

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**SISTEM APLIKASI IZIN PENGEMBANGAN KOMPETENSI BKD DIY
MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN
DOCKERFILENYA**



DISUSUN OLEH:

**NAMA ANGGOTA : JUDANTI CAHYANING TYAS 123170014
RIZKYANA KUSLIHAH 123170069**

KELAS : A

**ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.KOM
MUHAMMAD IMAM AL FATAH**

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

HALAMAN PENGESAHAN

SISTEM APLIKASI IZIN PENGEMBANGAN KOMPETENSI BKD DIY MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

Judanti Cahyaning Tyas

123170014

Rizkyana Kuslihah

123170069

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Al Fatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Assalamualaikum, Wr.Wb

Puji syukur saya panjatkan kepada Allah SWT yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY Menggunakan Ubuntu LAMPP dan Proses Pembuatan *Dockerfile*nya.

Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung mengenai pembahasan dan penerapan hosting cloud untuk Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY menggunakan Ubuntu LAMPP.

Laporan ini tentu saja masih jauh dari sempurna, karena masih memiliki banyak kekurangan. Oleh sebab itu, kritik dan saran dari semua pihak yang bersifat membangun selalu kami harapkan demi kesempurnaan laporan ini.

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

Wassalamualaikum, Wr.Wb

Yogyakarta, 24 Maret 2020

Penyusun

DAFTAR ISI

COVER.....	i
HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Proyek Akhir	1
1.2 Tujuan Proyek Akhir	3
1.3 Manfaat Proyek Akhir	4
1.4 Tahap Penyelesaian Proyek Akhir	5
BAB II ISI DAN PEMBAHASAN	7
2.1 Komponen yang Digunakan	7
2.2 Rancangan Arsitektur <i>Cloud Computing</i>	10
2.2.2 Rancangan Arsitektur Proyek Kedua.....	13
2.3 Parameter dan Konfigurasi	14
2.3.2 Parameter dan Konfigurasi Proyek Kedua.....	17
2.4 Tahap Implementasi	22
2.4.1 Tahap Implementasi Proyek Pertama	22
2.4.2 Tahap Implementasi Proyek Kedua	28
2.5 Hasil Implementasi	33
2.6 Pengujian Singkat.....	36
BAB III JADWAL Pengerjaan dan Pembagian Tugas	43
3.1 Agenda Pengerjaan.....	43
3.2 Keterangan Pembagian Tugas	43
BAB IV KESIMPULAN DAN SARAN	44
4.1 Kesimpulan	44
4.2 Saran.....	45
DAFTAR PUSTAKA	46
LAMPIRAN.....	47

BAB I

PENDAHULUAN

1.1. Latar Belakang Proyek Akhir

Teknologi *Cloud computing* adalah teknologi yang menjadikan internet sebagai pusat proses pengelolaan dan pengolahan daya komputasi yang dilakukan melalui jaringan. Teknologi *cloud* menyediakan kemudahan dan kecepatan layanan akses sistem dimanapun selama pengguna terhubung melalui jaringan internet. Karena dinilai dari banyaknya manfaat *cloud computing* seperti ketersediaan data yang disimpan pada penyimpanan di *server* terpusat dengan jaminan tingkat keamanan, ketersediaan listrik, fleksibilitas, skalabilitas yang tinggi serta fasilitas pendukung lainnya, menjadikan banyaknya perusahaan IT menerapkan penggunaan *cloud computing* dalam sistemnya, sejumlah perusahaan IT di antaranya adalah Google (google drive) dan IBM (blue cord initiative). Ide terciptanya *cloud computing* pada tahun 1995. Larry Ellison menawarkan ide "*Network Computing*" untuk menggantikan *PC desktop* dengan sebuah terminal yang langsung tersambung dengan sebuah server penyedia *environment* yang berisi berbagai kebutuhan *software* yang siap di akses oleh pengguna. *Cloud computing* ini mulai berkembang seiring dengan berkembangnya web dan internet.

Salah satu contoh penerapan teknologi *cloud* yang banyak digunakan saat ini adalah Google Drive. **Google Drive** adalah layanan penyimpanan daring milik Google yang diluncurkan pada 24 April 2012. Layanan ini merupakan ekstensi dari Google Docs dan akan mengganti URL docs.google.com dengan drive.google.com setelah diaktifkan. Google Drive memberikan layanan penyimpanan gratis sebesar 15 GB dan dapat ditambahkan dengan pembayaran tertentu. Dengan fitur unggulan yang sama seperti Dropbox, yaitu sinkronisasi data melalui folder khusus di dalam desktop atau lebih dikenal dengan Desktop Sync Clients. GDrive memberikan kapasitas gratis sebesar 5 GB dan tentunya fitur-fitur yang terintegrasi dengan layanan Google lainnya seperti: Gmail, G+ dan Google Search. Fitur yang bisa digaris bawahi dari GDrive adalah API's untuk para Developer. Hingga kini GDrive telah terhubung dengan puluhan aplikasi pihak ketiga. GDrive dapat digunakan menjamin keamanan data bagi penggunanya, jika ponsel hilang atau rusak, data yang ada di dalam ponsel tidak akan hilang selama keseluruhan data sudah ter-backup oleh GDrive. Selain itu, terdapat fitur yang memungkinkan pengaturan terhadap hak akses

pengguna pada file yang disimpan dalam GDrive, sehingga tidak sembarang orang dapat mengakses data tersebut tanpa seizin pemilik data.

Judul yang kami ulas yakni Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY menggunakan Ubuntu LAMPP dan Docker. Izin Pengembangan Kompetensi merupakan sebuah program yang diadakan oleh pemerintah bagi Aparatur Sipil Negara (ASN) agar memiliki kesempatan mengembangkan kompetensi melalui tugas Perjalanan Dinas ke Luar Negeri maupun Izin Belajar. Perjalanan Dinas ke luar negeri dilakukan dengan tujuan untuk menghadiri seminar, lokakarya, simposium, konferensi, peninjauan, studi perbandingan maupun untuk hal-hal yang mempunyai prioritas tinggi dan penting bagi Lembaga Negara atau Instansi Pemerintah. Pengembangan ini tidak hanya berfungsi sebagai peningkatan pengetahuan bagi ASN, tapi juga sudah menjadi kebutuhan organisasi untuk meningkatkan kinerja dengan Sumber Daya Manusia (SDM) yang kompeten. Dalam menyusun strategi dalam melakukan izin belajar dan menentukan jurusan, para pegawai harus menyesuaikan pengembangan kompetensi dengan kebutuhan organisasi.

Pengajuan izin pengembangan kompetensi membutuhkan surat pengantar yang diberikan oleh Lembaga Negara atau Instansi Pemerintah yang menaungi ASN dalam melaksanakan tugas tersebut. Beberapa berkas penting juga diperlukan sebagai syarat kelengkapan dalam mengajukan pelaksanaan izin pengembangan kompetensi. Proses input data pribadi beserta berkas ASN yang ditugaskan untuk melakukan izin pengembangan kompetensi masih banyak dilakukan menggunakan cara manual. Hal ini terkadang masih menimbulkan terjadinya kesalahan pada proses input data pengajuan izin. Untuk itu dibuatlah suatu sistem berbasis web yang digunakan untuk mempermudah proses input data pengajuan izin pengembangan kompetensi oleh *user*, dan juga mempermudah *admin* dalam proses pendataan surat pengajuan izin dan berkas kelengkapan ASN.

Dari masalah yang telah dijabarkan di atas terkait dengan perancangan sistem, masih terdapat beberapa kekurangan, diantaranya adalah penggunaan sistem yang tidak terintegrasi menggunakan teknologi *cloud* memiliki beberapa kelemahan terkait dengan keamanan informasi database ASN yang bersifat rahasia sehingga apabila terdapat kerusakan yang terjadi pada sistem akan berakibat fatal pada data yang ada di dalamnya karena tidak terdapat *backup server* menggunakan arsitektur *recovery*, sulitnya melakukan dokumentasi laporan administrasi dan proses pembuatan surat pengajuan izin yang masuk disebabkan karena keterbatasan akses sistem yang hanya bisa dilakukan melalui komputer kantor. Oleh karena

itu, solusi yang diusulkan pada penelitian ini dalam menyelesaikan permasalahan di atas adalah dengan penggunaan penerapan layanan berbasis *cloud computing* pada sistem.

Komponen *software* yang digunakan dalam merancang arsitektur *cloud* sistem adalah *software* VMware Workstation 15.0, ISO installer Ubuntu LAMPP, dan *Docker*. Data yang akan digunakan sementara menggunakan database Izin Luar Negeri yang telah diinput, untuk database Izin Belajar masih digunakan data percobaan sementara. Siapkan PC yang akan digunakan untuk membuat virtualisasi *cloud*, selanjutnya install Ubuntu server yang telah disiapkan dalam penyimpanan desktop dengan menggunakan *software* VMware Workstation 15.0. Setelah itu install *software* pengelola database untuk mendukung proses pembuatan layanan SaaS seperti Apache, PHP, Mysql, dll. Aplikasi yang sudah terinstall selanjutnya lakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Aplikasi Izin Pengembangan Kompetensi ketersediaannya / *availability*-nya maksimal. Terakhir, lakukan pengujian sistem pada semua komponen untuk mengetahui kesesuaian jalannya sistem dengan hasil yang diharapkan.

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 Aplikasi Izin Pengembangan Kompetensi yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04.4 dan LAMPP (Apache 2.4, PHP 7.3, MySQL 5.2).
2. Meminimalisir proses *maintenance* pada sistem layanan yang terintegrasi dengan teknologi *cloud computing*.
3. Meningkatkan kecepatan akses sistem sehingga waktu tunggu untuk memuat keseluruhan program ke dalam tampilan layar untuk diakses oleh *user* dan *admin* menjadi lebih cepat.
4. Meningkatkan keamanan data pada sistem dengan membatasi hak akses hanya pada pihak-pihak yang berwenang terhadap jalannya program.
5. Terhadap judul proyek akhir Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY maka dengan laporan ini akan dituliskan cara penyelesaian dengan menggunakan penerapan rancangan layanan *cloud computing* sistem aplikasi

berbasis web menggunakan VM Ubuntu yang di dalamnya terdapat layanan LAMPP untuk selanjutnya dilakukan pembuatan *Dockerfile*-nya, kemudian dipublikasikan sehingga akan menghasilkan program Sistem Aplikasi Izin dengan memanfaatkan sistem penyedia layanan *cloud* agar setiap *client* ataupun *user* dapat saling menggunakan aplikasi dari *database server* dimanapun, sehingga setiap perubahan dari sisi server maupun client akan dapat langsung tersimpan pada seluruh sistem.

1.3 Manfaat Proyek Akhir

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

1. Sistem Aplikasi input Izin Pengenbangan Kompetensi yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.
2. Perusahaan pengguna jasa *cloud* tidak perlu mempermasalahkan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.
3. Batasan memori penyimpanan data pada sistem menjadi tidak terbatas dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing*. Akses sistem yang lebih cepat karena semua data sudah tersimpan dalam penyimpanan yang tersedia. Sehingga waktu tunggu untuk memuat keseluruhan program ke dalam tampilan layar untuk diakses oleh *user* dan *admin* menjadi lebih cepat.
4. Permintaan data secara *realtime* dapat dilakukan secara terpusat maupun secara terdistribusi dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*. Sehingga jaringan pada sistem lebih terjamin keamanannya karena hanya pihak berwenang saja yang dapat mengakses sistem tersebut.
5. Mengurangi kesalahan dan meningkatkan efisiensi yang dilakukan pada proses input data ke dalam sistem karena semua data akan diproses dan didokumentasikan dalam database secara otomatis oleh sistem.

1.4 Tahap Penyelesaian Proyek Akhir

Program yang dirancang menggunakan Ubuntu LAMPP sebagai sarana hosting cloud yang dibuat berupa simulasi jaringan virtual di dalam sebuah PC yang telah terinstall aplikasi VMware Workstation 15.0 dan juga ISO *installer* Ubuntu LAMPP.

Dalam membangun rancangan sistem, dilakukan proses instalasi aplikasi Ubuntu LAMPP menggunakan ISO *installernya*. Setelah proses instalasi selesai, selanjutnya buat server baru untuk Ubuntu LAMPP dalam VMware. Ketika proses pembuatan server baru selesai dilakukan, server Ubuntu LAMPP telah berhasil dibuat dan siap untuk digunakan.

Upload berkas pada server hosting cloud, dilakukan melalui aplikasi WinSCP dengan cara drag and drop pada file yang hendak diupload pada bagian sebelah kiri (lokasi file di komputer) ke bagian sebelah kanan (lokasi server hosting). Lalu klik save.

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

1. Menganalisis kebutuhan dari sistem Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Menentukan konfigurasi yang tepat untuk pengaturan Ubuntu LAMPP dan pembuatan Dockerfile sehingga dapat digunakan sesuai *requirement* yang berupa:
 - a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Terdapat divisi pusat (admin) yang dapat masuk ke semua bagian sistem kecuali akun yang tidak terdaftar/yang bukan admin program.
 - c. Menggunakan Ubuntu dengan versi 18.04.4 LTS.
3. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
4. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Aplikasi Izin Pengembangan Kompetensi ketersediaannya/*availability*-nya maksimal.
5. Mengunggah berkas proyek yang ada pada Workstation ke dalam Server melalui aplikasi WinSCP.
6. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, kesesuaian hasil output sistem sebelum dan sesudah dilakukan hosting, dsb...

7. Menghasilkan Sistem Aplikasi Izin Pengembangan Kompetensi yang berbasiskan *cloud computing* sesuai standar ISO 9001.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Komponen yang digunakan pada arsitektur virtualisasi *cloud* yang digunakan pada tugas proyek akhir ini dibagi menjadi dua sesuai dengan yang telah dituliskan pada judul di awal yang merupakan integrasi yang berkesinambungan antara kedua topik permasalahan. Proyek pertama yakni “Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY Menggunakan Ubuntu LAMPP” dan proyek kedua “Proses Pembuatan *Dockerfile*nya”.

