

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**SISTEM PENDUKUNG KEPUTUSAN TRANSFER PEMAIN BOLA
MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN
DOCKERFILENYA**



DISUSUN OLEH:

NAMA ANGGOTA : RISKI MIDI W 123170035

M SETIAWAN W 123170043

KELAS : C

ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom

WAHYU AJI NUGROHO, S.Kom

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

2020

HALAMAN PENGESAHAN

SISTEM PENDUKUNG KEPUTUSAN TRANSFER PEMAIN BOLA MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

Riski Midi Wardana

123170035

M Setiawan W

123170043

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

Puji syukur kehadiran Allah SWT yang hingga saat ini masih memberikan kita nikmat iman dan kesehatan, sehingga saya diberi kesempatan yang luar biasa ini yaitu kesempatan untuk menyelesaikan laporan praktikum *Cloud Computing* dengan judul “Sistem Pendukung Keputusan Transfer Pemain Bola Menggunakan Ubuntu LAMPP Dan Proses Pembuatan Dockerfilenya” ini dengan tepat waktu.

Adapun penyusunan laporan percobaan ini adalah dengan maksud supaya dapat mengimplementasikan hasil belajar yang kami peroleh selama mengikuti kegiatan belajar di Informatika ini khususnya pada mata kuliah *Cloud Computing*, baik itu materi di kelas ataupun kegiatan praktikum yang dilaksanakan di laboratorium.

Dalam melakukan percobaan ini, tentunya banyak sekali hambatan yang kami rasakan, oleh sebab itu, kami berterimakasih kepada beberapa pihak terutama bapak pengajar mata kuliah *Cloud Computing* dan juga para asisten laboratorium kami yang telah membantu membina dan mendukung kami dalam mengatasi beberapa hambatan yang kami temui.

Selain itu kami juga sadar bahwa pada laporan kami ini dapat ditemukan banyak sekali kekurangan serta jauh dari kesempurnaan. Oleh sebab itu, kami benar-benar menanti kritik dan saran untuk kemudian dapat kami revisi dan kami tulis di masa yang selanjutnya, sebab sekali lagi kami menyadari bahwa tidak ada sesuatu yang sempurna tanpa disertai saran yang konstruktif. Dan semoga laporan percobaan ini dapat memberikan manfaat.

Yogyakarta, 30 April 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Proyek Akhir	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
2.3.1 Parameter dan Konfigurasi Pada Proyek Pertama.....	9
2.3.2 Parameter dan Konfigurasi Pada Proyek Kedua	10
2.4.1 Tahap Implementasi Proyek Pertama Tahapan implementasi proyek pertama	11
2.4.2 Tahap Implementasi Proyek Kedua Tahapan implementasi proyek pertama kedua	17
2.5.1 Hasil Implementasi Proyek Pertama.....	19
2.5.2 Hasil Implementasi Proyek Kedua	20
2.6 Pengujian Singkat.....	20
BAB III JADWAL Pengerjaan DAN PEMBAGIAN TUGAS.....	21
3.1 Agenda Pengerjaan.....	21
3.2 Keterangan Pembagian Tugas.....	21
BAB IV KESIMPULAN DAN SARAN.....	23
4.1 Kesimpulan	23
4.2 Saran.....	23
DAFTAR PUSTAKA	24
LAMPIRAN	25

BAB I

PENDAHULUAN

1.1. Latar Belakang Proyek Akhir

Cloud Computing merupakan istilah yang sudah sering kita dengar. Namun, masih sedikit yang mengerti apa itu Definisi, Fungsi dan cara kerja *Cloud Computing* dalam kehidupan sehari – hari. *Cloud Computing* adalah sebuah model komputasi dimana aktifitas pemrosesan, penyimpanan, perangkat lunak dan layanan lainnya disediakan layaknya sumber virtual terpadu pada suatu jaringan yang umumnya adalah internet (Loudon,2015). Pada perusahaan – perusahaan maupun organisasi besar pengaruh *Cloud Computing* untuk memudahkan dalam berinteraksi dari satu komputer dengan komputer lainnya tanpa memakan banyak biaya. *Cloud Computing* membuat semua pekerjaan serba praktis dan tidak perlu menggunakan Infrastruktur yang besar. *Cloud Computing* Bahkan dapat meminimalisir biaya pengeluaran dalam perusahaan. Tanpa kita sadari sehari – hari kita telah bertukar informasi dan data menggunakan *Cloud Computing*. Dalam penerapannya, *Cloud Computing* hanya membutuhkan jaringan internet, hal ini dikarenakan *Cloud Computing* memiliki server yang terpusat.

Contoh penerapan teknologi virtualisasi adalah Netflix. Netflix merupakan layanan *Video On Demand* yang memungkinkan user melihat film dimanapun dan kapanpun. Netflix menggunakan layanan Amazon Elastic Compute Cloud (Amazon EC2) untuk membuat *scalable encoding system* yang dapat melayani banyak pengguna. Contoh penerapan teknologi kontainerisasi adalah Ubisoft. Ubisoft merupakan perusahaan yang bergerak di bidang pengembangan aplikasi *game*. Ubisoft menggunakan kontainerisasi dengan memanfaatkan Amazon Elastic Container Service untuk menciptakan pengalaman *gaming* yang lancar.

Proyek perlu diselesaikan menggunakan virtualisasi dan kontainerisasi karena memungkinkan untuk mengirimkan kode lebih cepat, menstandarisasi operasi aplikasi, memindahkan kode dengan lancar, dan menghemat uang dengan meningkatkan pemanfaatan sumber daya. Kita dapat mendapatkan satu objek yang dapat dijalankan di mana saja.

Sistem informasi ini membutuhkan program PHP, Apache, dan MySQL yang kemudian akan dikonfigurasi menggunakan sistem operasi ubuntu di virtual machine. Setelah selesai di virtualisasikan menggunakan VMWare selanjutnya akan di kontainerisasi

menggunakan docker sehingga ketika kita akan menjalankan aplikasi akan lebih mudah tanpa menginstall aplikasi yang dibutuhkan di server secara manual.

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 Pendukung Keputusan Transfer Pemain Bola yang telah dibuat sebelumnya dengan menggunakan Ubuntu 20.4 dan LAMPP (Apache 8, PHP 7.4, MySQL 5.2).
2. Pemanfaatan *docker* untuk Sistem Pendukung Keputusan Transfer Pemain Bola yang telah dibuat sebelumnya menggunakan Ubuntu 20.4 dan LAMPP.

