

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI PERJALANAN DINAS MENGGUNAKAN UBUNTU
LAMPP DAN PROSES PEMBUATAN DOCKER FILE-NYA**



DISUSUN OLEH:

**NAMA ANGGOTA : ARIF MAULANA 123170010
ARCHAN HABIB S 123170024
KELAS : E
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.
WAHYU AJI NUGROHO, S.Kom.**

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

HALAMAN PENGESAHAN

SISTEM INFORMASI PERJALANAN DINAS MENGGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKER FILE-NYA

Disusun oleh :

Arif Maulana

123170010

Archan Habib Sya'bana

123170024

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Wahyu Aji Nugroho, S.Kom.

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Allah swt atas segala limpahan rahmat-Nya sehingga saya dapat menyelesaikan penyusunan laporan akhir praktikum teknologi *cloud computing*. Adapun isi dari laporan akhir ini adalah proyek akhir kami yang berjudul Sistem Informasi Perjalanan Dinas Menggunakan Ubuntu LAMPP dan Proses Pembuatan Docker file-nya yang membahas tentang informasi perjalanan dinas yang dimiliki oleh Disporapar Boyolali agar terciptanya suasana transparansi informasi perjalanan dinas dan menjaga integritas dinas itu sendiri.

Tak lupa pula saya mengucapkan banyak terima kasih kepada asisten dosen yang selalu membimbing dan mengajari saya dalam melaksanakan praktikum dan dalam penyusunan laporan akhir ini. Serta semua pihak yang telah membantu saya dalam penyusunan laporan akhir ini. Karena laporan akhir ini masih jauh dari sempurna oleh karena itu kritik dan saran yang membangun masih saya harapkan untuk penyempurnaan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan laporan akhir ini saya ucapkan terima kasih. Semoga laporan ini dapat dipergunakan sesuai dengan keperluan.

Yogyakarta, 31 Maret 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Tujuan Proyek Akhir	2
1.3. Manfaat Proyek Akhir	2
1.4. Tahap Penyelesaian Proyek Akhir	3
BAB II ISI DAN PEMBAHASAN	6
2.1. Komponen yang Digunakan	6
2.2. Rancangan Arsitektur <i>Cloud Computing</i>	5
2.3. Parameter dan Konfigurasi	8
2.4. Tahap Implementasi	10
2.5. Hasil Implementasi	18
2.6. Pengujian Singkat	19
BAB III JADWAL Pengerjaan dan Pembagian Tugas	19
3.1. Agenda Pengerjaan	23
3.2. Keterangan Pembagian Tugas	23
BAB IV KESIMPULAN DAN SARAN	24
4.1. Kesimpulan	24
4.2. Saran	24
DAFTAR PUSTAKA	25
LAMPIRAN	26

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Cloud computing adalah teknologi berbasis Internet di mana informasi disimpan di dalam server dan disediakan sebagai layanan sesuai permintaan yang diinginkan klien (Etro, 2014). Begitupula dengan *software* atau aplikasi lainnya sehingga memungkinkan satu komputer *server* untuk membagikannya dengan komputer lainnya yang terhubung. Hal ini akan menghemat biaya operasional juga menghemat waktu karena tidak memerlukan penyimpanan eksternal berkapasitas besar untuk menyimpan setiap *software*. Seperti misalnya data *Microsoft Word* yang cukup diunduh satu kali pada *server* kemudian bisa digunakan di komputer lainnya tanpa harus melakukan penginstalan kembali.

Saat ini penggunaan *cloud computing* sudah sangat luas, biasanya *cloud computing* digunakan dalam komputasi pada *website* supaya *website* tersebut dapat diakses oleh seluruh pengguna tidak hanya terbatas dalam satu jaringan yang sama dengan komputer *server* pemroses *website* tersebut (Postmedya, 2019). Selain digunakan dalam komputasi *website*, *cloud computing* juga biasa digunakan pada alat-alat elektronik yang sudah menerapkan teknologi IoT atau *Internet of Thing*. Dalam peralatan elektronik yang sudah menggunakan IoT, tentunya akan banyak menyimpan data yang dibutuhkan dalam alat tersebut (Indonesian Cloud, 2016). Oleh karena itu, *cloud computing* dalam hal tersebut digunakan sebagai tempat penyimpanan data-data yang digunakan oleh peralatan elektronik tersebut, supaya bisa diakses oleh semua pengguna alat tersebut. Selain kedua contoh penggunaan *cloud computing* tersebut sebenarnya masih sangat banyak contoh implementasi layanan *cloud computing* dalam kehidupan sehari-hari (Wow Technology, n.d.).

Sistem Informasi Perjalanan Dinas merupakan aplikasi yang dirancang untuk merekap dan membuat surat untuk perjalanan dinas di Disporapar Boyolali, dimana ketika pegawai ingin melakukan perjalanan dinas bagian sekretariat harus membuat surat perjalanan dinas untuk pegawai tersebut, didalam setiap perjalanan dinas terdapat 3 macam surat yang harus dibuat yaitu surat SP untuk kelompok perjalanan, surat SPPD untuk perorangan, dan rincian biaya. Aplikasi ini dibuat untuk mempermudah dalam pembuatan surat karena apabila suatu perjalanan dinas dilakukan oleh banyak pegawai atau terdapat banyak perjalanan dinas yang dilakukan didalam dinas tersebut maka pegawai pembuatan surat

akan kewalahan, karena bisa mencapai puluhan surat yang harus dibuat, dan hal itu sering terjadi di Disporapar Boyolali selain pembuatan surat sistem ini memiliki fungsi sebagai rekap biaya dimana ketika suatu perjalanan dilakukan maka biaya pada setiap pegawai akan di rekap tiap bulannya. Didalam sistem ini terdapat beberapa data yang dipakai sebagai acuan yaitu data pegawai, biaya, tipe perjalanan, dan kategori jabatan. Dalam proyek kali ini, kami akan menggunakan layanan *cloud computing* untuk tempat komputasi pemrosesan dari sebuah *website*, dan tempat penyimpanan data yang digunakan dalam *website* tersebut.

Dalam pengembangan *website* ini, tahap-tahap yang dilakukan untuk menyelesaikannya adalah pertama membuat *website sistem informasi perjalanan dinas* ini yang dipastikan sudah dapat dijalankan dengan normal tanpa kendap. Kemudian membuat *server* ubuntu yang akan digunakan sebagai tempat untuk *hosting* dari *website* tersebut. setelah dipastikan *website* dapat berjalan dengan normal dan layanan *hosting* yang kita buat menggunakan ubuntu LAMPP dapat bekerja dengan normal. maka kita dapat membuat *website* yang kita buat tersebut dihosting menggunakan layanan *hosting* yang telah kita buat. Sehingga nantinya *website* tersebut dapat diakses oleh pegawai di dinas tersebut dimana saja dan kapan saja, tidak hanya menjadi *website* internal yang hanya bisa diakses oleh pembuat *website* tersebut.

1.2 Tujuan Proyek Akhir

Berhubungan dengan judul proyek akhir yaitu sistem informasi perjalanan dinas Menggunakan Layanan Ubuntu LAMPP dan proses pembuatan Dockerfilenya, maka laporan ini akan menjelaskan cara penyelesaian sistem yang kita buat ini dengan menggunakan ubuntu dan LAMPP sebagai tempat *hosting* dari *website* yang kita buat. Dalam implementasinya penyelesaian yang terdapat di dalam laporan ini akan menghasilkan berupa *website* yang berisi layanan pembuatan surat dan informasi tentang perjalanan dinas yang telah dilakukan sehingga tiap pegawai atau bidang sekretariat akan dipermudah dalam pendataan atau pembuatan surat dan kepala dinas pun bisa memantau bagaimana kegiatan, biaya, dan pegawai yang melakukan perjalanan dinas.