2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY 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 dan bahasa HTML dengan versi minimal 3.3.2.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data izin kompetensi kepegawaian yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaannya ialah seluruh pegawai kantor Badan Kepegawaian di Yogyakarta. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet kantor Badan Kepegawaian dan memiliki akun untuk dapat mengakses laman web. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali pihak-pihak yang berkepentingan terhadap hak akses sistem.
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. Komponen penting yang terdapat pada perancangan *server* arsitektur *cloud computing* dalam LAMPP—yang merupakan sebuah paket perangkat lunak bebas untuk menjalankan aplikasi secara lengkap—yakni Apache dengan versi 2.4, PHP dengan versi 7.3, MySQL dengan versi 5.7 dan phpMyAdmin dengan versi 7.2.

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.	Merk Server	Virtual Machine dengan VMWare Workstation 15	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	i5 core @2.5Ghz	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: 192.168.190.130/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.190.130	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.190.130	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.	RAM	8GB	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 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Preprosesor konten <i>Web Server</i> dengan jenis PHP (ekstensi PHP)
		MySQL	Preprosesor layanan manajemen basis data
		phpMyAdmin	Preprosesor layanan manajemen database MySQL berbasis <i>Web GUI</i>
2.	HTML	Versi 3.2.2	Dukungan bahasa pemrograman yang digunakan oleh sistem izin kompetensi kepegawaian.
3.	MySQL	Versi 5.2	Dukungan penyimpanan database yang digunakan oleh sistem izin kompetensi kepegawaian.

4.	puTTY	Versi 0.73	<i>Software</i> untuk proses instalasi Apache, PHP, MySQL dan phpMyAdmin
5.	WinSCP	Hostname: 192.168.190.130 Username: projekakhir	<i>Software</i> untuk mengunggah berkas proyek dari penyimpanan workstation ke <i>server cloud</i>

2.1.2 Komponen pada Proyek Kedua

Untuk pembuatan *Dockerfile* “Program Aplikasi Izin Pengembangan Kompetensi” pada server LAMPP dengan menggunakan Ubuntu 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 dan bahasa HTML dengan versi minimal 3.3.2.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data izin kompetensi kepegawaian yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh pegawai kantor Badan Kepegawaian di Yogyakarta. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet kantor Badan Kepegawaian dan memiliki akun untuk dapat mengakses laman web. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali pihak-pihak yang berkepentingan terhadap hak akses sistem.
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. Komponen penting yang terdapat pada perancangan *server* arsitektur *cloud computing* dalam LAMPP—yang merupakan sebuah paket perangkat lunak bebas untuk menjalankan aplikasi secara lengkap—yakni Apache dengan versi 2.4, PHP dengan versi 7.3, MySQL dengan versi 5.7 dan phpMyAdmin dengan versi 7.2.

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 kedua

No.	Nama Parameter	Nilai	Keterangan
1.	Merk Server	Virtual Machine dengan VMWare Workstation 15	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	i5 core @2.5Ghz	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: 192.168.190.130/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.190.130	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.190.130	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.	RAM	8GB	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 *Docker* untuk proyek kedua

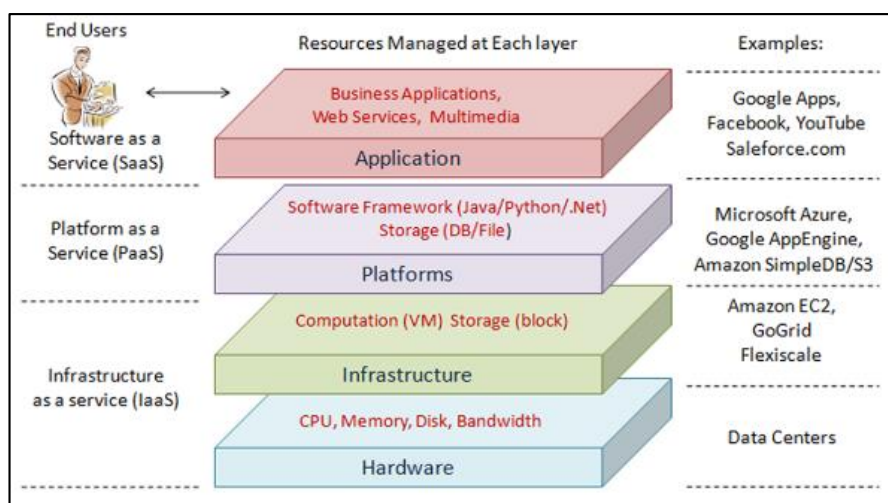
No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Preprosesor konten <i>Web Server</i> dengan jenis PHP (ekstensi PHP)
		MySQL	Preprosesor layanan manajemen basis data
2.	puTTY	Versi 0.73	<i>Software</i> untuk proses instalasi Apache, PHP, MySQL dan phpMyAdmin

2.2 Rancangan Arsitektur *Cloud Computing*

Rancangan arsitektur pada tugas proyek akhir ini dibagi menjadi dua sesuai dengan yang telah dituliskan pada judul di awal yang merupakan integrasi yang berkesinambungan antara kedua topik permasalahan. Proyek pertama yakni “Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY Menggunakan Ubuntu LAMPP” dan proyek kedua “Proses Pembuatan *Dockerfilenya*”.

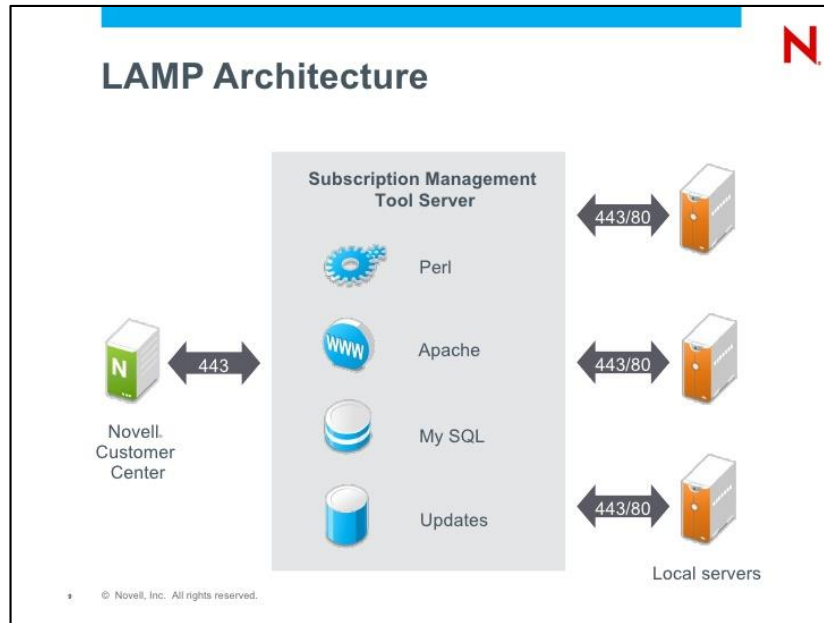
2.2.1 Rancangan Arsitektur Proyek Pertama

Pada proyek akhir ini digunakan rancangan arsitektur layanan IaaS, dengan menggunakan hardware laptop ASUS A442U sebagai layer utama yang terletak di bagian paling bawah struktur diagram dan berfungsi sebagai hardware perancangan arsitektur virtualisasi layanan *cloud computing*. Kemudian pada layer tengah—layer yang terletak di atas IaaS—merupakan layanan PaaS tempat berjalannya sistem operasi Windows 10 dan juga aplikasi VMware Workstation 15—yang berperan sebagai software perancangan *server cloud*. Sedangkan di layer paling atas terdapat layanan SaaS tempat berjalannya program web terkait dengan sistem aplikasi yang telah disimpan dalam layanan hosting *cloud*. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.2.1** berikut ini:



Gambar 2.2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya

Rancangan arsitektur LAMP dalam proyek ini terdiri dari alat *server* yang di dalamnya terdapat komponen sistem operasi Linux, PHP/Perl/Python (bahasa pemrograman yang dipakai), Apache *web server* dan sistem basis data MySQL. Pada *local server* terdapat *device* dengan akses menggunakan *port* 443 (digunakan untuk menjalankan *Secure Server Layer*) dan *port* 80 (*port* untuk menghubungkan koneksi antara *client* dengan *web server*). Untuk lebih jelasnya, dapat dilihat pada **Gambar 2.2.2** berikut ini:

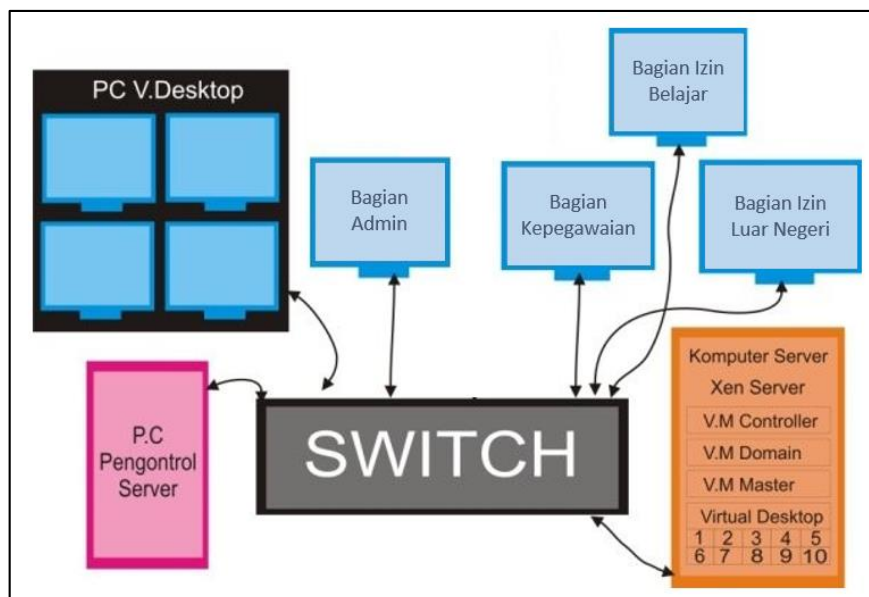


Gambar 2.2.2 Rancangan Arsitektur LAMPP

Pada perancangan virtualisasi untuk sistem web berbasis *cloud* ini digunakan satu buah *switch* untuk menghubungkan setiap PC agar bisa berinteraksi antara satu sama lain. Terdapat juga empat buah PC untuk masing-masing bagian yang berperan penting dalam akses web seperti contohnya bagian admin yang berwenang mengendalikan data sistem web. PC *client/user* ditunjukkan pada gambar yang terdapat di dalam kotak hitam. *Client* disini berperan sebagai *user* yang melakukan input data izin pengembangan kompetensi.

PC pengendali *server*—seperti yang ditunjukkan pada gambar kotak berwarna pink—memiliki tugas untuk mengendalikan keseluruhan aktivitas yang sedang berjalan dan tercatat di dalam *server* sistem, termasuk dalam pembaruan dan perawatan terhadap sistem. Virtual mesin sistem—pada gambar kotak berwarna oranye—merupakan layanan implementasi virtualisasi berbasis *cloud* yang digunakan pada *server*. Dan merupakan perangkat penting berfungsi untuk menyimpan program web yang telah di*hosting*.

Seperti yang dijelaskan pada paragraf sebelumnya, topologi jaringan tersebut dapat dilihat pada **Gambar 2.2.3** berikut ini:



Gambar 2.2.3 Topologi Jaringan server Ubuntu LAMPP

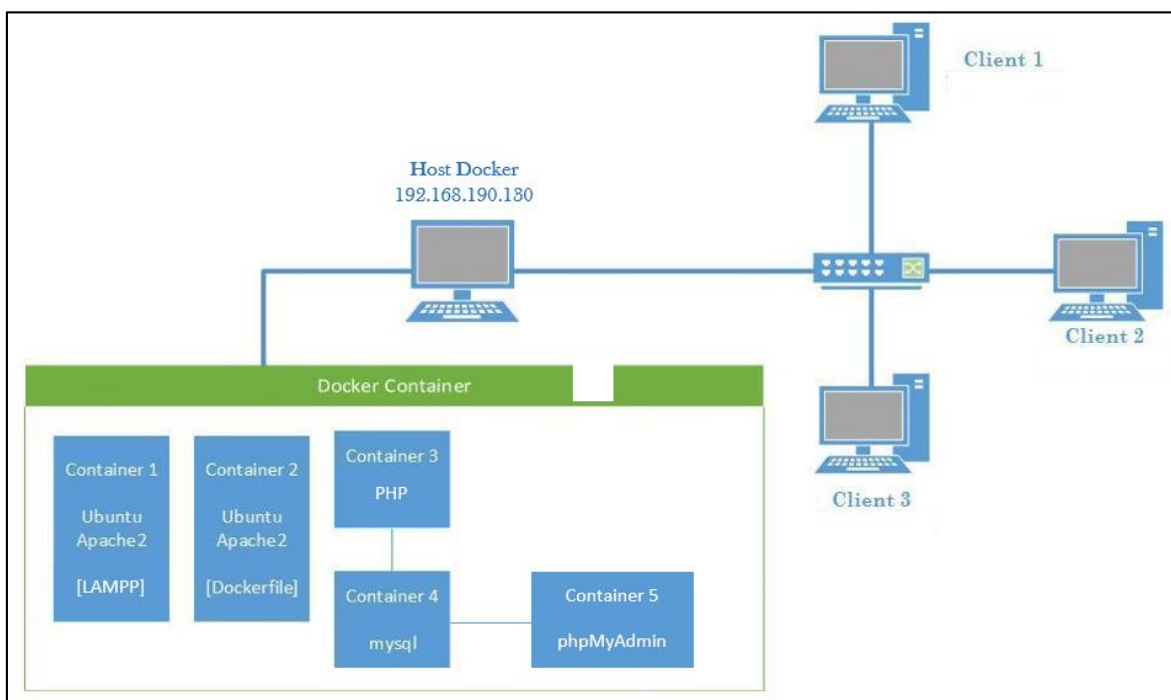
2.2.2 Rancangan Arsitektur Proyek Kedua

Rancangan arsitektur virtualisasi proyek kedua untuk sistem web berbasis *cloud* ini digunakan satu buah *switch* untuk menghubungkan antara *PC Host Docker* dan *client* agar bisa berinteraksi antara satu sama lain. *Host Docker* berperan sebagai *admin* pengelola *web server* yang berwenang mengendalikan dan memantau jalannya sistem beserta data yang ada di dalamnya. *Client* berperan sebagai *user* yang melakukan input data izin pengembangan kompetensi.

