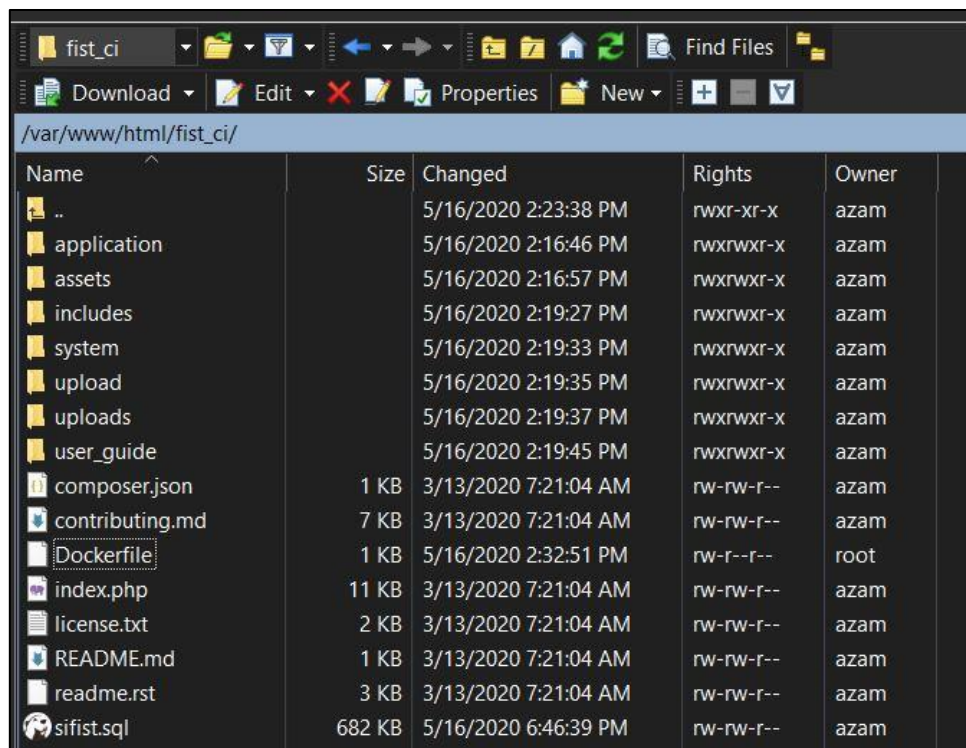
```

azam@aza-server:~$ sudo chmod +x /usr/local/bin/docker-compose
azam@aza-server:~$

```

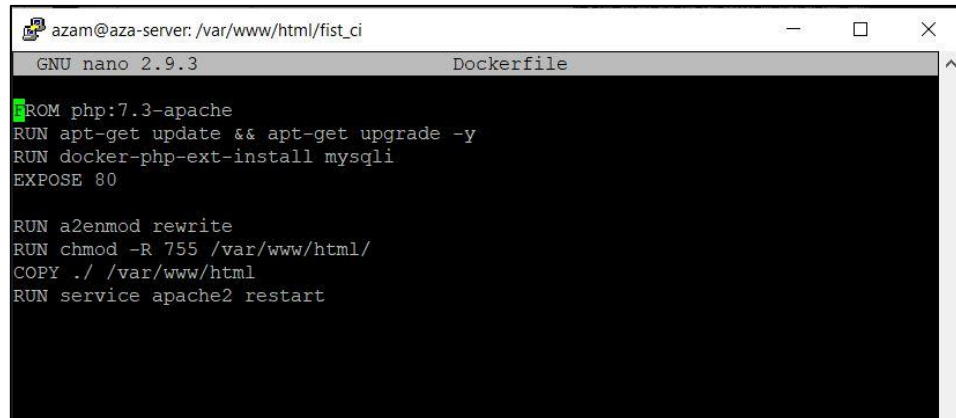
Gambar 2.50 Proses eksekusi instalasi docker compose

