

577

6 m>U i UbXU'DUfUa UUh\$*"+",z\$)#)#&\$&\$

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING LAPORAN PROYEK AKHIR

SISTEM MONITORING ACCOUNT MANAGER OS UBUNTU MENGGUNAKAN LAMPP DAN PROSES PEMBUATAN DOCKER FILE-NYA



DISUSUN OLEH:

NAMA ANGGOTA : RAHMAT ZUMARLI 123170011
FHREZHA ZEANETH 123170044
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

F9J=9K98

6 mIK U mi '5 ¶'Bi [fc\ c 'Uh%&.' %%&z\$)#)#&\$&\$

HALAMAN PENGESAHAN

SISTEM MONITORING ACCOUNT MANAGER OS UBUNTU MENGGUNAKAN LAMPP DAN PROSES PEMBUATAN DOCKER FILE-NYA

Disusun oleh :

Rahmat Zumarli

123170011

Fhrezha Zeaneth

123170044

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 syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul Sistem Monitoring Account Manager OS Ubuntu Menggunakan LAMPP dan Proses Pembuatan Docker filenya. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil proyek Kerja Praktek yang kami lakukan.

Sekilas mengenai proyek Kerja Praktek yang kami pilih merupakan proyek dari suatu divisi pada perusahaan PT. Gamatechno Indonesia. Dalam satu Divisi Marketing, Super Visi membutuhkan suatu sistem dimana dapat memantau kinerja dari staff mereka ,yaitu Account Manager (AM). Pemantauan ditujukan terhadap aktifitas dan perkembangan proyek setiap AM. Hak akses Super Visi dapat memantau seluruh data AM dan Customer, sedangkan AM hanya dapat mengakses akun mereka masing-masing. Proses CRUD akun AM dibawah kendali Super Visi, sedangkan proses CRUD *activity* dan *project* dikendalikan oleh masing-masing AM.

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

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

Yogyakarta, 15 April 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Proyek Akhir	1
1.2. Tujuan Proyek Akhir	6
1.3. Manfaat Proyek Akhir	6
1.4. Tahap Penyelesaian Proyek Akhir	7
BAB II ISI DAN PEMBAHASAN	8
2.1. Komponen yang Digunakan	8
2.2. Rancangan Arsitektur <i>Cloud Computing</i>	10
2.3. Parameter dan Konfigurasi	13
2.4. Tahap Implementasi	14
2.5. Hasil Implementasi	14
2.6. Pengujian Singkat	15
BAB III JADWAL Pengerjaan dan Pembagian Tugas	16
3.1. Agenda Pengerjaan	16
3.2. Keterangan Pembagian Tugas	16
BAB IV KESIMPULAN DAN SARAN	17
4.1. Kesimpulan	17
4.2. Saran	17
DAFTAR PUSTAKA	18
LAMPIRAN	19

(gunakan tombol update table (klik kanan daftar isi) lalu format ulang dengan format TNR 12, spacing row single, spacing before after 0 dan indentasi kanan 0, indentasi kiri biarkan)

BAB I

PENDAHULUAN

TANPA SPASI

1.1. Latar Belakang Proyek Akhir

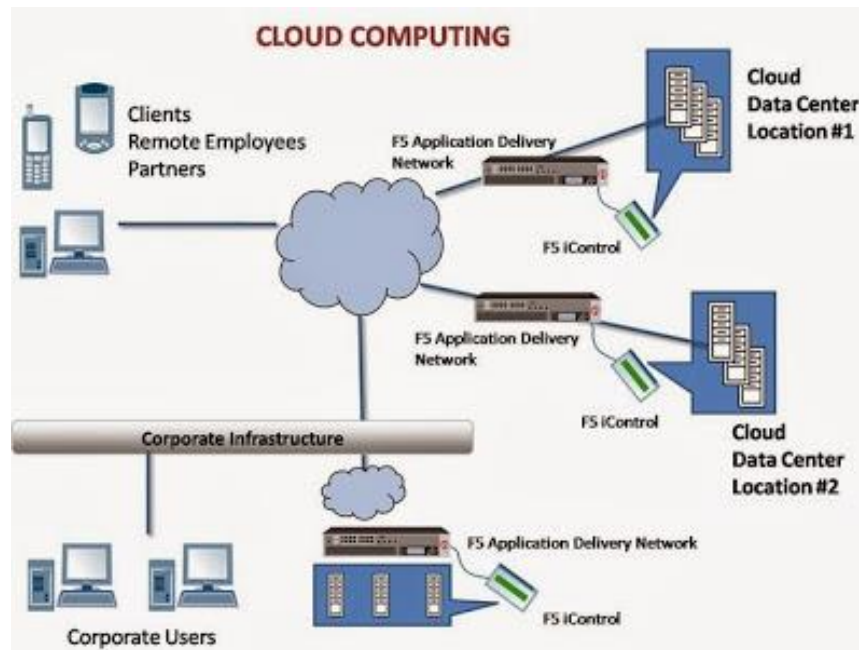
Cloud computing bermula pada tahun 1960-an oleh John McCarthy seorang pakar komputasi MIT dan salah satu pionir **intelligent** buatan. Beliau menyampaikan sebuah visi mengenai komputasi bahwa “suatu hari nanti, komputasi akan menjadi infrastruktur publik seperti listrik dan telepon”. Namun di tahun 1995 Larry Ellison seorang pendiri Oracle menciptakan sebuah ide yang disebut “**Network Computing**” dimana merupakan ajang **kampaye** untuk menggugat dominasi Microsoft yang menjadi raja dekstop **computing** dengan Windows 95- miliknya. Larry menawarkan ide dengan prinsip **user** tidak perlu banyak menggunakan **software**, mulai dari OS dan **software** lain akan dimasukkan dalam PC Dekstop **user**. PC Dekstop dapat digantikan oleh terminal yang terhubung dengan **server** penyedia **enviromtment** dan berisi berbagai kebutuhan **software** siap pakai. Ide dari Larry Ellison sempat menjadi sorotan publik bersamaan dengan munculnya beberapa pabrikan seperti Sun Microsystem dan Novell Netware yang menawarkan hal serupa yaitu **Network Computing client**. Namun karena kualitas jaringan komputer yang belum memadai dan membuat akses **Network Computing** menjadi sangat lambat maka **user** kembali menggunakan PC dan **Network Computing** hilang dengan perlahan. Tahun 2000, Amazon memiliki peran penting dalam pengembangan **cloud computing** dengan melakukan modernisasi pusat data seperti jaringan komputer dengan 10% kapasitas mereka pada satu waktu. Setelah menemukan arsitektur **cloud**, perkembangan mengalami peningkatan efisiensi internal, kemudian Amazon mengembangkan produk baru sebagai penyedia **cloud computing** untuk pelanggan eksternal dengan peluncuran Amazon Web Service (AWS) tahun 2006.