PC Host Docker server memiliki tugas untuk mengendalikan keseluruhan aktivitas yang sedang berjalan dan tercatat di dalam *server* sistem, termasuk dalam pembaruan dan perawatan terhadap sistem. *PC Host Docker* merupakan satu-satunya PC yang terhubung pada *Docker container* karenahanya *Host Docker* yang berhak mengakses bagian dalam program yang terdapat di dalam *Docker container* yang merupakan perangkat penting berfungsi untuk menyimpan program web yang telah dihosting.

Di dalam *Docker container* terdapat 5 *container* yang masing-masing *container* terdapat peran yang mendukung jalannya program pada *server Docker*. *Container 1* berisi layanan LAMPP yang digunakan pada *server*. *Container 2* berisi *Dockerfile*, *Container 3* berisi file web program yang akan diakses melalui *web browser* (file web PHP), *Container 4* berisi MySQL dan terakhir *Container 5* berisi file phpMyAdmin (file *database* program).

Seperti yang dijelaskan pada paragraf sebelumnya, topologi *Docker server* yang digunakan tersebut dapat dilihat pada **Gambar 2.2.4** berikut ini:



Gambar 2.2.4 Topologi Jaringan *server* Ubuntu LAMPP

2.3 Parameter dan Konfigurasi

Parameter dan konfigurasi yang digunakan pada arsitektur virtualisasi *cloud* yang digunakan pada tugas proyek akhir ini dibagi menjadi dua sesuai dengan yang telah dituliskan pada judul di awal yang merupakan integrasi yang berkesinambungan antara kedua topik permasalahan. Proyek pertama yakni “Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY Menggunakan Ubuntu LAMPP” dan proyek kedua “Proses Pembuatan *Dockerfile*nya”.

2.3.1 Parameter dan Konfigurasi Proyek Pertama

Proses yang dilakukan sebelum melakukan instalasi Apache, dilakukan *update* (pembaruan sistem). Proses instalasi Apache membutuhkan parameter yang digunakan untuk instalasi Apache dapat dilihat pada penjelasan **Modul 2.1** di bawah ini:

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

Keterangan:

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

Modul 2.1 Parameter instalasi Apache

Sintaks konfigurasi proses instalasi Apache yang diperlukan untuk pengaturan Firewall sehingga lalu lintas data dari Apache Web Server dengan tipe *protocol* HTTP (port 80) dan HTTP (port 443) diizinkan dapat dilihat pada penjelasan **Modul 2.2** berikut ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- ufw : parameter yang digunakan untuk memudahkan konfigurasi Firewall
- allow : perintah untuk mengizinkan akses

Modul 2.2 Konfigurasi instalasi Apache

Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan **Modul 2.3** berikut ini:

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

Keterangan:

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

Modul 2.3 Parameter instalasi MySQL

Proses pengaturan dasar pengamanan MySQL terkait validasi penggunaan dan pengaturan kata sandi, user, remote login, database yang secara default terpasang pada MySQL dan hak akses menggunakan sintaks seperti yang dapat dilihat pada penjelasan **Modul 2.4**. Konfigurasi tersebut diperlukan dalam melakukan proses instalasi MySQL untuk merubah validasi penggunaan kata sandi agar tidak menggunakan kombinasi yang kuat supaya kata sandi mudah diingat.

```
$ sudo mysql_secure_installation
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- mysql_secure_installation : script untuk meningkatkan keamanan instalasi MariaDB

Modul 2.4 Konfigurasi instalasi MySQL

Parameter instalasi PHP yang terdapat pada baris pertama merupakan parameter yang digunakan untuk proses instalasi PHP. Dan parameter di baris kedua adalah parameter yang digunakan untuk proses pembuatan sebuah file dengan format .php bernama info yang nantinya akan digunakan pada pengecekan keberhasilan proses instalasi PHP. Parameter instalasi PHP dapat dilihat pada penjelasan **Modul 2.5** berikut:

```
$ sudo apt install php libapache2-mod-php php-mysql
$ sudo nano /var/www/html/info.php
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- php : nama paket aplikasi untuk php
- libapache2-mod-php : paket yang dirancang dan dikompilasi untuk bekerja dengan Apache web server
- php-mysql : nama paket aplikasi untuk php mysql
- nano : media aplikasi yang digunakan untuk membuat berkas (sejenis aplikasi Notepad, namun berbasis CLI)
- /var/www/html/info.php : parameter yang digunakan untuk membuat sebuah berkas dengan format .php bernama info pada lokasi Web Server Apache yakni di /var/www/html/info.php

Modul 2.5 Parameter instalasi PHP

Parameter yang terdapat pada baris pertama digunakan untuk instalasi phpMyAdmin. Parameter yang berada di baris kedua memiliki fungsi untuk memperbaiki *error* yang disebabkan karena konfigurasi *user* root yang digunakan untuk *login* diatur dengan model *plugin Unix Socket based authentication*, sehingga perlu diatur *plugin* menggunakan *native MySQL authentication*. Sintaks *update* yang terdapat pada baris ketiga digunakan untuk mengatur parameter *plugin* dari *user* root. Parameter yang digunakan untuk instalasi phpMyAdmin dapat dilihat pada penjelasan **Modul 2.6** di bawah ini:

```
$ sudo apt install phpmyadmin php-mbstring php-gettext
$ sudo mysql -u root
UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('KATA_SANDI_ROOT_USER') WHERE User =
'root'
```

Keterangan:

- install : perintah untuk instalasi paket
- phpmyadmin : nama paket aplikasi untuk php
- php-mbstring : modul dalam php yang digunakan untuk mengkonversi string menjadi sintaks yang berbeda
- php-gettext : modul dalam php yang mendukung pengenalan bahasa pada server
- mysql : parameter akses ke mysql
- root : parameter untuk akses ke CLI dari konfigurasi mysql sebagai user root
- UPDATE mysql.user : sintaks untuk mengatur parameter plugin dari user

- `mysql_native_password` : default authentication plugin yang digunakan dalam pembuatan akun ketika kolom plugin kosong
- `authentication_string` : autentikasi user

Modul 2.6 Parameter instalasi phpMyAdmin

2.3.2 Parameter dan Konfigurasi Proyek Kedua

Proses instalasi *Docker* diawali dengan menghapus paket instalasi *Docker* yang dahulu pernah diinstall untuk diganti dengan paket *Docker Engine* dengan versi terbaru. Proses menghapus paket instalasi *Docker Engine* membutuhkan parameter yang dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

```
$ sudo apt-get apt-get remove docker docker-engine docker.io containerd
runc
```

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `apt-get` : parameter untuk mengatur paket aplikasi yang ada pada linux
- `remove` : perintah untuk menghapus paket
- `docker / docker-engine / docker.io` : versi lama Docker Engine yang sebelumnya pernah diinstall

Modul 2.7 Parameter upate dan install paket indeks

Selanjutnya, lakukan proses *update* (pembaruan sistem) pada indeks paket apt dilanjutkan dengan proses install indeks paket agar apt dapat diizinkan untuk mengakses repository melalui https. Proses *update* dan instalasi *Docker* membutuhkan parameter yang dapat dilihat pada penjelasan **Modul 2.8** berikut ini:

```
$ sudo apt-get update
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
```

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `apt-get` : parameter untuk mengatur paket aplikasi yang ada pada linux
- `update` : perintah untuk memperbarui paket
- `install` : perintah untuk instalasi indeks paket apt

Modul 2.8 Parameter update dan install indeks Docker

Selanjutnya unduh Docker melalui URL dengan menggunakan parameter seperti pada **Modul 2.9** berikut:

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

Keterangan:

- curl : untuk menguji konektivitas ke URL dan juga sebagai tool transfer data
- fsSL : untuk mengamankan transmisi data antara dua sistem yang berbeda
- https://download.docker.com/linux/ubuntu/gpg :
- apt-key : perintah pengelola daftar kunci yang digunakan oleh apt untuk mengautentikasi paket. Paket yang telah diautentikasi menggunakan kunci ini akan dianggap terpercaya

Modul 2.9 Parameter penambahann GPG key Docker

Setelah proses muat ulang daftar aplikasi yang tersedia dengan apt update, selanjutnya tambahkan GPG key Docker untuk memperbaiki error yang disebabkan karena GPG key tidak ditemukan/tidak valid dari repositori.

Periksa dan pastikan jika memiliki kunci dengan *fingerprint* 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, dengan melihat 8 karakter terakhir dari *fingerprint*, seperti yang dapat dilihat pada **Modul 2.10** di bawah ini:

```
$ sudo apt-key fingerprint 0EBFCD88
pub  rsa4096 2017-02-22 [SCEA]
    9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid          [ unknown] Docker Release (CE deb) <docker@docker.com>
sub  rsa4096 2017-02-22 [S]
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt-key : perintah pengelola daftar kunci yang digunakan oleh apt untuk mengautentikasi paket. Paket yang telah diautentikasi menggunakan kunci ini akan dianggap terpercaya

Modul 2.10 fingerprint GPG key Docker

Gunakan perintah untuk mengatur *stable repository* dengan menambahkan repositori apt baru ke dalam direktori seperti yang tercantum pada **Modul 2.11** di bawah ini:

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

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- add-apt : script untuk menambahkan repositori apt ke dalam direktori
- lsb_release -cs : mencetak informs LSB (Linux Standard Base) dan distribusi tertentu

Modul 2.11 Konfigurasi penambahan repositori

Perbarui paket indeks apt, dan install versi terbaru dari *Docker Engine* dan *containerd*, atau menuju ke tahap instalasi selanjutnya untuk menginstall versi spesifik saja dengan menggunakan parameter yang diperlihatkan pada **Modul 2.12** di bawah ini:

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

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `apt-get` : parameter untuk mengatur paket paket aplikasi yang ada pada linux
- `docker-ce / docker-ce-cli` : versi terbaru Docker Engine yang hendak diinstall

Modul 2.12 Parameter

Periksa jika *Docker Engine* telah diinstall pada *desktop* dengan menjalankan 'hello-world image'. Perintah ini akan mengunduh gambar uji dan menjalankannya pada *container*. Ketika *container* berjalan, maka akan mencetak pesan informasi. Perintah untuk menjalankan 'hello world' dapat dilihat pada **Modul 2.13** berikut:

```
$ sudo docker run hello-world
```

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `run` : perintah untuk menjalankan sebuah command pada Docker

Modul 2.13 Parameter untuk menjalankan hello world

Setelah instalasi *Docker* berhasil, dilanjutkan dengan proses menggunakan *Docker* untuk membuat LAMPP server untuk proyek tugas akhir. Gunakan parameter seperti yang dapat dilihat pada **Modul 2.14** berikut untuk menginstall curl yang tidak terinstall pada distribusi Linux.

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

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `install` : perintah untuk instalasi paket
- `curl` : perintah untuk

Modul 2.14 Parameter untuk install curl

Unduh komponen *Docker-compose* dengan menggunakan parameter seperti yang tercantum pada **Modul 2.15** di bawah ini:

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

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `install` : perintah untuk instalasi paket
- `curl` : perintah untuk

Modul 2.14 Parameter untuk install

Setelah *Docker-compose* berhasil diunduh, jalankan perintah dengan menggunakan parameter seperti yang dapat dilihat pada **Modul 2.15** baris pertama. Pastikan juga perintah *Docker-compose* bekerja dengan mengetikkan perintah seperti yang dituliskan pada **Modul 2.15** baris kedua berikut:

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

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `install` : perintah untuk instalasi paket
- `curl` : perintah untuk

Modul 2.15 Parameter untuk menjalankan *Docker-compose*

Buat direktori proyek **~/docker/lampp** dan **html** direktori di dalam direktori proyek untuk menyimpan file website (misal: PHP, HTML, dll.) menggunakan perintah pada **Modul 2.16** berikut:

```
$ mkdir -p ~/docker/lampp/html
```

Keterangan:

- `mkdir` : digunakan untuk membuat direktory atau folder baru

Modul 2.16 Parameter pembuatan direktori proyek

Navigasikan ke direktori proyek dan buat file **php.Dockerfile** ke dalam direktori **~/docker/lampp**. Dockerfile ini memungkinkan `mysqli` dan `PDO php extensions` pada **php:7.4.3-apache** image dari *Docker Hub* dan membangun kustomisasi *Docker image*. Parameter yang digunakan pada konten **php.Dockerfile** dapat dilihat pada **Modul 2.17** berikut:

```
$ cd ~/docker/lampp
$ nano.Dockerfile
```

Keterangan:

- `cd` : digunakan untuk berpindah direktori

Modul 2.17 Parameter pembuatan *php.Dockerfile*

Buat file **docker-compose.yml** di dalam direktori proyek **~/docker/lampp** menggunakan parameter pada **Modul 2.18**. Tampilan isi pada file **docker-compose.yml** dapat dilihat pada bagian **Tahap Implementasi Proyek Kedua**.

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

Keterangan:

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

Modul 2.18 Parameter pembuatan file docker-compose.yml

Pindah pada direktori **~/docker/lampp/html** untuk membuat file baru bernama **index.php** dengan menggunakan parameter yang terdapat pada **Modul 2.19** berikut ini:

```
$ cd html/
```

```
$ nano index.php
```

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `cd` : digunakan untuk berpindah direktori

Modul 2.19 Parameter pembuatan file index.php

Kemudian, lihat percabangan file program pada *Docker* dengan menggunakan parameter perintah seperti yang dapat dilihat pada **Modul 2.20** di bawah ini:

```
$ tree
```

Keterangan:

- `tree` : perintah untuk melihat percabangan file program

Modul 2.20 Parameter untuk melihat percabangan program

Untuk memulai menjalankan **web-server**, **mysql-server** dan **phpmyadmin** gunakan perintah yang tertera pada parameter yang terdapat di **Modul 2.21**. Semua layanan akan berjalan di *background*.

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

Keterangan:

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

Modul 2.21 Parameter *running Docker server*

Untuk melihat mapping port pada *Docker* dapat digunakan parameter yang terdapat pada **Modul 2.22** berikut ini:

```
$ sudo docker-compose ps
```

Keterangan:

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

Modul 2.22 Parameter *mapping port* pada *Docker*

Hasil tampilan instalasi dan pembuatan file program menggunakan *Docker* dapat dilihat pada bagian **Tahap Implementasi Proyek Kedua**.

2.4 Tahap Implementasi

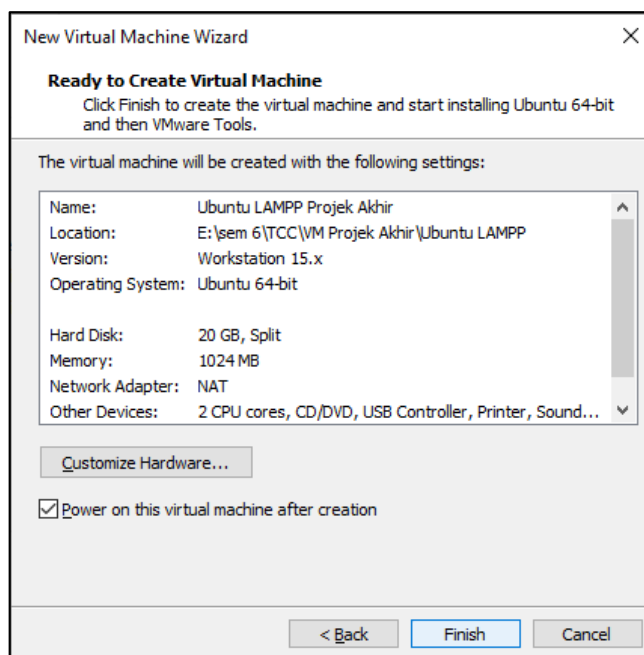
Tahap implemetasi pada tugas proyek akhir ini dibagi menjadi dua sesuai dengan yang telah dituliskan pada judul di awal yang merupakan integrasi yang berkesinambungan antara kedua topik permasalahan. Proyek pertama yakni “Sistem Aplikasi Izin Pengembangan Kompetensi BKD DIY Menggunakan Ubuntu LAMPP” dan proyek kedua “Proses Pembuatan *Dockerfilenya*”.

2.4.1 Tahap Implementasi Proyek Pertama

Tahap awal pembuatan rancangan layanan virtualisasi menggunakan layanan *cloud computing* VM pada VMware Workstation dengan memilih mode *custom (advanced)* untuk membuat mesin virtual dengan pengaturan yang bisa disesuaikan dengan kebutuhan. Selanjutnya pilih ISO *installer* yang tersimpan di dalam folder PC, dan isilah kolom nama, username dan juga password instalasi Ubuntu seperti yang ditunjukkan pada **Gambar 2.4.1**.

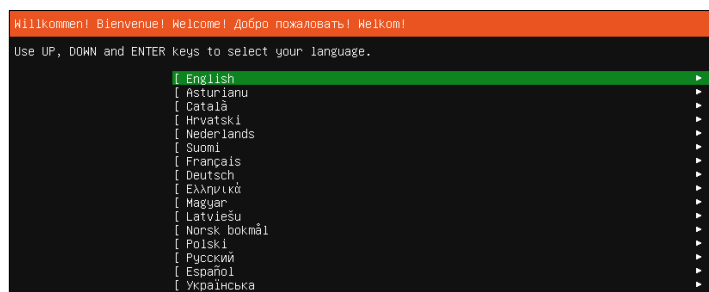
Gambar 2.4.1 Tampilan kolom input informasi *server*

Tentukan lokasi penyimpanan *server* dalam PC. Selanjutnya, tentukan jumlah prosesor, memori yang hendak digunakan pada *server*, tipe jaringan, tipe *controller* I/O, tipe disk dan juga kapasitas disk disesuaikan dengan OS yang digunakan. Setelah semua pengaturan mesin virtual sudah disesuaikan, selanjutnya akan muncul tampilan *wizard* pembuatan VM, dalam hal ini digunakan opsi konfigurasi *Custom* seperti pada **Gambar 2.4.1** berikut ini:



Gambar 2.4.2 Tampilan opsi pemilihan *mode wizard* pembuatan VM

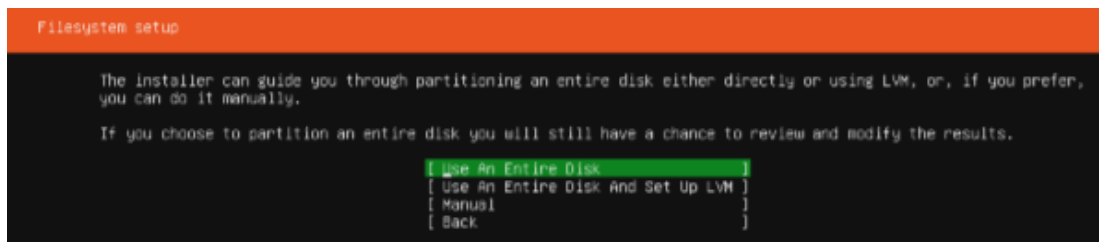
Setelah proses pembuatan virtual mesin pada *desktop* selesai, setelah itu akan muncul tampilan proses *booting* VM. Setelah proses *booting* selesai, berlanjut ke proses pembuatan server baru Ubuntu melalui *software* VMware Workstation. Sebelum instalasi, terlebih dahulu tentukan bahasa instalasi menggunakan bahasa inggris seperti yang ditunjukkan pada **Gambar 2.4.3** berikut:



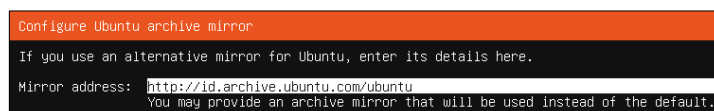
Gambar 2.4.3 Memilih bahasa yang digunakan pada server

Klik Install Ubuntu dengan menekan tombol enter, selanjutnya muncul *network connection* klik next hingga muncul halaman seperti yang tertera pada **Gambar 2.4.4** di

bawah ini. Dan pilih Use an Entire Disk. Klik next hingga muncul tampilan *Default Mirror Address* seperti pada **Gambar 2.4.5**. Sesudah itu, pilih lokasi harddisk tujuan untuk melakukan install Ubuntu *server*.

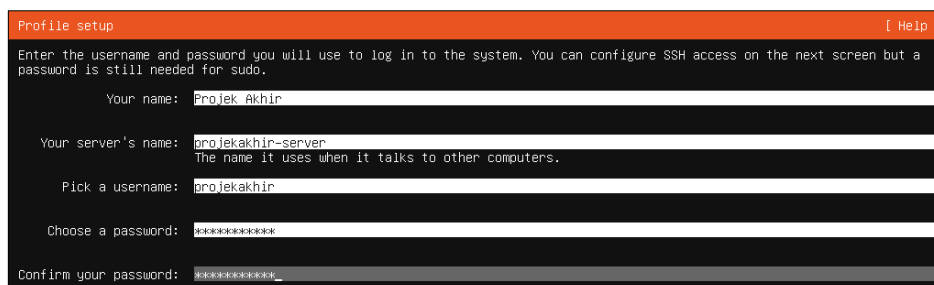


Gambar 2.4.4 Memilih disk *server*



Gambar 2.4.5 *Default Mirror Address*

Tahap selanjutnya yaitu isi dan tentukan nama pengguna *server*, nama *server*, username dan juga password. Username dan password tersebut nantinya akan digunakan untuk *login* ke dalam sistem. Pada proyek akhir ini mulai dari nama pengguna *server*, nama *server*, username dan password diatur menjadi projekakhir seperti yang dapat dilihat pada **Gambar 2.4.6**. Lalu, klik done dan lakukan *booting server* sebelum *server* dapat digunakan oleh *desktop*.



Gambar 2.4.6 Buat username dan password untuk akses server

Pada proses instalasi Ubuntu *server* ini akan dihasilkan IP address virtual pada server LAMPP. Jika sudah muncul tampilan proses instalasi *server* yang telah berhasil beserta IP address yang telah dihasilkan dapat dilihat pada **Gambar 2.4.7**, maka proses instalasi Ubuntu pada *desktop* sudah berhasil.

```

1 Mar 2020 13:54:06 +0000, datasource DataSourceNOCloud (seed=/var/lib/cloud/s
e-net]. Up 44.38 seconds
[ OK ] Started Execute cloud user/final scripts.
[ OK ] Reached target Cloud-init target.
projekakhir
Password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-91-generic x86_64)

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

System information as of Tue Mar 31 13:58:11 UTC 2020

System load:  0.25          Processes:      192
Usage of /:   19.2% of 19.56GB Users logged in:  0
Memory usage: 25%          IP address for ens33: 192.168.190.130
Swap usage:   0%

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See 'man sudo_root' for details.

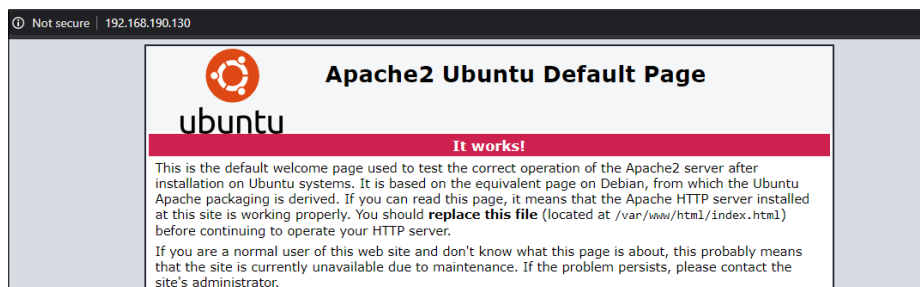
projekakhir@projekakhir-server:~$ _

```


Gambar 2.4.7 Proses instalasi server selesai

Setelah diperoleh IP address melalui proses instalasi server Ubuntu melalui *software* VMware, dilanjutkan menuju tahap selanjutnya yaitu install komponen-komponen yang dibutuhkan selama proses instalasi layanan LAMPP melalui *software* aplikasi puTTY. Sebelum melakukan instalasi komponen menggunakan puTTY, dilakukan koneksi terhadap server yang akan dikerjakan. Tahapannya diawali dengan proses login dengan menggunakan *username* dan *password* yang telah ditentukan pada proses instalasi server Ubuntu pada desktop melalui jendela input login puTTY, kemudian lanjutkan dengan proses instalasi komponen Apache, PHP, MySQL dan juga phpMyAdmin secara bertahap menggunakan parameter dan konfigurasi seperti yang telah dijelaskan pada bagian **Parameter dan Konfigurasi Proyek Pertama** yang menjelaskan secara terperinci mengenai proses instalasi komponen-komponen tersebut.

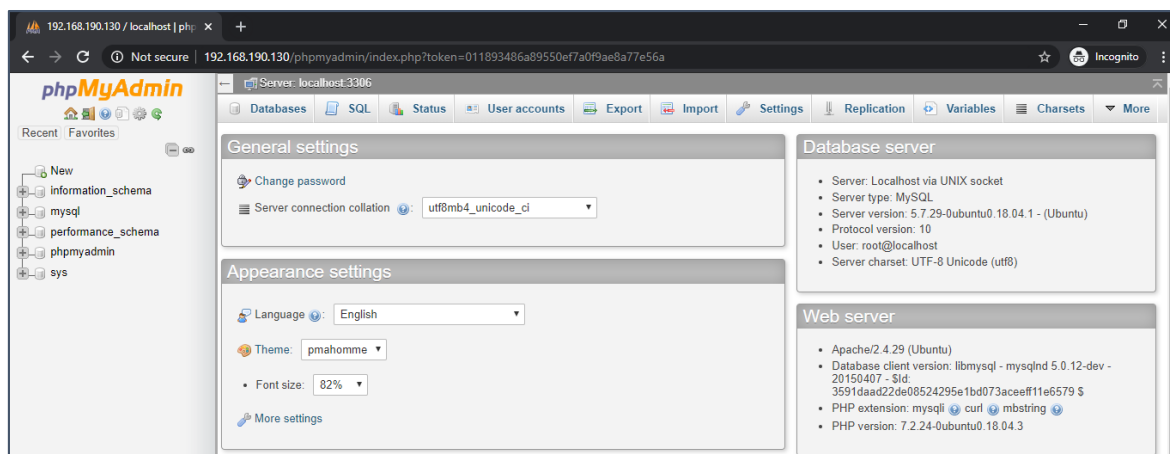
Jika proses instalasi komponen-komponen telah berhasil, maka selanjutnya lakukan uji keberhasilan instalasi komponen-komponen tersebut. Pada Apache dapat dilakukan uji dengan menggunakan IP address pada web browser. Proses akhir tampilan Apache yang berhasil terpasang adalah seperti yang terlihat pada **Gambar 2.4.8**. Tampilan berhasilnya proses instalasi PHP dapat dilihat pada **Gambar 2.4.9**. Sedangkan tampilan keberhasilan proses install phpMyAdmin ditunjukkan dengan tampilan pada **Gambar 2.4.10**.



Gambar 2.4.8 Tampilan hasil proses insatalasi APACHE

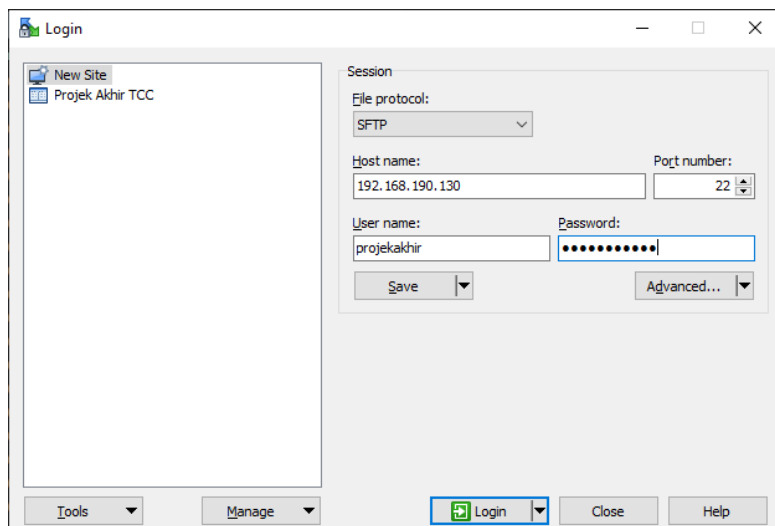
192.168.190.130/info.php	
PHP Version 7.2.24-0ubuntu0.18.04.3	
	
System	Linux projekakhir-server 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64
Build Date	Feb 11 2020 15:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2

Gambar 2.4.9 Tampilan hasil instalasi PHP



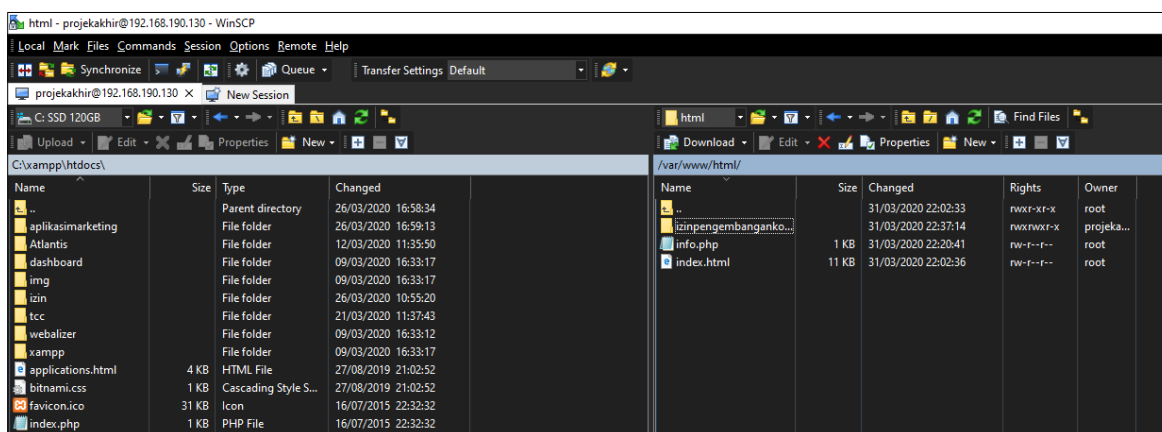
Gambar 2.4.10 Tampilan hasil proses instalasi phpMyAdmin

Server yang telah berhasil dirancang dapat digunakan untuk mengunggah berkas proyek ke *server cloud* dan meng-import database program ke dalam phpMyAdmin. Proses unggah berkas ke dalam *server* digunakan aplikasi WinSCP. Untuk dapat mengakses aplikasi WinSCP diharuskan *login* terlebih dahulu dengan alamat IP *desktop*, username dan password sesuai dengan yang telah ditentukan seperti yang dapat dilihat pada **Gambar 2.4.11** berikut:



Gambar 2.4.11 Login WinSCP

Untuk mengupload berkas kodingan file pada server hosting cloud, dapat dilakukan menggunakan aplikasi WinSCP yang berfungsi untuk transfer file antara windows dengan linux. Sebelum menjalankan aplikasi tersebut, dilakukan login terlebih dahulu dengan username dan password yang sebelumnya telah dibuat. Pengunggahan berkas ke server dengan cara drag and drop pada file yang hendak diupload pada bagian sebelah kiri (merupakan lokasi file yang terdapat di komputer) menuju ke bagian sebelah kanan tempat lokasi server hosting. Lalu klik save. Selanjutnya, buka aplikasi sistem web dengan mengetikkan alamat IP *desktop* pada *web browser*. Tampilan layar upload berkas untuk aplikasi WinSCP setelah berhasil login menggunakan username dan password seperti yang tercantum pada **Gambar 2.4.12** di bawah ini:



Gambar 2.4.12 Tampilan laman *upload* WinSCP setelah *login* berhasil

Setelah tahap *set-up* yang ditunjukkan dari tampilan keempat hasil akhir tahap instalasi telah berhasil dan juga berhasilnya proses unggah file melalui aplikasi WinSCP, *server* program bisa diakses dari *device* lokal menggunakan jaringan internet apapun tanpa

dibatasi oleh penyedia layanan jaringan. Hasil output dari proyek pertama dapat dilihat pada bagian **Hasil Implementasi Proyek Pertama**. Akses web menggunakan alamat IP dan diikuti dengan nama folder program tersebut dan nama file yang hendak diakses. Dalam proyek pertama alamat akses terhadap web program yakni: <http://192.168.190.130/>.

2.4.2 Tahap Implementasi Proyek Kedua

Tahap awal pembuatan rancangan layanan virtualisasi menggunakan layanan *Docker* yaitu dengan mengunduh dan melakukan proses instalasi docker yang akan digunakan dalam pembuatan *dockerfile* melalui aplikasi PuTTY. Pertama-tama lakukan proses menghapus paket instalasi *Docker* terdahulu yang pernah diinstall untuk diganti dengan paket *Docker Engine* dengan versi terbaru.

Selanjutnya, lakukan proses pembaruan paket dengan *update* dan dilanjutkan dengan proses install indeks paket apt, agar apt dapat diizinkan untuk mengakses repositori melalui https, kemudian tekan tombol enter pada keyboard untuk memulai proses instalasi indeks paket. Proses install indeks paket apt dapat dilihat pada **Gambar 2.4.10** di bawah ini:

```
projekakhir@projekakhir-server:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
0 upgraded, 2 newly installed, 0 to remove and 29 not upgraded.
Need to get 6,572 B of archives.
```

Gambar 2.4.10 Parameter instalasi indeks paket apt

Setelah proses instalasi indeks paket apt selesai, diperlukan penambahan *repository* baru pada daftar *repository Ubuntu* dengan menambahkan kunci GPG resmi *Docker* seperti pada **Gambar 2.4.11** berikut. Penambahan GPG *key* bertujuan untuk mengatasi pesan *error* yang disebabkan karena GPG *key* tidak valid dari *repository*.

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

Gambar 2.4.11 Penambahan GPG *key*

Selanjutnya lakukan proses pembaruan daftar aplikasi pada Ubuntu menggunakan parameter seperti yang telah dijelaskan pada bagian **Parameter dan Konfigurasi**. Langkah selanjutnya jika *repository* berhasil ditambahkan dan berhasil dimuat dengan sempurna lakukan kembali proses pembaruan sistem. Selanjutnya, tambahkan repositori apt baru ke dalam direktori dengan menggunakan perintah untuk mengatur *stable repository* seperti pada **Gambar 2.4.12**:

```
projekakhir@projekakhir-server:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:2 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.0 kB]
Hit:3 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 327 kB in 2s (147 kB/s)
Reading package lists... Done
projekakhir@projekakhir-server:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 252 kB in 11s (23.9 kB/s)
Reading package lists... Done
projekakhir@projekakhir-server:~$
```

Gambar 2.4.12 Pengaturan *stable repository*

Kemudian lakukan *update* paket indeks apt, dan install versi terbaru dari *Docker Engine* dan *containerd*. Terakhir pastikan *Docker Engine* dipasang dengan benar pada *desktop* dengan menjalankan 'hello-world image'. Perintah ini akan mengunduh gambar uji dan menjalankannya pada *container*. Ketika *container* berjalan, maka akan mencetak pesan informasi untuk menjalankan pesan 'hello world'. **Gambar 2.4.13** berikut merupakan tampilan akhir dari proses instalasi *Docker* yang telah berhasil dilakukan.

```
projekakhir@projekakhir-server:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdc26d7: Pull complete
Digest: sha256:8e3114318a995a1ee497790535e7b88365222a21771ae7e53687ad76563e8e76
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
projekakhir@projekakhir-server:~$
```

Gambar 2.4.13 Tampilan keberhasilan proses instalasi *Docker*

Setelah proses instalasi *Docker Engine* selesai dilakukan, lanjutkan dengan proses menggunakan *Docker* untuk membuat LAMPP server untuk proyek tugas akhir ini. Gunakan parameter untuk install curl (proses instalasi dilakukan sebab curl belum diinstall pada distribusi Linux). Kemudian unduh komponen *Docker-compose* melalui URL pada github seperti yang dapat dilihat pada bagian **Parameter dan Konfigurasi**. Jalankan perintah `chmod` setelah *Docker-compose* telah diunduh, dan pastikan untuk menguji perintah *Docker-compose* dapat bekerja dan menampilkan hasil uji seperti pada **Gambar 2.4.14** di bawah ini:

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

Gambar 2.4.14 Tampilan hasil uji *Docker-compose*

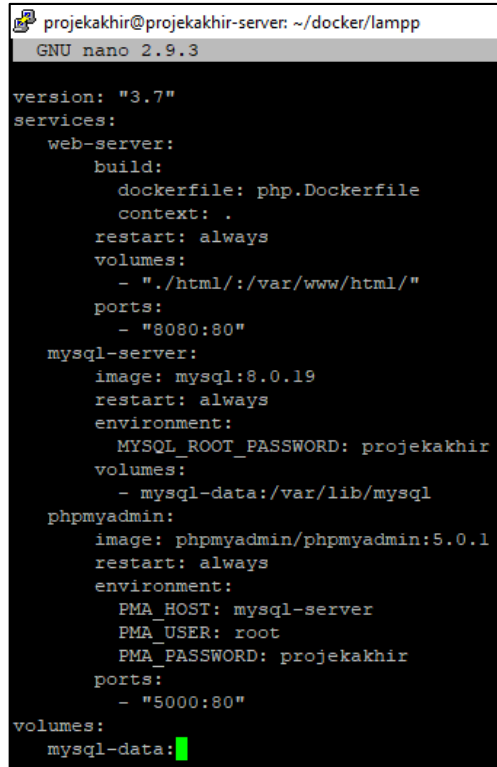
Buat direktori proyek `~/docker/lampp` dan **html** direktori di dalam proyek direktori untuk menyimpan file website (misal: PHP, HTML, dll.). Berikutnya, navigasikan untuk berpindah pada direktori `~/docker/lampp` menggunakan perintah `cd`. Buatlah sebuah file **php.Dockerfile** ke dalam direktori `~/docker/lampp`. Dockerfile ini memungkinkan `mysql` dan `PDO php extensions` pada **php:7.4.3-apache** image dari *Docker Hub* dan membangun kustomisasi *Docker image*. Isi konten file **php.Dockerfile** dapat dilihat pada **Gambar 2.4.15** berikut:

```
projekakhir@projekakhir-server: ~/docker/lampp
GNU nano 2.9.3

FROM php:7.4.3-apache
RUN docker-php-ext-install mysqli pdo pdo_mysql
```

Gambar 2.4.15 Tampilan isi file *php.Dockerfile*

Selanjutnya buat file **docker-compose.yml** di dalam direktori proyek `~/docker/lampp`, dan sesuaikan isi file tersebut sesuai dengan yang tertera pada **Gambar 2.4.16** setelah itu simpan file. Pada **Gambar 2.4.16** terdapat *script* yang digunakan untuk melakukan *port forwarding* dari **default port 80** (port asal aplikasi) ke port lain yang menjadi tujuan (dalam proyek pembuatan *Dockerfile* kali ini digunakan port **8080**—untuk mengakses *server*—dan port **5000**—untuk mengakses *database server*). Setelah dilakukan *port forwarding*, maka port default http akan di mapping ke *port proxy* yaitu 8080. *Port forwarding* dilakukan karena *web server* akan langsung berjalan di port 80. Akan tetapi, karena port ini tidak dapat langsung dibuka di localhost tanpa *port forwarding*.

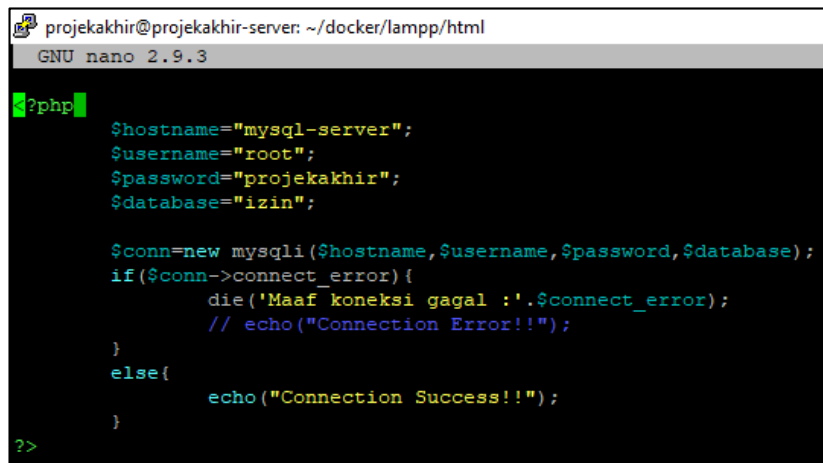


```
projekakhir@projekakhir-server: ~/docker/lampp
GNU nano 2.9.3

version: "3.7"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "./html:/var/www/html/"
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: projekakhir
    volumes:
      - mysql-data:/var/lib/mysql
  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: mysql-server
      PMA_USER: root
      PMA_PASSWORD: projekakhir
    ports:
      - "5000:80"
volumes:
  mysql-data:
```

Gambar 2.4.16 Tampilan isi file docker-compose.yml

Kemudian buat sebuah file bernama **index.php** yang disimpan di dalam direktori **html/** untuk menguji LAMPP server. Isi file tersebut seperti yang dapat dilihat pada **Gambar 2.4.17**. File **index.php** ini digunakan untuk menghubungkan file program web dengan *database server*.



```
projekakhir@projekakhir-server: ~/docker/lampp/html
GNU nano 2.9.3

<?php
$hostname="mysql-server";
$username="root";
$password="projekakhir";
$database="izin";

$conn=new mysqli($hostname,$username,$password,$database);
if($conn->connect_error){
    die('Maaf koneksi gagal :'.$connect_error);
    // echo("Connection Error!!");
}
else{
    echo("Connection Success!!");
}

?>
```

Gambar 2.4.17 Tampilan isi file index.php

Percabangan file direktori proyek **~/docker/lampp** dapat dilihat menggunakan parameter **tree** dan akan terlihat seperti tampilan pada **Gambar 2.4.18**.

```

projekakhir@projekakhir-server: ~/docker/lampp
projekakhir@projekakhir-server:~/docker/lampp$ tree
.
├── docker-compose.yml
├── html
│   └── index.php
└── php.Dockerfile

1 directory, 3 files
projekakhir@projekakhir-server:~/docker/lampp$ sudo docker-compose up -d

```

Gambar 2.4.18 Tampilan percabangan file program

Selanjutnya jalankan layanan **web-server**, **mysql-server** dan **phpmyadmin**. Semua layanan akan berjalan di *background*. Selanjutnya periksa pemetaan port pada *Docker* seperti pada **Gambar 2.4.19**. Seperti yang dapat dilihat, untuk layanan **web-server**, *Docker host* port-nya adalah **8080** yang memetakan ke *container* TCP port **80**. Untuk layanan **phpmyadmin**, *Docker host* port-nya yaitu **5000** yang memetakan ke *container* TCP port **80**. Jika hendak mengakses ke layanan LAMPP dari komputer lain dalam jaringan yang sama, harus mengetahui IP address pada *Docker host*.