1.3 Manfaat Proyek Akhir

Dengan dibuatnya proyek akhir ini, maka akan mempermudah dalam pembuatan surat perjalanan dinas dan proses rekap datanya. Sehingga ketika dinas ini memiliki banyak perjalanan dinas dan banyak pegawai yang melakukannya tidak perlu memerlukan

tenaga yang lebih ketika pembuatan suratnya. Sistem ini menggunakan *Docker* yang bisa menjadi contoh untuk pembuatan sistem lainnya dengan menggunakan *Docker* selain itu dengan menggunakan *Docker* maka akan mempermudah dalam hal pengoperasian sistem. Sistem ini dibuat ke dalam *cloud* dengan maksud agar biaya yang dikeluarkan untuk membangun sistem ini tidak begitu banyak karena *cloud* dapat mengurangi pengeluaran dalam hal pembelian infrastruktur, penyimpanan eksternal, serta biaya perawatannya (Binus Center Blog, 2013). Selain itu manfaat dengan adanya *cloud computing* dalam menjalankan website Sistem Informasi Perjalanan Dinas ini adalah website ini dapat diakses oleh semua pihak, tidak terbatas pada Batasan akses yang diberikan apabila website tersebut tidak dihosting dan hanya bisa dilakukan pengaksesan oleh *developer* yang membuat website tersebut. Karena memang itulah kelebihan dari layanan hosting yang menggunakan teknologi *cloud computing* dapat menjalankan komputasi dari sebuah website tersebut dimanapun dan kapanpun tidak terbatas oleh jaringan tertentu yang harus sama dengan jaringan yang digunakan *developer* dalam membuat dan mengembangkan website tersebut

1.4 Tahap Penyelesaian Proyek Akhir

Tahapan penyelesaian proyek akhir ini adalah sebagai berikut :

1. Menganalisis kebutuhan dari sistem untuk dibuat ke dalam arsitektur cloud computing.
2. Mengintegrasikan data pada *Docker* dengan sistem informasi perjalanan dinas.
3. Membuat atau mengkustomisasi konfigurasi yang sesuai pada pengaturan *Docker* sehingga dapat digunakan sesuai dengan kebutuhan yang berupa :
 - a. Dapat diakses oleh Jaringan/IP tertentu saja
 - b. Menggunakan PHP dengan versi 7.2.24
 - c. Menggunakan bahasa HTML dengan versi minimal 4.0.
 - d. Virtualisasi menggunakan *Docker*
4. Membuat topologi cloud computing untuk menggabungkan dua sub sistem yang berbeda sehingga dapat digunakan secara integrasi.
5. Membuat konfigurasi server untuk dapat digunakan sebagai *recovery* sehingga sistem informasi perjalanan dinas berjalan dengan maksimal dalam hal *availability*-nya.
6. Menguji keandalan arsitektur cloud computing yang dibangun dengan beberapa pengujian

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Dalam membangun “Sistem Informasi Perjalanan Dinas Menggunakan Ubuntu LAMPP Dan Proses Pembuatan Docker file-nya” yang berbasis konsep *cloud computing*, maka diperlukan analisis beberapa komponen yang digunakan. Berikut penjelasannya dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.2.24 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 untuk keperluan *website* Sistem Informasi *Kost*.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunanya ialah seluruh user/pegawai dan admin aplikasi Sistem Informasi *Perjalanan Dinas*. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *public*.

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 dan kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode NAT	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.12.133/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.12.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.12.255	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.4 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	<i>Harddisk</i>	20 GB	Alokasi <i>harddisk</i>
6.	<i>Memory</i>	1 GB	Alokasi <i>memory</i> untuk <i>guest OS</i>

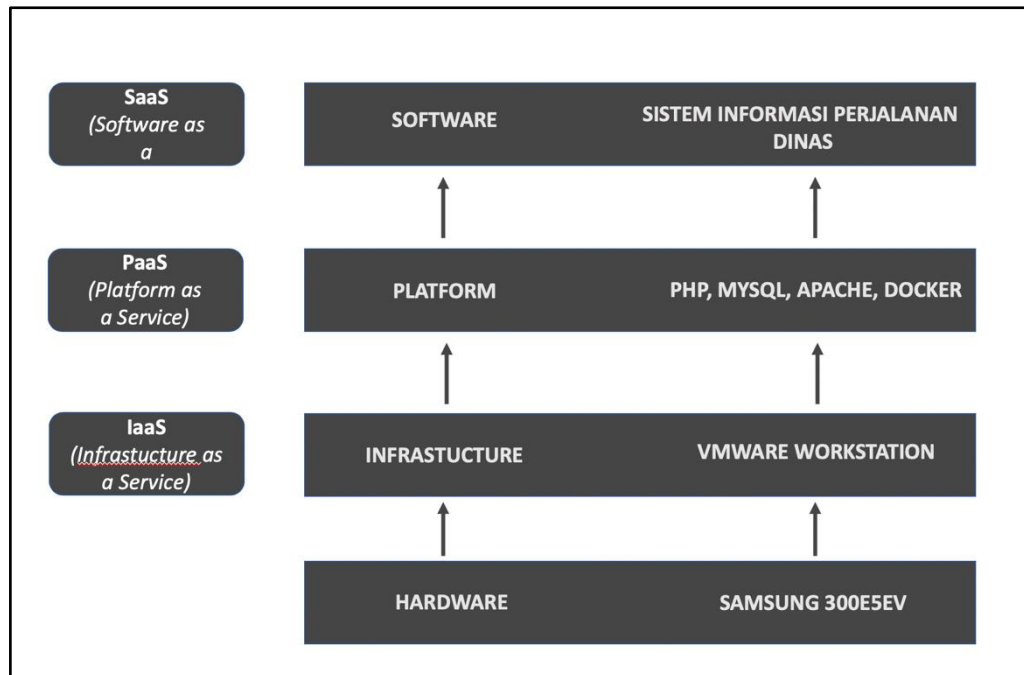
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

Tabel 2.2 Spesifikasi Ubuntu OS untuk proyek pertama dan kedua

No.	Nama Parameter	Nilai	Keterangan
1.	LAMP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.2.24	<i>Bahasa Pemrograman yang digunakan dalam membangun sistem informasi kost</i>
		MySQL	<i>Basis Data yang digunakan dalam menyimpan data sistem informasi perjalanan dinas</i>
		PHPMyAdmin	Perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL
2.	Docker	Docker 19.03.8	Platform terbuka agar dapat membangun dan menjalankan aplikasi di berbagai lokasi sebagai sebuah <i>container</i> yang ringan