Cloud computing merupakan teknologi yang menjadikan internet sebagai pusat *server* untuk mengelola data dan juga aplikasi pengguna dimana mengizinkan para pengguna untuk menjalankan program tanpa instalasi dan mengizinkan pengguna untuk mengakses data pribadi mereka melalui komputer dengan akses internet. Pada *cloud computing* sumber daya seperti *processor/computing power, storage, network, software* menjadi abstrak (*virtual*) dan diberikan sebagai layanan di jaringan/internet juga dapat menggabungkan beberapa perangkat komputer menjadi satu kesatuan (*cluster*) dan membuat membuat banyak *server* pada satu perangkat komputer dengan virtualisasi. *Cloud computing* tercipta

karena timbulnya kendala seperti keterbatasan atau pemborosan *resource* komputer yang menyebabkan terhambatnya beberapa kegiatan perkomputasian. Agar terciptanya efisiensi, perusahaan-perusahaan besar di bidang TI (Teknologi Informasi) pun sekarang beralih menggunakan teknologi *cloud computing*. Terdapat beberapa manfaat *cloud computing* yaitu :

1. Menyimpan semua data pada *server* yang terpusat sehingga pengguna tak perlu repot menyediakan infrastruktur seperti *data center*, media penyimpanan /*storage* dll karena semua telah tersedia secara virtual.
2. Keamanan data pengguna disimpan dengan aman melalui *server* yang disediakan penyedia layanan *Cloud Computing* seperti jaminan platform teknologi, jaminan ISO, data pribadi, dll.
3. Fleksibilitas dengan kemudahan data akses, kapan dan dimanapun kita berada dengan catatan bahwa pengguna (*user*) terkoneksi dengan internet. Pengguna juga dapat dengan mudah meningkatkan atau mengurangi kapasitas penyimpanan data tanpa perlu membeli peralatan tambahan seperti hardisk.
4. Menghemat biaya pembelian inventaris karena pengguna akan dikenakan biaya kompensasi rutin per bulan sesuai dengan paket layanan yang telah disepakati dengan penyedia layanan *Cloud Computing*.

Dilihat dari segi fisik, cloud merupakan sebuah kumpulan komputer dan server yang dapat diakses secara publik menggunakan internet. Salah satu hal yang penting dalam cloud computing adalah arsitekturnya, dimana merupakan kemampuan pemrosesan dari sistem yang bukan hanya tampilan dekstop, dan semua user dapat mengakses komputasi yang telah disediakan oleh server cloud selama menggunakan jaringan internet. Arsitektur cloud computing dapat dijelaskan melalui **gambar 1.1**. Arsitektur cloud computing sangat sederhana dan tidak membutuhkan manajemen khusus untuk menghubungkan semua komputer dan melakukan alokasi pemrosesan task ke user. User hanya perlu mengakses menggunakan user interface di komputer, kemudian akan otomatis dapat mengakses task. Request dari user dapat dilewatkan ke manajemen sistem. Layanan ini akan mengambil sumber daya yang ada di cloud kemudian menjalankan aplikasi web tertentu dan aplikasi web lainnya. Setelah aplikasi web dijalankan, sistem akan memonitor dan mengontrol fungsi penggunaan cloud sehinggalah sumber daya dapat terbagi berdasar kebutuhan.



Gambar 1.1 Arsitektur Cloud Computing

Dalam kehidupan nyata, penggunaan *cloud computing* dapat dilihat pada penggunaan *Amazon Web Services* (AWS) yang disediakan oleh *Amazon* dan dapat diakses di <https://aws.amazon.com/id/about-aws/>. *Amazon Web Services* memiliki lebih banyak layanan, dan fitur dalam layanannya dibandingkan penyedia *cloud* lainnya, mulai dari teknologi infrastruktur seperti penghitungan, penyimpanan, dan *database* hingga teknologi yang berkembang, seperti kecerdasan buatan, data lake dan analitik, dan *Internet of Things*. Hal ini membuat seluruh proses komputasi data menjadi lebih cepat, lebih mudah, dan lebih hemat biaya. AWS juga memiliki fungsionalitas detail dalam semua layanannya. Misalnya, AWS menawarkan ragam database paling luas yang dibangun untuk beragam jenis aplikasi sehingga dapat memilih *tools* yang tepat untuk pekerjaan tersebut dengan mendapatkan biaya dan kinerja terbaik. Biaya yang dikenakan juga mengikuti penggunaan member. AWS menyediakan dukungan untuk kedua solusi *open-source* dan komersial *Docker*. Ada sejumlah cara untuk menjalankan kontainer di AWS, termasuk Amazon Elastic Container Service (ECS) yang merupakan layanan pengelolaan kontainer berkinerja tinggi dan sangat mudah diskalakan. AWS Fargate adalah teknologi untuk Amazon ECS yang memungkinkan untuk menjalankan kontainer di produksi tanpa menerapkan atau mengelola infrastruktur. Amazon Elastic Container Service for Kubernetes (EKS) memudahkan *user* dalam menjalankan Kubernetes di AWS. AWS Fargate adalah teknologi untuk Amazon ECS yang memungkinkan *user* menjalankan kontainer tanpa menyediakan atau mengelola server. Amazon Elastic Container Registry (ECR) adalah

repositori kontainer pribadi yang tersedia dengan sangat baik dan aman sehingga memudahkan menyimpan dan mengelola gambar kontainer Docker *user*, mengenkripsi dan mengompresi gambar saat istirahat sehingga cepat ditarik dan aman. AWS Batch memungkinkan *user* dalam menjalankan beban kerja pemrosesan *batch* yang sangat mudah diskalakan menggunakan kontainer *Docker*. Terdapat tiga penawaran dari *Amazon Web Services* yang disebut solusi, yaitu berdasar kasus penggunaan, industri, dan jenis organisasi. Contoh pada kasus penggunaan yaitu pengarsipan data, pencadangan dan pemulihan, *blockchain*, dan lain sebagainya. Contoh pada industri yaitu periklanan dan pemasaran, media dan hiburan, telekomunikasi dan lain-lain. Sedangkan pada jenis organisasi yaitu seperti *start-up*.

Projek pada penggunaan cloud computing ini berjudul Sistem Monitoring *Account Manager OS Ubuntu Menggunakan LAMPP dan Proses Pembuatan Docker* filenya. dimana merupakan sebuah sistem yang dapat memantau kinerja staff dari satu divisi. Pemantauan kinerja dalam *activity* dan *project*. Setiap staff memiliki akun masing-masing dan begitu juga dengan kepala divisi tersebut. Hak akses staff hanya pada akun masing-masing, staff dapat membuat, mengubah, menghapus, dan melihat *activity* serta *project* mereka masing-masing, namun tidak dapat melakukan modifikasi terhadap kinerja staff lain. Kepala divisi berhak dalam pembuatan atau penghapusan akun staff. Namun hanya dapat melihat (*monitoring*) *activity* serta *project* seluruh staff. Sistem ini hanya untuk satu divisi yaitu marketing. Dengan menggunakan *cloud computing*, sistem Monitoring *Account Manager OS Ubuntu Menggunakan LAMPP dan Proses Pembuatan Docker* filenya. dapat digunakan dimana saja. *Cloud computing* membantu sistem monitoring agar dapat bekerja lebih cepat dengan tingkat produktivitas yang tinggi. Disisi lain, dengan dibentuknya sistem monitoring dengan cloud computing maka akan memberikan kelebihan seperti *reability* dan *cost*. *Reability* dalam artian *cloud computing* dapat membuat cadangan data, dan sinkronisasi penggunaan untuk user lebih mudah dan lebih murah karena data dapat dicerminkan di beberapa situs berlebihan di jaringan penyedia *cloud*. Sistem Monitoring *Account Manager OS Ubuntu Menggunakan LAMPP dan Proses Pembuatan Docker* filenya dengan *cloud computing* juga memberikan kemudahan bagi *user* yang menggunakan seperti jika mendadak akan menggunakan aplikasi maka aplikasi dapat diakses dimana saja, dengan syarat sinyal memadai, maka dengan ini kita tidak memerlukan satu komputer atau satu *handphone* untuk mengaplikasikan Sistem Monitoring *Account Manager OS Ubuntu Menggunakan LAMPP dan Proses Pembuatan Docker* filenya melainkan dapat digunakan

dimana saja. Dengan penggunaan *Docker* juga semakin mempermudah pengembangan terhadap kode, penyaluran pada pipa, sampai menyediakan berbagai lingkungan yang lebih konsisten dalam aplikasi pengembangan hingga ke tahan produksi. Kelebihan *Docker* seperti penempatan lingkungan serta konfigurasi menuju kode dan menyebarkannya menjadikan *Docker* dapat digunakan juga dalam banyak lingkungan, dengan memisahkan kebutuhan infrastruktur pada lingkungan aplikasi tersebut karena konfigurasi yang sama. Adanya *Docker* juga semakin membuat *developer* bebas dalam menjalankan berbagai aplikasi dalam beberapa IaaS/PaaS dengan tidak menggunakan *tweak* tambahan. *Docker* dapat mempersingkat waktu pemrosesan sehingga dapat memberikan kepastian kepada *client* dengan hasil yang sesuai keinginan mereka. Hal menarik lainnya yang dapat dirasakan yaitu penggunaan multi-tenance yang mana dapat menghindarkan penulisan ulang pada aplikasi yang utama. Kemampuan yang dimiliki oleh aplikasi isolasi *Docker* dapat menggabungkan banyak server, dengan demikian dapat menghemat pengeluaran. Adanya *Docker* juga memberikan konsolidasi server yang lebih padat dibandingkan apa yang bisa dilakukan di *virtual machine*.

