

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**APLIKASI PEMBELIAN TIKET KERETA ONLINE BERBASIS WEB
MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN
DOCKERFILENYA**



DISUSUN OLEH:

**NAMA ANGGOTA : P.H.CAN ENJOY SIMBOLON 123170065
ALIZA HANUM ANGGANI 123170090**

KELAS : B

**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

APLIKASI PEMBELIAN TIKET KERETA ONLINE BERBASIS WEB MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

P.H.Can Enjoy Simbolon

123170065

Aliza Hanum Anggani

123170090

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. 12160119

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 Allah SWT atas segala limpahan rahmat, kemudahan, dan karunia-Nya, sehingga kelompok kami dapat menyelesaikan laporan akhir praktikum Teknologi Cloud Computing ini tanpa adanya halangan yang berarti. Penulisan laporan akhir ini disusun dengan sistematika penulisan laporan akhir yang sesuai pada panduan.

Laporan akhir ini merupakan tanggung jawab dan tugas kami dalam menjalani kuliah Informatika. Laporan ini berisi tentang laporan proyek akhir Praktikum Teknologi Cloud Computing kami yang berjudul “Aplikasi Pembelian Tiket Kereta Online Berbasis Web Menggunakan Ubuntu Lamppp Dan Proses Pembuatan Dockerfilenya”.

Laporan ini tentu saja masih jauh dari kata sempurna. Laporan ini masih jauh dari yang diharapkan untuk memenuhi tugas akhir praktikum Teknologi Cloud Computing, oleh karena itu kritik serta saran yang membangun masih kami harapkan untuk penyempurnaan laporan akhir ini.

Akhir kata semoga laporan ini berguna dan bermanfaat bagi semua pihak dan penyusun sendiri pada khususnya. Atas perhatian dari semua pihak yang membantu penulisan laporan proyek akhir ini kami 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 | 4 |
| 2.1 Komponen yang Digunakan | 4 |
| 2.2 Rancangan Arsitektur <i>Cloud Computing</i> | 6 |
| 2.3 Parameter dan Konfigurasi | 6 |
| 2.4 Tahap Implementasi..... | 13 |
| 2.5 Hasil Implementasi | 20 |
| 2.6 Pengujian Singkat | 21 |
| BAB III JADWAL Pengerjaan dan Pembagian Tugas | 26 |
| 3.1 Agenda Pengerjaan | 26 |
| 3.2 Keterangan Pembagian Tugas..... | 26 |
| BAB IV KESIMPULAN DAN SARAN | 27 |
| 4.1 Kesimpulan | 27 |
| 4.2 Saran | 27 |
| DAFTAR PUSTAKA | 28 |
| LAMPIRAN..... | 29 |

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Cloud computing berawal pada tahun 1960-an, dimana John McCarthy, pakar komputasi MIT yang dikenal juga sebagai salah satu pionir intelegensi buatan, menyampaikan pendapat bahwa “komputerisasi pada suatu saat akan diorganisasikan sebagai sebuah kebutuhan masyarakat”. *Cloud computing* dipicu pertama kali ketika munculnya konsep Web 2.0, dimana pengguna bisa saling berinteraksi. Hal ini memungkinkan karena *bandwidth* yang bisa digunakan oleh masyarakat sudah semakin besar. Dan hal ini menjadi sebuah peluang oleh *developer* untuk membuat suatu sistem komputasi yang terdistribusi secara masal. Diawali oleh kemunculan situs www.salesforce.com pada awal 1999 yang menyediakan aplikasi *enterprise* melalui sebuah web yang sangat simple, pengguna menggunakan sistem langganan untuk bisa mengakses aplikasi di salesforce.com ini.

Cloud Computing (komputasi awan) adalah teknologi yang memanfaatkan layanan internet menggunakan pusat server yang bersifat virtual dengan tujuan pemeliharaan data dan aplikasi (Indra, 2010). Komputasi awan terdiri dari tiga elemen dasar yang menyusunnya (Mitchell, 2010), yaitu *Software as a Service (SaaS)*, *Platform as a Service (PaaS)*, dan *Infrastructure as a Service (IaaS)*. SaaS merupakan layanan cloud computing dimana pelanggan dapat menggunakan perangkat lunak yang disediakan oleh provider. Pengguna tidak perlu membeli lisensi perangkat lunak lagi, cukup dengan berlangganan dan membayar sesuai dengan permintaan. Contohnya: Office365, Adobe Creative Cloud, Facebook, Twitter, YahooMessenger, Skype, Gtalk, dan lainnya. PaaS merupakan layanan yang menyediakan hardware sehingga pengembang aplikasi tidak perlu memikirkan operating system, infrastructure scaling, load balancing dan lainnya. Pengembang dapat fokus pada aplikasi yang akan dikembangkan karena “tempat” untuk aplikasi sudah menjadi tanggung jawab provider. Contohnya: Microsoft Azure. IaaS

merupakan layanan yang menyediakan sumber daya teknologi informasi dasar yang dapat digunakan oleh penyewa untuk menjalankan aplikasi yang dimilikinya. Model ini seperti penyedia data center yang menyewakan ruangan, tapi ini lebih ke level mikronya. Keuntungan jenis layanan ini adalah kita tidak perlu membeli komputer fisik tetapi kita dapat melakukan konfigurasi komputer virtual yang dapat diubah dengan mudah. Contohnya: Amazon EC2, Windows Azure, dsb (Cancer & Alim, 2016). Manfaat cloud computing diantaranya adalah :

1. Semua Data Tersimpan di Server Secara Terpusat
2. Menjamin Keamanan Data Penggunaanya.
3. Fleksibilitas dan Skalabilitas yang Tinggi
4. Memberikan Jaminan Investasi Jangka Panjang

Proyek yang berjudul “Aplikasi Pembelian Tiket Kereta Online Berbasis Web Menggunakan Ubuntu Lampp Dan Proses Pembuatan Dockerfilenya” ini adalah aplikasi yang bertujuan untuk memudahkan pemesanan tiket kereta. Pengguna dapat membuat akun, melihat jadwal kereta, memesan tiket kereta, hingga memilih metode pembayaran. Proyek ini diasumsikan akan mengambil database dari PT KAI yang tentunya melibatkan proses cloud computing di dalamnya.

Tahap penyelesaian proyek akhir ini yang pertama menyiapkan dan menginstall software yang akan digunakan seperti VMWare Workstation, ISO Ubuntu, PuTTY, WinSCP, dll. Setelah itu dilakukan instalasi PHP-Apache-MySQL untuk menjalankan basis data dari program. Lalu memindahkan file web yang telah dibuat ke dalam LAMPP dan mengecek seluruh fungsionalitasnya. Setelah itu dari tahapan yang sudah dilaksanakan, akan dilanjutkan dengan pembuatan dockerfile untuk menciptakan *container* yang lebih ringan untuk diakses.

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 Aplikasi Pembelian Tiket Kereta Online yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.0.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Menjalankan Aplikasi Pembelian Tiket Kereta Online berbasis web di Ubuntu Server (LAMPP).

3. Membuat *dockerfile* untuk Aplikasi Pembelian Tiket Kereta Online berbasis web agar aplikasi tersebut dapat dibangun, dikemas, dan dijalankan di berbagai lokasi sebagai sebuah *container* yang ringan.

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. PT KAI 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 terdistribusi dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.
5. Admin atau developer dapat membangun, mengemas, dan menjalankan aplikasi di berbagai lokasi sebagai sebuah *container* yang ringan dengan dibuatnya *dockerfile*.

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari Aplikasi Pembelian Tiket Kereta Online untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Menentukan konfigurasi yang tepat untuk pembuatan *dockerfile* 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.

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. Proyek yang pertama ialah “Aplikasi Pembelian Tiket Kereta Api Online berbasis web menggunakan Ubuntu LAMPP” dan proyek yang kedua adalah “Proses Pembuatan Dockerfilenya”. Berikut akan dibahas keduanya masing-masing dalam sub bab terkait.