```

Digest: sha256:d2191935bda8c825bcf51d1c659efcd4d80a4f5da907ea72c8517c8144136de4
Status: Downloaded newer image for phpmyadmin/phpmyadmin:5.0.1
Creating lampp_web-server_1 ... done
Creating lampp_phpmyadmin_1 ... done
Creating lampp_mysql-server_1 ... done
projekakhir@projekakhir-server:~/docker/lampp$ sudo docker-compose ps

```

Name	Command	State	Ports
lampp_mysql-server_1	docker-entrypoint.sh mysqld	Up	3306/tcp, 33060/tcp
lampp_phpmyadmin_1	/docker-entrypoint.sh apac ...	Up	0.0.0.0:5000->80/tcp
lampp_web-server_1	docker-php-entrypoint apac ...	Up	0.0.0.0:8080->80/tcp

```

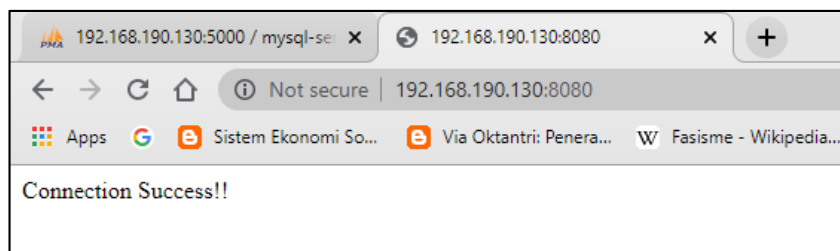
projekakhir@projekakhir-server:~/docker/lampp$

```

Gambar 2.4.19 Tampilan pemetaan layanan *Docker* yang berjalan

Selanjutnya lakukan proses uji dengan jalankan program melalui *web browser* menggunakan alamat IP yang sudah diketahui pada saat proses pengaturan layanan LAMPP pada *Docker*. Untuk mengakses phpMyAdmin, buka web browser dan kunjungi <http://localhost:5000> dari *Docker host* atau kunjungi <http://192.168.190.130:5000> dari komputer yang terhubung dengan jaringan internet. Tampilan laman phpMyAdmin yang dijalankan menggunakan *Docker host* dapat dilihat pada **Hasil Implementasi Proyek Kedua**.

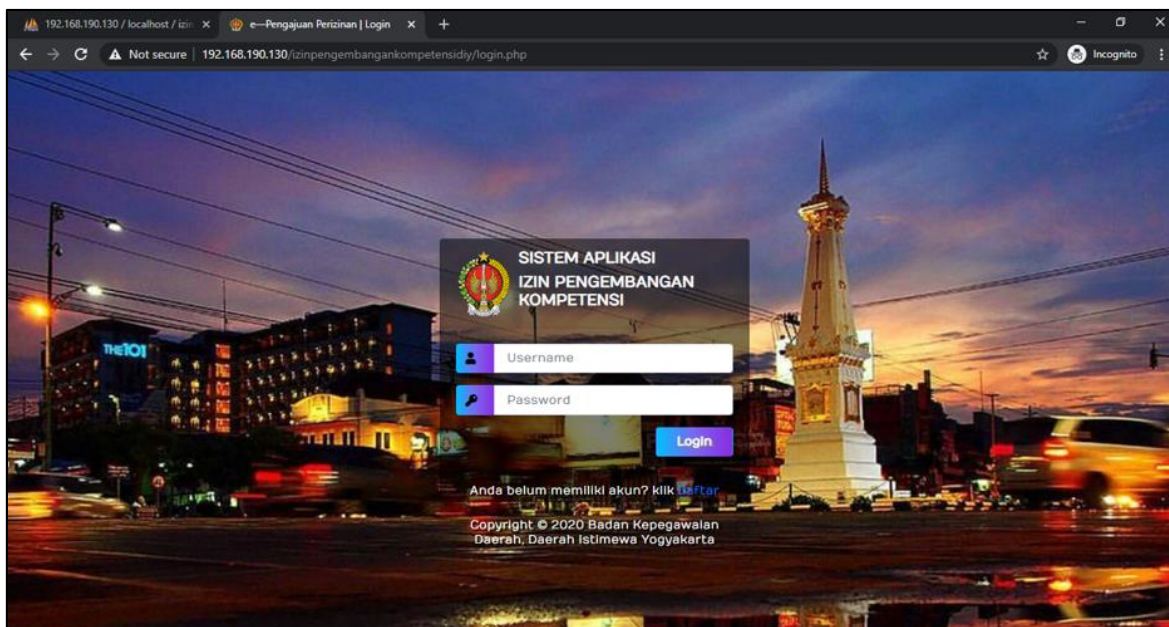
Untuk dapat mengakses file proyek dapat kunjungi <http://localhost:8080> dari *Docker host* atau kunjungi <http://192.168.190.130:8080> dari komputer yang terhubung dengan jaringan internet untuk mengakses layanan web. Jika terdapat pesan alert “**Connection Success!!**” yang menyatakan bahwa file PHP dapat berjalan dengan lancar, seperti yang terdapat pada **Gambar 2.4.20**. Artinya, PHP berjalan dan layanan database MySQL dapat diakses dari *web-server container*. Secara teknis, layanan LAMPP dapat berfungsi.



Gambar 2.4.20 Tampilan keberhasilan akses PHP dari *container*

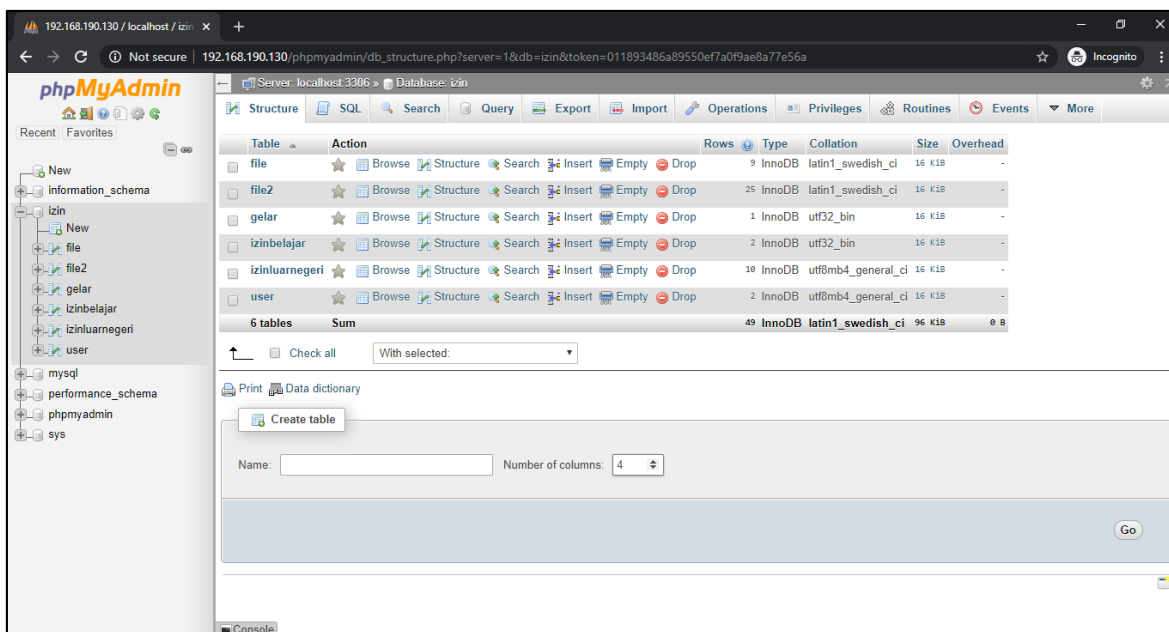
2.5 Hasil Implementasi

Hasil implementasi pembuatan rancangan layanan *cloud computing* menggunakan basis layanan SaaS melalui aplikasi VMware Workstation 15 dan sistem operasi Windows telah mencapai tahap akhir pengunggahan berkas ke dalam server Ubuntu LAMPP. File yang telah diupload dapat dibuka menggunakan IP Address sesuai dengan PC yang digunakan sebagai sarana hosting *cloud*. Program juga dapat dijalankan menggunakan jaringan internet apapun dengan lancar seperti saat ketika mengakses web sebelum dilakukan hosting *cloud*. Setelah pengguna mengetik URL alamat IP *desktop* pada *address bar browser* dan menekan tombol Enter, pengguna akan diperlihatkan sebuah tampilan *form login* dari program virtualisasi yang diakses melalui *server cloud* seperti dapat dilihat pada **Gambar 2.5.1** di bawah ini.



Gambar 2.5.1 Tampilan laman *login* dari server Ubuntu

Untuk tampilan database phpMyAdmin program yang diakses dari server Ubuntu seperti yang terlihat pada **Gambar 2.5.2** di bawah ini:

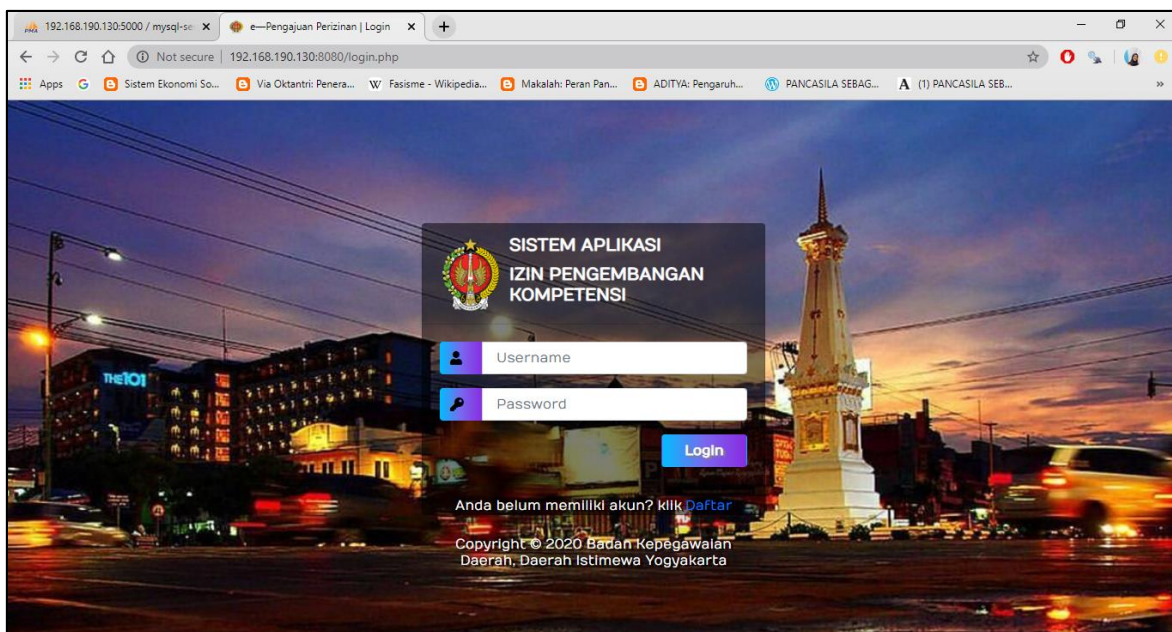


Gambar 2.5.2 Tampilan *database* dari server Ubuntu

Parameter dan konfigurasi yang diperlukan pada proses instalasi pembuatan layanan SaaS berdasarkan Apache, PHP, MySQL dan phpMyAdmin—yang termasuk komponen penting dalam OS Ubuntu LAMPP—telah terpasang dan diatur dengan benar. Sehingga pada saat proses instalasi berjalan hanya menemui sedikit kendala yang dapat terselesaikan permasalahannya dengan baik sehingga web program dapat diakses melalui *web browser*. Masing-masing parameter dalam komponen instalasi Apache, PHP, MySQL dan phpMyAdmin disesuaikan dengan kebutuhan sistem.

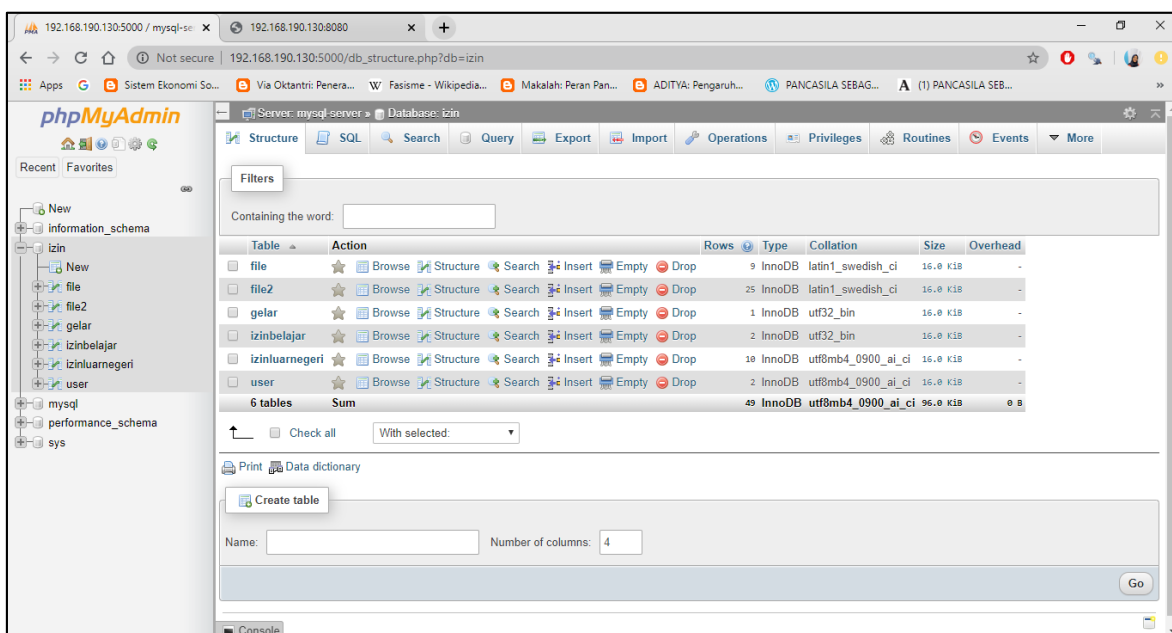
Hasil implementasi untuk Proyek Kedua yakni Pembuatan *Dockerfile* dari Proyek Pertama memiliki hasil tampilan yang serupa dengan Proyek Pertama, karena menggunakan file program yang sama. Yang **membedakan** antara Proyek Pertama dengan Proyek Kedua adalah di bagian **URL pada address bar browser** yang digunakan untuk mengakses program melalui *web browser*. Pada Proyek Pertama terlihat tampilan URL program setelah *web* berhasil diakses yakni <http://192.168.190.130/izinpengembangankompetensidiy/login.php>.

Sementara pada Proyek Kedua, tampilan URL program setelah *web* berhasil diakses adalah <http://192.168.190.130:8080/login.php>. Seperti yang terlihat pada **Gambar 2.5.3** yang terlampir berikut:



Gambar 2.5.3 Tampilan laman *login* dari server *Docker*

Tampilan pada laman phpMyAdmin program yang dijalankan menggunakan *Docker* *host* akan terlihat seperti pada **Gambar 2.4.20** berikut:



Gambar 2.5.4 Tampilan *database* dari server *Docker*

Tampilan *database* phpMyAdmin antara proyek pertama yang menggunakan server Ubuntu dan proyek kedua yang menggunakan *Docker*, juga terlihat perbedaan di bagian **URL pada address bar browser** yang digunakan untuk mengakses *database* program melalui *web browser*. Pada proyek pertama tampilan URL *database* setelah diakses yakni http://192.168.190.130/phpmyadmin/db_structure.php?server=1&db=izin&token=011893

486a89550ef7a0f9ae8a77e56a, sedangkan pada URL *database* proyek kedua yaitu http://192.168.190.130:5000/db_strucuture.php?db=izin.

Yang membedakan pada penulisan URL dari kedua program adalah akses *server* melalui *web browser*. Pada proyek pertama *web server* dapat secara otomatis mengakses program menggunakan **default port 80**, sebab itu hanya perlu dituliskan alamat IP *desktop* untuk dapat mengakses program tersebut. Sedangkan pada proyek kedua, karena langsung menggunakan perintah **docker run**, *web server* akan langsung berjalan di port 80. Akan tetapi, karena port ini tidak dapat langsung dibuka di localhost. Untuk dapat membukanya perlu dilakukan **port forwarding** dari **default port 80** (port asal aplikasi) ke port lain yang menjadi tujuan (dalam proyek pembuatan *Dockerfile* kali ini digunakan port **8080**—untuk mengakses *server*—dan port **5000**—untuk mengakses *database server*). Setelah dilakukan *port forwarding*, maka port default http akan di mapping ke *port proxy* yaitu 8080. Hal ini yang menjadikan penulisan URL menjadi berbeda, penulisan URL pada *Docker* setelah memanggil file yang terdapat pada alamat IP *desktop*, ia akan memanggil **port 8080** (untuk akses *web server*) atau **port 5000** (untuk akses *database server*) yang akan diakses pada *Host Container*.