Dalam pembuatan projek akhir Sistem Monitoring Account Manager OS Ubuntu Menggunakan LAMP dan Proses Pembuatan *Docker* filenya ini, beberapa tahapan yang perlu dilakukan yaitu :

1. Menentukan judul projek akhir dan rancangan sistem projek akhir yang akan Dibuat.
2. Mempersiapkan komponen software yang dibutuhkan seperti *VMWare Workstation* atau *Virtual Box*, *Docker*, *ISO Ubuntu*, LAMP dll.
3. Melakukan instalasi *Virtual Machine* yang akan digunakan sebagai *Virtual Machine* Sistem Operasi Ubuntu.
4. Melakukan instalasi *ubuntu server* yang akan digunakan sebagai sistem operasi penunjang *cloud computing*.
5. Menginstall *software* yang akan digunakan sebagai media instalasi dan konfigurasi
6. Melakukan installasi *PHP*, *Mysql*, dan *Apache* yang akan digunakan sebagai media installasi dan software pada ubuntu
7. Melakukan konfigurasi *PHP*, *Mysql*, dan *Apache* dll sesuai dengan kebutuhan.
8. *Testing software* sesuai kebutuhan

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 Monitoring Account Manager OS Ubuntu Menggunakan LAMPP dan Proses Pembuatan *Docker* filenya yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04.3 LTS dan LAMPP (Apache 2.4, PHP 7.3, MySQL).

2. Mengintegrasikan Sistem *Monitoring Account Manager* yang berada di Ubuntu Serve LAMPP dengan platform produk layanan yang menggunakan virtualisasi tingkat OS *Docker* untuk digunakan sebagai media pengiriman perangkat lunak dalam paket yang disebut wadah.

Tanpa Spasi

1.3 Manfaat Proyek Akhir

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

1. Sistem Monitoring yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.

2. Perusahaan tidak perlu memperlakukan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.

3. Batasan memori penyimpanan multimedia menjadi tidak terbatas dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing*.

4. Permintaan data secara *realtime* dapat dilakukan secara terpusat maupun secara terpilih.

5. Biaya dalam perawatan sistem juga tidak mahal, hanya biaya rutin sesuai penggunaan, dan mengurangi pengeluaran perusahaan karena tidak ada lagi biaya tambahan untuk pembelian inventaris seperti infrastruktur, hardisk, dll.

6. Dengan menggunakan *Docker* juga dapat memberi manfaat berupa kemudahan dalam pengoperasian sistem, dan dapat sebagai acuan atau contoh dalam pembuatan sistem menggunakan *Docker*.

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari Sistem Monitoring Account Manager untuk ditransformasikan ke dalam arsitektur cloud computing menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Mengintegrasikan penyimpanan data record pada *Docker* dengan Sistem Monitoring Account Manager yang berada di Ubuntu Server.
3. Menentukan konfigurasi yang tepat untuk pengaturan *Docker* sehingga dapat digunakan sesuai *requirement* yang berupa:
 - a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Terdapat divisi pusat yang dapat masuk ke semua akun kecuali akun Z.
 - c. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
 - d. Menggunakan PHP 7.3
 - e. Virtualisasi menggunakan *docker*
4. Merancang topologi cloud computing untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
5. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Monitoring Account Manager ketersediaannya/availability-nya maksimal.
6. Menguji keandalan arsitektur cloud computing yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dsb.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir ini dijabarkan sebagai berikut :

1. Virtual Box *Graphical User Interface* 6.0.18 r136238 (Qt5.6.2)
2. Ubuntu 18.04.3 LTS
3. LAMPP
4. Apache
5. Docker 2.2.0.5 (44384) *stable*

Berikut akan dibahas keduanya masing-masing dalam sub bab terkait.

2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Sistem *Monitoring Account Manager OS Ubuntu* Menggunakan LAMPP dan Proses Pembuatan Docker filenya” 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 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data yang dibutuhkan oleh sistem tersebut.
3. Untuk virtualisasi menggunakan Virtual Box *Graphical User Interface* versi 6.0.18 r136238 (Qt5.6.2) dimana *virtual machine* ini digunakan untuk melakukan instalasi ubuntu server.
4. Apache
5. Untuk membangun aplikasi, mengemas dan menjalankan aplikasi pada project ini adalah Docker dengan versi 2.2.0.5 (44384) *stable*.
6. Untuk target pengguna dengan konsep *cloud computing*, maka penggunanya ialah seluruh karyawan divisi marketing di PT.Gamatechno Indonesia yang ber-kantor di Yogyakarta. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet PT.Gamatechno Indonesia. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang termasuk dalam divisi marketing.

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

Tabel 2.1 Spesifikasi VM *cloud computing* untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	VirtualBox Graphical User Interface	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	Intel(R) Core(TM) @ 3.60GHz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.43.164/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 127.0.0.53	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.8.1	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	2GB	Alokasi RAM untuk <i>guest OS</i>
6.	Penyimpanan Data	15GB	Penyimpanan 15 GB digunakan untuk root

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	bahasa pemrograman <i>script server-side</i> yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum
		MySQL	sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (<i>Database Management System</i>) yang multialur dan multipengguna.
2.	Docker	Versi 2.2.0.5 (44383)	aplikasi yang bersifat <i>open source</i> yang berfungsi sebagai wadah/ <i>container</i> untuk mengepak/memasukkan sebuah <i>software</i> secara lengkap beserta semua hal lainnya yang dibutuhkan oleh software tersebut dapat berfungsi

2.1.2 Komponen pada Proyek Kedua

Penjelasannya sama seperti contoh sub bab sebelumnya, namun ini untuk proyek kedua. Penjelasan di bab 2.1 seputar konfigurasi VMware yang digunakan, kebutuhan/requirement cloud computingnya seperti apa, Ubuntu setelah diinstal akan diisi apa saja, setelah itu targetnya mau diconfig seperti apa, dsb...