2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Aplikasi Pembelian Tiket Kereta Online Berbasis Web 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 7.3.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data jadwal kereta, harga tiket, dan data penumpang kereta yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh orang yang ingin membeli tiket kereta api secara *online*. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *public*, dapat diakses secara bebas oleh semua orang.
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 Workstation | Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> . |
| 2. | Prosesor | 14ISK Intel Core i3-6006U CPU @ 2GHz | Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> . |
| 3. | Konfigurasi Jaringan <i>Guest OS</i> | Mode Bridge | Mode adapter jaringan VM <i>guest</i> yang digunakan. |
| | | IP: 192.168.138.128/22 | Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> . |
| | | DNS: 192.168.138.0 | Alamat IP untuk DNS <i>guest OS</i> . |
| | | GW: 192.168.138.0 | Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar. |
| 4. | Versi Ubuntu | Ubuntu 18.04.3 LTS | ISO Ubuntu yang digunakan untuk <i>guest OS</i> . |
| 5. | RAM | 1 GiB Memory | Alokasi RAM 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 2.0 | Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS. |
| | | PHP 7.2.24 | Prosesor konten Web Server |
| | | MySQL 5.2 | Basis layanan manajemen basis data |
| | | PHPMyAdmin | Layanan manajemen database MySQL berbasis Web GUI |
| 2. | PHP | Versi 7.3 | Dukungan bahasa pemrograman yang digunakan oleh aplikasi pembelian tiket kereta api online |

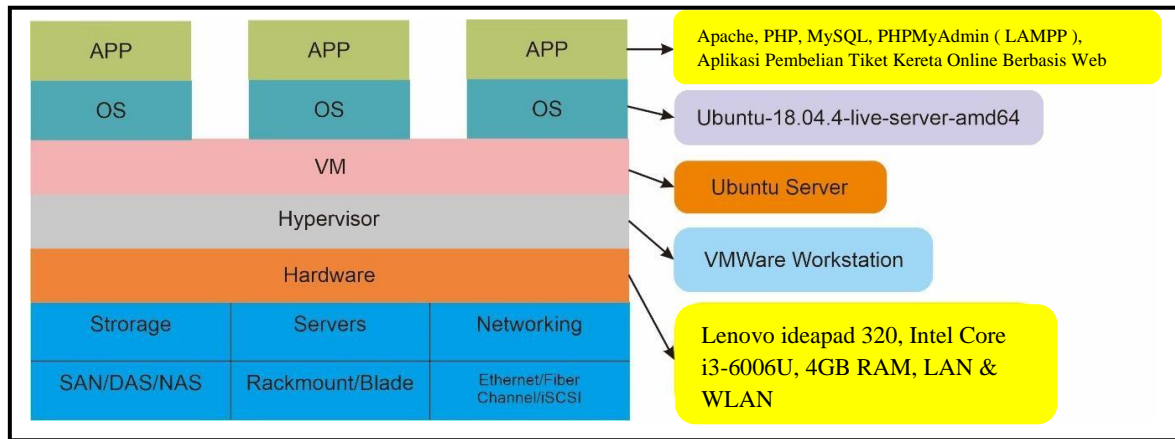
2.1.2 Komponen pada Proyek Kedua

Tabel 2.3 Spesifikasi Docker untuk proyek kedua

| No. | Nama Parameter | Nilai | Keterangan |
|-----|----------------|---------------------|--|
| 1. | Docker Engine | Docker Server 19.03 | <i>Docker engine server</i> yang digunakan dalam menjalankan container. |
| | | Docker Client 19.03 | <i>Docker engine client</i> yang digunakan untuk berkomunikasi dengan <i>docker server</i> . |
| 2. | Docker Compose | Docker Compose 1.17 | Dukungan alat untuk mendefinisikan dan menjalankan <i>multi-container</i> . |

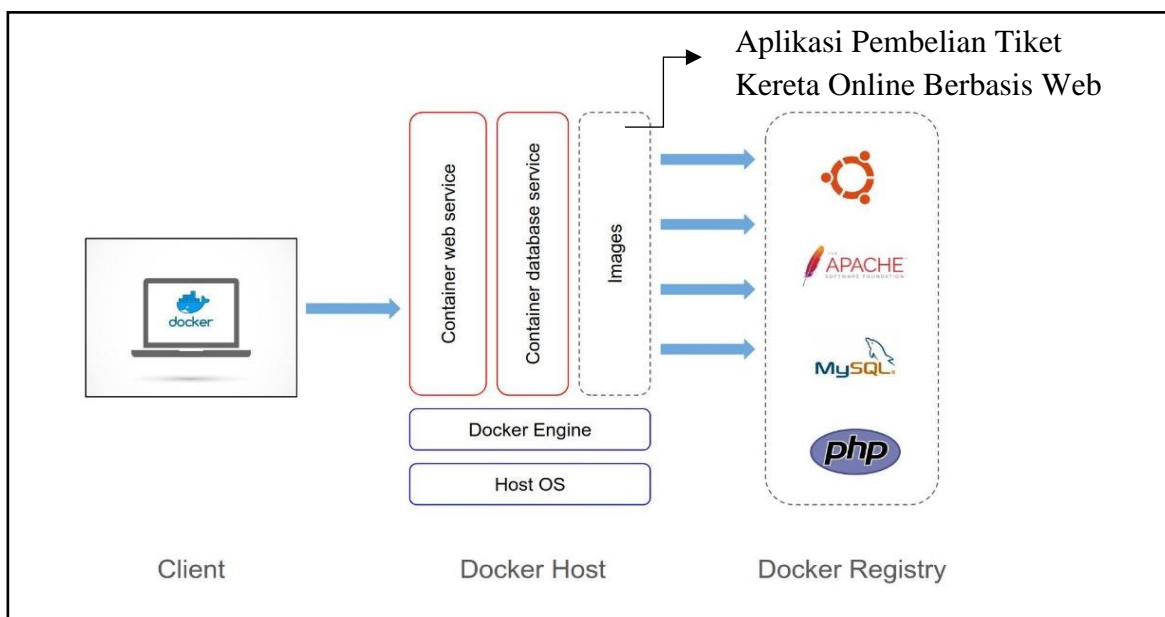
2.2 Rancangan Arsitektur Cloud Computing

Pada proyek akhir ini digunakan bentuk rancangan arsitektur IaaS di mana hardware laptop Lenovo ideapad 320 sebagai layer utama di bagian bawah, kemudian di atasnya adalah SaaS atau sistem operasi Windows dan aplikasi VMware Workstation yang berjalan.



Gambar 2.1 Arsitektur Ubuntu LAMPP

Selain menggunakan ubuntu LAMPP, pada proyek ini juga dilakukan proses pembuatan dockerfile yang bertujuan untuk membuat aplikasi sebagai *container* yang lebih ringan. Pada hal ini, sistem atau aplikasi kita terletak pada bagian *image*. Pengaturan software atau aplikasi beserta file dan hal pendukung lainnya akan menjadi sebuah *image* (istilah oleh docker). Arsitektur untuk docker dapat dilihat pada **Gambar 2.2** berikut.



Gambar 2.2 Arsitektur Docker

2.3 Parameter dan Konfigurasi

2.3.1 Parameter dan Konfigurasi Proyek Pertama

Untuk dapat digunakan, maka VMware Workstation perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada **Tabel 2.3** berikut ini

Tabel 2.3 Konfigurasi pada VMware Workstation yang digunakan

| No. | Nama Parameter | Nilai | Keterangan |
|-----|----------------------------|--------------------------------------|---|
| 1. | VMware ESXi <i>version</i> | 15.5.2 | Keterangan versi yang digunakan. |
| 2. | VMware ESXi <i>build</i> | 15785246 | Keterangan <i>build (patch) number</i> . |
| 3. | IPv4 <i>hypervisor</i> | IP: 192.168.138.128 (Static) | IP <i>number</i> untuk akses manajemen <i>hypervisor</i> melalui <i>web based</i> . Pengaturan IP <i>address</i> menggunakan mode statik. |
| | | SM: 255.255.255.0 | Kelas IP/ <i>subnet mask</i> yang digunakan. |
| | | DNS: 192.168.138.0 | Alamat IP untuk fungsionalitas DNS. |
| | | GW: 192.168.138.0 | Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar. |
| 4. | <i>Processor info</i> | 14ISK Intel Core i3-6006U CPU @ 2GHz | Jenis <i>processor</i> yang digunakan pada <i>hypervisor</i> . |
| 5. | <i>RAM info</i> | 1 GiB Memory | Kapasitas RAM pada <i>hypervisor</i> . |

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 2.2 berikut ini:

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

Keterangan:

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

Modul 2.2 Parameter instalasi MySQL

Pada tahap instalasi MySQL, terdapat error “can’t connect to local MySQL server through socket ‘/var/run/mysqld/mysqld.sock(2)’” hal ini lalu diatasi dengan meng un install MySQL dengan perintah “apt purge mysql*” lalu menginstall ulang MySQL dengan perintah yang sudah disebutkan diatas.

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 libapache2-mod-php php-mysql : nama paket aplikasi untuk PHP

Modul 2.3 Parameter instalasi PHP

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

```
$ sudo apt install php 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
- php phpmyadmin php-mbstring php-gettext : nama paket aplikasi untuk PHPMyAdmin

Modul 2.4 Parameter instalasi PHPMyAdmin

Parameter yang digunakan untuk mengatur parameter plugin dari user root dapat dilihat pada penjelasan **Modul 2.5** berikut ini:

```
UPDATE mysql.user SET plugin = 'mysql_native_password',  
authentication_string = PASSWORD ('hanum123') WHERE User = 'root';
```

Modul 2.5 Parameter pengaturan parameter plugin dari user root

Setelah langkah diatas dilakukan, untuk dapat mengakses phpmyadmin diperlukan login terlebih dahulu, namun saat login muncul error “#1698 – Access denied for user ‘root’@localhost” hal ini diatasi dengan menjalankan perintah “Flush privileges;”

2.3.2 Parameter dan Konfigurasi pada Proyek Kedua

Langkah pertama dalam install docker adalah meng update list packages. Parameter yang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