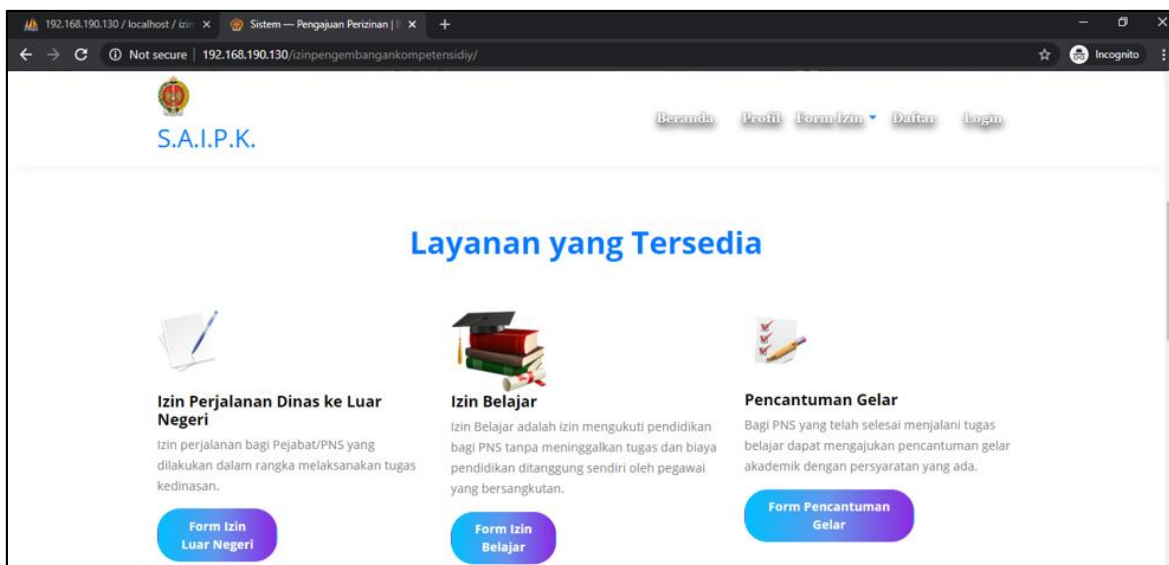
2.6 Pengujian Singkat

Pengujian sistem pada saat *web* program yang terdapat pada virtualisasi *server* Ubuntu dan *Docker* akan menampilkan *form login* pegawai. Setiap pegawai memiliki username dan password sendiri untuk login ke dalam sistem. Username masing-masing pegawai berbeda satu sama lain sesuai dengan NIK / NIP para pegawai, dan password akun pengguna dapat ditentukan secara bebas oleh pemilik akun melalui *form* registrasi. Pegawai diwajibkan untuk mengisi *form login* terlebih dahulu untuk dapat masuk dan mengakses sistem. Karena keseluruhan layanan yang terdapat dalam sistem tidak dapat diakses jika pengguna tidak memiliki akun. Bagi pengguna yang bukan admin, setelah melalui proses login berhasil pengguna akan dialihkan menuju tampilan beranda web seperti yang dapat dilihat pada **Gambar 2.6.1**.



Gambar 2.6.1 Tampilan laman *header* beranda setelah login berhasil

Pada bagian **Layanan yang Tersedia** terdapat tiga jenis layanan yang dapat digunakan oleh pengguna. Layanan pertama yaitu “Layanan Izin Perjalanan Dinas ke Luar Negeri”, “Layanan Izin Belajar”. Dan layanan terakhir yakni “Layanan Pencantuman Gelar”, layanan pengajuan Pencantuman Gelar ini tidak dapat diakses oleh pengguna dan hanya dapat diakses oleh admin pengelola sistem. Layanan Pencantuman Gelar hanya bisa dilakukan input oleh admin melalui data **Izin Belajar** yang telah ditempuh oleh pengguna. Pencantuman gelar dapat diajukan pengguna melalui admin dengan syarat pengguna harus menempuh / mengikuti Izin Belajar terlebih dahulu. Jika pengguna belum melakukan Izin belajar, maka pengguna tidak dapat mengajukan pencantuman gelar kepada admin. Jenis layanan tersebut dapat dilihat pada **Gambar 2.6.2** berikut:



Gambar 2.6.2 Daftar layanan yang tersedia pada sistem



Gambar 2.6.3 Fitur panduan layanan pada sistem

Dashboard admin memiliki 9 buah *button* seperti yang dapat dilihat pada **Gambar 2.6.4** di bawah ini. Masing-masing *button* jika dipilih akan menampilkan data-data hasil input yang dikelompokkan sesuai dengan jenis layanannya. Hanya admin yang bisa melakukan proses *editing* dan *delete* pada keseluruhan data. Proses **edit** dan **hapus** data pada **Ubuntu server** dan **Docker container** dapat berjalan lancar dengan data yang terdapat di dalam database ikut berubah pula. Admin juga dapat mengunduh dokumen surat dalam format pdf. Untuk melengkapi laporan, terdapat tombol berwarna hijau untuk mengunduh data tabel secara keseluruhan yang dalam format xlsx untuk dijadikan data laporan izin yang telah diproses.

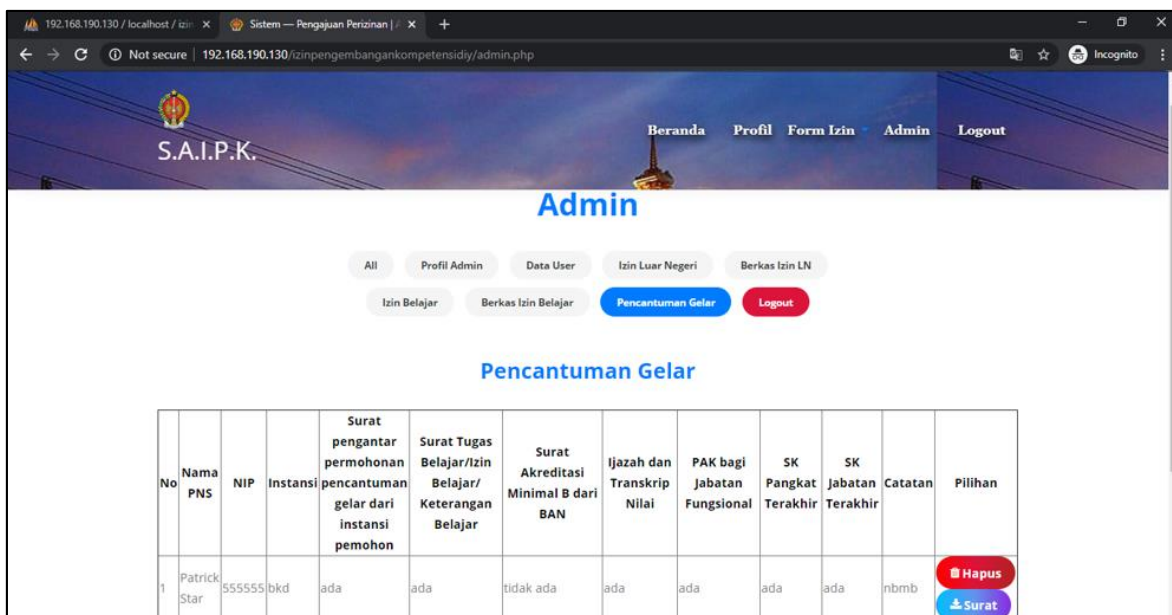
No	Nama Pejabat	NIP / jabatan	Negara Tujuan	Tujuan	Tanggal Berangkat	Tanggal Kepulangan	Keperluan Izin / Lainnya	Sumber Dana / Lainnya	Action
1	ROSE SUTIKNO, SH, MM	19630222 199203 1 006 / Kabid Industri Pariwisata DIY	Vietnam		2020-01-05	0000-00-00	/ -	/ -	Edit Hapus Surat
	WIWIK	19731015 199802 2 003 / Kepala	Beberapa						Edit Hapus

Gambar 2.6.4 Data Izin Luar Negeri pada Admin

Pada tampilan data Izin Belajar terdapat sebuah tombol berwarna hitam yang digunakan admin untuk melakukan input Pencantuman Gelar. Layanan Pencantuman Gelar hanya bisa diakses oleh admin pengelola sistem. Jika pengguna hendak meminta pengajuan pencantuman gelar, diharap untuk menghubungi admin dengan syarat pengguna telah menyelesaikan Izin Belajar terlebih dahulu. Karena pencantuman gelar hanya bisa dilakukan setelah menempuh dan menyelesaikan izin belajar. Tampilan data Izin Belajar seperti yang seperti pada **Gambar 2.6.5**, dan data untuk Pencantuman Gelar seperti pada **Gambar 2.6.6**.

No	Nama PNS	NIP / jabatan	Asal Surat	Tanggal Surat	Pangkat / Golongan	Instansi	Fakultas	Program Studi	Universitas	Pilihan	Pencantuman Gelar
1	Patrick Star	123456789 / bos	tegal	2020-01-07	5	BKD	Teknik Industri	Teknik Informatika	UPN	Hapus Edit Surat	Pencantuman Gelar
2	Adminila Prilla	1111111 / Guru Madya Dinas Pendidikan Kota	Semarang	2020-02-02	5	BKD	Teknik Industri	Sistem Informasi	UPN	Hapus Edit	Pencantuman Gelar

Gambar 2.6.5 Data Izin Belajar pada Admin

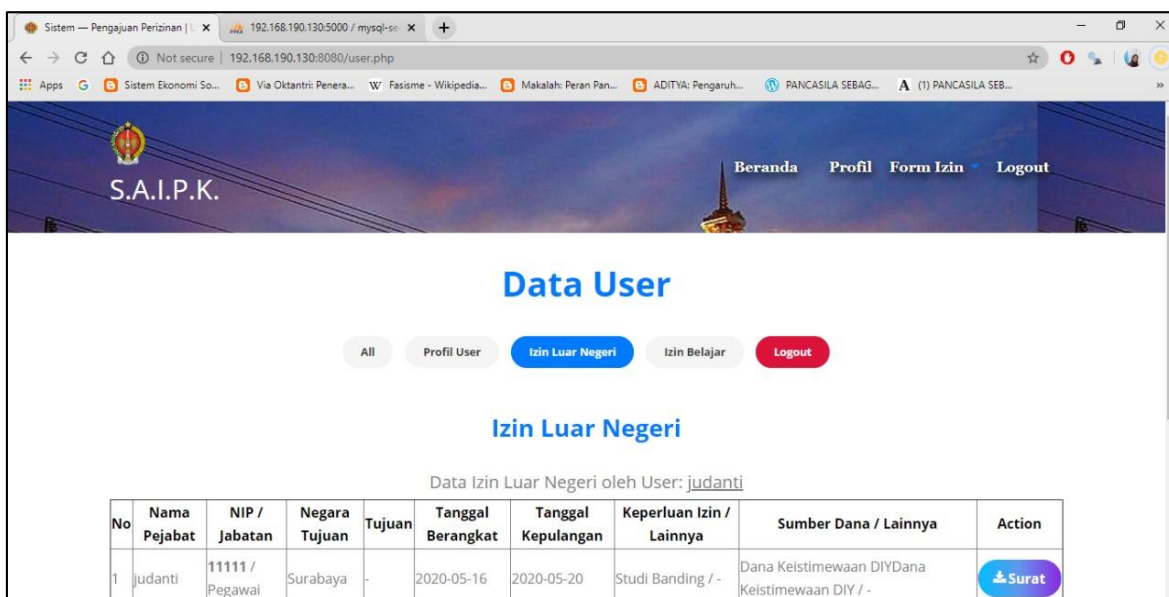


Gambar 2.6.6 Data Pencantuman Gelar pada Admin

Bagi pengguna bukan admin, akses terhadap sistem dapat login menggunakan akun yang telah dibuat saat proses registrasi. **Gambar 2.6.7** di bawah merupakan tampilan form input Izin Luar Negeri yang diakses melalui *Docker*. Di bagian atas form tepatnya di bawah judul *layout* terdapat pesan *alert* yang menyatakan bahwa *submit* data beserta berkas Izin Perjalanan Dinas ke Luar Negeri hanya dapat dilakukan selambat-lambatnya 17 hari sebelum jadwal keberangkatan peserta dinas. Hal ini karena dalam *script* program telah diatur sedemikian rupa agar data hanya dapat *submit* dalam jangka waktu sesuai dengan yang tertera pada pesan tersebut. Jika data yang diinput terlambat dari waktu yang telah ditentukan, maka sistem menolak input data tersebut secara otomatis.

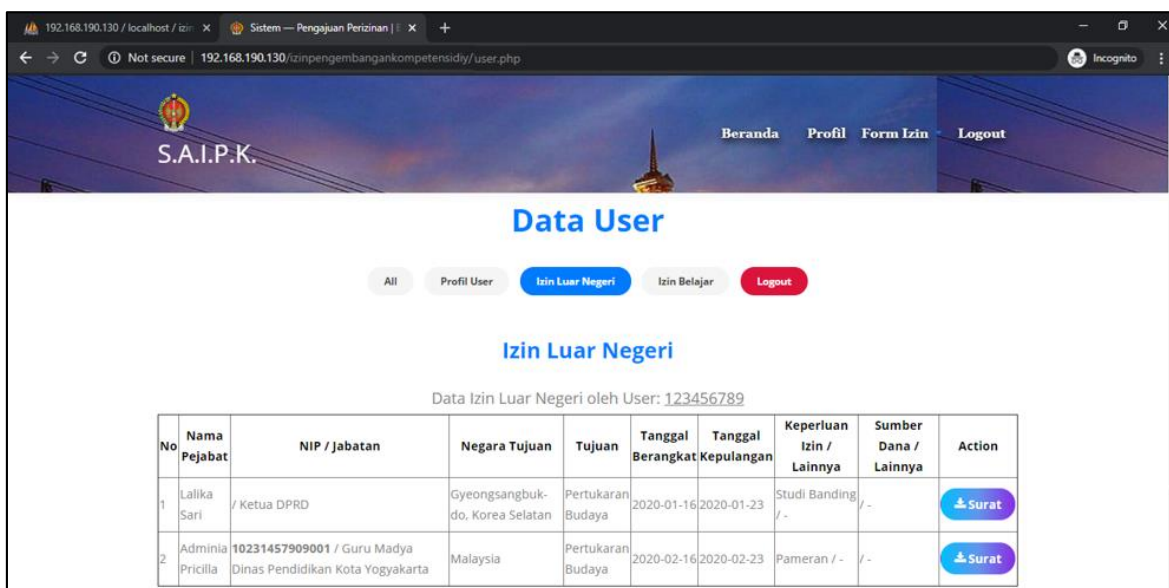
Gambar 2.6.7 Tampilan *Form Izin Luar Negeri* melalui *Docker*

Setelah data input Izin Luar Negeri tercatat dalam *database* admin, pengguna akan dibawa menuju tampilan data user. Dalam data *user* terdapat 5 buah *button* masing-masing menampilkan data-data yang dikelompokkan sesuai dengan jenis layanannya. **Gambar 2.6.8** di bawah merupakan contoh tampilan menu user setelah data berhasil diinput pada form Izin Luar Negeri oleh pengguna yang diakses melalui *Docker container*. Sebagaimana terlihat penggunaan port pada akses URL *address bar*.



Gambar 2.6.8 Tampilan Data Izin luar Negeri melalui *Docker*

Sedangkan, pada **Gambar 2.6.9** menampilkan data hasil pengujian input milik pengguna dengan NIP 123456789 yang diakses melalui Ubuntu.



Gambar 2.6.9 Tampilan Data Izin Luar Negeri melalui VM

Berbeda dengan admin yang terdapat tombol untuk edit dan hapus data, di akhir tabel hanya terdapat tombol untuk mengunduh surat. Pengguna tidak memiliki wewenang untuk

merubah dan menghapus data yang telah diinput. Jika terdapat kesalahan dalam data dan ingin dilakukan perubahan pada data yang dimaksud, hendaknya menghubungi admin. Untuk itu data yang diinput harus sudah benar dengan dilakukan cek ulang. Karena database kepegawaian yang dikelola oleh admin bersifat rahasia dan hanya dapat diakses oleh admin pengelola.

Hasil pengujian yang dilakukan terhadap alur jalannya program sesuai dan tidak mengalami kendala dalam menjalankan sistem. Percobaan input data pengajuan izin sudah berhasil dilakukan baik melalui *server Ubuntu* yang dibuktikan dengan keberhasilan pengujian input data ke dalam sistem yang terdapat pada **Gambar 2.6.4** dan melalui *Docker* yang dibuktikan dengan tampilan halaman user setelah keberhasilan proses input data ke dalam sistem yang terdapat pada **Gambar 2.6.8**, dengan munculnya pesan alert data berhasil diinputkan ke dalam database sistem. Proses edit dan hapus data yang dilakukan oleh admin juga berhasil merubah dan menghapus data dalam *database* sistem. Hal ini membuktikan bahwa sistem dapat berjalan dengan baik. Waktu yang diperlukan bagi *browser* untuk memuat program lebih cepat dibandingkan menggunakan *hosting* lokal.

BAB III

JADWAL Pengerjaan dan Pembagian Tugas

3.1 Agenda Pengerjaan

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

Tabel 3.1 Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Perancangan dan Pengembangan LAMPP								
3.	Penyusunan Laporan Project								
4.	Revisi Laporan Project								
5.	Perancangan dan Pengembangan Docker								
6.	Melanjutkan Penyusunan Laporan								
7.	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 (LAMPP)	Rizkyana
2.	Pengembangan Arsitektur Cloud Computing (LAMPP)	Judanti
3.	Pengujian LAMPP	Judanti
4.	Perancangan Arsitektur Dockerfile	Rizkyana
5.	Pengembangan Arsitektur Dockerfile	Judanti
6.	Pengujian Dockerfile	Judanti
7.	Perbaikan kode program sistem yang diupload di LAMPP dan Docker	Judanti, Rizkyana
8.	Penyusunan Laporan	Rizkyana

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan permasalahan yang sudah dipaparkan pada bagian **Latar Belakang** pembuatan layanan *cloud computing* berbasis SaaS menggunakan *server* Ubuntu dan *Docker* sesuai harapan telah memenuhi *point* yang dirinci pada bagian **Tujuan Proyek Akhir**. Program dapat dibuka melalui *web browser* menggunakan jaringan internet lokal dengan lancar seperti saat ketika mengakses web menggunakan hosting lokal.

Parameter dan konfigurasi yang diperlukan pada proses instalasi pembuatan layanan Ubuntu LAMPP dan *Docker* telah terpasang dan diatur dengan benar. Sehingga pada saat proses berjalan hanya menemui sedikit kendala yang akhirnya dapat terselesaikan permasalahannya dengan baik sehingga *web* program dapat diakses dengan lancar.

Berdasarkan pengujian, sistem yang dijalankan dengan virtualisasi *cloud* membuahkan hasil perancangan layanan *cloud computing* yang diterapkan dalam sistem web pengajuan Izin Pengembangan Kompetensi dapat berjalan sesuai kebutuhan yang diinginkan. Para ASN yang hendak melakukan pengajuan izin dapat mengajukan permohonan tanpa terpusat pada admin dalam melakukan input data ke dalam *database* sistem. Sehingga tidak menghabiskan banyak waktu dalam melakukan proses input data berkat sistem yang terintegrasi dengan layanan *cloud* memungkinkan kemudahan dan kecepatan akses web dari *server* melalui berbagai *platform* yang digunakan.

Selain memudahkan sistem dalam memuat laman web melalui *server* utama menggunakan jaringan internet apapun tanpa dibatasi oleh penyedia layanan jaringan, dengan dibuatnya rancangan virtualisasi berbasis *cloud*, sistem aplikasi tersebut dapat mengurangi beban kerja *storage* (penyimpanan) dalam melakukan proses komputasi, agar proses yang bekerja di dalam sistem menjadi lebih cepat sehingga setiap perubahan data yang tersimpan dari sisi *server* maupun *client* akan dapat langsung tersimpan pada seluruh sistem penyimpanan *cloud*.

Berdasarkan pembagian tugas proyek pada Bab III, hasilnya sudah cukup baik. Dengan pembagian pengerjaan tugas dan manajemen waktu pengerjaan yang telah ditentukan sedemikian rupa, proyek dapat diselesaikan dengan tepat waktu dan tidak

mengalami banyak kendala ketika proses pengoperasian server. Hal ini didukung juga oleh proses pengerjaan—mulai dari perancangan arsitektur virtualisasi, pembuatan *server cloud* hingga pada tahap implementasi dan pengujian proyek—sesuai prosedur yang terdapat pada referensi.

4.2 Saran

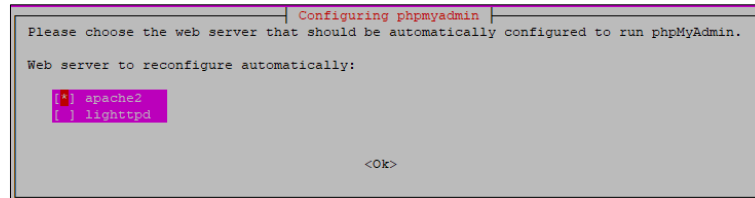
Saran yang dapat diberikan oleh penulis untuk proyek akhir ini adalah sistem aplikasi web ini agar dapat dimanfaatkan lebih maksimal lagi ke depannya dan harapan dengan adanya pengembangan di kemudian hari agar lebih baik lagi dari sistem yang telah dibuat saat ini. Serta penyempurnaan fitur layanan yang ada agar lebih ditingkatkan lagi supaya baik admin maupun pengguna merasa lebih nyaman dalam pemakaian aplikasi.

DAFTAR PUSTAKA

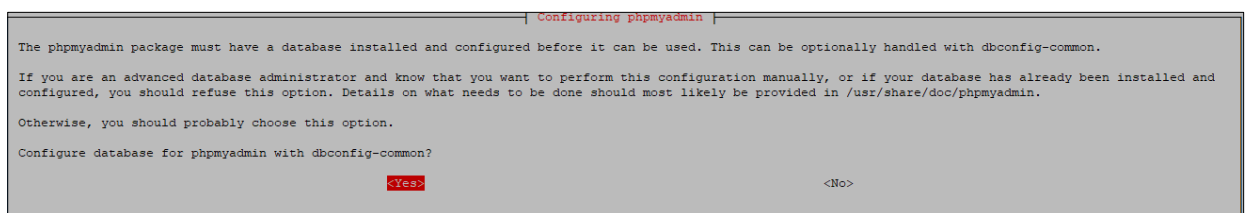
- [1] Kurniawan, Hendri (2017). Pengertian dan Manfaat *Cloud Computing* yang Wajib Anda Ketahui. Diambil 26 Maret 2020, dari: <https://www.indoworx.com/pengertian-dan-manfaat-cloud-computing-yang-wajib-anda-ketahui/amp/>
- [2] Anggi, Fersisilia (2016). Pengertian, Manfaat, Cara Kerja dan Contoh *Cloud Computing*. Diambil 26 Maret 2020, dari: <http://pusatteknologi.com/pengertian-manfaat-cara-kerja-dan-contoh-cloud-computing.html>
- [3] Wikipedia (2019). Google Drive. Diambil 9 Mei 2020, dari: https://id.wikipedia.org/wiki/Google_Drive
- [4] HUMAS MENPANRB (2019). Pengembangan Kompetensi Harus Kompak dengan Kebutuhan Organisasi. Diambil 9 Mei 2020, dari: <https://www.menpan.go.id/site/berita-terkini/pengembangan-kompetensi-harus-kompak-dengan-kebutuhan-organisasi>
- [5] Ramadhan, Dwi Rendra (2017). Peran Virtualisasi pada *Cloud Computing*. Diambil 26 Maret 2020, dari: <http://dwireramdhan.blogspot.com/2017/04/peran-virtualisasi-pada-cloud-computing.html?m=1>
- [6] Maulana, Halim. (2016, September). Analisis dan Perancangan Sistem Replikasi Database MySQL dengan Menggunakan VMware pada Sistem Operasi *Open Source*, Jurnal Nasional Informatika dan Teknologi Jaringan, No. 1 Vol. I. 60-80.
- [7] Shovan, Shahriar. (2020). Set Up a LAMPP Server with Docker. Diambil 28 April 2020, dari https://linuxhint.com/lamp_server_docker/

LAMPIRAN

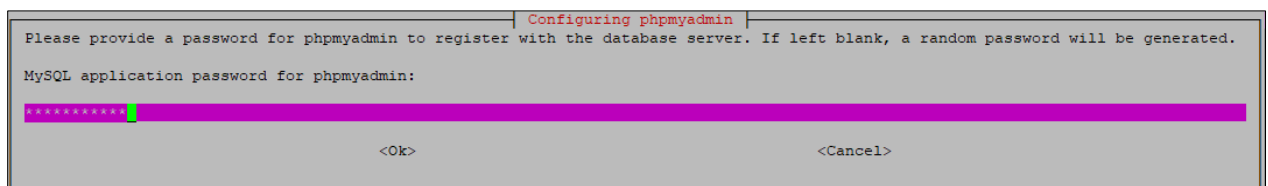
Tampilan proses instalasi phpMyAdmin ditunjukkan dengan beberapa screenshot pada **Gambar 2.13** hingga **Gambar 2.15**.



Gambar 2.13 Memilih web server untuk menjalankan PHPMYAdmin



Gambar 2.14 Persetujuan konfigurasi database PHPMYAdmin



Gambar 2.15 Input *password*