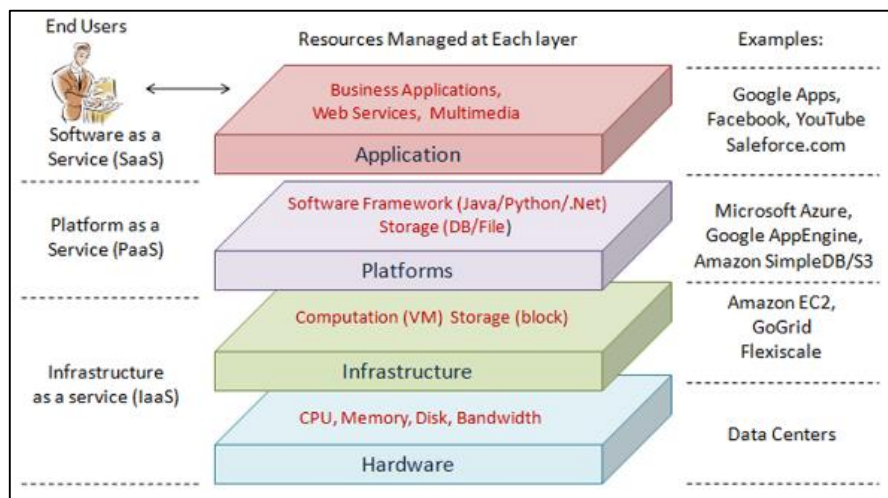
2.2 Rancangan Arsitektur *Cloud Computing*

Sama seperti penjelasan sebelumnya, untuk bagian ini juga dijelaskan menjadi dua sub bab bila sebelumnya seperti itu. Bilamana sebelumnya cuma satu sub bab karena judulnya jadi satu/satu topik kesatuan, maka di sini juga cukup 2.2 saja, tidak perlu dipecah menjadi dua. Begitu pula untuk sub bab-sub bab di bab dua semuanya seperti itu.

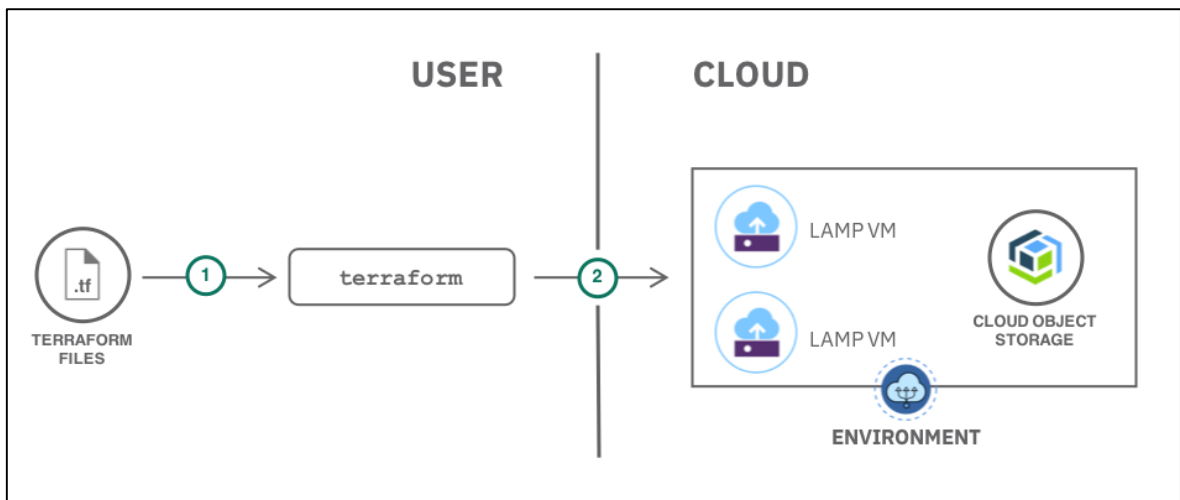
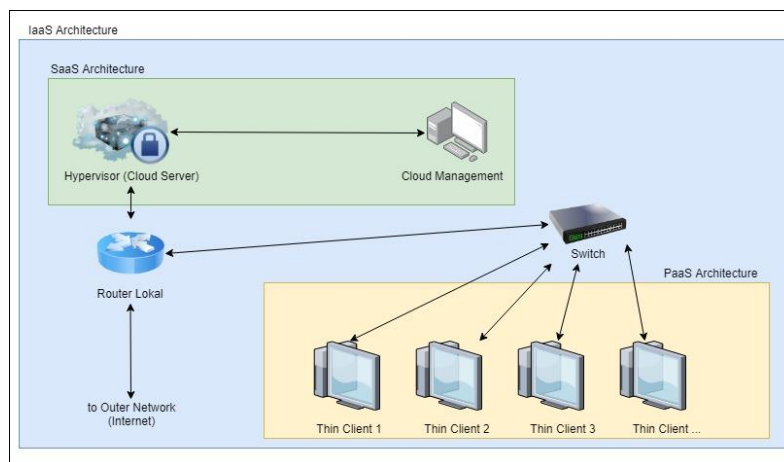
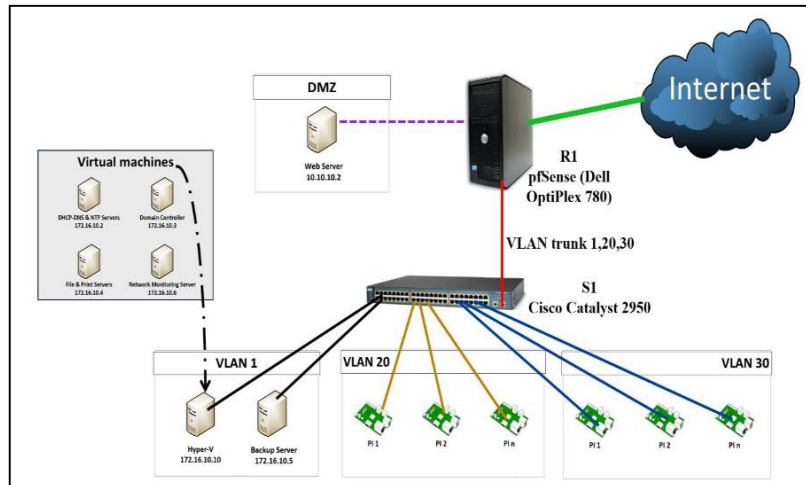
Pada bagian ini, buatlah gambar mengenai penyusunan komponen sebelumnya yang telah dilist. Gambar tersebut dapat berupa topologi jaringan terhadap alat yang digunakan, keterkaitan antar alat dan bahan yang digunakan, atau gambaran mengenai lingkup arsitektur (iaas/paas/saas) terhadap alat dan bahan. Gunakan kreasi dan kreativitas Anda sehingga pembaca memahami bentuk arsitektur yang Anda ingin buat. Format penulisan: penjelasan terlebih dahulu dalam bentuk poin-poin yang menggambarkan arsitektur, kemudian diacu pada gambar sekian yang berada di bawah penjelasan.