Setelah menginstall docker compose, selanjutnya kita akan memindahkan folder web yang akan kita *hosting* ke dalam folder `/var/www/html/` yang dapat diakses menggunakan WinSCP sehingga folder web yang akan kita *hosting* dalam kasus ini adalah `fist_ci` dapat ditampilkan pada folder `/var/www/html/` seperti **Gambar 2.51** berikut :



Gambar 2.51 Folder `fist_ci` yang telah dimasukkan ke `/var/www/html/`

Setelah `fist_ci` sudah berada pada folder yang ditentukan, selanjutnya kita akan berpindah direktori ke tempat `fist_ci` tersebut menggunakan parameter yang dijelaskan pada **Modul 2.9** kemudian kita akan membuat `dockerfile` untuk instalasi `php` dan aplikasi yang dibutuhkan untuk proses *hosting* yang parameternya dapat dilihat pada **Modul 2.10**. Kemudian isi `dockerfile` sesuai dengan yang ditunjukkan oleh **Gambar 2.52** berikut:



```

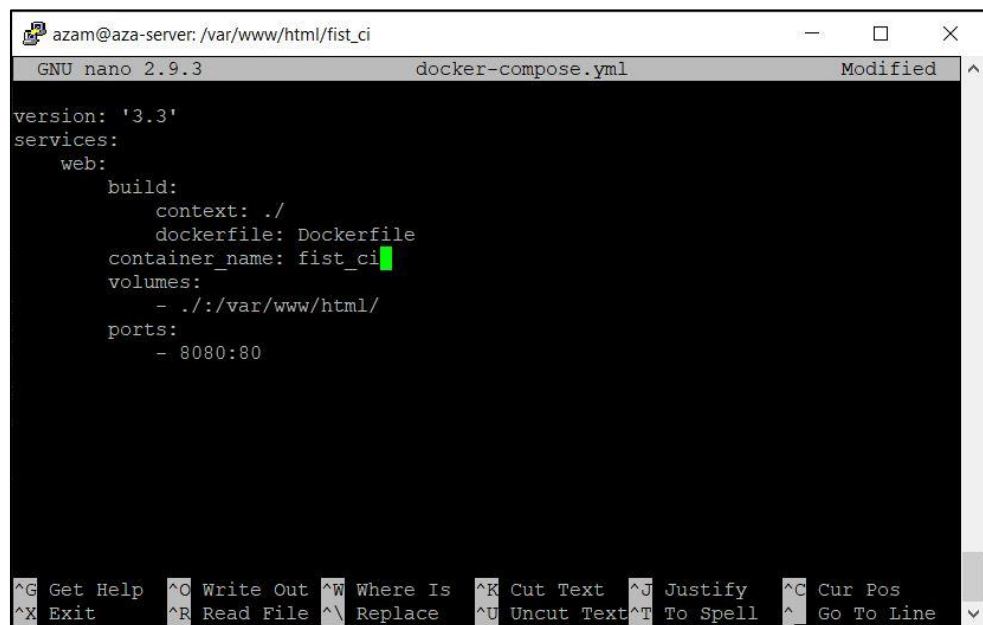
azam@aza-server: /var/www/html/fist_ci
GNU nano 2.9.3 Dockerfile
FROM php:7.3-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli
EXPOSE 80

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

```

Gambar 2.52 Isi dari dockerfile

Setelah pembuatan dockerfile, yang selanjutnya dibuat adalah docker-compose.yml yang mana docker-compose akan menjalankan dockerfile maupun docker images yang ada pada direktori tersebut secara bersamaan. Parameter untuk menjalankannya terdapat pada **Modul 2.11**. Karena pada *web* yang akan dijalankan databasenya merupakan database online maka pada docker-compose tidak perlu menjalankan database seperti yang ditampilkan oleh **Gambar 2.53**, jika menggunakan database offline maka dapat diisikan seperti yang ditampilkan oleh **Gambar 2.54** berikut :