1.3 Manfaat Proyek Akhir

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

1. Pengurangan Biaya Investasi *Hardware*. Investasi *hardware* dapat ditekan lebih rendah karena virtualisasi hanya mendayagunakan kapasitas yang sudah ada. Tak perlu ada penambahan perangkat komputer, server dan peripheral secara fisik. Walaupun ada penambahan kapasitas harddisk dan memori, itu lebih ditujukan untuk mendukung stabilitas kerja komputer induk, yang jika dihitung secara finansial, masih jauh lebih hemat dibandingkan investasi *hardware* baru.
2. Kemudahan *Backup & Recovery*. Server-server yang dijalankan didalam sebuah mesin virtual dapat disimpan dalam 1 buah image yang berisi seluruh konfigurasi sistem. Jika satu saat server tersebut *crash*, kita tidak perlu melakukan instalasi dan konfigurasi ulang. Cukup mengambil salinan image yang sudah disimpan, merestore data hasil backup terakhir dan server berjalan seperti sedia kala. Hemat waktu, tenaga dan sumber daya.
3. Kemudahan Deployment. Server virtual dapat dikloning sebanyak mungkin dan dapat dijalankan pada mesin lain dengan mengubah sedikit konfigurasi. Mengurangi beban kerja para staf IT dan mempercepat proses implementasi suatu sistem Mengurangi Biaya Space. Semakin sedikit jumlah server berarti semakin sedikit pula ruang untuk menyimpan perangkat. Jika server ditempatkan pada suatu co-location server/data center, ini akan berimbas pada pengurangan biaya sewa

4. Kemudahan *Maintenance* & Pengelolaan. Jumlah server yang lebih sedikit otomatis akan mengurangi waktu dan biaya untuk mengelola. Jumlah server yang lebih sedikit juga berarti lebih sedikit jumlah server yang harus ditangani

5. Kemudahan *Replacement*. Proses penggantian dan upgrade spesifikasi server lebih mudah dilakukan. Jika server induk sudah overload dan spesifikasinya tidak mencukupi lagi, kita bisa dengan mudah melakukan upgrade spesifikasi atau memindahkan virtual machine ke server lain yang lebih powerful

1.4 Tahap Penyelesaian Proyek Akhir

Gunakan format berikut dan ubah beberapa bagian seperlunya saja atau pilih beberapa poin yang relevan, bila tidak ada dapat ditambahkan sendiri...

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

1. Menganalisis kebutuhan dari sistem ABC untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS. (contoh penulisan dalam penomoran formatnya rapi seperti ini)
2. Mengintegrasikan penyimpanan data *medical record* pada FreeNAS dengan Sistem Rekam Medis yang berada di Ubuntu Server.
3. Menentukan konfigurasi yang tepat untuk pengaturan FreeNAS sehingga dapat digunakan sesuai *requirement* yang berupa:
 - a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Terdapat divisi pusat yang dapat masuk ke semua akun kecuali akun Z.
 - c. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
4. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
5. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Pencatatan Kendaraan ketersediaannya/*availability*-nya maksimal.
6. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dsb...
7. Menghasilkan Sistem Pelacakan Kereta Api yang berbasis *cloud computing* sesuai standar ISO 9001.

(bagian ini merupakan penjelasan lebih mendetail terhadap poin-poin teknis untuk pengerjaan cloudnya seperti yang telah dijelaskan secara singkat pada bagian 1.1 paragraf ketiga dan keempat)

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir ini terbagi menjadi dua sesuai dengan judul yang telah dituliskan sebelumnya. Komponen yang pertama ialah “Sistem Pendukung Keputusan Transfer Pemain Bola Menggunakan Ubuntu LAMPP” dan yang kedua “Sistem Pendukung Keputusan Transfer Pemain Bola Menggunakan Ubuntu LAMPP Dengan Docker”. Berikut akan dibahas keduanya masing-masing dalam sub bab terkait.

2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Sistem Pendukung Keputusan Transfer Pemain Bola Menggunakan Ubuntu LAMPP” yang berbasiskan konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.4 dan bahasa HTML dengan versi minimal 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data pemain bola yang ingin di transfer.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh jalaran orang yang ingin mencari pemain bola di Indonesia.
4. Sistem tersebut juga harus dapat digunakan sewaktu-waktu dan ditargetkan memiliki nilai *uptime* SLA 99,9% sehingga diperlukan minimal dua buah Ubuntu Server, satu sebagai *primary server* dan salah satunya sebagai *backup server*. Mekanisme untuk peralihan antar *server* membutuhkan *proxy server* dan semuanya menggunakan sistem operasi Ubuntu.

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.	Merek Server	Virtual Machine dengan VMWare Fusion	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .

2.	Prosesor	1 core @2.4Ghz	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: 192.168.43.156/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.43.156	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.43.255	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
		PORT: 80	Port yang dapat di akses
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	4GB	Alokasi RAM untuk <i>guest OS</i>
6.	Storage	20GB	Alokasi Storage untuk <i>guest OS</i>

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

Tabel 2.2 Spesifikasi Ubuntu OS untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.4	Bahasa pemrograman yang digunakan untuk membuat sistem
		MySQL 5.2	Database yang digunakan untuk penyimpanan data

2.1.2 Komponen pada Proyek Kedua

Untuk membangun “Sistem Pendukung Keputusan Transfer Pemain Bola Menggunakan Ubuntu LAMPP Dengan Docker” yang berbasis konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.4 dan bahasa HTML dengan versi minimal 4.0.

2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data pemain bola yang ingin di transfer.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh jajaran orang yang ingin mencari pemain bola di Indonesia.
4. Sistem tersebut juga harus dapat digunakan sewaktu-waktu dan ditargetkan memiliki nilai *uptime* SLA 99,9% sehingga diperlukan minimal dua buah Ubuntu Server, satu sebagai *primary server* dan salah satunya sebagai *backup server*. Mekanisme untuk peralihan antar *server* membutuhkan *proxy server* dan semuanya menggunakan sistem operasi Ubuntu.
5. Sistem harus mudah untuk di install di berbagai platform dengan konfigurasi yang minimal.
6. Sistem harus compatibel dan dapat berjalan di berbagai platform.