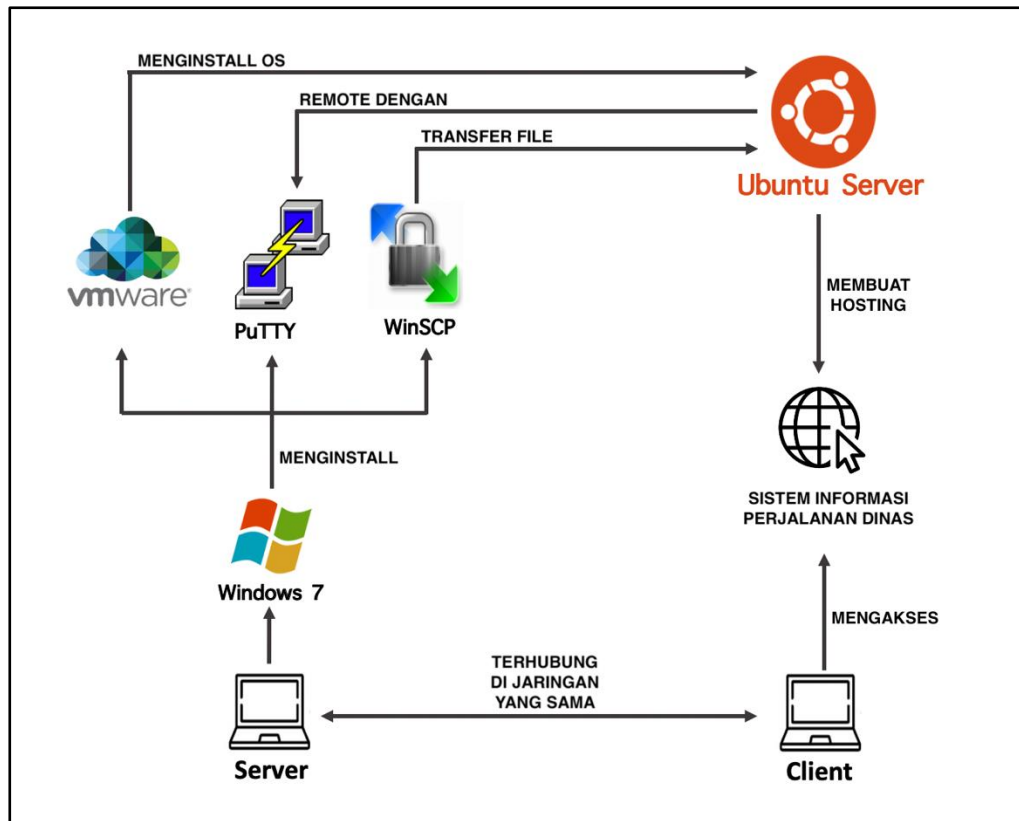
2.2 Rancangan Arsitektur Cloud Computing

Pada proyek ini, kami menggunakan Laptop SAMSUNG 300E5EV sebagai hardware yang kemudian di layer atasnya digunakan hypervisor berupa VMware Workstation. Untuk virtual machine sendiri kami menggunakan Ubuntu Server yang akan digunakan untuk virtualisasi Ubuntu 18.04 LTS. Selanjutnya pada Ubuntu 18.04 LTS dilakukan instalasi app berupa Apache, PHP, MySQL, PHPMyAdmin, Docker dll. Arsitekturnya dapat dilihat pada **gambar 2.1** dibawah ini:

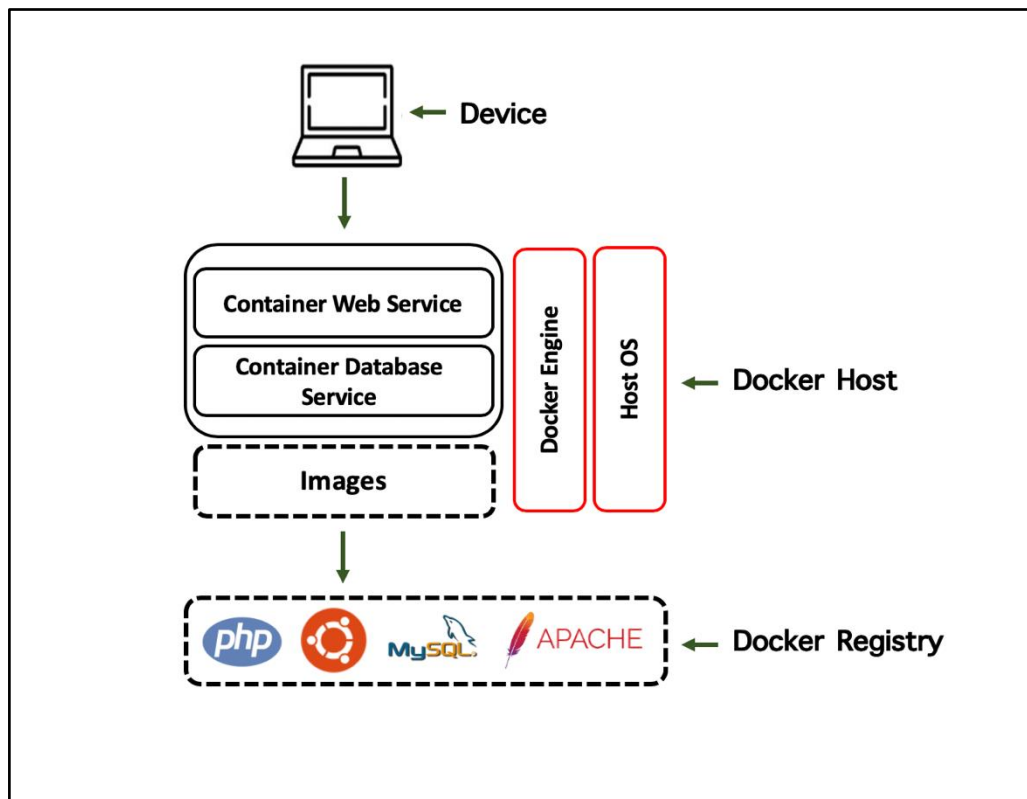


Gambar 2.1 Topologi Layer By Layer

Pada proyek pertama koneksi LAN (Indihome) digunakan untuk menghubungkan Device 1 yang bertugas sebagai host dan Device 2 yang bertugas sebagai User. Tujuan menghubungkan kedua device dengan LAN adalah agar website juga dapat diakses oleh Device 2 tidak hanya Device 1 saja. Device 1 yang bertugas sebagai host yang memakai OS Windows kemudian dilakukan instalasi VMware Workstation, PuTTY dan WinSCP. Kemudian pada VMWare Workstation dilakukan instalasi Ubuntu Server yang akan dilakukan remote oleh software PuTTY. Kemudian dalam Ubuntu Server akan dilakukan instalasi Apache, MySQL, PHPMyAdmin sehingga dapat menjalankan *website*. Source Code dari *website* yang akan digunakan kemudian di copy ke dalam directory yang terdapat pada Ubuntu Server dengan menggunakan WinSCP yang sudah tersambung dengan IP Address dari Ubuntu Server. Arsitekturnya dapat dilihat pada **gambar 2.2** dibawah ini:



Gambar 2.2 Topologi Proyek Akhir



Gambar 2.3 Topologi Proyek Kedua

2.3 Parameter dan Konfigurasi

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

$ sudo ufw allow in "Apache Full"
Keterangan: Untuk mengatur Firewall agar port 80 dan port 443 diizinkan oleh sistem Ubuntu.
```

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

$ sudo mysql_secure_installation
Keterangan: Untuk mengatur keamanan pada MySQL contohnya username dan password
```

Modul 2.2 Parameter instalasi MySQL

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

$ sudo nano /var/www/html/info.php
Keterangan: Untuk membuat file baru bernama info.php yang berada di direktori var/www/html
```

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
- phpMyAdmin php-mbstring php-gettext: nama paket untuk PHPMyAdmin

$ sudo mysql -u root
Keterangan: Untuk masuk ke MySQL sebagai user root.

mysql> UPDATE mysql.user SET plugin = 'mysql_native_password', authentication_string = PASSWORD('19121998') WHERE User = 'root';
Keterangan: Untuk mengubah password dari user 'root' menjadi '19121998'

mysql> FLUSH PRIVILEGES;
Keterangan: Untuk merefresh akun yang terkoneksi dengan phpmyadmin

$ sudo chown archanhs /var/www/html
Keterangan: Untuk memberikan akses ke User archanhs agar dapat mengakses direktori var/www/html

```

Modul 2.4 Parameter instalasi PHPMyAdmin

Parameter yang digunakan untuk melihat *IP address* yang digunakan dapat dilihat pada penjelasan **Modul 2.5** berikut ini :

```

$ ip addr

Keterangan :
Untuk melihat ip address.
- ip : perintah untuk menjalankan ip yang akan menampilkan seluruh interface jaringan yang ada beserta konfigurasinya seperti ip address
- addr : perintah dari ip untuk menampilkan ip address pada setiap interface jaringan yang ada

```

Modul 2.5 Melihat *IP address*

Parameter yang digunakan untuk proses instalasi docker dapat dilihat pada **Modul 2.6** seperti dibawah ini :

```

$ sudo apt-get update
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
Keterangan: Untuk memperbarui indeks paket apt dan instal paket untuk memungkinkan apt menggunakan repositori melalui HTTPS:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg
| sudo apt-key add -
$ sudo apt-key fingerprint 0EBFCD88
Keterangan: Untuk menambahkan Docker Official GPG Key

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

```

Keterangan: Untuk mengatur repository menjadi stable

```
$ sudo apt-get update
```

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

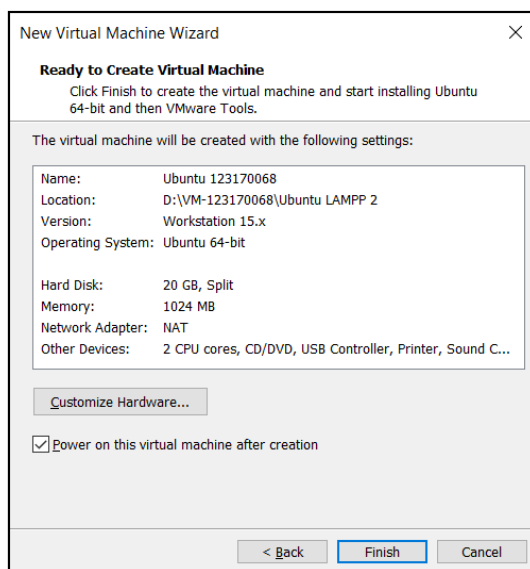
Keterangan: Untuk memperbarui indeks paket apt dan install versi terbaru Docker Engine

Modul 2.6 Parameter instalasi Docker

2.4 Tahap Implementasi

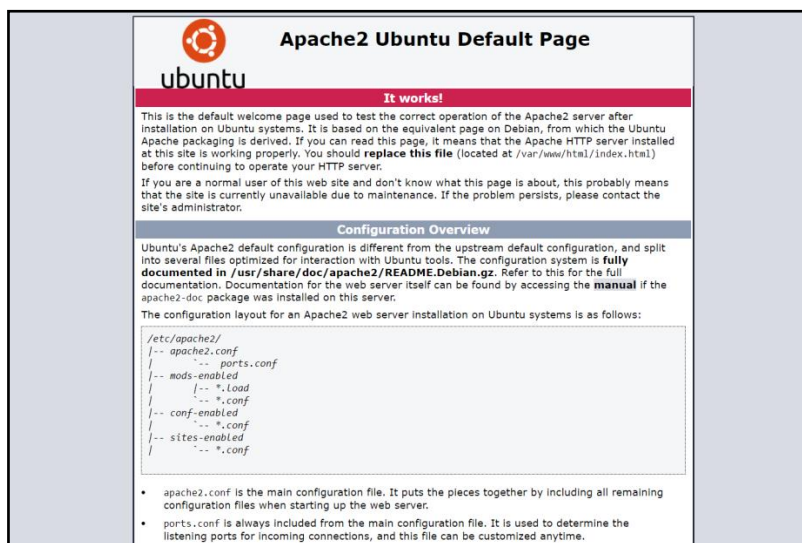
Tugas 1 :

Hasil dari implementasi **Tabel 2.1** tentang konfigurasi untuk VMware Workstation untuk Virtualisasi Ubuntu yang dibuat dapat dilihat pada **Gambar 2.3** berikut ini:



Gambar 2.4 Hasil Pembuatan Virtual Machine dengan OS Ubuntu

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



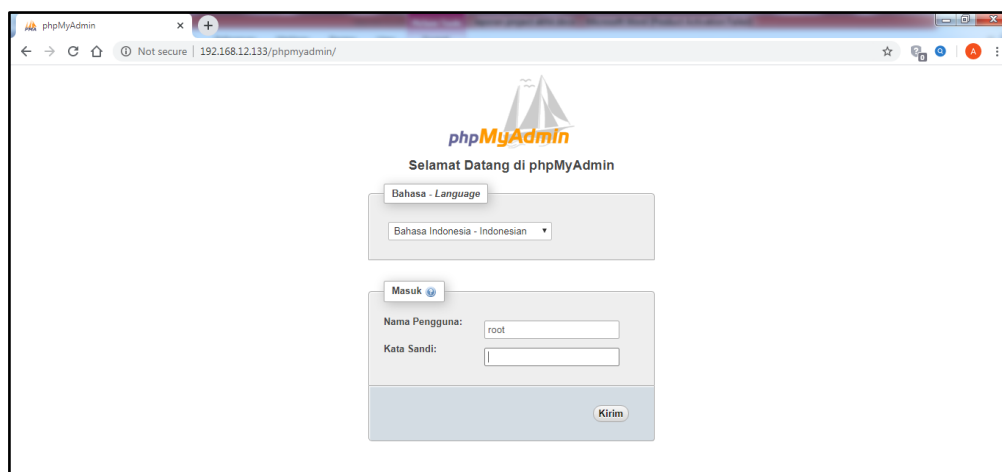
Gambar 2.5 Hasil Instalasi Apache

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

PHP Version 7.2.24-0ubuntu0.18.04.3	
	
System	Linux fahmi-server 4.15.0-76-generic #96-Ubuntu SMP Fri Jan 17 17:24:28 UTC 2020 x86_64
Build Date	Feb 11 2020 15:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-apache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-bz2.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-xml.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip

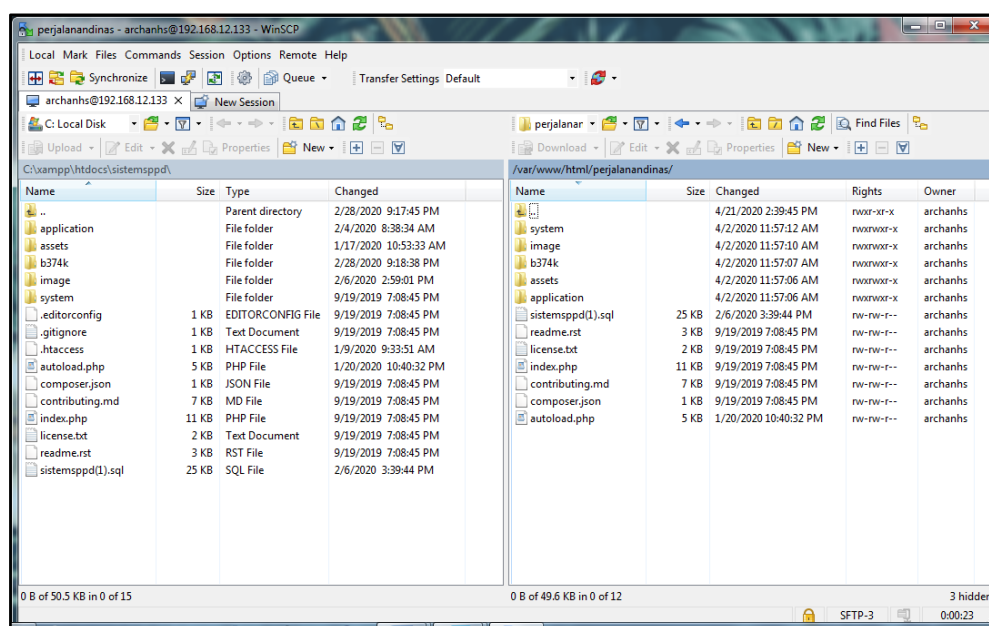
Gambar 2.6 Hasil Instalasi PHP

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



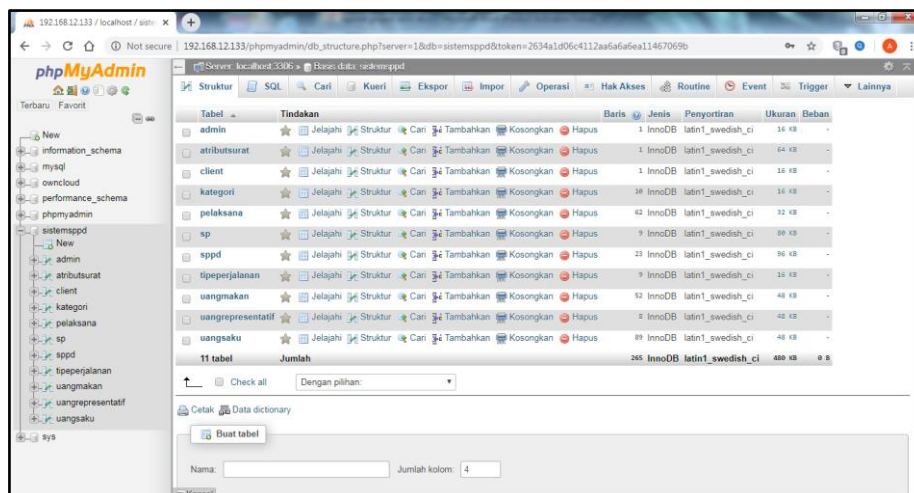
Gambar 2.7 Hasil Instalasi PHPMyAdmin

Setelah dilakukan instalasi LAMPP maka Langkah selanjutnya adalah memindahkan *source code website* yang digunakan dari laptop ke server ubuntu dengan menggunakan WinSCP. WinSCP sendiri membutuhkan *IP Address*, *Username*, dan *Password* dari Ubuntu untuk dapat mengakses direktorinya. Setelah itu baru dilakukan pemindahan *source code* menuju ke direktori `var/www/html` seperti **gambar 2.8** dibawah ini:



Gambar 2.8 Proses Pemindahan Source Code

Setelah dilakukan *source code* berhasil dimasukkan ke dalam direktori `var/www/html` maka langkah selanjutnya adalah melakukan *import database website* yang digunakan seperti pada **gambar 2.9** dibawah ini:



Gambar 2.9 Hasil Import Database

Tugas 2 :

Instalasi Docker

1. Hasil Instalasi Docker. Seperti **listing program 1.0** dan **gambar 3.0** dibawah ini :

```
sudo docker -v
```

Listing program 1.0 Listing update server dan install Docker

```
archanhs@archanhs:~$ sudo docker -v
[sudo] password for archanhs:
Docker version 19.03.6, build 369ce74a3c
archanhs@archanhs:~$ _
```

Gambar 3.0 cek versi docker

2. Setelah docker terinstall maka tahap selanjutnya melihat apakah service docker sudah berjalan dengan listing seperti dibawah ini maka akan terlihat keterangan apakah service docker sudah active atau tidak. Seperti **Listing program 1.1** dan **Gambar 3.1** dibawah ini :

```
systemctl status docker
```

Listing program 1.1 Listing melihat status Docker

```

archanhs@archanhs:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-05-13 09:01:09 UTC; 1min 16s ago
     Docs: https://docs.docker.com
    Main PID: 1196 (dockerd)
      Tasks: 12
   CGroup: /system.slice/docker.service
           └─1196 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

May 13 09:01:07 archanhs dockerd[1196]: time="2020-05-13T09:01:07.007277192Z" level=warning msg="You
May 13 09:01:07 archanhs dockerd[1196]: time="2020-05-13T09:01:07.009454685Z" level=warning msg="You
May 13 09:01:07 archanhs dockerd[1196]: time="2020-05-13T09:01:07.012024207Z" level=warning msg="You
May 13 09:01:07 archanhs dockerd[1196]: time="2020-05-13T09:01:07.015329771Z" level=info msg="Loadin
May 13 09:01:08 archanhs dockerd[1196]: time="2020-05-13T09:01:08.845735882Z" level=info msg="Default
May 13 09:01:09 archanhs dockerd[1196]: time="2020-05-13T09:01:09.166243167Z" level=info msg="Loadin
May 13 09:01:09 archanhs dockerd[1196]: time="2020-05-13T09:01:09.632286972Z" level=info msg="Docker
May 13 09:01:09 archanhs dockerd[1196]: time="2020-05-13T09:01:09.635540391Z" level=info msg="Daemon
May 13 09:01:09 archanhs dockerd[1196]: time="2020-05-13T09:01:09.812162930Z" level=info msg="API li
May 13 09:01:09 archanhs systemd[1]: Started Docker Application Container Engine.
lines 1-19/19 (END)

```

Gambar 3.1 Status Docker

- Setelah *service* docker sudah aktif maka tahap selanjutnya jika ingin menjalankan docker tanpa sudo (Opsional) karena default docker command harus pakai sudo, jadi agar tidak perlu menggunakan sudo buat group docker dan assign user ke group docker terlebih dahulu. Seperti **Listing program 1.2** dan **Gambar 3.2** dibawah ini :

```

sudo groupadd docker
sudo usermod -aG docker $USER

```

Listing program 1.2 Group Docker dan assign user

```

archanhs@archanhs:~$ sudo groupadd docker
[sudo] password for archanhs:
archanhs@archanhs:~$ sudo usermod -aG docker $USER

```

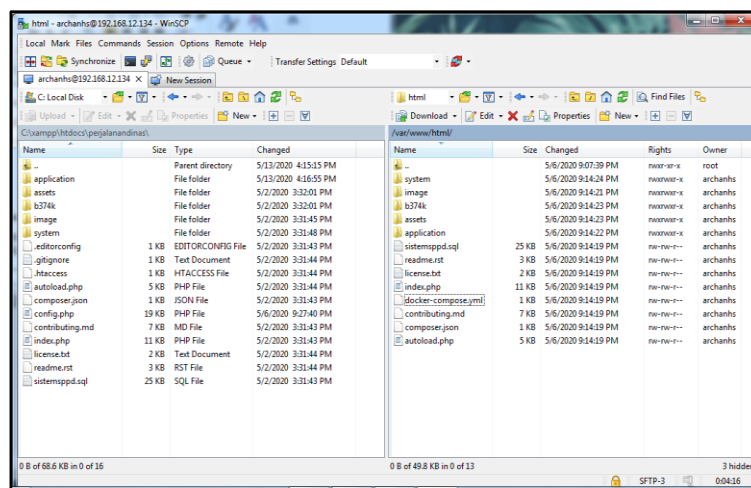
Gambar 3.2 Group Docker dan assign user

Membuat dan menjalankan Image

Untuk membuat file web yang sudah kita hosting agar bisa dijalankan melalui docker, ada beberapa tahapan sebagai berikut :

- Buka winscp, kemudian copy kan file web ke dalam /var/www/html. Seperti

Gambar 3.3 dibawah ini :



Gambar 3.3 Direktori html

2. Masuk ke direktori pada /var/www/html. Seperti **Listing program 1.3** dan **Gambar 3.4** dibawah ini :

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

Listing program 1.3 Pindah direktori

```
archanhs@archanhs:~$ cd /var/www/html
archanhs@archanhs:/var/www/html$ _
```

Gambar 3.4 Folder html

3. Setelah masuk ke direktori html buat file Dockerfile di dalam direktori tersebut, dimana didalam direktori tersebut juga ada file websie yang akan digunakan, isikan Seperti **Listing program 1.4** dibawah ini pada Dockerfilenya :

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

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

Listing program 1.4 Isi Dockerfile

4. Buat docker image dari Dockerfile yang sudah kita buat tadi. Saya buat Dengan nama web_dev2 dan tag versi 1.0, pastikan ketika melakukan build image posisi direktori sama dengan Dockerfilenya berada Seperti **Listing program 1.5** dan **Gambar 3.5** dibawah ini :

```
$ docker build . -t web_dev2:1.0
```

Listing program 1.5 *Build image Dockerfile*

```
archanhs@archanhs:~$ cd /var/www/html
archanhs@archanhs:/var/www/html$ docker build . -t web_dev2:1.0
Sending build context to Docker daemon 19.02MB
Step 1/8 : FROM php:7.3-apache
--> f0d98b8dd966
Step 2/8 : RUN apt-get update && apt-get upgrade -y
--> Using cache
--> 592bbfbbb519
Step 3/8 : RUN docker-php-ext-install mysqli
--> Using cache
--> 930ccb34ef61
Step 4/8 : EXPOSE 80
--> Using cache
--> b0ead16caa51
Step 5/8 : RUN a2enmod rewrite
--> Using cache
--> e9e77c794aa1
Step 6/8 : RUN chmod -R 755 /var/www/html/
--> Running in d08a2662f854
Removing intermediate container d08a2662f854
--> 910064695c14
Step 7/8 : COPY ./ /var/www/html
--> 0817bcd68adf
Step 8/8 : RUN service apache2 restart
--> Running in 357a028b71ee
Restarting Apache httpd web server: apache2AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
.
Removing intermediate container 357a028b71ee
--> 068f4533f310
Successfully built 068f4533f310
Successfully tagged web_dev2:1.0
```

Gambar 3.5 *Build image Dockerfile*

5. Setelah image webnya sudah terbuat , selanjutnya yaitu membuat image mysql untuk databasenya. Seperti **Listing program 1.6** dan **Gambar 3.6** dibawah ini :

```
$ docker pull mysql:5.6
```

Listing program 1.6 *Pull image mysql*

```
archanhs@archanhs:~$ docker pull mysql:5.6
```

Gambar 3.6 *build pull image mysql*

6. setelah image web dan mysql sudah ada maka selanjutnya adalah membuat container untuk menjalankan image mysql serta membuat dan mengimport database , pastikan direktori sesuai dengan posisi database saat melakukan import database. Seperti **Listing program 1.7** dan **Gambar 3.7** dibawah ini.

```
$ docker run --name=mysql_service1 -e MYSQL_ROOT_PASSWORD=mysql
-d mysql:5.6

$ docker exec -i mysql_service1 mysql -u root -pmysql -e "create
database sistemspdd"

$ docker exec -i mysql_service mysql -u root -pmysql sistemspdd <
./sistemspdd.sql
```

Listing program 1.7 Membuat dan mengimport database

```
archanhs@archanhs:~$ docker run --name=mysql_service1 -e MYSQL_ROOT_PASSWORD=mysql -d mysql:5.6
38342d34a2b9303ef1d316617c7bca83b71c433734045b994612171482c480d0
archanhs@archanhs:~$ docker exec -i mysql_service1 mysql -uroot -pmysql -e "create database sistemsp
pd"
Warning: Using a password on the command line interface can be insecure.
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
archanhs@archanhs:~$ docker exec -i mysql_service1 mysql -u root -p mysql -e "create database sistem
spdd"
Enter password: 19121998
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
archanhs@archanhs:~$ docker exec -i mysql_service1 mysql -u root -p mysql -e "create database sistem
spdd"
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
archanhs@archanhs:~$ docker exec -i mysql_service1 mysql -u root -p -e "create database sistemspdd"
Enter password: 19121998
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
archanhs@archanhs:~$ docker exec -i mysql_service1 mysql -u root -pmysql -e "create database sistem
spdd"
Warning: Using a password on the command line interface can be insecure.
archanhs@archanhs:~$

archanhs@archanhs:/var/www/html$ docker exec -i mysql_service1 mysql -u root -pmysql sistemspdd < ./
sistemspdd.sql
Warning: Using a password on the command line interface can be insecure.
```

Gambar 3.7 Membuat dan mengimport database

7. setelah container mysql terbuat dan database berhasil diimport , maka selanjutnya membuat dan menjalankan container untuk webnya yang nantinya akan dihubungkan dengan container mysql yang telah dibuat tadi dan membuat portnya menjadi 8000 jadi jika kita ingin mengakses web ini maka alamat_ip:8000, lalu jalankan container tersebut, Seperti **Listing program 1.8** dan **Gambar 3.8** dibawah ini :

```
$ docker run --name=web_dev --link=mysql_service1 -p 8000:80 -d
web_dev2:1.0

$ docker container start web_dev
```

Listing program 1.8 Membuat dan menjalankan container web

```
archanhs@archanhs:/var/www/html$ docker run --name=web_dev --link=mysql_service1 -p 8000:80 -d web_d
ev2:1.0
45492e0251a7c0a68f991acf166b72e5d6df01a098323d70627ae7b4101c4da5
archanhs@archanhs:/var/www/html$ docker container start web_dev
web_dev
archanhs@archanhs:/var/www/html$
```

Gambar 3.8 Membuat dan menjalankan container web

2.5 Hasil Implementasi

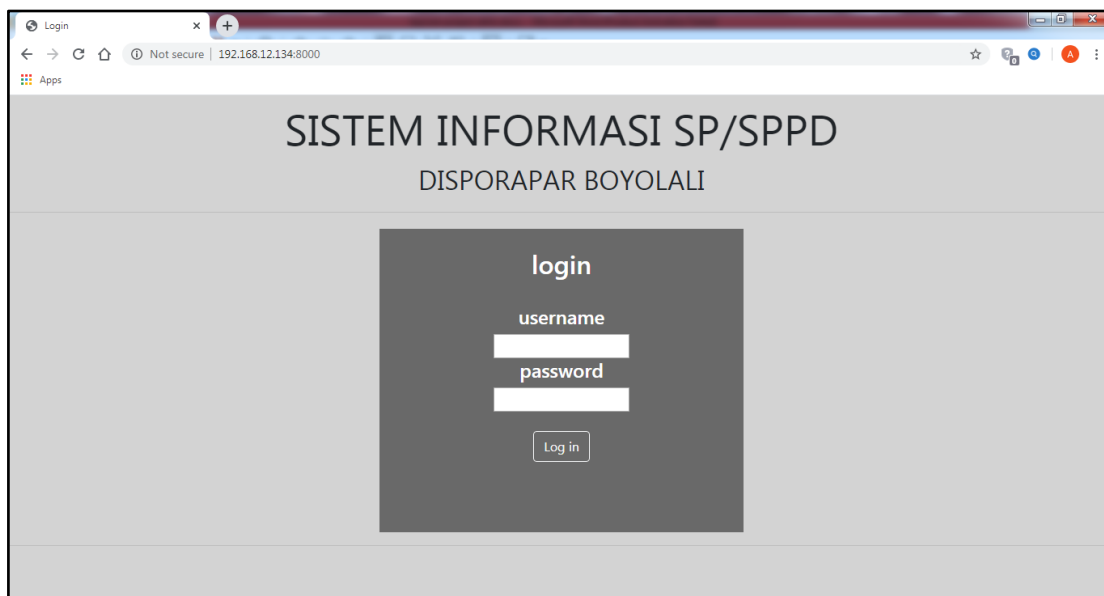
Setelah konfigurasi dan instalasi selesai, hasil dari hosting *website* menggunakan LAMPP pada Ubuntu seperti pada **Gambar 3.9** berikut ini:



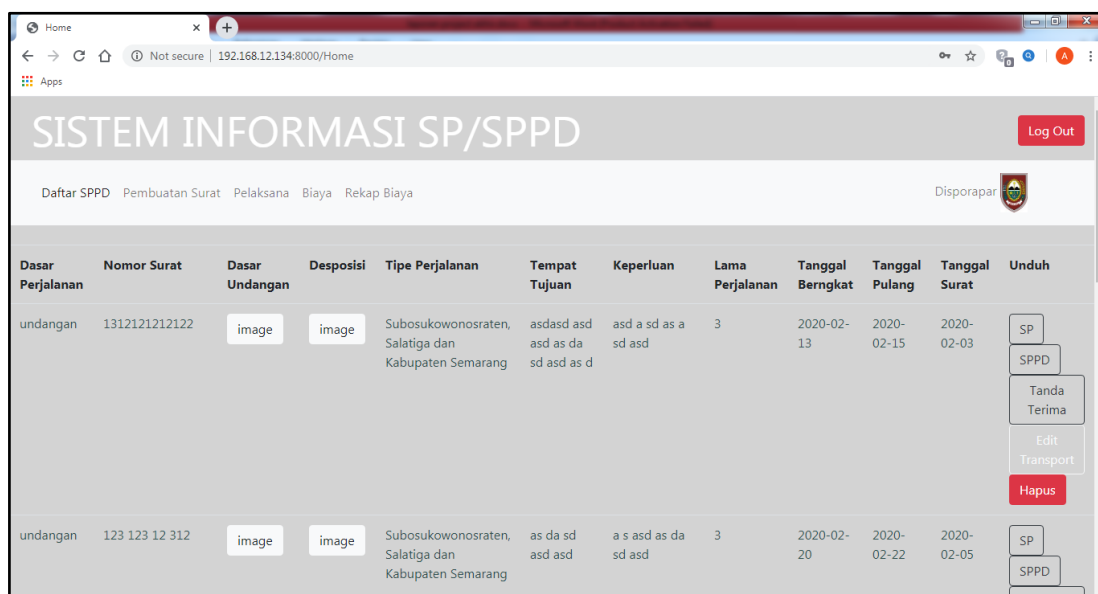
Gambar 3.9 Hasil *website* yang sudah dihosting dengan LAMPP

Untuk hasil dari docker yang sudah kita kerjakan pada tahap sebelumnya, dapat kita jalankan dengan membuka browser dan mengetikkan alamat_ip:8000. Maksud dari :8000 yaitu sesuai dengan perintah ketika kita membuat container dari image web_dev yaitu di -p 8000:80 dengan begitu ketika kita ingin mengakses website dari docker maka harus menambahkan port yang telah dibuat tadi setelah alamat ip, hasil dari webnya memiliki performa yang sama dengan web yang dihosting dengan LAMPP hanya saja web ini berjalan di container docker, dan web dalam bentuk image ini bisa dijalankan walaupun pada ubuntu tidak terinstall php,mysql dan aplikasi pendukung lainnya, karena

semua berjalan menggunakan image yang sudah dibuat tadi melalui container docker. Dokumentasi dari hasil dapat dilihat pada **gambar 4.0** dan **gambar 4.1** di bawah ini :



Gambar 4.0 Hasil website docker



Gambar 4.1 Hasil halaman home website docker

2.6 Pengujian Singkat

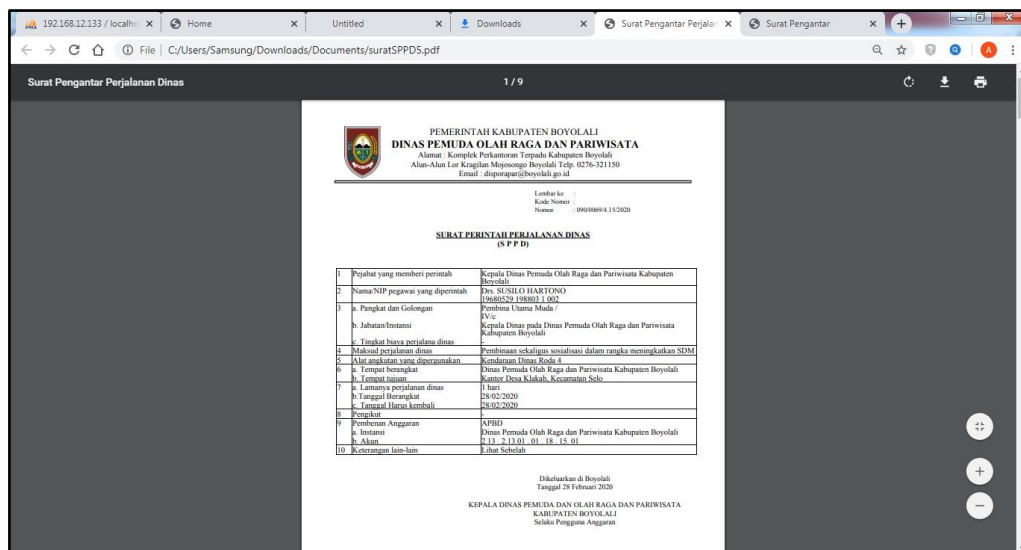
Pengujian Tugas1 :

Pada sistem informasi yang telah dirancang, akan dilakukan pengujian terhadap permasalahan yang tertuang pada bab pertama, yaitu bagaimana mempercepat proses pembuatan surat perjalanan dinas pada **Gambar 4.2** terlihat form untuk inputan komponen surat.

Gambar 4.2 form untuk pembuatan surat

Setelah proses penginputan pembuatan surat maka surat akan langsung terbuat 3 macam surat seperti pada **Gambar 4.3** yang merupakan surat SP dan **Gambar 4.4** yang merupakan surat SPPD