```
sudo apt update
```

Modul 2.6 Parameter update list packages

Langkah kedua dalam install docker adalah install beberapa package yang diperlukan agar apt bisa diakses HTTPS. Parameter yang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

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

Modul 2.7 Parameter package apt

Langkah ketiga dalam install docker adalah menambahkan GPG key dari untuk Docker repository ke sistem kita. Parameter yang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.8** berikut ini:

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

Modul 2.8 Parameter menambahkan GPG key dari untuk Docker repository ke sistem

Langkah keempat dalam install docker adalah menambahkan Docker repository ke APT. Parameter yang digunakan untuk menambahkan docker repository ke APT dapat dilihat pada penjelasan **Modul 2.9** berikut ini:

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

Modul 2.9 Parameter menambahkan Docker repository ke APT

Langkah kelima dalam install docker adalah update package database. Parameter yang digunakan untuk meng update package database dapat dilihat pada penjelasan **Modul 2.10** berikut ini:

```
sudo apt update
```

Modul 2.10 Parameter update package database

Langkah keenam dalam install docker pastikan kita akan menginstall Docker repo ke dalam ubuntu. Parameter yang digunakan untuk meng update list packages dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

```
apt-cache policy docker-ce
```

Modul 2.11 Parameter memastikan docker repo

Langkah ketujuh dalam install docker adalah install docker itu sendiri. Parameter yang digunakan untuk menginstall docker dapat dilihat pada penjelasan **Modul 2.12** berikut ini:

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

Modul 2.12 Parameter untuk install docker

Langkah kedelapan dalam install docker adalah memastikan bahwa docker sudah berjalan. Parameter yang digunakan untuk memastikan bahwa docker sudah berjalan dapat dilihat pada penjelasan **Modul 2.13** berikut ini:

```
sudo systemctl status docker
```

Modul 2.13 Parameter untuk memastikan docker sudah berjalan

Setelah docker berjalan, harus menginstall docker compose. Parameter yang digunakan untuk *download docker-compose binary file* dapat dilihat pada penjelasan **Modul 2.14** 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
```

Modul 2.14 Parameter untuk *download docker-compose binary file*

Parameter yang digunakan untuk install curl dapat dilihat pada penjelasan **Modul 2.15** berikut ini:

```
sudo apt install curl -y
```

Modul 2.15 Parameter untuk install curl

Parameter yang digunakan untuk install docker compose dapat dilihat pada penjelasan **Modul 2.16** berikut ini:

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

Modul 2.16 Parameter untuk install docker compose

Parameter yang digunakan untuk memastikan docker compose sudah berjalan dengan baik dapat dilihat pada penjelasan **Modul 2.17** berikut ini:

```
docker-compose version
```

Modul 2.17 Parameter untuk memastikan docker compose sudah berjalan

Setelah docker dan docker compose sudah dipastikan berjalan dengan baik, tahap selanjutnya adalah mengatur docker-compose untuk proyek yang akan dikerjakan.

Parameter yang digunakan untuk masuk ke folder proyek dapat dilihat pada penjelasan **Modul 2.18** berikut ini:

```
cd /var/www/html/projek/
```

Modul 2.18 Parameter untuk masuk ke folder proyek

Parameter yang digunakan untuk membuat Dockerfile dapat dilihat pada penjelasan **Modul 2.19** berikut ini:

```
nano Dockerfile
```

Modul 2.19 Parameter membuat Dockerfile

Selain file dengan nama dockerfile, juga dibutuhkan file bernama docker-compose.yml.

Parameter yang digunakan untuk membuat file docker-compose.yml dapat dilihat pada penjelasan **Modul 2.20** berikut ini:

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

Modul 2.20 Parameter membuat file docker-compose.yml

Parameter yang digunakan menjalankan web-server, mysql-server, dan phpmyadmin dapat dilihat pada penjelasan **Modul 2.21** berikut ini:

```
docker-compose up
```

Modul 2.21 Parameter menjalankan web-server, mysql-server, dan phpmyadmin

Pada saat menjalankan perintah diatas, ditemukan *error* yaitu "Couldn't reliably determine the servers fully qualified domain" dan hal ini diatasi dengan menjalankan perintah "echo "Servername localhost" | sudo tee /etc/apache2/conf-available/fqdn.conf".

Kendala selanjutnya adalah "Can't serve directory;no matching directory index" dan diatasi dengan mengubah file tampilan awal php yang semula bernama "beranda.php" menjadi "index.php"

Langkah terakhir adalah mengimport *database* kereta.sql ke dalam docker, dengan perintah yang dapat dilihat pada **Modul 2.22**, **Modul 2.23** dan **Modul 2.24** dibawah ini

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

Modul 2.22 Parameter mengimport *database*

```
docker ps
```

Modul 2.23 Parameter mengimport *database*

```
docker exec -i kereta-db mysql -uahanum -phanum123 kereta <./kereta.sql
```

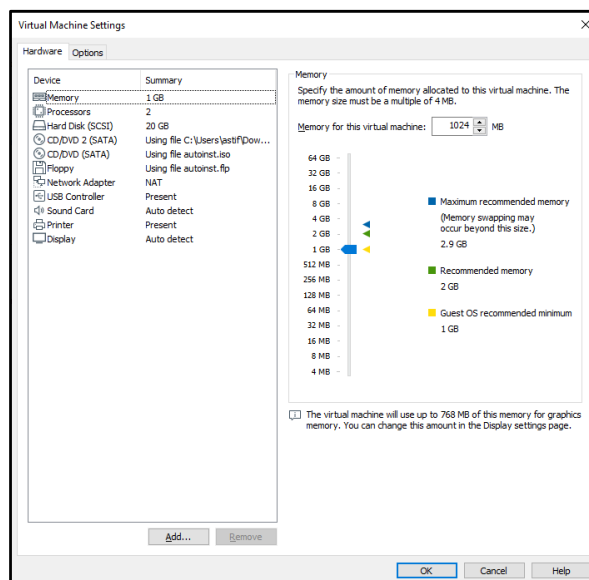
Modul 2.24 Parameter mengimport *database*

Pada langkah terakhir ini terdapat *error* yang berbunyi "ERROR 1050 (42S01) at line 31: Table 'akun' already exists"

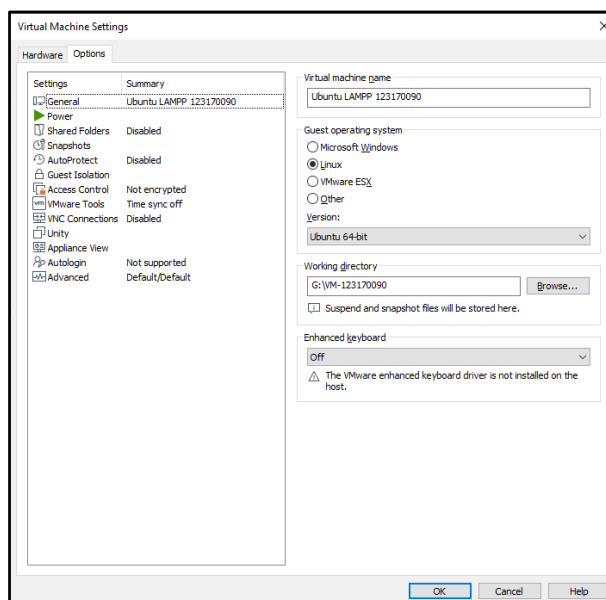
2.4 Tahap Implementasi

2.4.1 Tahap Implementasi Proyek Pertama

Hasil dari implementasi **Tabel 2.1** mengenai konfigurasi untuk VMware Workstation dapat dilihat pada **Gambar 2.3** dan **Gambar 2.4** berikut ini :