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

Tabel 2.2 Spesifikasi VM *cloud computing* untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Fusion	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	1 core @2.4Ghz	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: 192.168.43.156/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.43.156	Alamat IP untuk DNS <i>guest OS</i> .
		PORT: 8000	Port yang dapat di akses
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	4GB	Alokasi RAM untuk <i>guest OS</i>
6.	Storage	20GB	Alokasi Storage untuk <i>guest OS</i>

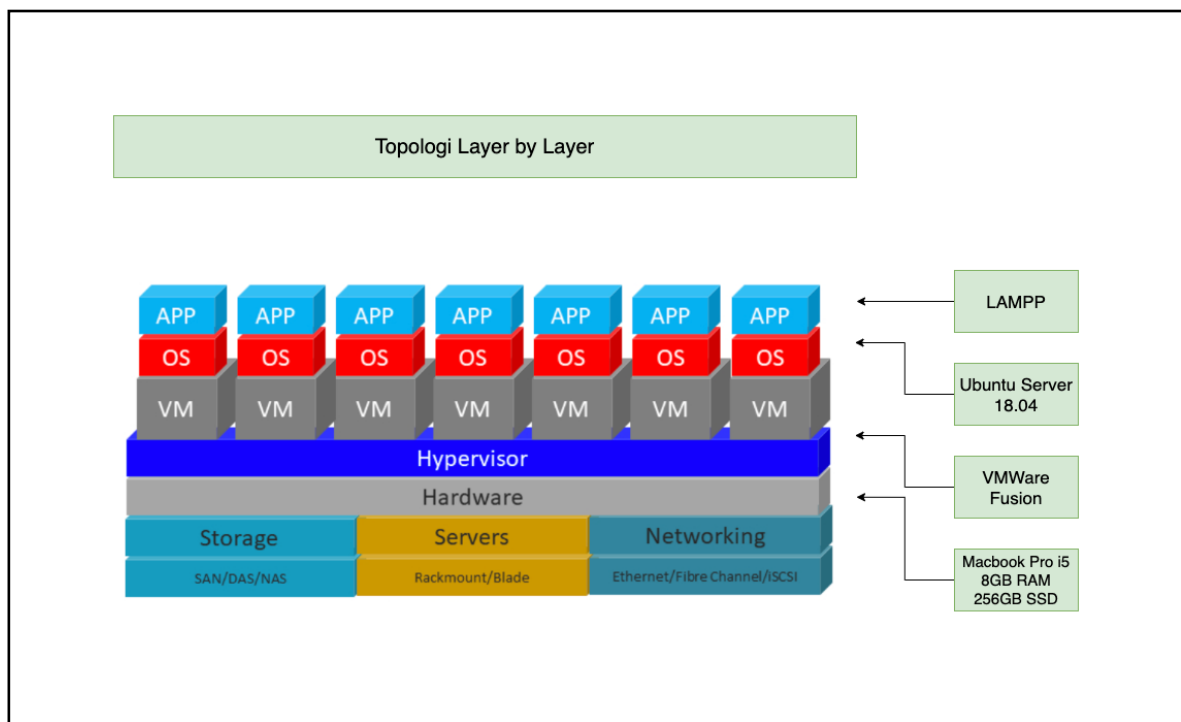
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

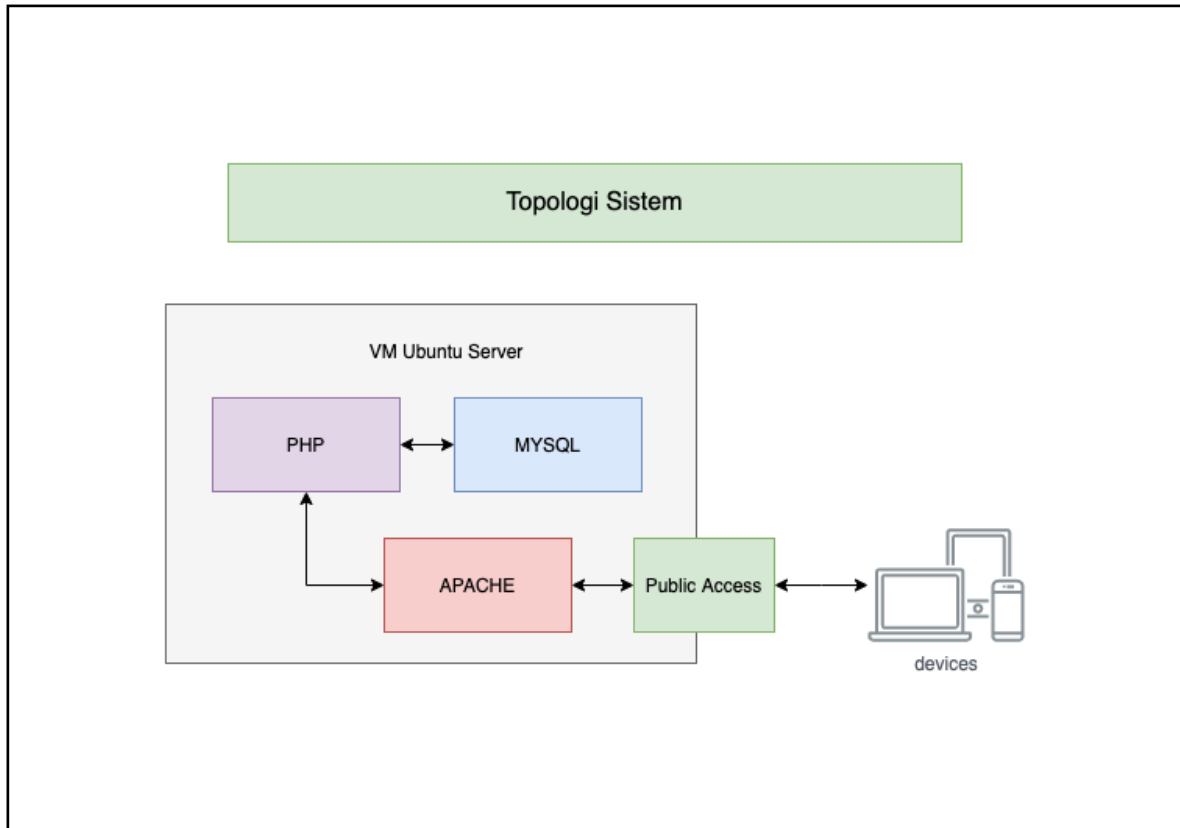
Tabel 2.3 Spesifikasi Ubuntu OS untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preproesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.4	Bahasa pemrograman yang digunakan untuk membuat sistem
		MySQL 5.2	Database yang digunakan untuk penyimpanan data
2	Docker	Docker 19.03.5	Platform untuk kontainerisasi
		Docker Compose 1.24.1	Kontainer Orkestrator