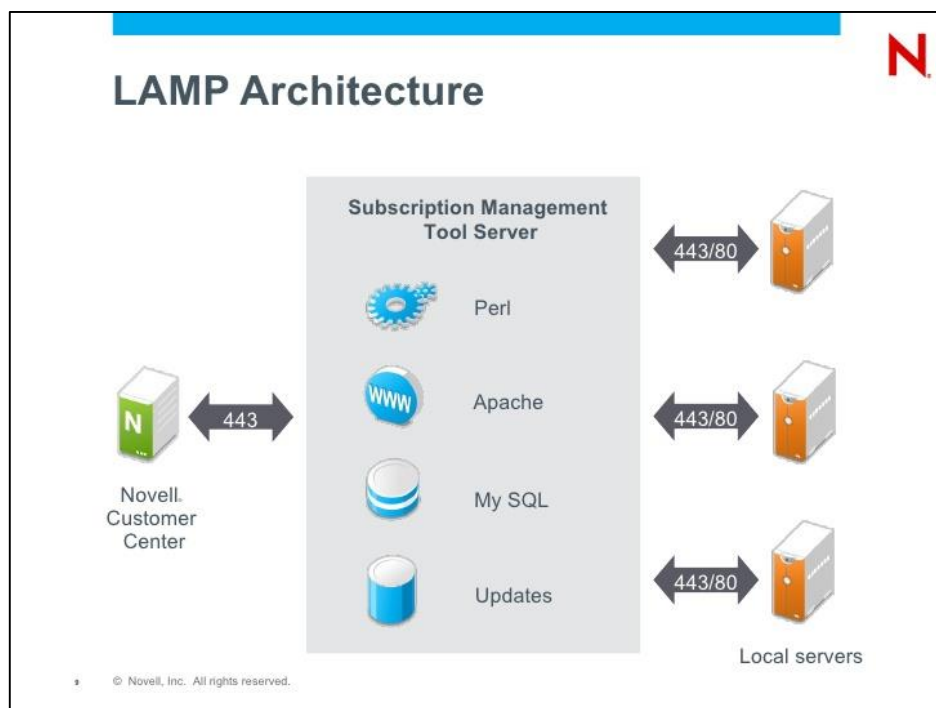
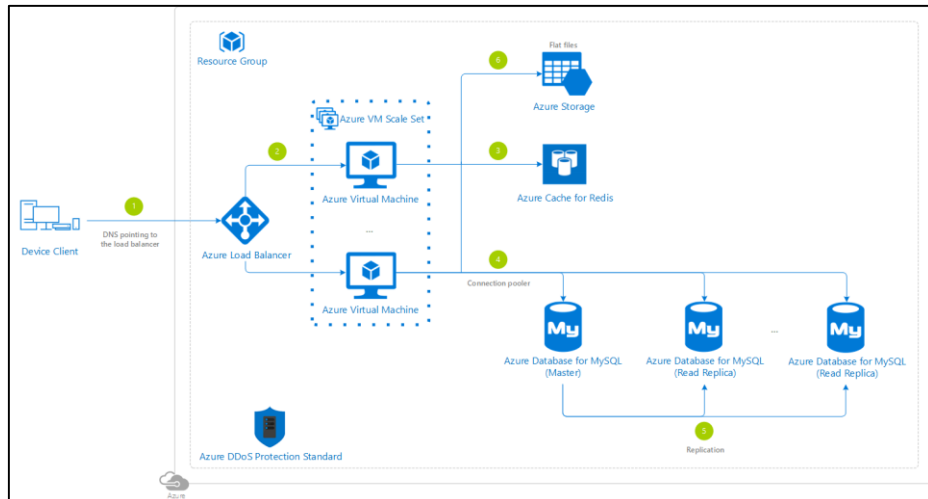
Dijelaskan dulu gambarnya seperti apa (komponen penyusunnya) baru gambarnya di bawah

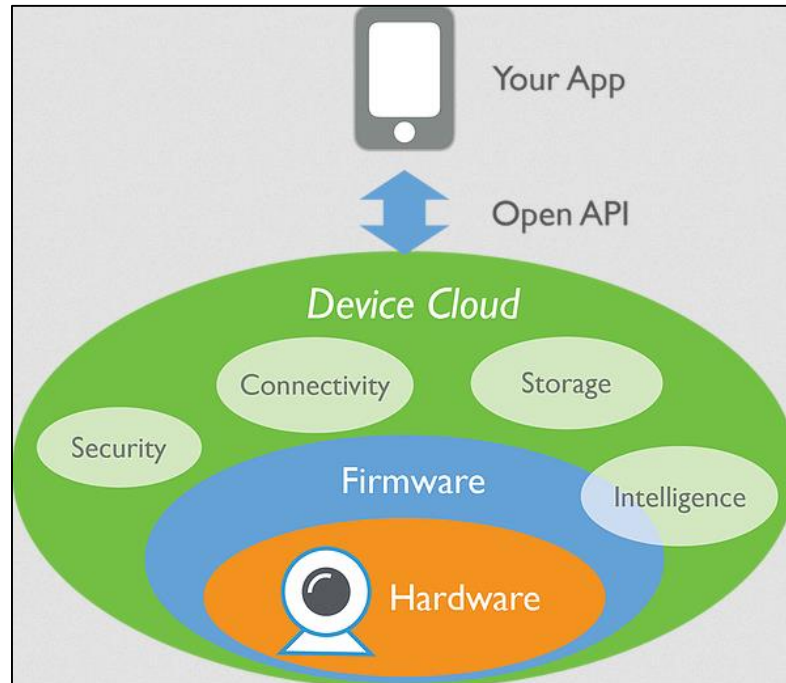
Contoh: pada project akhir ini digunakan bentuk rancangan arsitektur IaaS di mana hardware laptop ASUS ROG sebagai layer utama di bagian bawah, kemudian di atasnya adalah PaaS atau sistem operasi Windows dan aplikasi VMware Workstation yang berjalan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



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







2.3 Parameter dan Konfigurasi

Pada bagian ini, tuliskan dalam bentuk tabel dan penjelasannya mengenai isian parameter dan konfigurasi terhadap komponen alat dan bahan yang Anda gunakan. Jelaskan juga bilamana alat dan bahan yang digunakan tersebut perlu diolah (dikonfigurasi) terlebih dahulu sehingga siap pakai untuk digunakan pada tahap berikutnya (misal dikonfigurasi). Format tabel yakni: penjelasan terlebih dahulu mengenai jenis konfigurasi atau parameter, kemudian buat tabel, lalu berikan tulisan penjelasan mengenai keterangan di dalam tabel atau di bawah tabel bilamana diperlukan.

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

```
$ sudo apt install apache2
```

Keterangan:

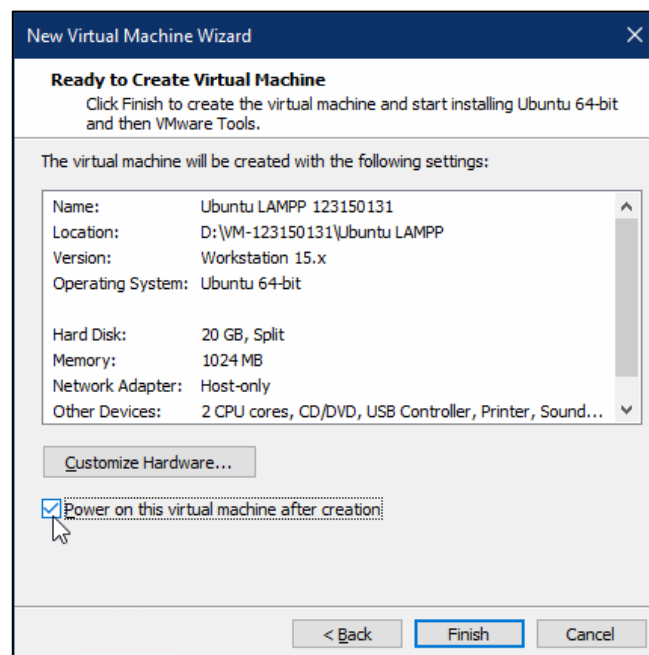
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- apache2 : nama paket aplikasi untuk Apache

Modul 2.1 Parameter instalasi Apache

2.4 Tahap Implementasi

Pada tahap ini jelaskan mengenai implementasi Anda terhadap penyelesaian soal yang diberikan. Buatlah dalam format yang terstruktur sesuai flow pengerjaan Anda. Beberapa yang perlu Anda tuliskan pada bagian ini di antaranya:

- a. Tahapan mengenai langkah-langkah kerja dan screenshootnya, semisal untuk tahap awal pada *wizard* pembuatan VM pada VMware Workstation digunakan opsi konfigurasi *Custom* seperti pada **Gambar 2.3** berikut ini:

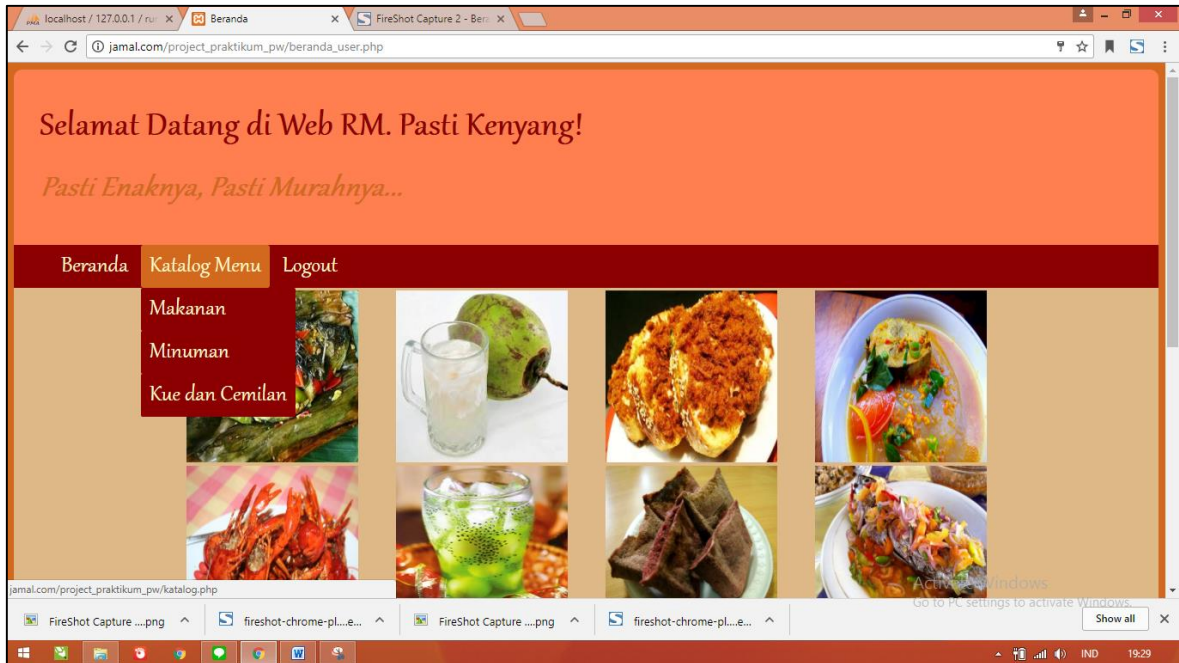


Gambar 2.3 Tampilan opsi pemilihan *mode wizard* pembuatan VM

- b. Kodingan atau settingan dari implementasi tahap sebelumnya dan disertakan screenshootnya bilamana memungkinkan
- c. Diteruskan sampai semua konfigurasi telah diatur semuanya sesuai rancangan/list di tahap sebelumnya

2.5 Hasil Implementasi

Pada tahap ini jelaskan mengenai hasil implementasi Anda bahwa apa yang telah Anda buat telah menyelesaikan persoalan yang ada. Jelaskan juga bahwa parameter dan konfigurasi yang Anda buat telah terpasang dan diatur dengan benar, selain itu juga dapat ditampilkan mengenai keadaan akhir hasil dari pekerjaan Anda (contoh koneksi internet telah ada, bisa diakses dari lokal, dsb)



2.6 Pengujian Singkat

Pada tahap ini buktikan bahwa rancangan yang Anda buat telah dapat digunakan terhadap beberapa pengujian singkat. Buat sendiri **dua** masalah yang **sesuai dengan penjelasan di latar belakang/tujuan proyek akhir, tentunya yang terkait dengan cloud computing**. Bagian ini akan dibuktikan secara langsung pada saat presentasi proyek akhir. Tuliskan pada bagian ini dengan format: masalahnya atau hasil yang diharapkan, eksekusi penyelesaian masalah, hasil yang didapat. Bila hasil yang didapat tidak sesuai, maka jelaskan apa kendalanya, tidak diwajibkan hasil harus sesuai dengan harapan. Berikan juga screenshootnya.

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			
		1	2	3	4	1	2	3	4	1	2	3	4
1.	Analisa Persoalan												
2.	Pembagian Tugas												
3.	Pemilihan judul projek dan pembuatan rancangan												
4.	Install VMWare, Pembuatan Laporan Bab I												
5.	Install Ubuntu, PHP MyAdmin, PHP, MySQL, Apache2												
6.	Revisi Laporan Bab I												
7.	Install Docker												
8.	Pembuatan Laporan Bab II												
9.													
10.													
11.													
12.	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	Fhrezha Rahmat
2.	Penginstalan Ubuntu	Fhrezha
3.	Penginstalan Package	Rahmat
4.	Agenda Pengerjaan Proyek	Fhrezha
5.	Instalasi Docker	Rahmat
6.		
7.		
8.		

(Pembagian tugas proyek akan diverifikasi oleh asisten praktikum, setiap penanggung jawab wajib bertanggungjawab terhadap tugasnya. Penilaian akan berdasarkan nilai individu dan tim.)

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Pada bagian ini, jelaskan mengenai hasil dari proyek yang Anda buat. Anda dapat mencontoh penulisan pada bagian 2.5 dan bagian 2.6. Penulisan pada bagian ini setidaknya memuat:

- a. Berdasarkan masalah, kemudian dikerjakan, maka didapatkan hasil yang seperti apa. Apakah dapat dikerjakan, terdapat kendala, atau hal teknis lainnya.
- b. Berdasarkan pengujian masalah secara acak, apa yang didapatkan.
- c. Berdasarkan pembagian tugas proyek pada bab 3, bagaimana hasilnya.

4.2 Saran

Pada bagian ini, jelaskan mengenai saran-saran yang mengacu pada bagian 4.1 mengenai kesimpulan proyek. Contohnya: pembagian tugas seharusnya lebih merata, spesifikasi laptop seharusnya dengan ram 10 GB, waktu yang tersedia tidak cukup dikarenakan bla bla bla, dan sebagainya.

DAFTAR PUSTAKA

Tambahkan daftar pustaka dengan format yang digunakan di IF, yakni APA Style. Setidaknya ada 5 daftar pustaka yang Anda gunakan untuk menyelesaikan proyek ini. Spacing dalam satu paragraf single, tambahkan spasi/jarak antar paragraf

PostMedya, < <http://www.postmedya.com/teknologi/cloud-computing-pengertian-sejarah-serta-manfaatnya/> >, (2018, accessed 29 April 2020)

Restu Adyatma, < <http://aboutdoublr.blogspot.com/2013/11/memahami-arsitektur-cloud-computing.html> >, (2014, accesseed 29 April 2020)

Development & Security, Web & Development, < <https://idcloudhost.com/mengenal-apa-itu-docker-definisi-fungsi-keunggulan-dan-cara-kerjanya/> >, (6 Febuari 2020, accessed 27 April 2020)

Mobnasesemka.com, < <https://mobnasesemka.com/keunggulan-docker/> >, (24 Juli 2016, accessed 28 April 2020)

_____, < <https://aws.amazon.com/id/docker/> >, (____, accessed 28 April 2020)

Juni Yadi, < <https://juniyadi.id/apa-itu-docker-serta-kelebihan-dan-kekurangan-docker/> >, (28 Maret 2019, accessed 28 April 2020)

Wang, Shulong., Hou, Yibin., Gao, Fang., & Ji, Xinrong. 2016. “A Novel IoT Access Architecture for Vehicle Monitoring System”. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT).

_____, <<https://idcloudhost.com/pengertian-internet-of-things-iot/>>, (23 Jun 2016, accessed 10 Mei 2019)

Erick, Jan Solem. 2012. *Programming Computer Vision with Python*.

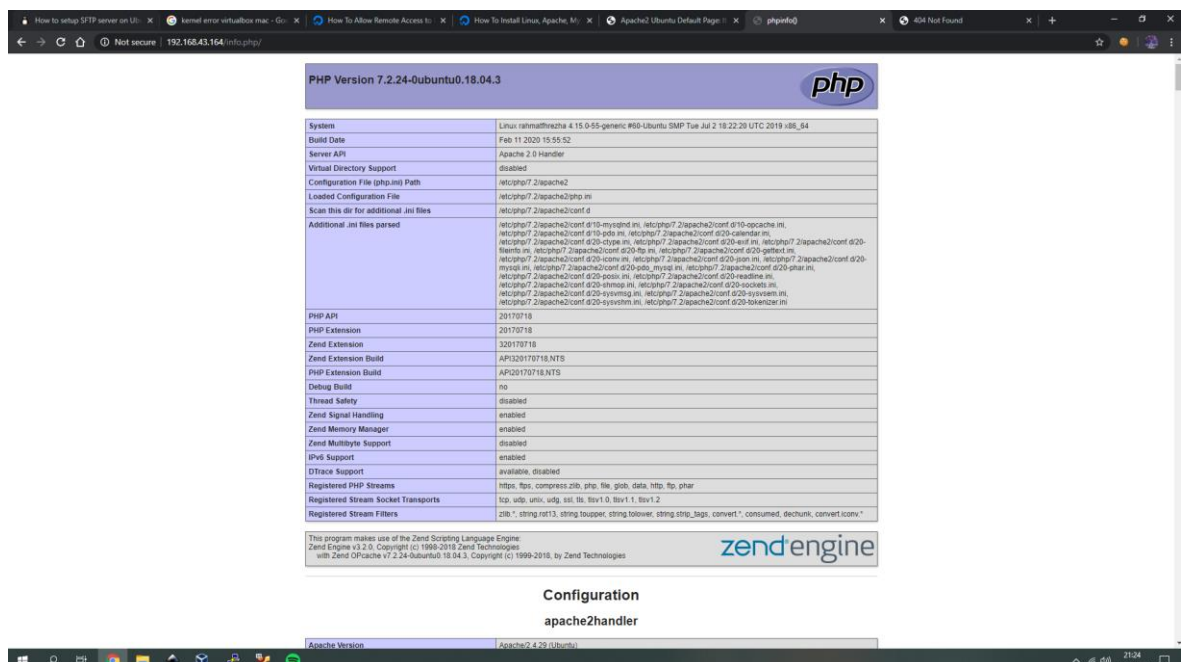


LAMPIRAN

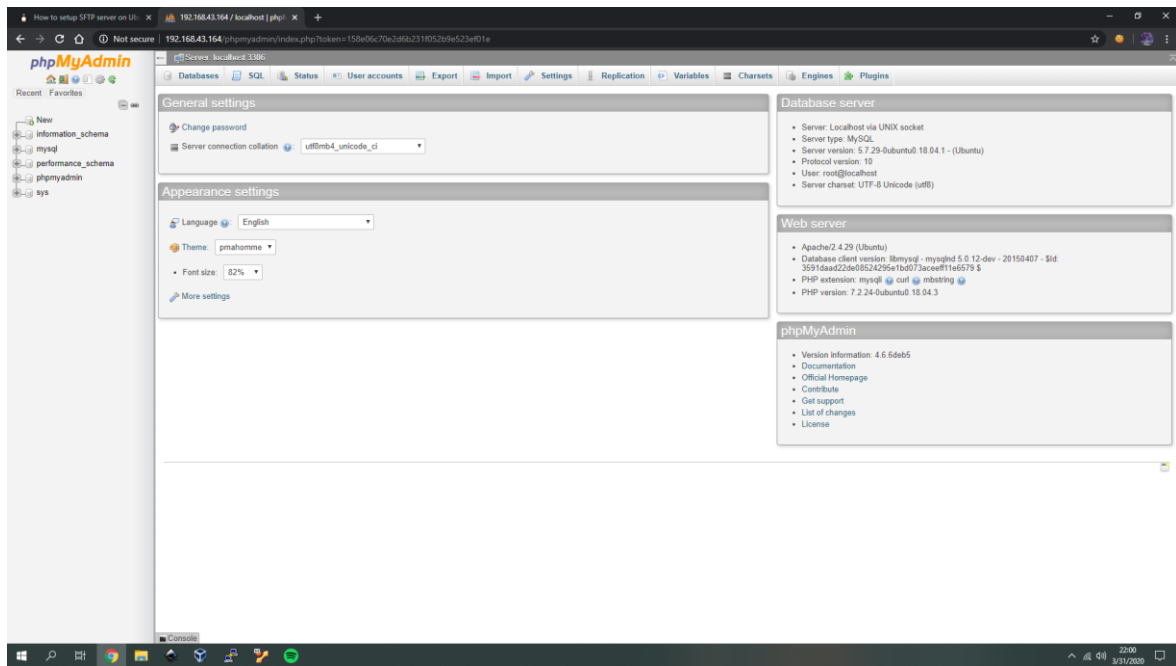
Lampiran pada bagian ini dapat berupa screenshoot, listing program yang terlalu panjang, dan sebagainya, atau dapat juga tugas bilamana diminta oleh asisten praktikum.