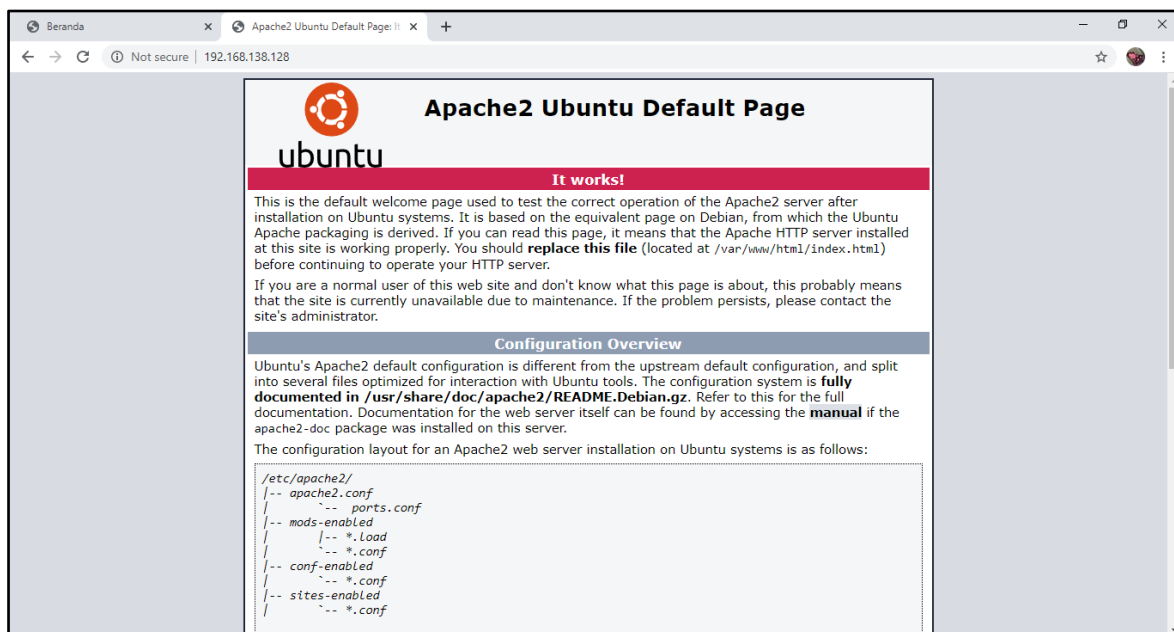


Gambar 2.3 Konfigurasi VMWare Workstation



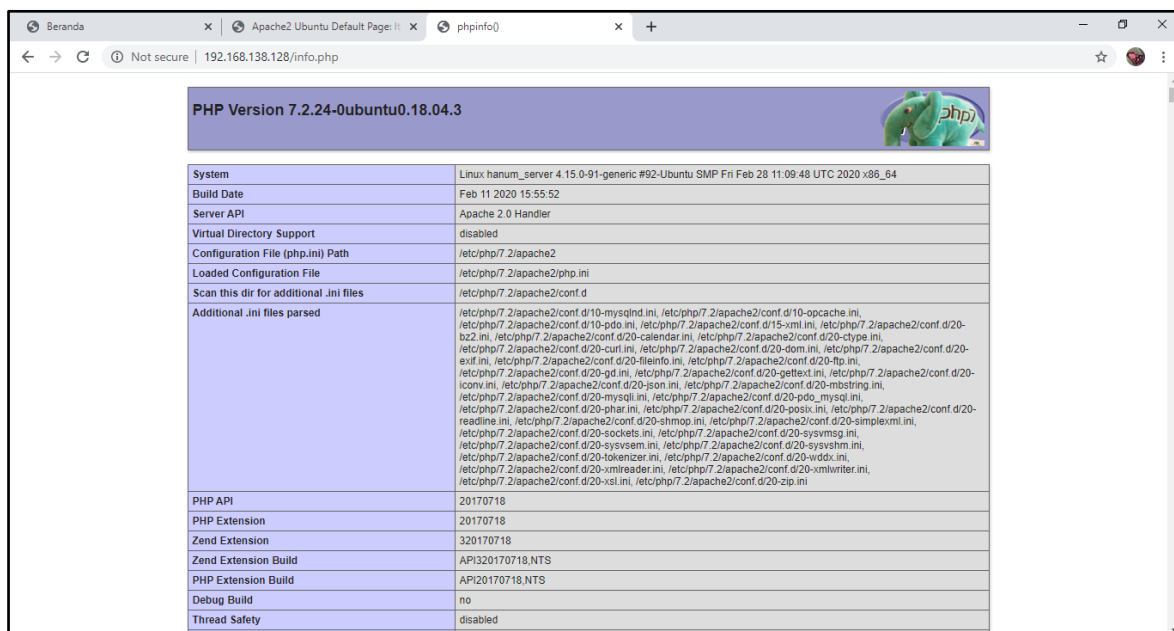
Gambar 2.4 Konfigurasi VMWare Workstation

Hasil dari implementasi **Modul 2.1** mengenai instalasi Apache dapat dilihat pada **Gambar 2.5** berikut ini :



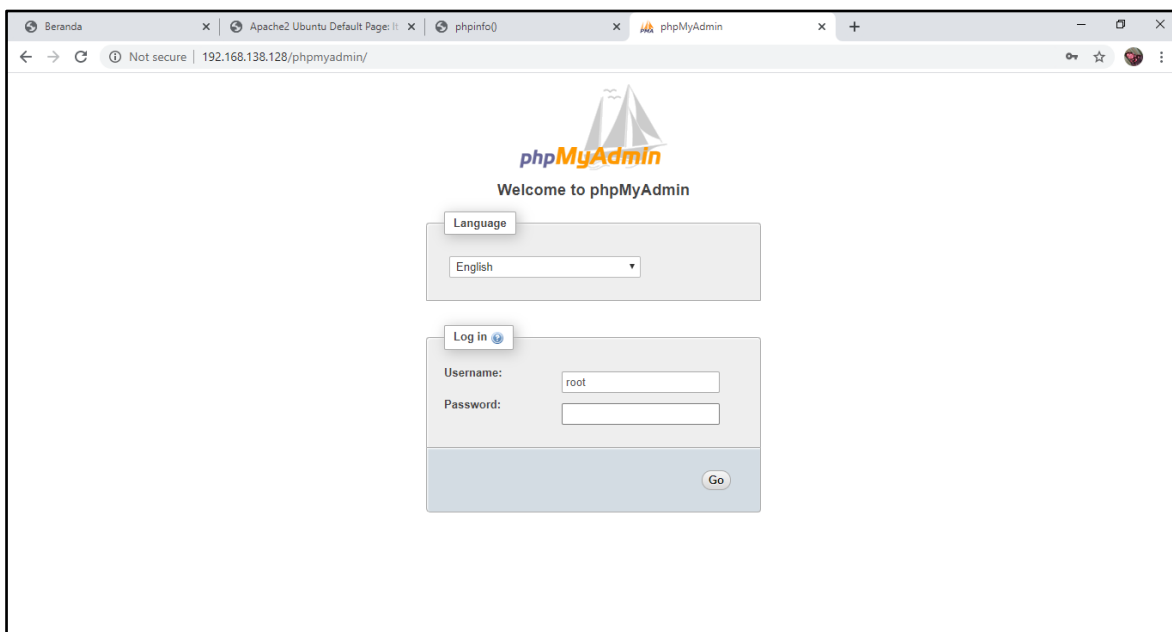
Gambar 2.5 Tampilan Apache setelah Ter-Install

Hasil dari implementasi **Modul 2.3** mengenai instalasi PHP dapat dilihat pada **Gambar 2.6** berikut ini :



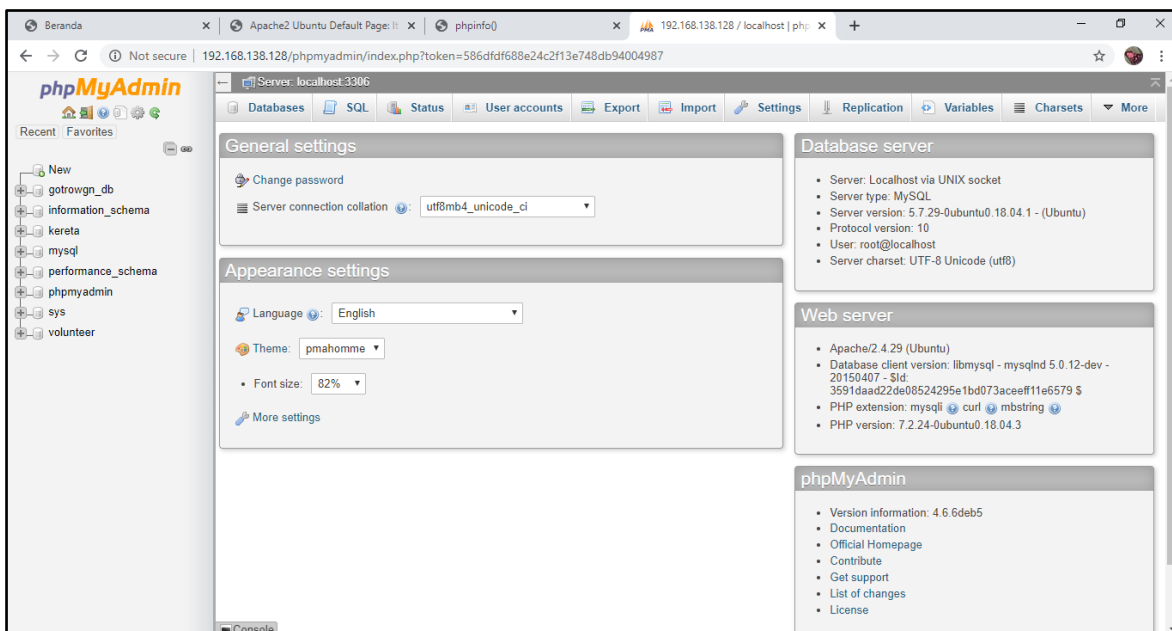
Gambar 2.6 Tampilan PHP setelah Ter-Install

Hasil dari implementasi **Modul 2.4** mengenai instalasi PHPMyAdmin dapat dilihat pada **Gambar 2.7** berikut ini :



Gambar 2.7 Tampilan PHPMyAdmin setelah Ter-Install

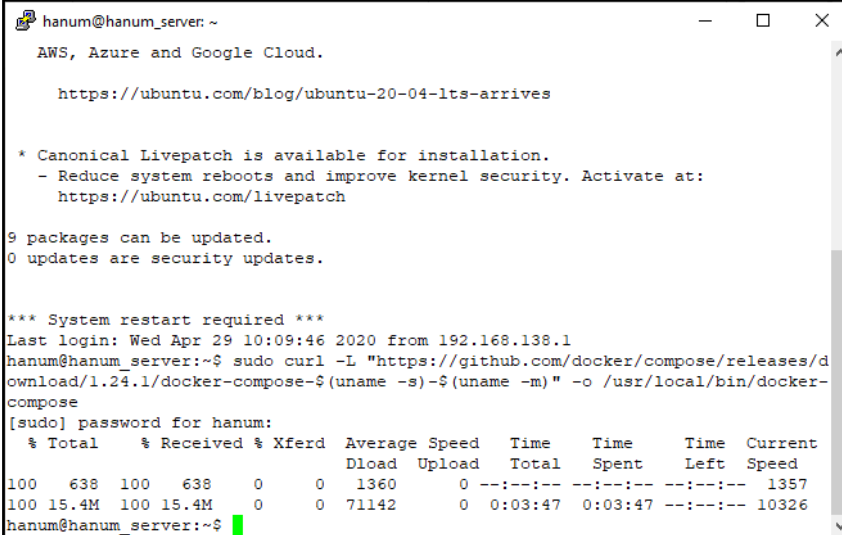
Hasil dari implementasi **Modul 2.5** mengenai pengaturan parameter plugin dari user root dapat dilihat pada **Gambar 2.8** berikut ini :