2.2.1 Rancangan Arsitektur *Cloud Computing* Proyek Pertama

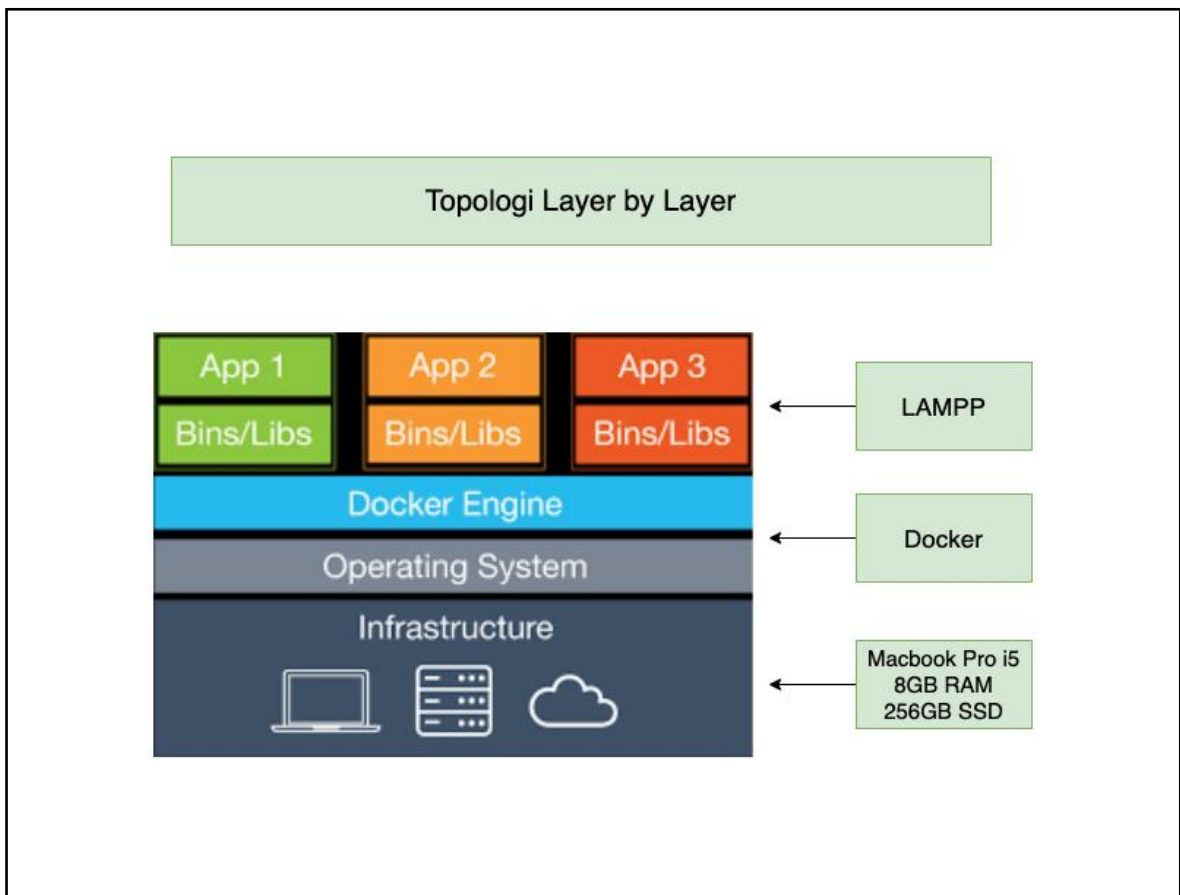
Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya (font 11, TNR)



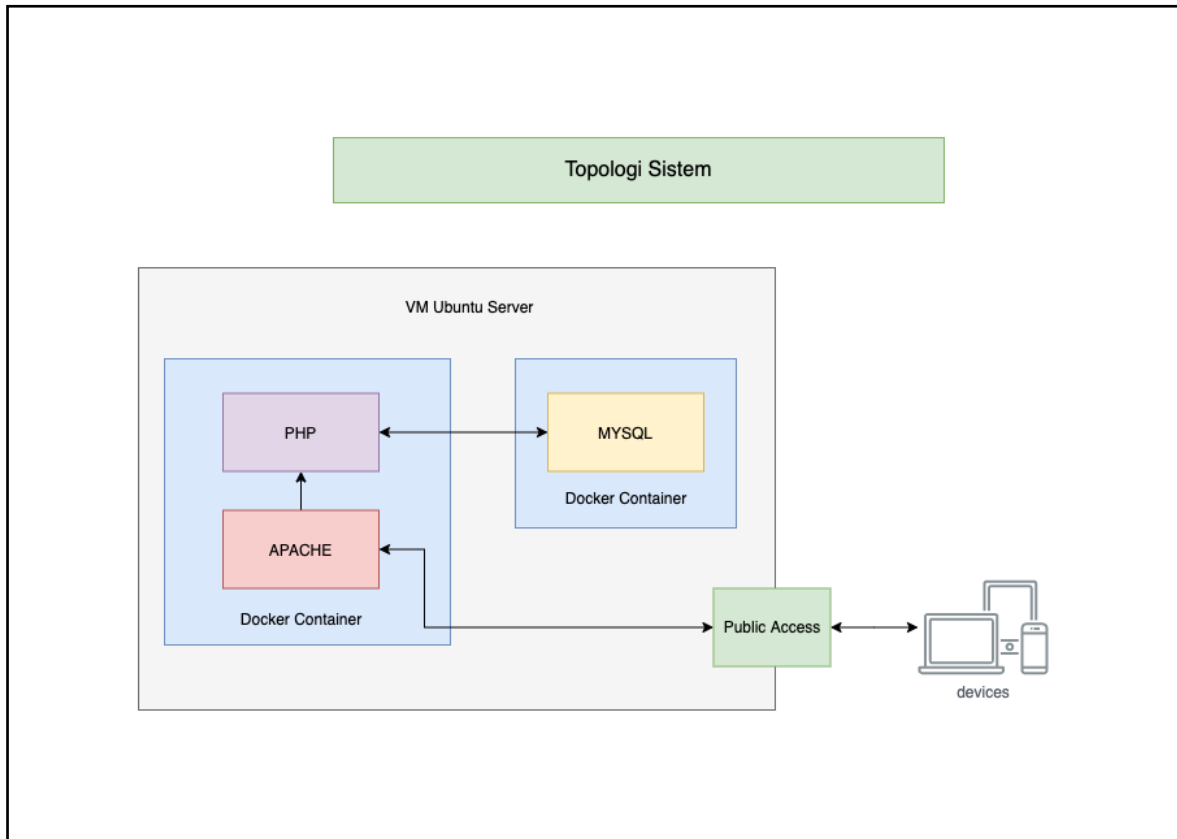


Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya (font 11, TNR)

2.2.2 Rancangan Arsitektur *Cloud Computing* Proyek Kedua



Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya (font 11, TNR)



Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya (font 11, TNR)

2.3.1 Parameter dan Konfigurasi Pada Proyek Pertama

Pada proyek ini dibutuhkan aplikasi utama yaitu apache2 untuk web server, mysql untuk database, dan php untuk bahasa pemrograman, sehingga perlu di install terlebih dahulu.