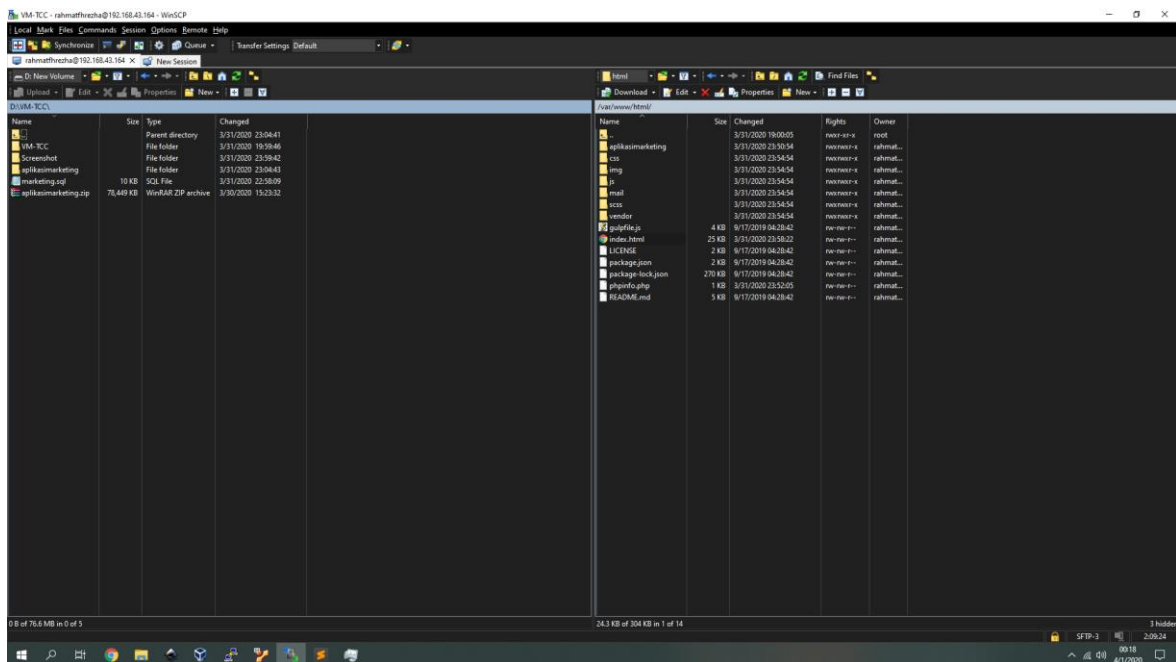
Gambar 1.2 Instalasi Apache



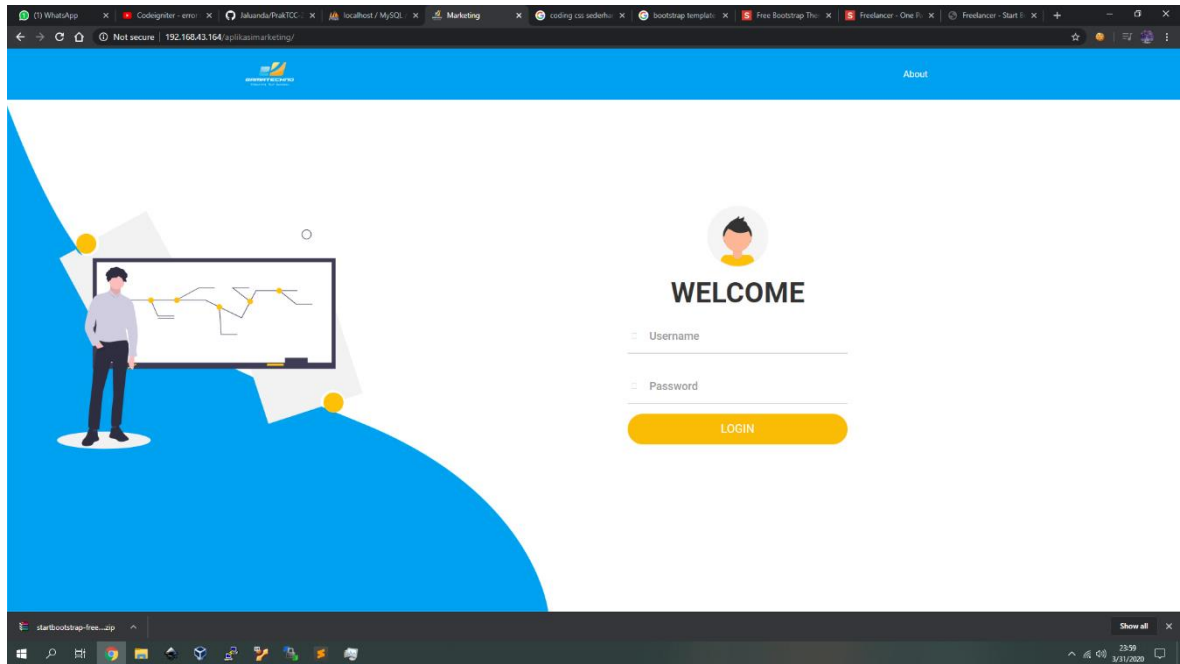
Gambar 1.3 Instalasi PHP



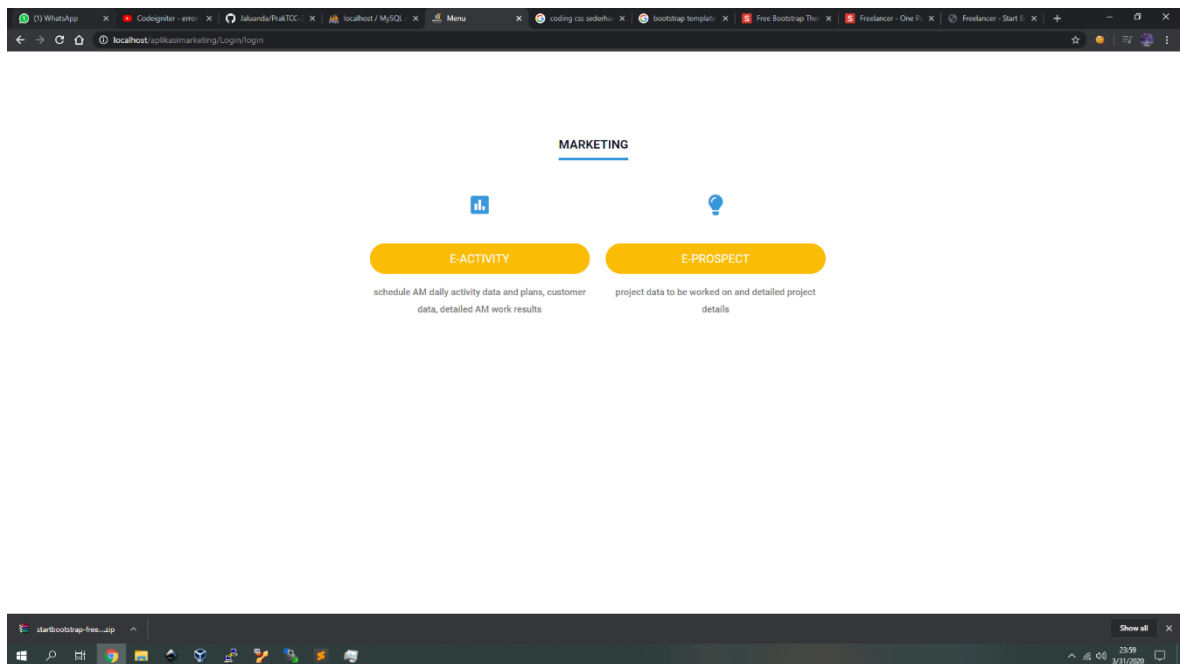
Gambar 1.4 Hasil instalasi PHPMYAdmin



Gambar 1.5 Transfer File menggunakan WinSCP



Gambar 1.6 Hasil hosting di LAMPP



Gambar 1.7 Hasil hosting pada LAMP

CONTOH FORMAT PENULISAN

(hapus bagian ini)

Beri penjelasan mengenai gambar sebelum mencantumkan gambar. Setiap gambar wajib dengan border hitam tipis. Jangan lupa acu gambar tersebut, yakni pada **Gambar 2.1** berikut:



Gambar 2.1 Font 11 Center

Contoh penulisan tabel sama seperti gambar, buat dulu penjelasannya kemudian acu pada **Tabel 1.2** berikut ini. Format yakni spacing single dan font size 10:

<i>Field</i>	<i>Tipe Data</i>	<i>Keterangan</i>
Nama Field 1	Tipe Data 1	<i>Jenis Constraint</i>
Nama Field 2	Tipe Data 2	<i>Jenis Constraint</i>
Nama Field 3	Tipe Data 3	
Nama Field 4	Tipe Data 4	
Nama Field 5	Tipe Data 5	

Tabel 1.2 Tabel Nama Tabel 2

Keterangan dituliskan bila perlu, mengacu pada **Tabel 1.2** yakni:

- Format pengisian pada kolom field yakni nama field
- Keterangan dsb dsb

Untuk penulisan listing program atau kodingan jelaskan terlebih dahulu maksud listing tersebut lalu acu pada **Listing 2.1** berikut:

```
CANTUMKAN QUERY CREATE TABLE 1 DENGAN FONT COURIER SIZE 10 SPACING  
SINGLE TANPA JUSTIFY  
  
public function blabla(){  
    exit();  
}
```

Listing Program 2.1 Listing Create Table Nama Tabel 1

Bila ingin menyatakan pengetikan parameter semisal `sudo apt install apache2` maka jelaskan terlebih dahulu maksud perintah tersebut lalu gunakan format pada **Perintah Program 2.1** berikut ini:

```
$ sudo apt install apache2
root@server:~# sudo apt install apache2
```

Perintah Program 2.1 Operasi instalasi paket aplikasi *apache2*

Penjelasan ketentuan penulisan Laporan Proyek Akhir secara umum:

1. Huruf Times New Roman 12
2. Margin Top Left 3, Right Bottom 2.5.
3. Spacing 1,5
4. Penomoran Halaman, pada setiap **awal bab** di **bawah tengah**. Setiap **isi bab**, di **atas kanan**.
Font TNR 11
5. Jatah nilai untuk format penulisan adalah 25%, bilamana format acak-acakan maka jatah nilai lainnya gugur, sehingga nilai proyek akhir = 0. Laporan merupakan bukti bahwa Anda telah mengerjakan tugas sesuai dengan apa yang Anda presentasikan

Selamat mengerjakan, semoga sukses skripsi dan lulus cepat dengan nilai memuaskan.