Gambar 2.8 Tampilan PHPMYAdmin setelah Berhasil Login

2.4.2 Tahap Implementasi Proyek Kedua

Hasil dari implementasi **Modul 2.14** mengenai download docker compose dapat dilihat pada **Gambar 2.9** berikut ini :



```

hanum@hanum_server: ~
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:
https://ubuntu.com/livepatch

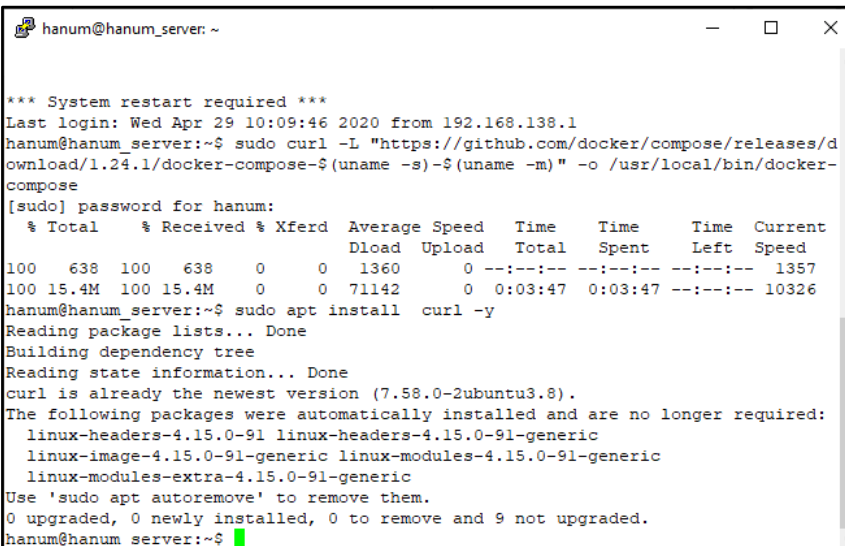
9 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Wed Apr 29 10:09:46 2020 from 192.168.138.1
hanum@hanum_server:~$ 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
[sudo] password for hanum:
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100  638    100  638    0     0   1360      0  --:--:-- --:--:-- --:--:--   1357
100 15.4M    100 15.4M    0     0  71142      0  0:03:47  0:03:47 --:--:--  10326
hanum@hanum_server:~$

```

Gambar 2.9 Tampilan PUTTY saat selesai download docker compose

Hasil dari implementasi **Modul 2.15** mengenai install curl dapat dilihat pada **Gambar 2.10** berikut ini :



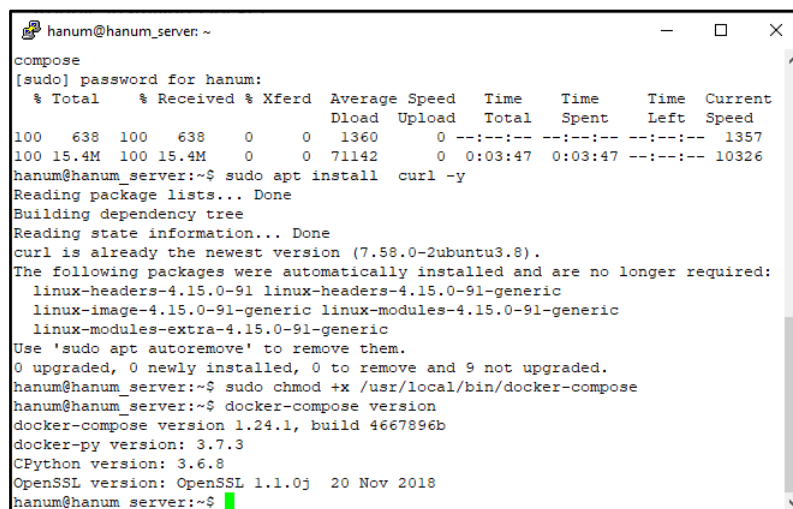
```

hanum@hanum_server: ~
*** System restart required ***
Last login: Wed Apr 29 10:09:46 2020 from 192.168.138.1
hanum@hanum_server:~$ 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
[sudo] password for hanum:
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100  638    100  638    0     0   1360      0  --:--:-- --:--:-- --:--:--   1357
100 15.4M    100 15.4M    0     0  71142      0  0:03:47  0:03:47 --:--:--  10326
hanum@hanum_server:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-91 linux-headers-4.15.0-91-generic
  linux-image-4.15.0-91-generic linux-modules-4.15.0-91-generic
  linux-modules-extra-4.15.0-91-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
hanum@hanum_server:~$

```

Gambar 2.10 Tampilan PUTTY saat selesai install curl

Hasil dari implementasi **Modul 2.17** yang berisi cara memastikan docker compose sudah terinstall dapat dilihat pada **Gambar 2.11** berikut ini :



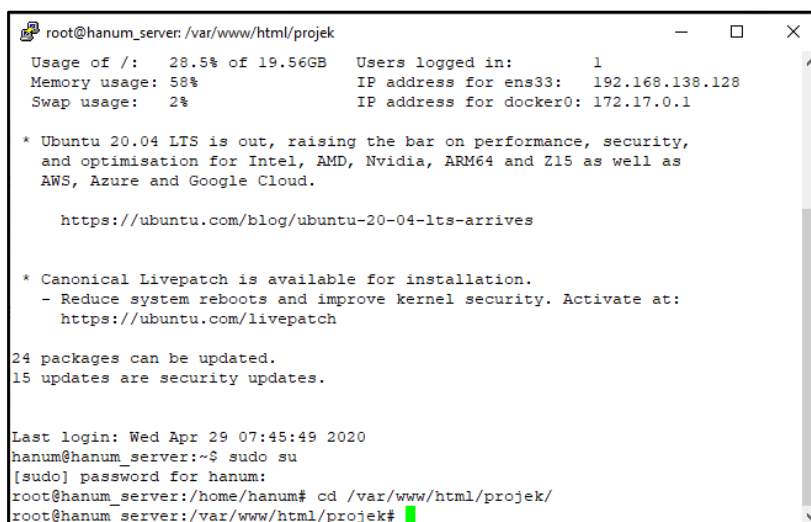
```

hanum@hanum_server: ~
compose
[sudo] password for hanum:
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100    638    100    638    0     0   1360      0  --:--:-- --:--:-- --:--:--   1357
100 15.4M    100 15.4M    0     0  71142      0  0:03:47 0:03:47 --:--:-- 10326
hanum@hanum_server:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-91 linux-headers-4.15.0-91-generic
  linux-image-4.15.0-91-generic linux-modules-4.15.0-91-generic
  linux-modules-extra-4.15.0-91-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
hanum@hanum_server:~$ sudo chmod +x /usr/local/bin/docker-compose
hanum@hanum_server:~$ docker-compose version
docker-compose version 1.24.1, build 4667896b
docker-py version: 3.7.3
CPython version: 3.6.8
OpenSSL version: OpenSSL 1.1.0j  20 Nov 2018
hanum@hanum_server:~$

```

Gambar 2.11 Tampilan yang menunjukkan bahwa docker-compose sudah terinstall

Hasil dari implementasi **Modul 2.18** yang berisi cara masuk ke dalam folder proyek dapat dilihat pada **Gambar 2.12** berikut ini :



```

root@hanum_server: /var/www/html/projek
Usage of /: 28.5% of 19.56GB
Memory usage: 58%
Swap usage: 2%
Users logged in: 1
IP address for ens33: 192.168.138.128
IP address for docker0: 172.17.0.1

* 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.

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

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

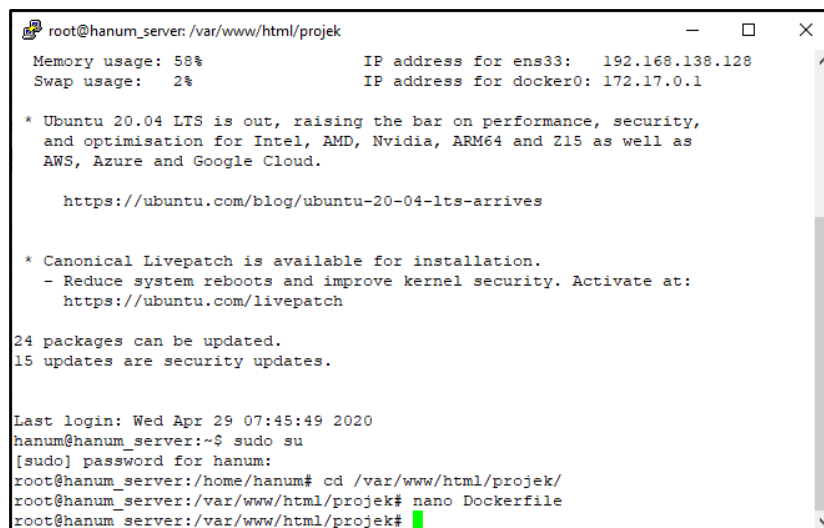
24 packages can be updated.
15 updates are security updates.

Last login: Wed Apr 29 07:45:49 2020
hanum@hanum_server:~$ sudo su
[sudo] password for hanum:
root@hanum_server:/home/hanum# cd /var/www/html/projek/
root@hanum_server:/var/www/html/projek#

```

Gambar 2.12 Tampilan masuk ke folder proyek

Hasil dari implementasi **Modul 2.19** yang berisi cara membuat Dockerfile dan yang diisikan ke dalam file dockerfile dapat dilihat pada **Gambar 2.13** dan **Gambar 2.14** berikut ini :



```

root@hanum_server: /var/www/html/projek
Memory usage: 58%          IP address for ens33: 192.168.138.128
Swap usage: 2%            IP address for docker0: 172.17.0.1

* 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.

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

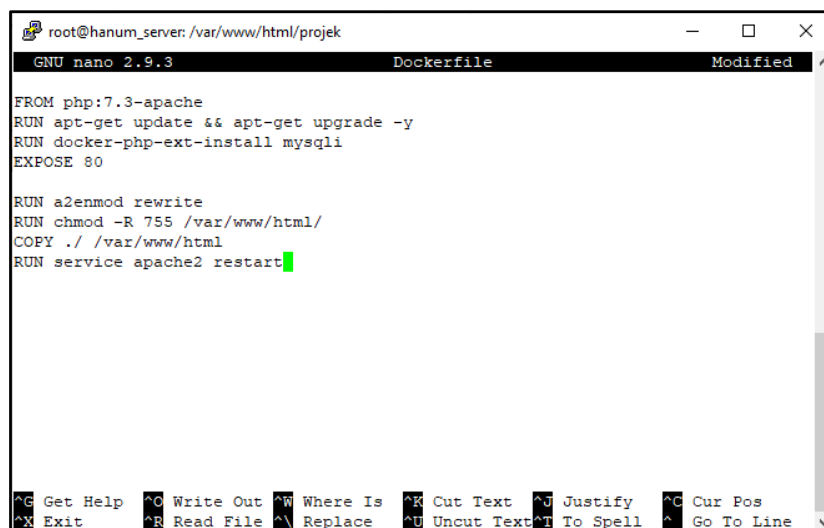
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

24 packages can be updated.
15 updates are security updates.

Last login: Wed Apr 29 07:45:49 2020
hanum@hanum_server:~$ sudo su
[sudo] password for hanum:
root@hanum_server:/home/hanum# cd /var/www/html/projek/
root@hanum_server:/var/www/html/projek# nano Dockerfile
root@hanum_server:/var/www/html/projek#

```

Gambar 2.13 Membuat Dockerfile



```

GNU nano 2.9.3      Dockerfile      Modified
FROM php:7.3-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli
EXPOSE 80

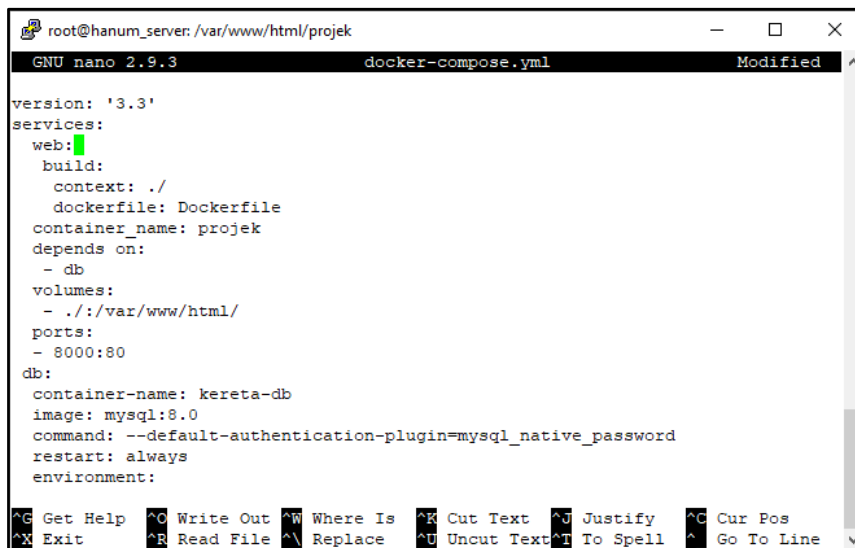
RUN a2enmod rewrite
RUN chmod -R 755 /var/www/html/
COPY ./ /var/www/html
RUN service apache2 restart

```

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

Gambar 2.14 Isi dockerfile

Hasil dari implementasi **Modul 2.20** yang berisi cara membuat file dockerfile-compose.yml dan yang diisikan ke dalam file dockerfile-compose.yml dapat dilihat pada **Gambar 2.15** berikut ini :



```

root@hanum_server: /var/www/html/projek
GNU nano 2.9.3      docker-compose.yml      Modified

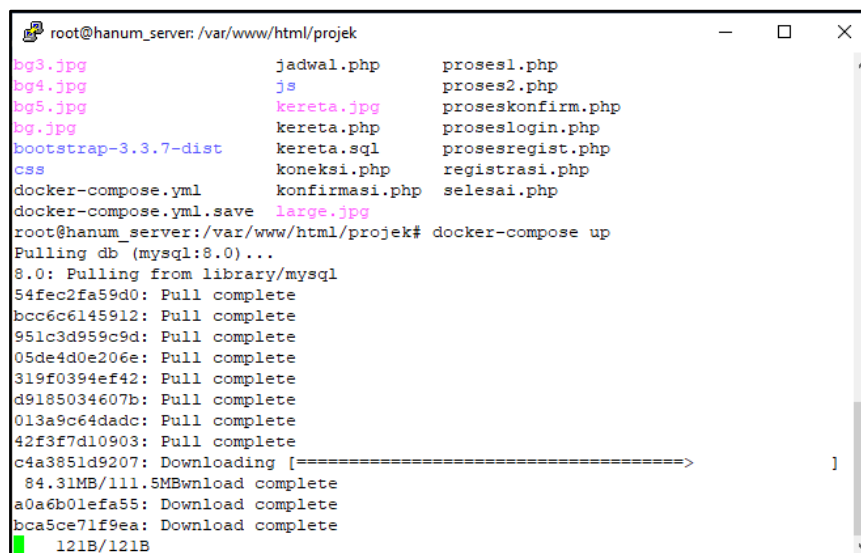
version: '3.3'
services:
  web:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: projek
    depends on:
      - db
    volumes:
      - ../var/www/html/
    ports:
      - 8000:80
  db:
    container-name: kereta-db
    image: mysql:8.0
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:

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

```

Gambar 2.15 Isi file docker-compose.yml

Hasil dari implementasi **Modul 2.21** yang berisi cara menjalankan web-server,mysql-server, dan phpmyadmin dapat dilihat pada **Gambar 2.16** berikut ini :



```

root@hanum_server: /var/www/html/projek

bg3.jpg          jadwal.php       proses1.php
bg4.jpg          js               proses2.php
bg5.jpg          kereta.jpg       proseskonfirmasi.php
bg.jpg           kereta.php       proseslogin.php
bootstrap-3.3.7-dist kereta.sql        prosesregist.php
css              koneksi.php      registrasi.php
docker-compose.yml konfirmasi.php     selesai.php
docker-compose.yml.save large.jpg

root@hanum_server: /var/www/html/projek# docker-compose up
Pulling db (mysql:8.0)...
8.0: Pulling from library/mysql
54fec2fa59d0: Pull complete
bcc6c6145912: Pull complete
951c3d959c9d: Pull complete
05de4d0e206e: Pull complete
319f0394ef42: Pull complete
d9185034607b: Pull complete
013a9c64dad3: Pull complete
42f3f7d10903: Pull complete
c4a3851d9207: Downloading [=====]
 84.31MB/111.5MBwnload complete
a0a6b01efa55: Download complete
bca5ce71f9ea: Download complete
 121B/121B

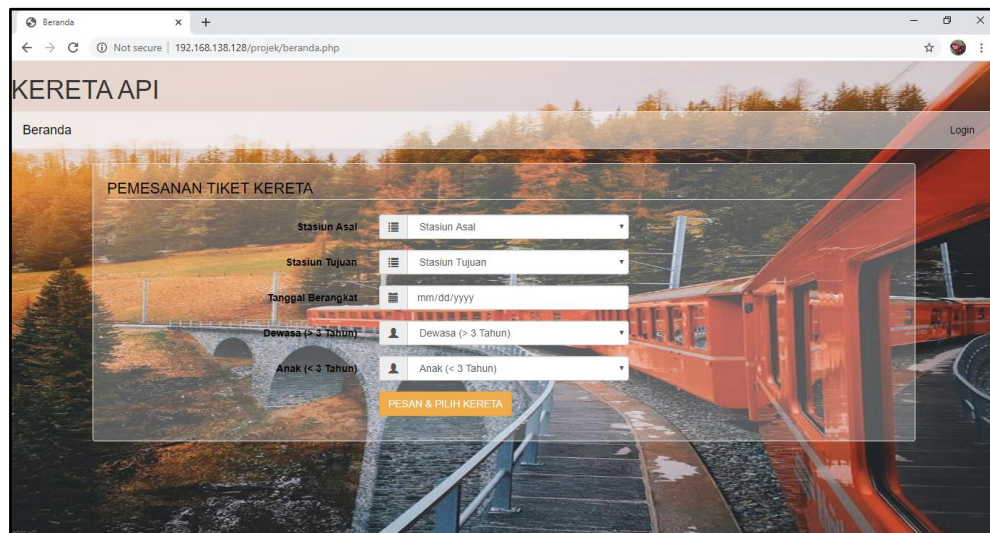
```

Gambar 2.16 Menjalankan web-server, mysql-server, dan phpmyadmin

2.5 Hasil Implementasi

2.5.1 Hasil Implementasi Proyek Pertama

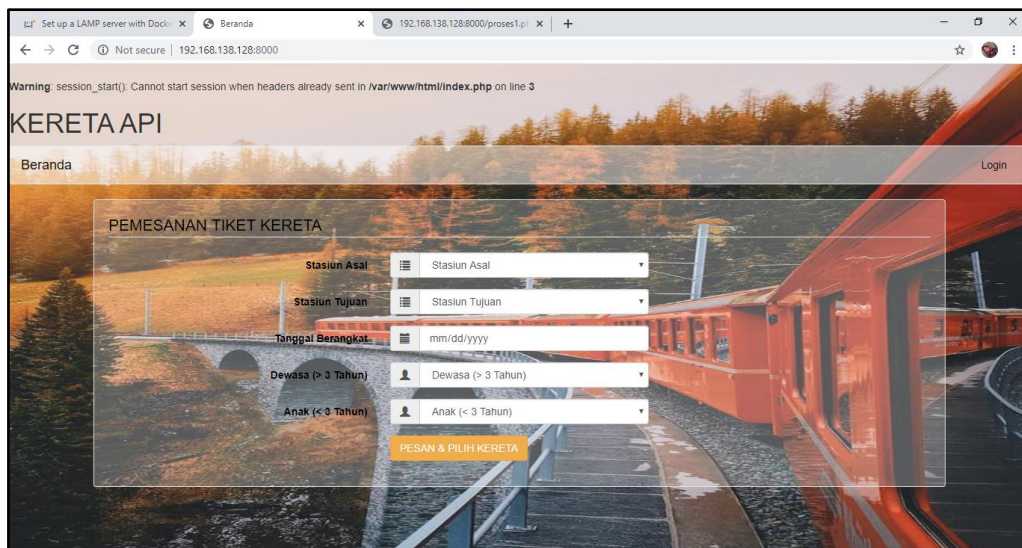
Aplikasi Pembelian Tiket Kereta Api Online sudah dapat berjalan di lokal, terlihat pada **Gambar 2.17**



Gambar 2.16 Tampilan Beranda Aplikasi pada Lokal

2.5.2 Hasil Implementasi Proyek Kedua

Aplikasi Pembelian Tiket Kereta Api Online sudah dapat berjalan saat diakses dengan ip:port (192.168.138.128:8000) terlihat pada **Gambar 2.18**



Gambar 2.18 Tampilan Beranda Aplikasi saat Diakses dengan ip:port

2.6 Pengujian Singkat

2.6.1 Pengujian Singkat Proyek Pertama

Terhadap aplikasi yang telah dirancang, akan dilakukan pengujian terhadap permasalahan yang tertuang pada bab pertama, yaitu bagaimana meningkatkan efektivitas waktu dan kemudahan pada pembelian tiket kereta api online

Pertama-tama pengguna melakukan registrasi dengan memasukkan data diri pada dashboard sistem seperti yang terlihat pada **Gambar 2.19** berikut.

Gambar 2.19 Dashboard Registrasi

Tahap selanjutnya adalah pengguna akan menggunakan nomer telepon dan password yang sudah didaftarkan untuk login seperti yang terlihat pada **Gambar 2.20**

Gambar 2.20 Dashboard Login

Setelah berhasil login, pengguna dapat memilih stasiun asal, stasiun tujuan, tanggal keberangkatan, dan banyak penumpang seperti terlihat pada **Gambar 2.21**

KERETA API

Beranda Aliza Hanum Anggani

PEMESANAN TIKET KERETA

Stasiun Asal:

Stasiun Tujuan:

Tanggal Berangkat:

Dewasa (> 3 Tahun):

Anak (< 3 Tahun):

PESAN & PILIH KERETA

Gambar 2.21 Dashboard Beranda

Lalu, pengguna akan diarahkan untuk memilih beberapa pilihan kereta yang tersedia sesuai dengan tanggal keberangkatan dan tujuan yang sudah dipilih sebelumnya seperti pada **Gambar 2.22**. Pada tahapan ini pengguna juga sekaligus mengetahui jumlah tagihan dari tiap-tiap pilihan kereta.

KERETA API

Beranda Aliza Hanum Anggani

Pilih Kereta

| Kereta Api | Tanggal Keberangkatan | Berangkat | Datang | Durasi | Jumlah Penumpang | Harga Satuan | Total Harga | Action |
|-------------------|-----------------------|-----------|----------|--------|------------------|--------------|-------------|-----------------------|
| FAJAR UTAMA YK | 2020-04-16 | 06:15:00 | 14:35:00 | 8j 20m | 3 | 260000 | 520000 | Pilih |
| GAJAHWONG | 2020-04-16 | 06:45:00 | 14:53:00 | 8j 8m | 3 | 220000 | 440000 | Pilih |
| JAYAKARTA PREMIUM | 2020-04-16 | 13:20:00 | 21:54:00 | 8j 34m | 3 | 220000 | 440000 | Pilih |
| SENJA UTAMA YK | 2020-04-16 | 19:00:00 | 03:13:00 | 8j 13m | 3 | 265000 | 530000 | Pilih |
| BOGOWONTO | 2020-04-16 | 21:45:00 | 05:44:00 | 7j 59m | 3 | 320000 | 640000 | Pilih |
| SENJA UTAMA SOLO | 2020-04-16 | 22:00:00 | 05:53:00 | 7j 53m | 3 | 400000 | 800000 | Pilih |

Gambar 2.22 Dashboard Pemilihan Kereta Api

Setelah memilih kereta, pengguna akan diminta untuk mengisi identitas masing-masing penumpang (**Gambar 2.23**)

Konfirmasi Kereta Api

Nama Kereta: JAYAKARTA PREMIUM

Tanggal Berangkat: 2020-04-16

Waktu Berangkat: 22:00:00

Waktu Datang: 05:53:00

Durasi Perjalanan: 7j 53m

Jumlah Penumpang: 3

Harga: \$ 800000

Silahkan Isi Data Penumpang Terlebih Dahulu

| | | | |
|------------------|--|-----------------|--|
| 1 Nama Lengkap : | | No. Identitas : | |
| 2 Nama Lengkap : | | No. Identitas : | |
| 3 Nama Lengkap : | | No. Identitas : | |

Gambar 2.23 Dashboard Pengisian Identitas Penumpang

Selanjutnya akan ditampilkan data konfirmasi dan pilihan untuk metode pembayaran yang ditawarkan kepada pengguna pada **Gambar 2.24**

Pembayaran

Kode Pembayaran: KRT-001

INFO PERJALANAN

No Kereta: 25

Nama Kereta: LODAYA

Berangkat Kereta: 20:08:00

Datang Kereta: 04:15:00

DATA PENUMPANG

Nama Penumpang: Arya

ID Penumpang: 123

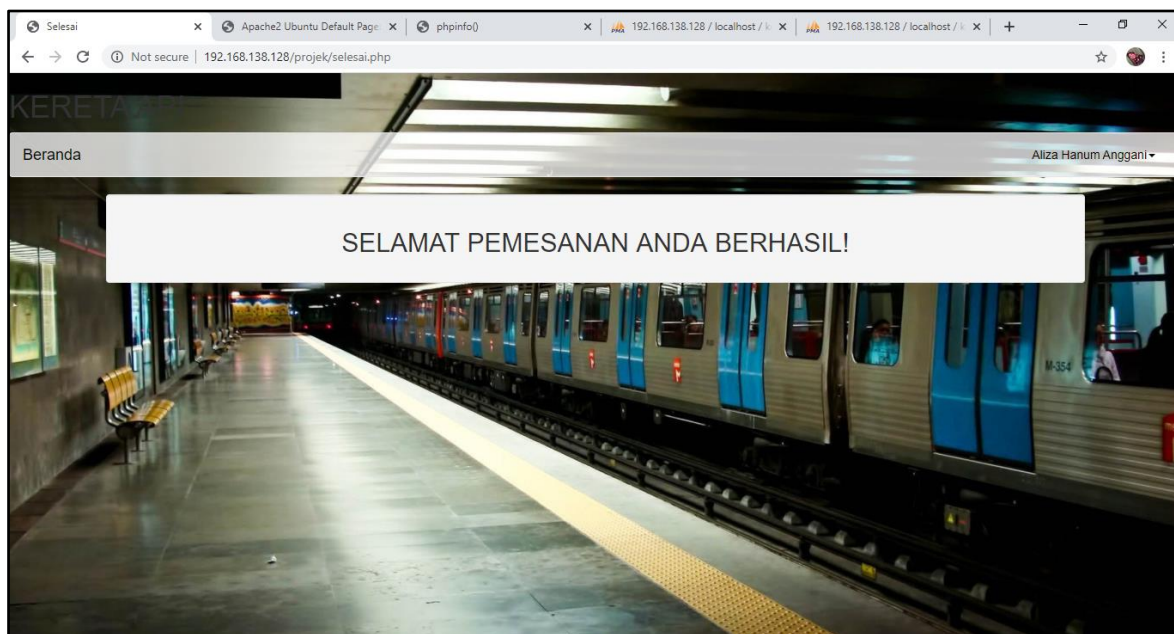
PILIH PEMBAYARAN

☐ Indomaret
☐ Alfamart
☒ ATM

Bayar

Gambar 2.24 Dashboard Konfirmasi Pembayaran

Tahap terakhir, akan muncul pemberitahuan bahwa pemesanan tiket telah berhasil (**Gambar 2.25**)