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

```
$ sudo apt install apache2
```

Keterangan:

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

Modul 2.1 Parameter instalasi Apache

Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan

Modul 2.1 berikut ini:

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

Keterangan:

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

Modul 2.1 Parameter instalasi Apache

Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan **Modul**

2.1 berikut ini:

```
$ sudo apt install php
```

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

Modul 2.1 Parameter instalasi Apache

Parameter yang digunakan untuk migrasi table di database dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ export $(cat .env)
$ make migrate
```

Keterangan:

- export: untuk mengexport variabel ke terminal
- cat: untuk menampilkan data dari awal sampai akhir
- make: aplikasi untuk menjalankan script dari makefile
- migrate: parameter script dari makefile yang dibuat

Modul 2.1 Parameter instalasi Apache

2.3.2 Parameter dan Konfigurasi Pada Proyek Kedua

Pada proyek ini dibutuhkan aplikasi utama yaitu docker dan docker-compose sehingga perlu di install terlebih dahulu.

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

```
$ sudo snap install docker
```

Keterangan:

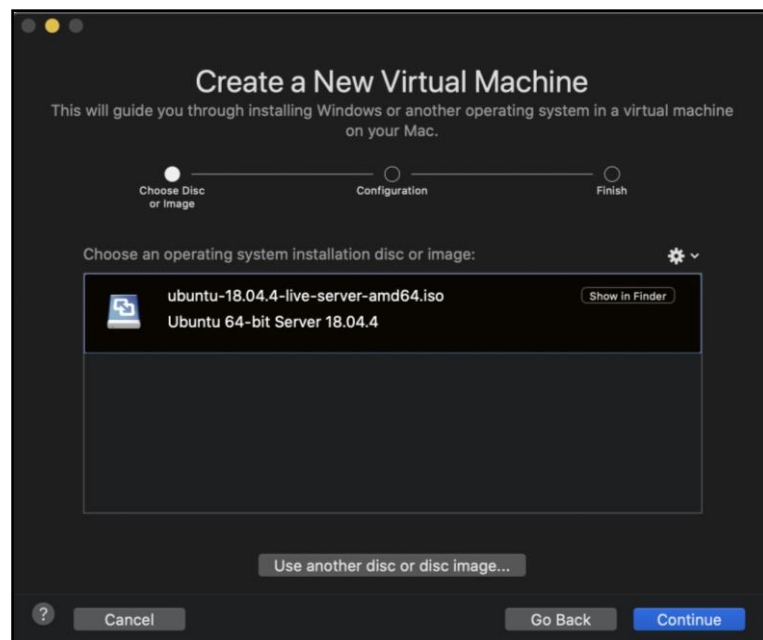
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- snap : merupakan alternative package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- docker : nama paket aplikasi untuk docker

Modul 2.1 Parameter instalasi Apache

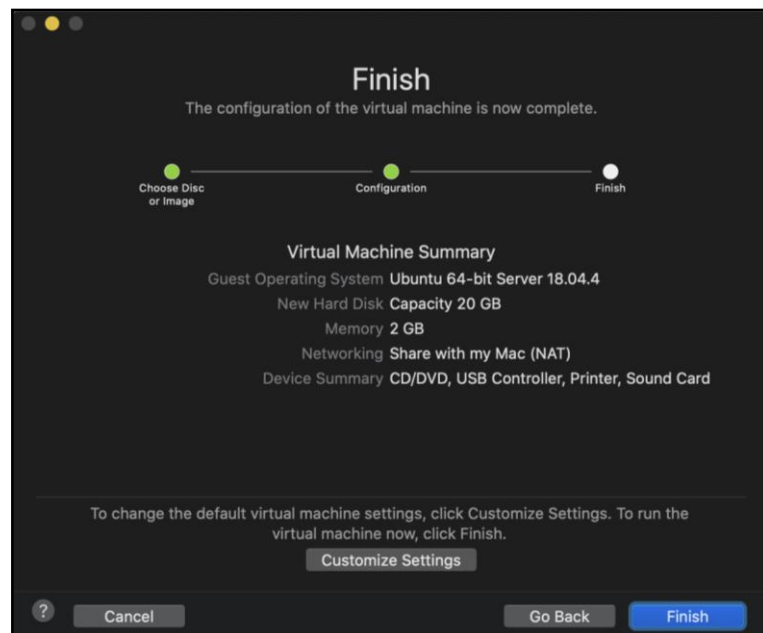
2.4.1 Tahap Implementasi Proyek Pertama

Tahapan implementasi proyek pertama sebagai berikut:

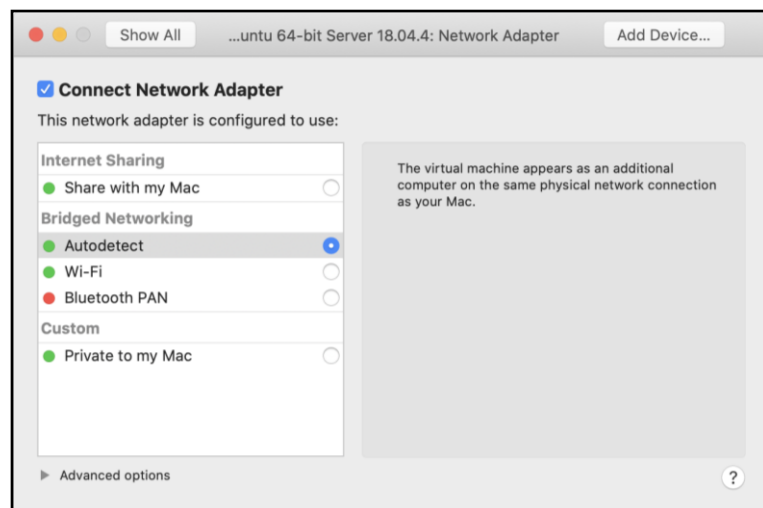
- a. Buat virtual machine di VMWare Fusion dengan image Ubuntu Server 18.04 yang Sudah di download



- b. Setelah itu akan ditampilkan rincian konfigurasi berdasarkan parameter-parameter yang telah ditentukan sebelumnya, dan juga akan didapatkan pesan/tombol finish untuk menyelesaikan konfigurasinya



- c. Ubah network adapter menggunakan bridge



- d. Masukkan username dan password untuk ubuntu server

Profile setup [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: Riski Midi Wardana

Your server's name: ubuntu-18-04
The name it uses when it talks to other computers.

Pick a username: riskimidiw

Choose a password: *****

Confirm your password: *****

[Done]

- e. Hubungkan terminal Mac dengan virtual machine yang Sudah dibuat menggunakan SSH

```
riskimidiw — riskimidiw@ubuntu-18-04: ~ — ssh riskimidiw@192.168.43.156...
➜ ~ ssh riskimidiw@192.168.43.156
riskimidiw@192.168.43.156's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-99-generic x86_64)