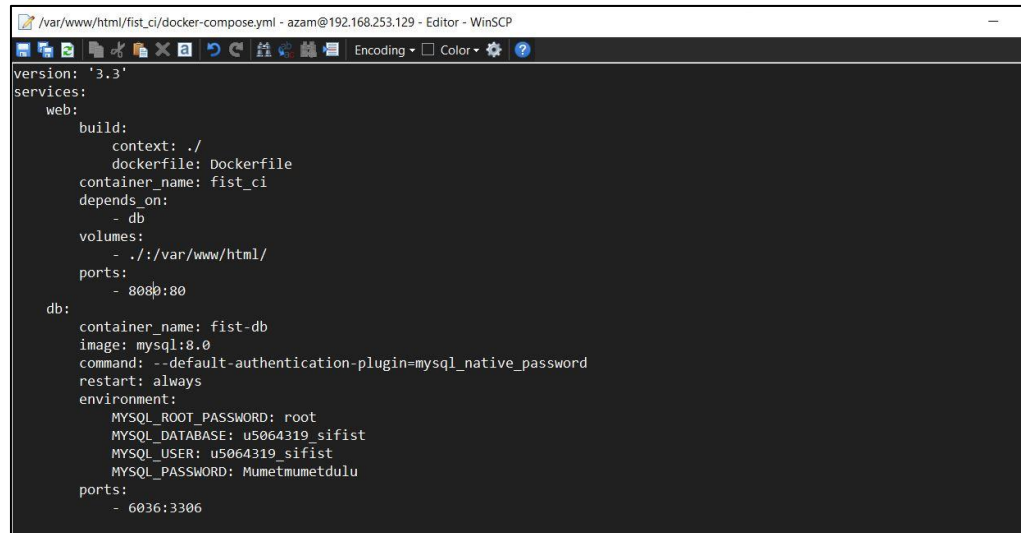


```

azam@aza-server: /var/www/html/fist_ci
GNU nano 2.9.3 docker-compose.yml Modified
version: '3.3'
services:
  web:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: fist_ci
    volumes:
      - ./:/var/www/html/
    ports:
      - 8080:80

```

Gambar 2.53 Docker-compose database online



Gambar 2.54 Docker-compose database offline

Sebelum menjalankan docker-compose. Pada folder web kita karena menggunakan *framework code igniter* maka perlu dilakukan konfigurasi terlebih dahulu pada config.php dan databasenya yang merupakan database online. Selain itu kita juga harus mengubah konfigurasi *htaccessnya* dengan mengakses parameter seperti yang telah dijelaskan pada **Modul 2.14**. Konfigurasi config dapat dilihat pada **Gambar 2.55** dan **Gambar 2.56**, konfigurasi database dapat dilihat pada **Gambar 2.57** dan konfigurasi *.htaccess* dapat dilihat pada **Gambar 2.58** berikut :

```
$config['base_url'] = 'http://'.$_SERVER['HTTP_HOST'].str_replace(basename($_SERVER['SCRIPT_NAME']),"",$_SERVER['SCRIPT_NAME']);
```

Gambar 2.55 Konfigurasi base_url

```

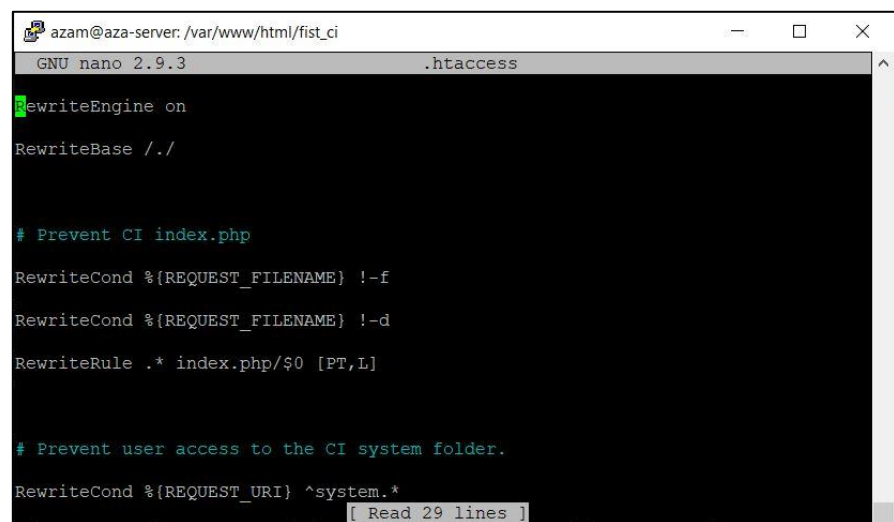
*/
$config['sess_driver'] = 'files';
$config['sess_cookie_name'] = 'ci_session';
$config['sess_expiration'] = 7200;
$config['sess_save_path'] = sys_get_temp_dir();
$config['sess_match_ip'] = FALSE;
$config['sess_time_to_update'] = 300;
$config['sess_regenerate_destroy'] = FALSE;

```

Gambar 2.56 Konfigurasi sess_save_path

```
$db['default'] = array(
    'dsn' => '',
    'hostname' => '153.92.10.57',
    'username' => 'u5064319_sifist',
    'password' => 'Mumetmumetdulu',
    'database' => 'u5064319_sifist',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Gambar 2.57 Konfigurasi database



```
azam@aza-server: /var/www/html/fist_ci
GNU nano 2.9.3 .htaccess

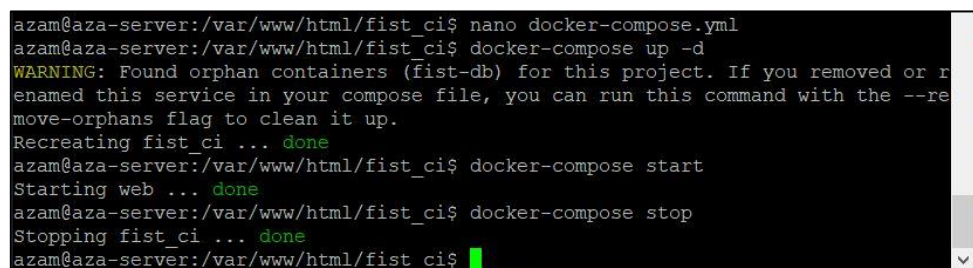
RewriteEngine on
RewriteBase ./

# Prevent CI index.php
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule .* index.php/$0 [PT,L]

# Prevent user access to the CI system folder.
RewriteCond %{REQUEST_URI} ^system.*
[ Read 29 lines ]
```

Gambar 2.57 Konfigurasi .htaccess

Setelah semua konfigurasi telah selesai diatur, selanjutnya kita dapat membangun, menjalankan, maupun menghentikan docker-compose dengan parameter yang dijelaskan pada **Modul 2.12** dan **Modul 2.13** yang hasilnya seperti pada **Gambar 2.58** sebagai berikut :

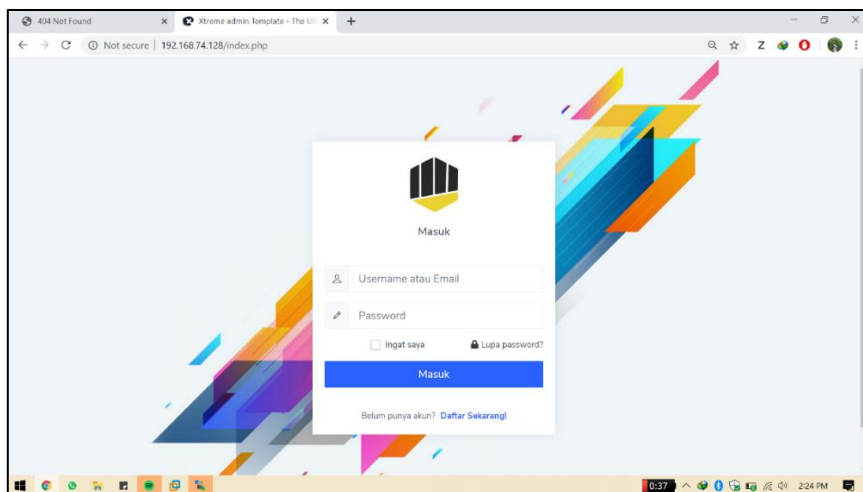