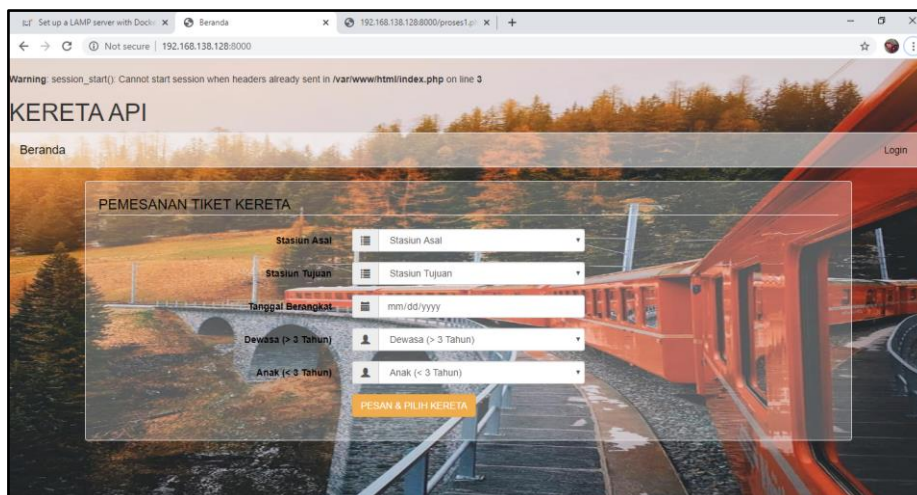
Gambar 2.25 Dashboard Notifikasi Pembelian Tiket Sukses

Hasil dari apa yang telah dilakukan pengguna tersebut adalah membeli tiket kereta api sama seperti cara konvensional namun lebih cepat dan hemat tenaga. Hal ini dapat terlihat bahwa waktu yang dibutuhkan hanya 5 menit dan dapat dilakukan dari mana saja selama ada koneksi internet.

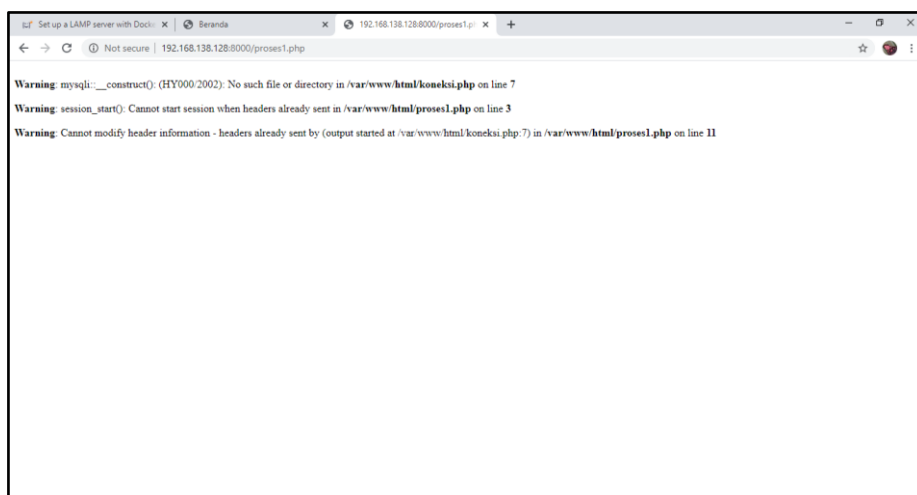
Kesimpulan dari percobaan tersebut bahwa sistem yang dibuat mampu memberikan layanan yang sangat bermanfaat bagi pengguna kereta api, selain dapat diakses dari mana saja, biaya perbaikan berkala jauh lebih murah.

2.6.2 Pengujian Singkat Proyek Kedua

Untuk hasil pengujian singkat pada proyek kedua yaitu Pembuatan dockerfilenya, dapat terlihat pada **Gambar 2.26** dan **Gambar 2.27**. Saat diakses menggunakan ip:port (dalam hal ini 192.168.138.128:8000) halaman beranda sudah muncul tetapi karena pada proses *import database* masih terdapat kendala, maka saat mengakses halaman yang membutuhkan *database* masih menunjukkan error seperti pada **Gambar 2.27**.



Gambar 2.26 Tampilan Beranda



Gambar 2.27 Error pada Bagian Database

BAB III

JADWAL Pengerjaan dan Pembagian Tugas

3.1 Agenda Pengerjaan

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

Tabel 3.1 Agenda Pengerjaan Proyek

| No. | Jenis Tugas | Waktu Pengerjaan | | | | | | | | | |
|-----|---|------------------|---|---|---|-------|---|---|---|-----|---|
| | | Maret | | | | April | | | | Mei | |
| | | 2 | 3 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 1. | Analisa Persoalan | | | | | | | | | | |
| 2. | Pembagian Tugas | | | | | | | | | | |
| 3. | Perancangan Program | | | | | | | | | | |
| 4. | Perancangan Struktur Database | | | | | | | | | | |
| 5. | Instalasi Ubuntu, LAMPP, Vmware Workstation | | | | | | | | | | |
| 6. | Konfigurasi Hosting Lokal | | | | | | | | | | |
| 7. | Submisi Progress Pertama | | | | | | | | | | |
| 8. | Instalasi Docker | | | | | | | | | | |
| 9. | Pembuatan Dockerfile | | | | | | | | | | |
| 10. | Submisi Progress Kedua | | | | | | | | | | |
| 11. | Penyelesaian Database Dockerfile | | | | | | | | | | |
| 12. | Submisi Final | | | | | | | | | | |

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. | Analisa Persoalan | Hanum |
| 2. | Pembagian Tugas | Hanum |
| 3. | Perancangan Program | Can |
| 4. | Perancangan Struktur Database | Can |
| 5. | Instalasi Ubuntu, LAMPP, Vmware Workstation | Hanum |
| 6. | Konfigurasi Hosting Lokal | Can & Hanum |
| 7. | Submisi Progress Pertama | Hanum |
| 8. | Instalasi Docker | Can & Hanum |
| 9. | Pembuatan Dockerfile | Hanum |
| 10. | Submisi Progress Kedua | Hanum |
| 11. | Penyelesaian Database Dockerfile | Hanum |
| 12. | Submisi Final | Can & Hanum |

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan permasalahan yang terdapat di awal , yaitu membuat aplikasi pembelian tiket kereta online berbasis web menggunakan ubuntu lampp dan proses pembuatan dockerfilenya diperoleh hasil yang cukup baik. Aplikasi pembelian tiket kereta online dapat berjalan dengan cukup baik saat dijalankan melalui *hosting local* maupun melalui Docker *container*. Dari pengujian sistem, hampir semua skenario yang dibuat dapat berjalan dengan baik sesuai dengan yang diharapkan, kecuali pada bagian database docker. Database belum bisa di *import* dengan baik.

4.2 Saran

Saran untuk pengerjaan proyek ini adalah seharusnya menggunakan hardware atau laptop yang lebih memadai dari segi RAM, spesifikasi dan yang lainnya sehingga tidak sering terjadi *not responding* saat proses pengerjaan.

DAFTAR PUSTAKA

- Cancer, Y., & Alim, Z. (2016). PLATFORM AS A SERVICE (PaaS) SEBAGAI LAYANAN SISTEM OPERASI CLOUD COMPUTING. *Jurnal TIMES*, Vol. V No(6), 381–384. <https://doi.org/10.1007/s12599-011-0183-3>
- Saputra, Didi Indra. (2010, Maret 12). Definisi Komputer Awan (Cloud Computing). Diakses dari <https://didiindra.wordpress.com/tag/definisi-komputasi-awan-cloud-computing/>
- Mitchell, Erik. 2010. *Using Cloud Services for Library IT Infrastructure* dalam code4lib Journal. Tersedia dalam <http://journal.code4lib.org/articles/2510>. [Diakses pada 29 Maret 2020]

LAMPIRAN