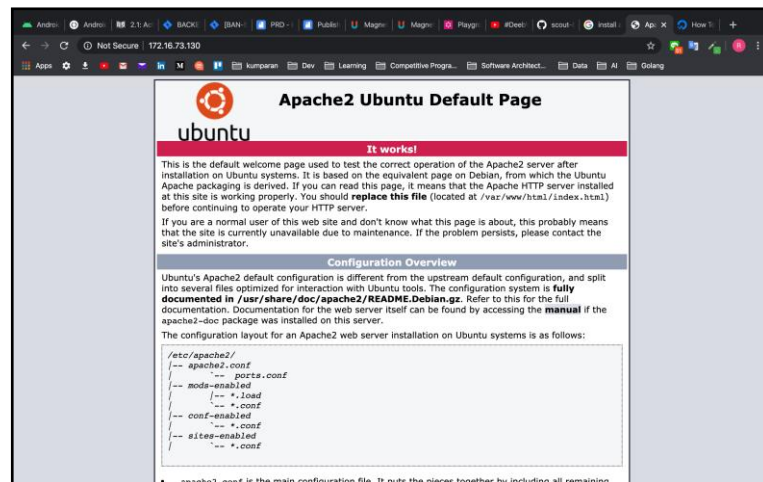
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri May 1 04:34:38 UTC 2020

System load:            0.79
Usage of /:              36.3% of 19.56GB
Memory usage:           31%
Swap usage:             0%
Processes:              205
Users logged in:        1
IP address for ens33:    192.168.43.156
IP address for br-1004830e4c3f: 172.19.0.1
IP address for docker0:  172.17.0.1
IP address for br-85e3cf090dcf: 172.18.0.1

 * 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.
```

f. Install Apache2



g. Install MYSQL

```
riskimidiw@ubuntu-18-04: ~ — ssh 172.16.73.130 — 80x24
riskimidiw@ubuntu-18-04:~$ sudo 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 22 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]
```

h. Install PHP

```

riskimidiw — riskimidiw@ubuntu-18-04: ~ — ssh 172.16.73.130 — 95x48
riskimidiw@ubuntu-18-04:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: No
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.

riskimidiw — riskimidiw@ubuntu-18-04: ~ — ssh 172.16.73.130 — 80x24
riskimidiw@ubuntu-18-04:~$ sudo apt install php libapache2-mod-php php-mysql php
-intl -y
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-intl php7.2-json php7.2-mysql php7.2-opcache 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-intl
  php-mysql php7.2 php7.2-cli php7.2-common php7.2-intl php7.2-json
  php7.2-mysql php7.2-opcache php7.2-readline
0 upgraded, 15 newly installed, 0 to remove and 22 not upgraded.
Need to get 4258 kB of archives.
After this operation, 18.6 MB of additional disk space will be used.
0% [Working]

```

```

riskimidiw@ubuntu-18-04:~$ git clone https://github.com/riskimidiw/scout-bot
Cloning into 'scout-bot'...
Username for 'https://github.com': riskimidiw
Password for 'https://riskimidiw@github.com':
remote: Enumerating objects: 491, done.
remote: Counting objects: 100% (491/491), done.
remote: Compressing objects: 100% (216/216), done.
remote: Total 491 (delta 269), reused 491 (delta 269), pack-reused 0
Receiving objects: 100% (491/491), 1.52 MiB | 853.00 KiB/s, done.
Resolving deltas: 100% (269/269), done.
riskimidiw@ubuntu-18-04:~$ ls
scout-bot  snap
riskimidiw@ubuntu-18-04:~$

```

i. Clone repository dari proyek

```

riskimidiw@ubuntu-18-04:~$ sudo chmod 777 /var/www/html/
riskimidiw@ubuntu-18-04:~$ cp -r scout-bot/ /var/www/html/
riskimidiw@ubuntu-18-04:~$ cd /var/www/html/scout-bot/
riskimidiw@ubuntu-18-04:/var/www/html/scout-bot$

```

j. Konfigurasi server

```

# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/scout-bot/

SetEnv SERVER_BASE_URL http://172.16.73.130/
SetEnv MYSQL_HOST localhost
SetEnv MYSQL_DATABASE scoutbot
SetEnv MYSQL_USERNAME admin
SetEnv MYSQL_PASSWORD admin

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

-- INSERT --

```

k. Migrate database

```
riskimidiw — riskimidiw@ubuntu-18-04: /var/www/html/scout-bot — ssh 172.1...
[riskimidiw@ubuntu-18-04:/var/www/html/scout-bot$ vim .env
[riskimidiw@ubuntu-18-04:/var/www/html/scout-bot$ export $(cat .env)
[riskimidiw@ubuntu-18-04:/var/www/html/scout-bot$ make migrate
Migrations run successfully
[riskimidiw@ubuntu-18-04:/var/www/html/scout-bot$ ]
```

2.4.2 Tahap Implementasi Proyek Kedua

Tahapan implementasi proyek pertama sebagai berikut:

- a. Buka virtual machine yang Sudah di buat di proyek pertama, kemudian hubungkan dengan ssh

```
riskimidiw — riskimidiw@ubuntu-18-04: ~ — ssh riskimidiw@192.168.43.156...
ssh riskimidiw@192.168.43.156
riskimidiw@192.168.43.156's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-99-generic x86_64)

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

System information as of Fri May 1 04:34:38 UTC 2020

System load:                0.79
Usage of /:                  36.3% of 19.56GB
Memory usage:               31%
Swap usage:                 0%
Processes:                  205
Users logged in:             1
IP address for ens33:        192.168.43.156
IP address for br-1004830e4c3f: 172.19.0.1
IP address for docker0:      172.17.0.1
IP address for br-85e3cf090dcf: 172.18.0.1

 * 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.
```

b. Install docker

```
riskimidiw — riskimidiw@ubuntu-18-04: ~ — ssh riskimidiw@192.168.43.156...
riskimidiw@ubuntu-18-04:~$ sudo snap install docker
```

```
riskimidiw — riskimidiw@ubuntu-18-04: ~ — ssh 172.16.73.130 — 80x24
riskimidiw@ubuntu-18-04:~$ sudo chmod 666 /var/run/docker.sock
riskimidiw@ubuntu-18-04:~$ docker version
Client:
 Version:      18.09.9
 API version:  1.39
 Go version:   go1.13.4
 Git commit:   1752eb3
 Built:        Sat Nov 16 01:05:26 2019
 OS/Arch:      linux/amd64
 Experimental: false

Server:
 Engine:
  Version:      18.09.9
  API version:  1.39 (minimum version 1.12)
  Go version:   go1.13.4
  Git commit:   9552f2b
  Built:        Sat Nov 16 01:07:48 2019
  OS/Arch:      linux/amd64
  Experimental: false
riskimidiw@ubuntu-18-04:~$
```

c. Jalankan container dengan docker-compose