```
azam@aza-server:/var/www/html/fist_ci$ nano docker-compose.yml
azam@aza-server:/var/www/html/fist_ci$ docker-compose up -d
WARNING: Found orphan containers (fist-db) for this project. If you removed or r
enamed this service in your compose file, you can run this command with the --re
move-orphans flag to clean it up.
Recreating fist_ci ... done
azam@aza-server:/var/www/html/fist_ci$ docker-compose start
Starting web ... done
azam@aza-server:/var/www/html/fist_ci$ docker-compose stop
Stopping fist_ci ... done
azam@aza-server:/var/www/html/fist_ci$
```

Gambar 2.58 Menjalankan dan menghentikan docker compose

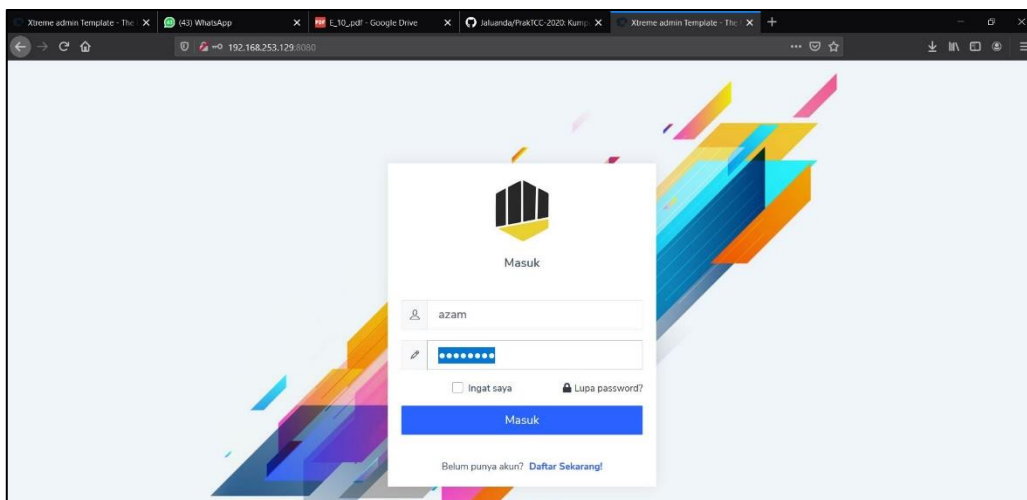
2.5 Hasil Implementasi

Setelah menyelesaikan tahapan-tahapan pada proses pengerjaan proyek 1 atau instalasi Linux OS, instalasi Ubuntu Server, dan tahap-tahap instalasi untuk layanan Hosting dengan benar, maka proyek yang telah diunduh dapat diakses melalui browser seperti pada **Gambar 2.59** berikut.



Gambar 2.59 Project 1 sudah dapat diakses melalui browser

Sementara hasil dari pengerjaan proyek 2 atau *hosting* menggunakan docker dengan cara mengakses IPHost:8080 ditampilkan pada **Gambar 2.60** sebagai berikut :



Gambar 2.60 Project 2 sudah dapat diakses melalui browser

2.6 Pengujian Singkat

Terhadap sistem informasi yang telah dirancang, akan dilakukan pengujian terhadap permasalahan yang tertuang pada bab pertama, yaitu bagaimana sistem dapat mempermudah pengelolaan layanan yang disediakan salah satunya adalah dengan pengelolaan kelas *training*.

Pertama-tama setelah admin berhasil *login*, admin menuju halaman kelas *training* dan meng-klik *button* Tambah Kelas. Hasil yang diharapkan adalah akan tampil *form* untuk *input* data kelas. Setelah data diisi maka kelas baru berhasil ditambahkan.

Berikut **Gambar 2.61** dan **Gambar 2.62** merupakan hasil pengujian singkat pada percobaan menambah data kelas:

The screenshot shows a web browser window with the URL `192.168.74.128/index.php/training/form_class`. The page title is 'FIST EFFECT'. The navigation bar includes 'Konsultasi', 'Training', '<> DEV', 'Seminar & Workshop', 'Feedback', and 'Pengaturan'. The main content area is titled 'Registrasi' and 'Tambah Kelas Training'. The form contains the following fields:

- Nama Kelas:
- Nama Paket:
- Nama Tentor:
- Status Kelas:

At the bottom of the form are two buttons: 'Batal' and 'Simpan'.

Gambar 2.61 Form tambah kelas

The screenshot shows a web browser window with the URL `192.168.74.128/index.php/training`. The page title is 'FIST EFFECT'. The navigation bar is the same as in Gambar 2.61. The main content area is titled 'Registrasi' and 'Daftar Kelas'. A 'Tambah Kelas' button is in the top right corner. The table below shows the list of classes:

No	Nama Kelas	Nama Mentor	Nama Paket	Pertemuan	Status Kelas	Aksi
1	BCD	Tentor 2	HTML & CSS - Basic	0/12	Sedang Berlangsung	[Icons]
2	BCD	Tentor 331	HTML & CSS - Basic	0/12	Sedang Berlangsung	[Icons]
3	B	Tentor 2	Dasar Web - Basic	0/14	Sudah Seleksi	[Icons]
4	Z	Tentor 5	Building Web & CI - Private	0/12	Sudah Seleksi	[Icons]
5	F	Tentor 2	HTML & CSS - Basic	0/12	Sedang Berlangsung	[Icons]
6	Uji Coba	Tentor Fist 1	Building Web & CI - Private	0/12	Belum Berlangsung	[Icons]
7	G	Tentor Fist 2	HTML & CSS - Basic	0/12	Sudah Seleksi	[Icons]

Gambar 2.62 Halaman Daftar Kelas

Hasil dari uji coba yang telah dilakukan adalah sistem sudah berjalan dengan baik dan dapat mempermudah admin dalam mengelola kelas *training*.

Pengujian proyek 2 dilakukan hampir sama seperti pengujian pada proyek 1. Pengujian pertama yang dilakukan adalah dengan menampilkan data yang ada pada database online seperti yang ditampilkan pada **Gambar 2.63** berikut :

No	Nama Kelas	Nama Mentor	Nama Paket	Pertemuan	Status Kelas	Aksi
1	F	Tentor 2	HTML & CSS - Basic	0/12	Sedang Berlangsung	[Icons]
2	Uji Coba	Tentor Fist 1	Building Web & CI - Private	0/12	Belum Berlangsung	[Icons]
3	G	Tentor Fist 2	HTML & CSS - Basic	0/12	Sudah Selesai	[Icons]

Gambar 2.63 Halaman Daftar Kelas

Pengujian selanjutnya adalah dengan menambahkan kelas dengan cara mengisi form tambah kelas seperti yang diperlihatkan pada **Gambar 2.64** dan hasilnya diperlihatkan pada **Gambar 2.65** berikut :

Gambar 2.64 Halaman Form Tambah Kelas

Registrasi
Training > Kelas

Daftar Kelas

+ Tambah Kelas

Tampilkan 10 entri Cari:

No	Nama Kelas	Nama Mentor	Nama Paket	Pertemuan	Status Kelas	Aksi
1	G	Tentor Fist 2	HTML & CSS - Basic	0/12	Sudah Selesai	
2	MantabMantab	Tentor 331	Building Web & CI - Private	0/12	Belum Berlangsung	

Gambar 2.64 Halaman Setelah Tambah Kelas

Kesimpulan dari percobaan tersebut bahwa sistem yang dibuat mampu berjalan sesuai dengan fungsionalitasnya tanpa ada kendala setelah dilakukan *hosting* baik melalui Ubuntu server secara langsung menggunakan LAMPP maupun menggunakan docker. Sistem juga dapat memberikan layanan yang sangat bermanfaat bagi admin, karena dapat mempermudah administrasi *training* yang dapat diakses dari mana saja, juga bagi perusahaan karena 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 hingga Mei 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
1.	Analisa Persoalan										
2.	Pembagian Tugas										
3.	Instalasi Linux OS dan Ubuntu Server										
4.	Penyusunan Laporan Bab I										
5.	Instalasi dan Hosting menggunakan LAMPP										
6.	Revisi Laporan 1										
	Instalasi dan Hosting menggunakan Docker										
7.	Penyusunan Laporan Bab II										
8.	Revisi Laporan 2										
9.	Penyelesaian Proyek dan Laporan										

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	Azam
2.	Latar Belakang Masalah	Ayu
3.	Tujuan dan Manfaat Proyek	Azam
4.	Instalasi dan Hosting menggunakan LAMPP	Ayu
5.	Instalasi dan Hosting menggunakan Docker	Azam
6.	Pengujian Singkat	Azam
7.	Penyusunan Laporan Bab II	Azam dan Ayu
8.	Penyusunan Laporan Bab III	Ayu
9.	Penyusunan Laporan Bab IV	Ayu

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan pengerjaan proyek akhir dengan judul Sistem Informasi Fist Effect bagia Training ini bisa dikatakan berhasil karena baik *hosting* yang dilakukan langsung menggunakan Ubuntu LAMPP dan juga Docker dapat berjalan sesuai fungsionalitasnya tanpa ada kendala baik pada fungsionalitas maupun pada koneksi database. Masalah yang dihadapi seperti penyesuaian *framework code igniter* dan *online database* juga dapat teratasi seperti yang telah dibuktikan pada **Bab II** sub **2.4** hingga **2.6**. Keberhasilan proyek ini juga dapat memberi manfaat kepada fist effect terutama bagian admin karena dapat mempermudah pekerjaan dan juga penyimpanan data-data yang dapat digunakan kapanpun dan dimanapun karena menggunakan teknologi *cloud computing*.

Pembagian peran kelompok pada proyek ini kami rasa kurang efektif karena baik *hosting* langsung pada ubuntu lamp dan juga docker harus dikerjakan di 1 laptop sehingga pembagiannya bukan kerja sama melainkan pembagian tugas seperti yang sudah dijelaskan pada **Bab III** sub **3.2** . Komunikasi antar anggota juga kurang maksimal dengan adanya *study from home* yang sedang berlaku saat ini sehingga dapat memberatkan masing-masing anggota.

4.2 Saran

Docker yang cukup sulit dimengerti seharusnya terdapat tutorial langsung dari pihak kampus karena dengan pembagian tugas yang seperti ini terdapat anggota kelompok yang mungkin kurang mengerti tentang penggunaannya apalagi dengan sistem yang terjadi sehingga membuat pembagian tugasnya bukan kerja kelompok melainkan pembagian tugas murni seperti yang sudah dijelaskan sebelumnya.

DAFTAR PUSTAKA

- Puspitasari, D. (2020). Cloud Computing. Retrieved from <https://dianpuspita0903.wordpress.com/2016/05/01/cloud-computing/>.
- Fajrin, T. (2012). Analisis Sistem Penyimpanan Data Menggunakan Sistem Cloud Computing Studi Kasus SMK N 2 Karanganyar. *IJNS-Indonesian Journal on Networking and Security*, 1(1).
- Adiputra, F. (2015). Container Dan Docker: Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi Web. *Jurnal Simantec*, 4(3).
- Elang, N. (2020). Pengertian Lamp Server secara umum. Retrieved 17 May 2020, from <https://nandaelang21.blogspot.com/2018/03/pengertian-lamp-server-secara-umum.html>
- Al Amien, J. (2019). Membangun Cloud Menggunakan Docker Pada Implementasi Load Balancing dan Pengujian Algoritma Round Robin Pada Web Server. In *Prosiding Seminar Nasional Computation Technology and its Application* (Vol. 1, No. 1, pp. 128-132).
- Apridayanti, S., Isnawaty, I., & Saputra, R. A. (2018). DESAIN DAN IMPLEMENTASI VIRTUALISASI BERBASIS DOCKER UNTUK DEPLOYMENT APLKASI WEB. *semanTIK*, 4(2), 37-46