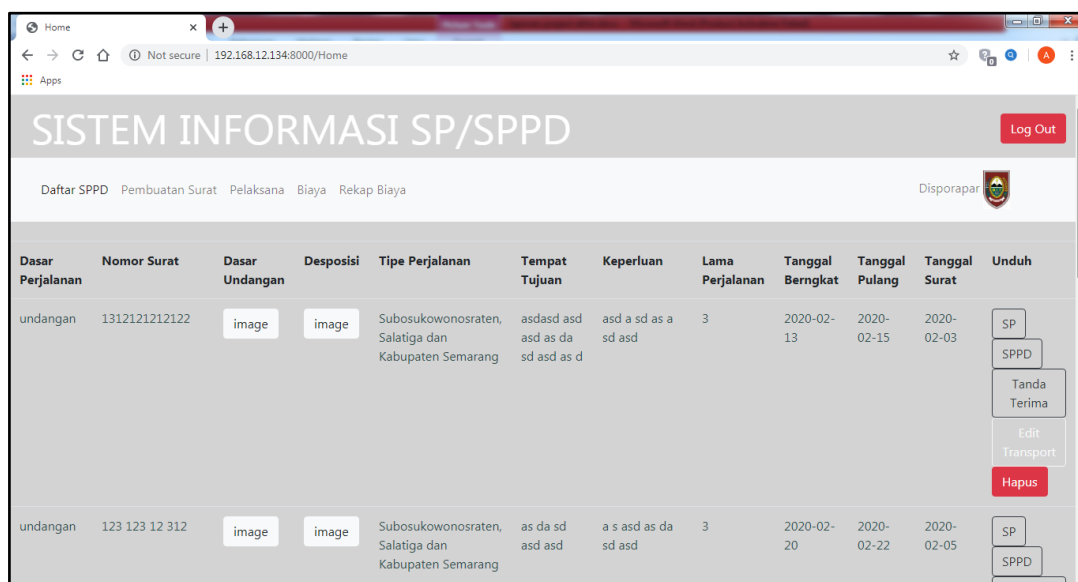
Gambar 4.3 Hasil surat SP



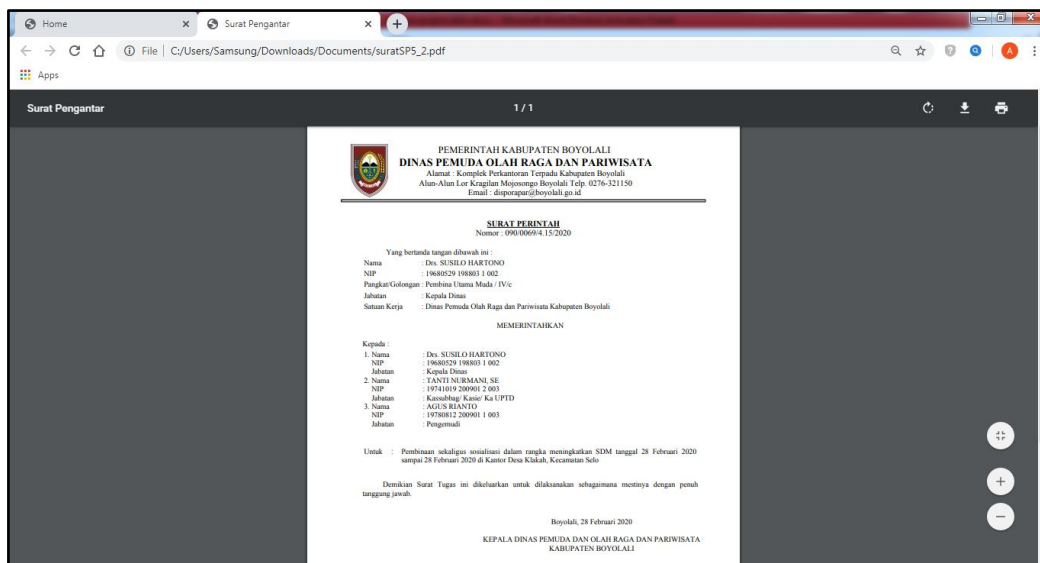
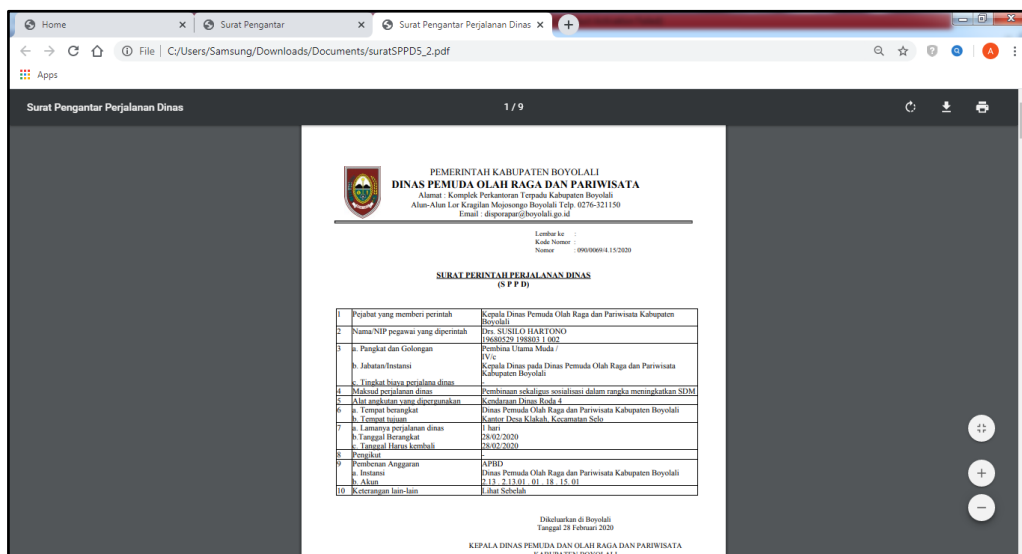
Gambar 4.4 Hasil surat SPPD

Pengujian Tugas 2:

Pada tugas 2 ini pengujiannya sama dengan tugas1 akan tetapi web yang diakses berbeda karena pada tugas 2 ini mengakses web yang dijalankan oeh container docker, dan pengujian serta performanyapun sama dengan web yang ada di tugas 1 bisa di lihat di **Gambar 4.5** , **Gambar 4.6** ,**Gambar 4.7** dan **Gambar 4.8** dibawah ini.



Gambar 4.5 Home page website docker

Gambar 4.6 Hasil surat SP *website* dockerGambar 4.7 Hasil surat SPPD *website* docker

Daftar SPPD Pembuatan Surat Pelaksana Biaya Rekap Biaya

Bulan --pilih bulan-- Tahun --pilih tahun-- [Lihat Data](#)

Rekap Bulan 01 Tahun 2020 [Download](#)

Nama	Total Biaya Makan	Total Biaya Saku	Total Biaya Representatif	Total Biaya Transport	Total Biaya Penginapan	Total
Drs. SUSILO HARTONO	600.000	1.800.000	600.000	0	1.411.200	4.411.200
THEODURUS WIDI NOEGROHO, BA	450.000	1.050.000	0	0	1.411.200	2.911.200
SRI SUPAMI, SE	0	195.000	0	1.800.000	1.500.000	3.495.000
RA. ARI MUSTIKAWATI	0	159.000	0	0	0	159.000
ERY WIDARINI	0	159.000	0	0	0	159.000
AGUS RIANTO	375.000	450.000	0	679.500	0	1.504.500

Gambar 4.8 Hasil rekap biaya *website* docker

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, April, dan 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.	Pengerjaan Tugas 1(LAMPP)										
4.	Pembuatan Laporan Submisi										
5.	Pengerjaan Tugas 2 (Docker)										
6.	Melanjutkan pembuatan laporan										
7.	Revisi										
8.	Presentasi Proyek Akhir										

3.2 Keterangan Pembagian Tugas

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

Tabel 3.2 Pembagian Tugas Proyek

No.	Keterangan Tugas	Penanggung Jawab
1.	Perancangan Arsitektur Cloud Computing	Archan & Arif
2.	Latar Belakang Masalah	Arif
3.	Agenda Pengerjaan Proyek	Arif
4.	Pembuatan Tugas 1 (LAMPP)	Archan
5.	Pembuatan Tugas 2 (Docker)	Archan
6.	Pengujian Singkat	Archan
7.	Pembuatan Laporan	Archan & Arif

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan permasalahan yang ada yaitu pembuatan sistem informasi perjalanan dinas dengan Ubuntu LAMPP beserta pembuatan docker file-nya, sistem ini dapat berjalan dengan baik dan pengujian sistem juga sesuai dengan yang diharapkan. Serta keberhasilan dari proyek akhir ini dapat memberikan manfaat bahwa sistem informasi perjalanan dinas ini dapat diimplementasikan kedalam docker dimana hal tersebut dapat memudahkan dalam pembuatan sistem informasi perjalanan dinas ini.

Pembagian kelompok dalam proyek ini menurut kami kurang efektif dikarenakan untuk teknologi docker ataupun virtualisasi machine ini perlu dipahami oleh setiap anggota kelompok. Sehingga dalam pengerjaan proyek ini mempunyai beberapa kendala karena kondisi *work from home*

4.2 Saran

Berdasarkan proses pengerjaan tugas proyek akhir ini, kami mendapatkan banyak kendala terutama untuk materi *Docker* yang tidak mudah untuk dipahami. Akan lebih baik jika materi disampaikan dengan menggunakan video tutorial yang bisa di share di *platform* media *online* karena materi tentang *Docker* tidak cukup banyak di internet.

DAFTAR PUSTAKA

- Etro, F. (2014). The economics of cloud computing. *Cloud Technology: Concepts, Methodologies, Tools, and Applications*, 4(February), 2135–2148.
<https://doi.org/10.4018/978-1-4666-6539-2.ch101>
- Wow Technology. (n.d.). *Cloud Computing dan Pemanfaatannya bagi Kehidupan Sehari-Hari*. Retrieved from wowrack: <http://blog.wowrack.co.id/2017/05/cloud-computing-dan-pemanfaatannya-bagi.html>
- Indonesian Cloud. (2016, 06 03). *Internet of Things (IoT) & Cloud Computing*. Retrieved from <https://indonesiancloud.com/internet-of-things-iot-cloud-computing/>
- Binus Center Blog. (2013, March 14). *Keuntungan dari Cloud Computing*. Retrieved from Binuscentreblog: <https://binuscenterblog.wordpress.com/2013/03/14/keuntungan-dari-cloud-computing/>
- Postmedya. (2019, January 25). *Cloud Computing : Pengertian, Sejarah serta Manfaatnya*. Retrieved from Postmedya: <http://www.postmedya.com/teknologi/cloud-computing-pengertian-sejarah-serta-manfaatnya/>

LAMPIRAN