```
riskimidiw — riskimidiw@ubuntu-18-04: ~/scout-bot — ssh 172.16.73.130 — 8...
riskimidiw@ubuntu-18-04:~$ cd scout-bot/
riskimidiw@ubuntu-18-04:~/scout-bot$ sudo systemctl stop apache2 mysql
riskimidiw@ubuntu-18-04:~/scout-bot$ docker-compose up -d
Creating network "scout-bot_scoutbot-network" with the default driver
Pulling mysql (mysql:5.7)...
5.7: Pulling from library/mysql
123275d6e508: Pulling fs layer
27cddf5c7140: Downloading [=====>]
27cddf5c7140: Downloading [=====>]
123275d6e508: Downloading [>]
277.6kB/27.1MBDownloading [>]
42.8kB/4.178MBiting
123275d6e508: Downloading [==>]
1.117MB/27.1MBDownloading [=====>]
c17d442e14c9: Downloading [=====>]
960.9kB/4.178MBiting
003c42612f11: Waiting
```

d. Migrate database

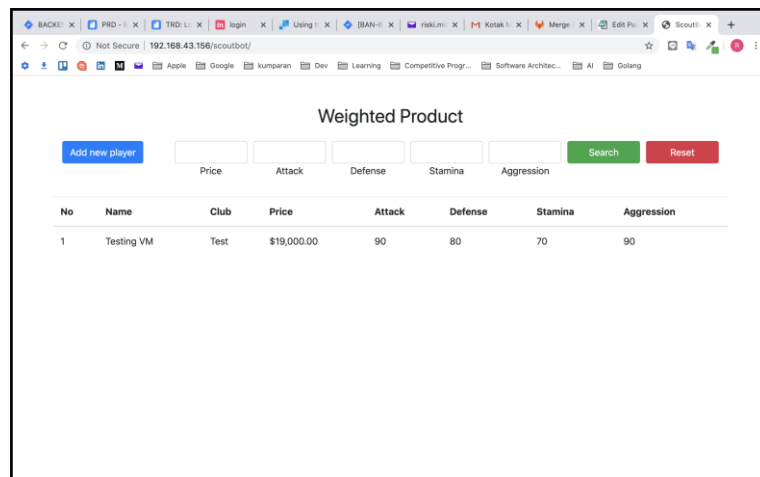
```
riskimidiw@ubuntu-18-04: ~/scout-bot — ssh 172.16.73.130 — 8...
find . -name \*.la -o -name \*.a | xargs rm -f
find . -name \*.so | xargs rm -f
find . -name .libs -a -type d|xargs rm -rf
rm -f libphp.la modules/* libs/*
Removing intermediate container 9281347209e5
--> b87e272b4d3b
Step 4/4 : RUN a2enmod rewrite
--> Running in efca6539776f
Enabling module rewrite.
To activate the new configuration, you need to run:
    service apache2 restart
Removing intermediate container efca6539776f
--> 6f3e04f235b7

Successfully built 6f3e04f235b7
Successfully tagged scout-bot:latest
WARNING: Image for service scoutbot was built because it did not already exist.
To rebuild this image you must use `docker-compose build` or `docker-compose up
--build`.
Creating scout-bot_mysql_1 ... done
Creating scout-bot_scoutbot_1 ... done
riskimidiw@ubuntu-18-04:~/scout-bot$ ./migrate.sh
Migrations run successfully
riskimidiw@ubuntu-18-04:~/scout-bot$
```

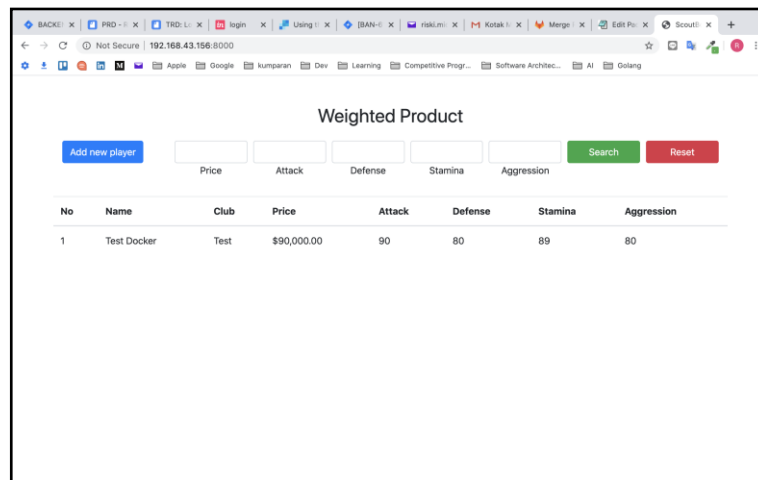
2.5 Hasil Implementasi

Pada sistem yang kami buat ini, telah berhasil kami terapkan kedalamnya teknologi *cloud computing* dan juga telah memanfaatkan sistem *docker* dalam pengerjaannya. Karena dengan menggunakan teknologi *cloud computing*, sistem yang kami buat ini tidak hanya bisa diakses dari perangkat laptop tempat dibuatnya sistem ini. Akan tetapi dapat diakses melalui perangkat-perangkat yang terhubung dengan jaringan lokal yang sama dengan tempat atau perangkat dibuatnya sistem ini.

2.5.1 Hasil Implementasi Proyek Pertama



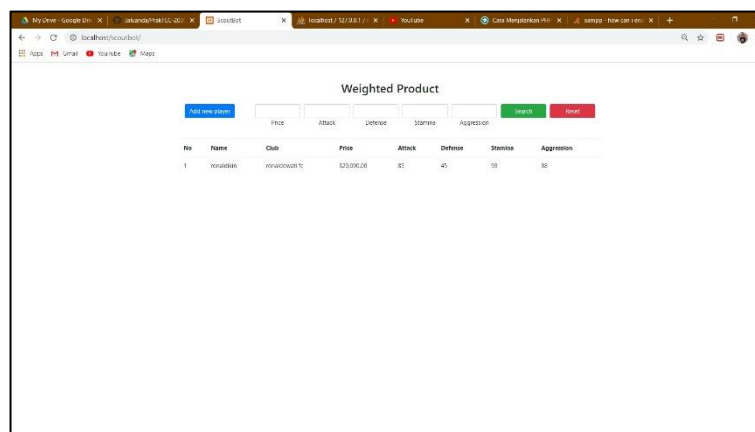
2.5.2 Hasil Implementasi Proyek Kedua



2.6 Pengujian Singkat

Dengan menerapkan teknologi *cloud computing*, kami mencoba membuat suatu sistem pendukung keputusan *transfer* pemain bola yang terintegrasi dengan teknologi *cloud* dan juga kita mencoba memanfaatkan *docker* dalam sistem ini. Dalam pengujian dan pembuatannya kami menggunakan Ubuntu 20.4 dan LAMPP.

Dalam pengerjaannya memang kami menemukan beberapa kendala, akan tetapi kendala yang kami hadapi masih bisa kami selesaikan berkat dokumentasi yang kami buat setiap pertemuan di kelas praktikum. Untuk proyek ini berhasil kami selesaikan sesuai dengan tujuan yang ingin kami capai. Sistem pendukung keputusan yang kami buat telah terintegrasi dengan teknologi *cloud* sehingga bisa diakses dari perangkat lain yang berada dalam 1 jaringan yang sama. Untuk hasil akhir dari sistem ini bisa dilihat pada **Gambar 2.2** berikut ini :



Gambar 2.2 Tampilan Akhir Dari Sistem

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	
		3	2	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
4.	Pengerjaan Proyek Pertama								
5.	Pengerjaan Laporan Progres 1								
6.	Pengerjaan Proyek Kedua								
7.	Memperbaiki Laporan Progres 1								
8.	Pengerjaan Laporan Progres 2								
9.	Memperbaiki dan Finalisasi 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	Midi
2.	Pengujian Singkat	Setiawan
3.	Latar Belakang Masalah	Midi
4.	Agenda Pengerjaan Proyek	Midi
5.	Pengerjaan Proyek Pertama	Setiawan
6.	Pengerjaan Laporan Progres 1	Midi
7.	Perbaikan Laporan Progres 1	Setiawan
8.	Pengerjaan Proyek Kedua	Midi
9.	Pengerjaan Laporan Progres 2	Midi

10.	Perbaikan dan Finalisasi Laporan	Setiawan
-----	----------------------------------	----------

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Pada zaman sekarang ini, segala sesuatu yang berhubungan dengan IT dituntut agar bisa bekerja sebaik mungkin dan mudah diakses oleh banyak orang. Oleh karena itu kami berusaha menghadirkan suatu sistem pendukung keputusan transfer pemain bola dengan memanfaatkan teknologi *cloud computing* dan *docker*.

Dalam pengerjaan proyek ini, kami juga menemukan beberapa kendala kecil seperti *ip* tidak terdeteksi, belum bisa akses internet, dll. Akan tetapi kami berhasil menyelesaikan kendala tersebut dengan menggunakan dokumentasi yang kami buat disetiap pertemuan praktikum *cloud computing*. Kekurangan dari sistem yang kami buat ini adalah karena singkatnya waktu dan ada masalah pada salah satu laptop kami, maka sistem yang kami buat memang belum sempurna. Akan tetapi untuk tujuan yang ingin kami capai dari proyek ini, keduanya sudah berhasil kami capai atau kami selesaikan berkat pembagian tugas yg sesuai

4.2 Saran

Oleh karena sistem yang kami hasilkan belum sempurna, alangkah baiknya para pembaca bisa membantu kami dalam menyempurnakan sistem pendukung keputusan ini. Dan juga karena ini merupakan kerja berkelompok, sangat disarankan untuk selalu berkomunikasi dalam pengerjaannya. Terlebih di masa pandemi ini yang menyarankan seluruh masyarakat untuk sebisa mungkin dirumah saja.

DAFTAR PUSTAKA

Cinderatama, T. A., Yunhasnawa, Y., & Alhamri, R. Z. (2018). Desain Dan Implementasi Hybrid Cloud Computing Sebagai Infrastruktur Untuk Analisis Big Data Menggunakan Analytic Hierarchy Process (AHP). *Techno. Com*, 17(4), 404-414.

Adiputra, F. (2015). Container Dan Docker: Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi Web. *Jurnal Simantec*, 4(3).

Bik, R., & Fadlulloh, M. (2017). Implementasi Docker Untuk Pengelolaan Banyak Aplikasi Web (Studi Kasus: Jurusan Teknik Informatika Unesa). *Jurnal Manajemen Informatika*, 7(2).

Oktavianus, Y. L. (2013). Membangun Sistem Cloud Computing dengan Implementasi Load Balancing dan Pengujian Algoritma Penjadwalan Linux Virtual Server pada FTP Server. *Jurnal Nasional Teknik Elektro*, 2(1), 25-30.

Ashari, A., & Setiawan, H. (2011). Cloud Computing: Solusi ICT?. *Jurnal Sistem Informasi*, 3(2).

LAMPIRAN

Lampiran 1. Script Makefile

```
Makefile

SHELL:=/bin/bash

load_env:
    @export $(cat .env)

migrate:
    @php index.php migration migrate

migration:
ifdef name
    @php index.php migration create $(name)
else
    @echo "[ERROR] : Required name"
    @echo "[USAGE] : make migration name=your_migration"
endif
endif
```

Lampiran 2. File .env

```
SERVER_BASE_URL=http://192.168.43.156:8000/
MYSQL_HOST=mysql
MYSQL_DATABASE=scoutbot
MYSQL_USERNAME=admin
MYSQL_PASSWORD=admin
```

Lampiran 3. File 000-default.conf

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/scoutbot

    # Set apache environment variables
    SetEnv SERVER_BASE_URL ${SERVER_BASE_URL}
    SetEnv MYSQL_HOST ${MYSQL_HOST}
    SetEnv MYSQL_DATABASE ${MYSQL_DATABASE}
    SetEnv MYSQL_USERNAME ${MYSQL_USERNAME}
    SetEnv MYSQL_PASSWORD ${MYSQL_PASSWORD}
</VirtualHost>
```

Lampiran 4. File migrate.sh

```
docker-compose exec scoutbot \sh -c "make migrate"
```

Lampiran 5. File Dockerfile

```
FROM php:7.4.1-apache

COPY . /var/www/html/scoutbot/

COPY ./000-default.conf /etc/apache2/sites-available/

RUN apt-get update && \
    apt-get install -y libicu-dev && \
    docker-php-ext-configure intl && \
    docker-php-ext-install intl && \
    docker-php-ext-install mysqli

RUN a2enmod rewrite
```

Lampiran 6. File docker-composer.yml

```
version: "3.7"

services:
  mysql:
    image: mysql:5.7
    restart: always
    volumes:
      - ~/scoutbot-data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: admin
      MYSQL_DATABASE: scoutbot
      MYSQL_USER: admin
      MYSQL_PASSWORD: admin
    networks:
      - scoutbot-network

  scoutbot:
    build: .
    restart: always
    env_file:
      - .env
    depends_on:
      - mysql
    networks:
      - scoutbot-network
    ports:
      - "8000:80"

networks:
```

scoutbot-network:

