

577

6 m>Ui UbXUDUfUa UUn% \$%-2\$-#)/#B\$&\$

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING LAPORAN PROYEK AKHIR

**SISTEM INFORMASI ODE BENTARA BUDAYA MENGGUNAKAN UBUNTU LAMP DAN PROSES
PEMBUATAN DOCKER FILE NYA**



DISUSUN OLEH:

**NAMA ANGGOTA : EDWINA AYU C 123170095
MUHAMMAD MA'RUF A R 123170101
KELAS : B
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.KOM
IMAM ALFATAH**

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

HALAMAN PENGESAHAN

SISTEM INFORMASI ODE BENTARA BUDAYA MENGGUNAKAN UBUNTU LAMPP DAN PROSES
PEMBUATAN DOCKER FILE NYA

Disusun oleh :

Edwina Ayu Christy

123170095

Muhammad Ma'ruf Al Rizqi

123170101

Telah diperiksa dan disetujui oleh Asisten Praktikum Teknologi Cloud Computing

pada tanggal :

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123456789

Mengetahui,
Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul *Sistem Informasi Ode Bentara Budaya menggunakan Ubuntu Lampp Dan Proses Pembuatan Dockerfile-nya*. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

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

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

Yogyakarta, 01 April 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2 Tujuan Proyek Akhir	1
1.3 Manfaat Proyek Akhir	1
1.4 Tahap Penyelesaian Proyek Akhir	2
BAB II ISI DAN PEMBAHASAN	3
2.1 Komponen yang Digunakan	3
2.2 Rancangan Arsitektur <i>Cloud Computing</i>	3
2.3 Parameter dan Konfigurasi	4
2.4 Tahap Implementasi	5
2.5 Hasil Implementasi	6
2.6 Pengujian Singkat	6
BAB III JADWAL Pengerjaan dan Pembagian Tugas	8
3.1 Agenda Pengerjaan	8
3.2 Keterangan Pembagian Tugas	8
BAB IV KESIMPULAN DAN SARAN	9
4.1 Kesimpulan	9
4.2 Saran	9
DAFTAR PUSTAKA	10
LAMPIRAN	11

BAB I

PENDAHULUAN

1.1. Latar Belakang Proyek Akhir

Sebuah komputer, memiliki beberapa kecenderungan yang selalu digunakan, maupun dibutuhkan seperti Ketersambungan, kemampuan berbagi, Operasi tanpa henti, dan butuh mengecil, namun menyebar. Cloud Computing : “Cloud” yang artinya Awan (internet); “Computing” (komputasi); merupakan pengembangan teknologi komputer, memanfaatkan internet dengan *software* lengkap dan sistem operasional yang juga tersedia secara online. Seluruh nya terkait dengan internet, sehingga terminal yang digunakan juga menjadi terpusat. Internet dianggap sebagai awan besar berisi komputer yang saling tersambung, dengan awan itu (Stevan Greve). Cloud Computing memiliki kapasitas virtual yang tidak terbatas dengan data dan lokasi penyimpanan yang berkualitas tinggi, beberapa fitur yang lengkap harus mengeluarkan biaya. Teknologi ini harus didukung dengan koneksi internet yang cukup memadai, dan beberapa fitur disesuaikan sesuai dengan apa yang kita bayarkan kepada perusahaan pendukung Cloud Computing.

Salah satu penggunaan cloud computing digunakan dalam penyimpanan Website di Internet. Pada proyek kali ini, website yang diambil adalah ODE Bentara. Ode Bentara merupakan sebuah komunitas yang isinya adalah seluruh paduan suara yang ada di Daerah Istimewa Yogyakarta. Di dalamnya terdapat kamus istilah musik, dan juga forum untuk melakukan diskusi antar paduan suara. Database yang digunakan sebelumnya menggunakan PHP My Admin. Sebelumnya sudah pernah di unggah dengan domain .000webhostapp, dengan URL <https://odebentara.000webhostapp.com/> yang merupakan layanan domain gratis. URL di atas sudah tidak dapat diakses dikarenakan tenggat waktu yang sudah habis, karena tidak membayar domain tadi. Untuk mengaktifkannya kembali, maka perlu diupload kembali dengan menggunakan domain lain, yang akan dikerjakan menggunakan

Dalam proses deteksi terkadang masih menemui kesalahan yaitu terpisahnya objek deteksi karena terdapat bagian dari objek yang memiliki derajat keabuan yang mirip dengan *background*. Pada penelitian *monitoring* yang lain, kendaraan dalam lalu lintas juga dapat terdeteksi dengan baik dengan menggunakan metode *multiblob* (Augusta, Sari, & Adikara, 2019). Penggunaan *multiblob* mampu menghasilkan tingkat akurasi sebesar 73,75% untuk *monitoring* lalu lintas dalam kondisi lalu lintas lengang, namun penggunaan *multiblob* sangat bergantung pada tingkat kepadatan lalu lintas dalam area pemantauan, sehingga ketika lalu lintas berada dalam keadaan ramai, sistem sering kali melakukan kesalahan dengan membaca *segerombolan* kendaraan menjadi satu kesatuan besar. (di paragraf ini mulai jelaskan kendala apa yang ada di lapangan sehingga perlu dibuat sistemnya/produknya (tidak harus kendala yang berkaitan dengan cloud), setelah itu jelaskan bagaimana cara membawa sistem yang telah dibuat tersebut ke dalam cloud, konsepnya bagaimana... misalnya: ketika dibawa ke cloud maka SOP-nya akan berubah sehingga diperlukan sosialisasi lagi... contoh lain: perlu mendefinisikan server yang sesuai kebutuhan sehingga tidak kurang resource maupun boros resource/beli terlalu mahal namun tidak digunakan maksimal...)

Dari beberapa penelitian yang telah dijabarkan, masih terdapat beberapa kekurangan, diantaranya adalah penggunaan algoritma yang memiliki keterbatasan pada proses deteksi, yaitu hanya mampu mendeteksi satu objek pada satu proses deteksi. Kekurangan lainnya adalah penggunaan teknik *preprocessing* yang masih sederhana, sehingga sistem belum mampu menghasilkan representasi objek dengan baik yang berpengaruh pada ketepatan proses deteksi. Sehingga dari permasalahan yang telah dijabarkan, solusi yang diusulkan pada penelitian ini adalah penggunaan *multi-frame detection* untuk proses deteksi kendaraan sehingga mampu mendeteksi kendaraan dengan jumlah yang lebih banyak serta dapat mengetahui kecepatan yang dihasilkan dari kendaraan yang melintasi area pemantauan, dan penggunaan *cascaded mask* untuk melakukan proses *filtering* citra sehingga mampu menghasilkan data citra yang baik untuk berbagai kondisi pencahayaan yang berbeda. (di paragraf terakhir ini setelah menganalisis

'konsep' untuk membawanya ke cloud pada paragraf sebelumnya, dilanjutkan dengan bagaimana 'teknis singkat' cara pengerjaannya nanti di project ini... jelaskan poin-poin singkatnya saja bukan step how to ini itu nya... semisal pertama menganalisis kebutuhan hardware, kemudian menentukan versi software yang akan digunakan, kemudian pembuatan vm berdasarkan analisis konfigurasi yang telah dibuat sebelumnya, dst. hingga hasil akhir sistem diuji berhasil...

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 Informasi Ode Bentara Budaya* dengan menggunakan Ubuntu 20.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Mengimplementasikan aritektur cloud computing untuk *Sistem Informasi Ode Bentara Budaya* menggunakan Docker (Docker 19.03.8, PhPMyAdmin, dan MySQL 5.2)

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 dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.

(bagian ini cenderung menjelaskan manfaat yang didapatkan ketika sistem telah beralih ke arsitektur cloud computing, cobalah mendeskripsikan manfaat yang lebih

spesifik, bukan yang umum seperti dapat diakses dari mana saja, tetapi lebih jauh setelah diakses dari mana saja lalu apa manfaat lebih jauh dari itu..... jelaskan manfaat yang mengenai cloud computingnya saja)

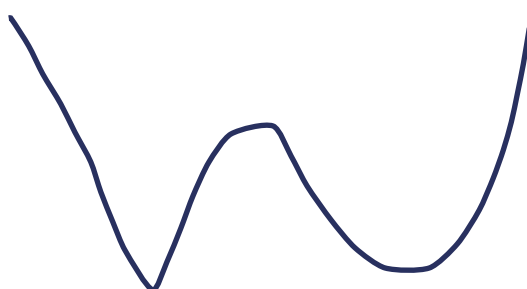
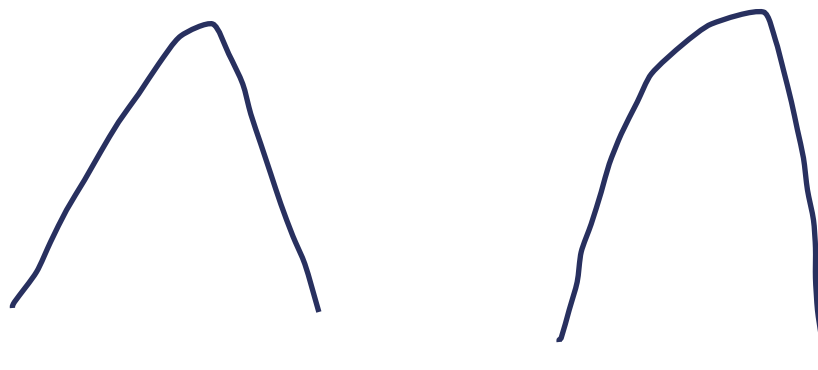
1.4 Tahap Penyelesaian Proyek Akhir

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

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

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

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



BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir ini terbagi menjadi dua sesuai dengan judul yang telah dituliskan sebelumnya. Atau komponen dituliskan jadi satu kesatuan jika judulnya nyambung antar dua topik dan bentuknya cenderung integrasi antar kedua topik. Bila dibagi menjadi dua, jelaskan terlebih dahulu yang pertama ialah “Sistem Pelacakan Kendaraan Bermotor di Indonesia menggunakan Ubuntu LAMPP” dan yang kedua “Penyimpanan data mentahnya (RAW) di FreeNAS”. Berikut akan dibahas keduanya masing-masing dalam sub bab terkait.

2.1.1 Komponen pada Proyek Pertama

Untuk membangun “Sistem Pelacakan Kendaraan Bermotor di Indonesia 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 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data kendaraan bermotor yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh kantor kepolisian di Indonesia dan kantor kepolisian pusat di Jakarta. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet kepolisian. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang berkepentingan.
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. Dan seterusnya...

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

Tabel 2.1 Spesifikasi VM *cloud computing* untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.110.2/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.110.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.110.1	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	4GB	Alokasi RAM untuk <i>guest OS</i>
6.	dsb	dsb	dsb

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	dsb
		dsb	dsb
		dsb	dsb

2.	Phython	Versi 2.7	Dukungan bahasa pemrograman yang digunakan oleh sistem pelacakan kendaraan.
3.	dsb	dsb	dsb

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, Ubuntunya setelah diinstal akan diisi apa saja, setelah itu targetnya mau diconfig seperti apa, dsb...

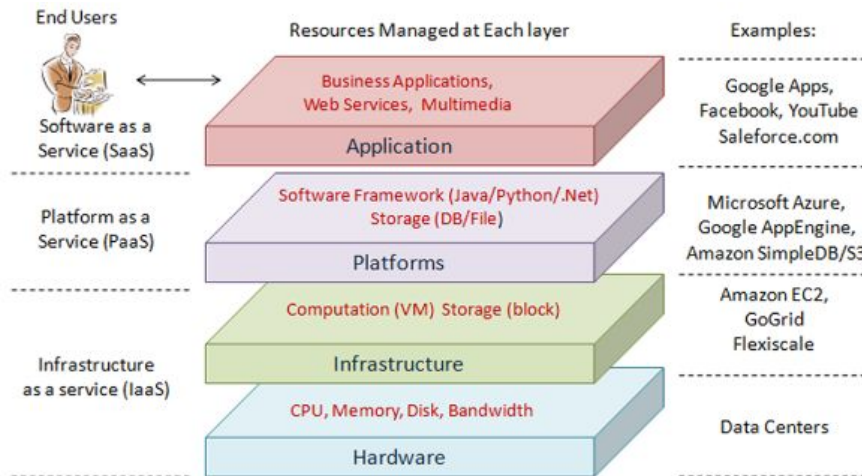
2.2 Rancangan Arsitektur *Cloud Computing*

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.

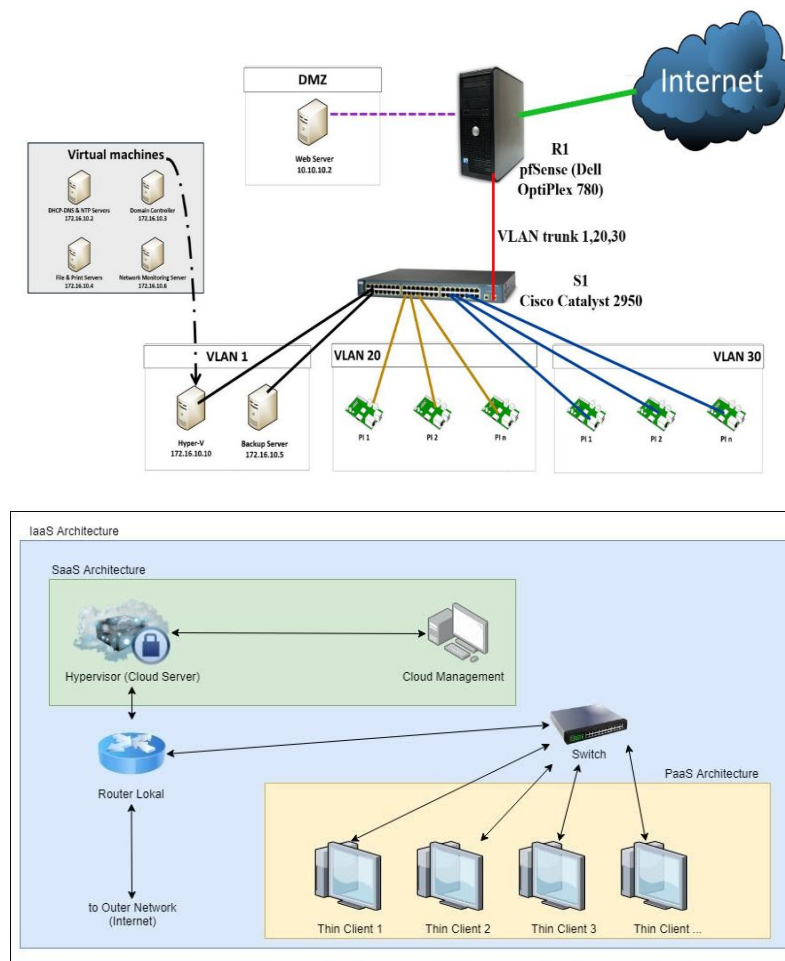
Pada bagian ini, buatlah gambar mengenai penyusunan komponen sebelumnya yang telah dilist. Gambar tersebut dapat berupa topologi jaringan terhadap alat yang digunakan, keterkaitan antar alat dan bahan yang digunakan, atau gambaran mengenai lingkup arsitektur (iaas/paas/saas) terhadap alat dan bahan. 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.

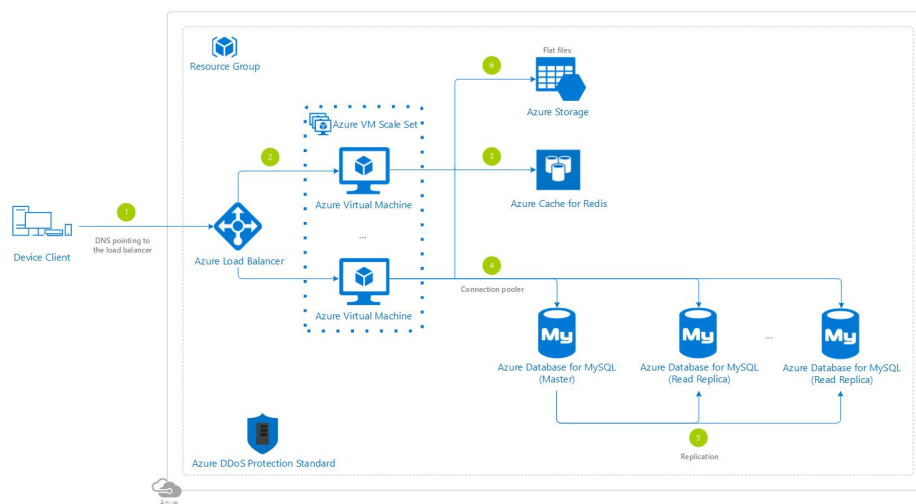
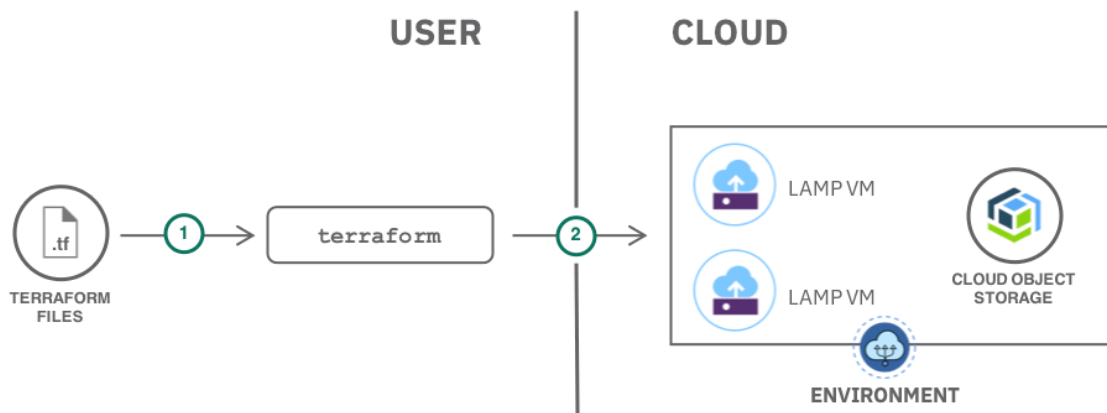
Dijelaskan dulu gambarnya seperti apa (komponen penyusunnya) baru gambarnya di bawah

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:

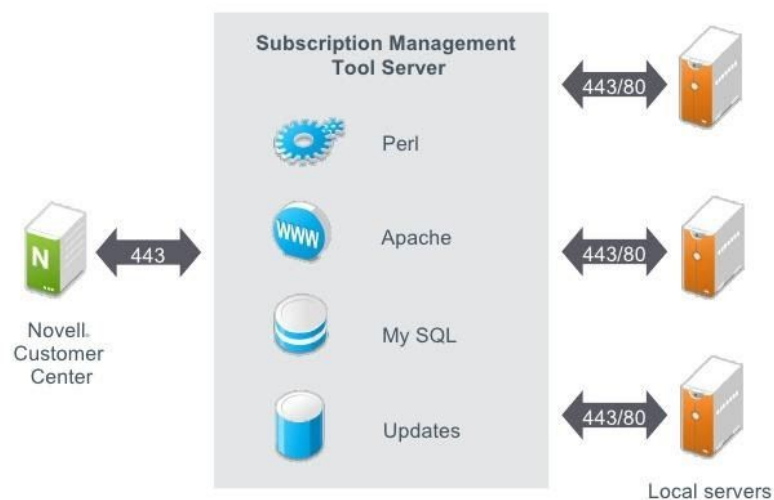


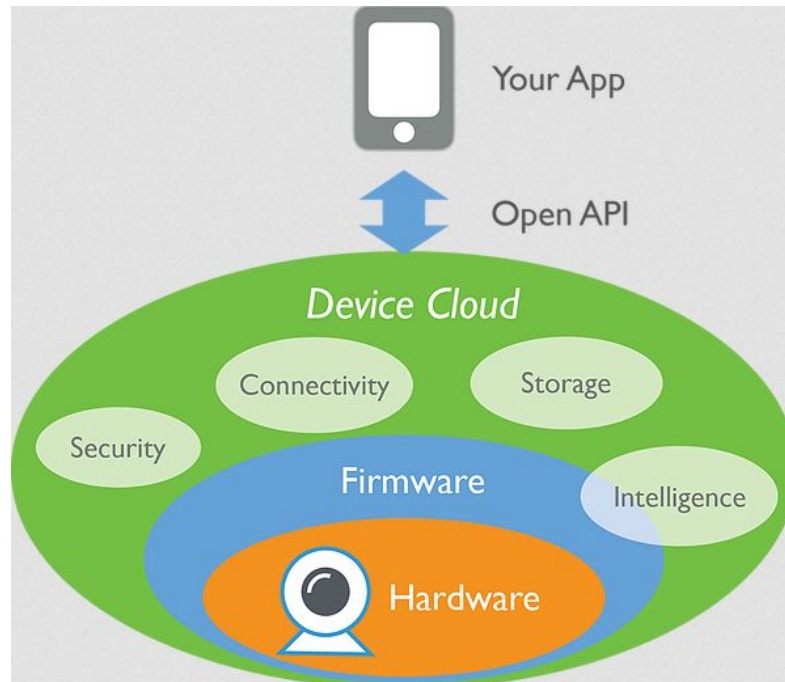
Gambar 2.1 Penjelasan layer arsitektur terhadap komponen penyusunnya (font 11, TNR)





LAMP Architecture





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

2.4.1. Sistem Informasi Ode Bentara Budaya Menggunakan Ubuntu LAMPP

2.4.2 Sistem Informasi Ode Bentara Budaya Menggunakan Dockerfile

2.4.2.1 Instalasi Docker

2.4.2.2. Instalasi MySQL dan PhPMyAdmin

1. Buat file dengan nama docker-compose.yml.
2. Pada terminal, masuk ke direktori file tersebut disimpan. Lalu ketikkan perintah ini, satu – persatu

```
touch docker-compose.yml  
open docker-compose.yml
```

Lalu akan membuka text editor default anda, isikan listing berikut untuk MySQL dan PHPMyAdmin:

```
version: "2"  
services:  
  db:  
    container_name: mysql  
    hostname: mysql  
    image: mysql  
    ports:  
      - "3306:3306"  
    environment:  
      MYSQL_DATABASE: dockerDB  
      MYSQL_USER: dockerDB  
      MYSQL_PASSWORD: dockerDB  
      MYSQL_ROOT_PASSWORD: dockerDB  
    volumes:  
      - persistent:/var/lib/mysql  
      - ./files:/files  
    networks:  
      - default  
  phpmyadmin:  
    container_name: phpmyadmin  
    image: phpmyadmin/phpmyadmin  
    hostname: phpmyadminDocker  
    links:  
      - db:db  
    ports:  
      - 8000:80
```



```

environment:
  MYSQL_USER: dockerDB
  MYSQL_PASSWORD: dockerDB
  MYSQL_ROOT_PASSWORD: dockerDB
volumes:
  persistent:

```

Simpan file docker-compose.yml

3. Lakukan Instalasi, dengan menggunakan perintah

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

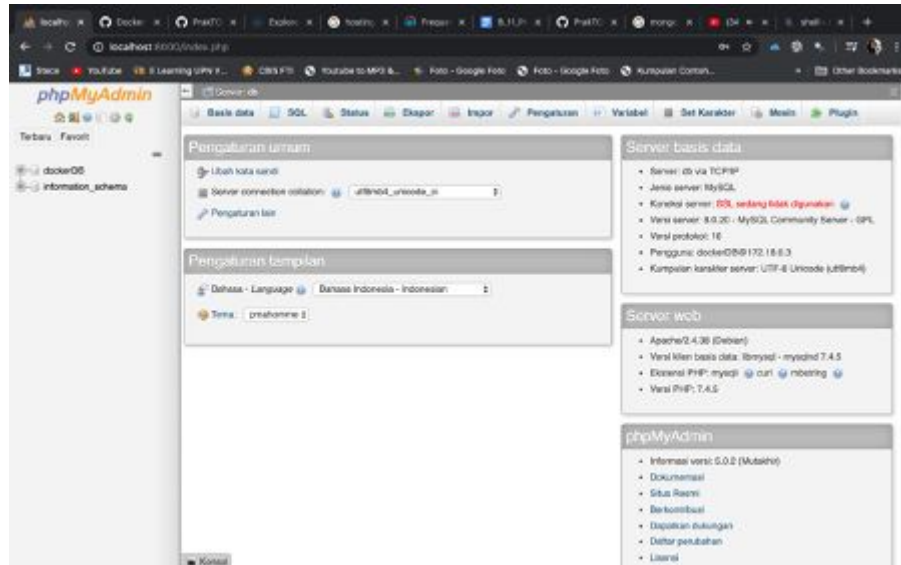
4. Maka, Container yang berjalan ada 2 dalam docker, yaitu mysql, dan phpmyadmin, dapat dilihat dengan perintah

```
docker-compose ps
```

Name	Command	State	Ports
mysql	docker-entrypoint.sh mysqld	Up	0.0.0.0:3306->3306/tcp, 33060/tcp
phpmyadmin	/docker-entrypoint.sh apache2	Up	0.0.0.0:8000->80/tcp

5. Dari instalasi ini, maka hasilnya dapat dilihat dengan membuka <http://localhost:8000/> pada jendela browser, akan muncul.

Password dan user disesuaikan dengan pengaturan di file docker-compose.yml sebelumnya.



2.4.2.3. HOSTING PHP

1. Buka Text Editor (sublime, Visual Studio Code, dll)

Buat File Php dalam folder (file lebih baik di luar)

Dalam kasus ini : `ed > ode > src > index.php`

2. Buat Dockerfile dengan nama “Dockerfile” jangan ada tambahan lainnya, karena tidak akan terbaca. Isi Dockerfile dengan listing sebagai berikut :

```
FROM php:7.4-cli
COPY . /usr/src/myapp
WORKDIR /usr/src/myapp
CMD [ "php", "./index.php" ]
```

Simpan, pindah ke jendela terminal

3. Masuk ke dalam direktoripenyimpanan Dockerfile, lalu kita buat image dockerfile tersebut, dalam proyek ini, image akan dibuat dengan nama “ode” sehingga untuk membuat image, gunakan perintah:

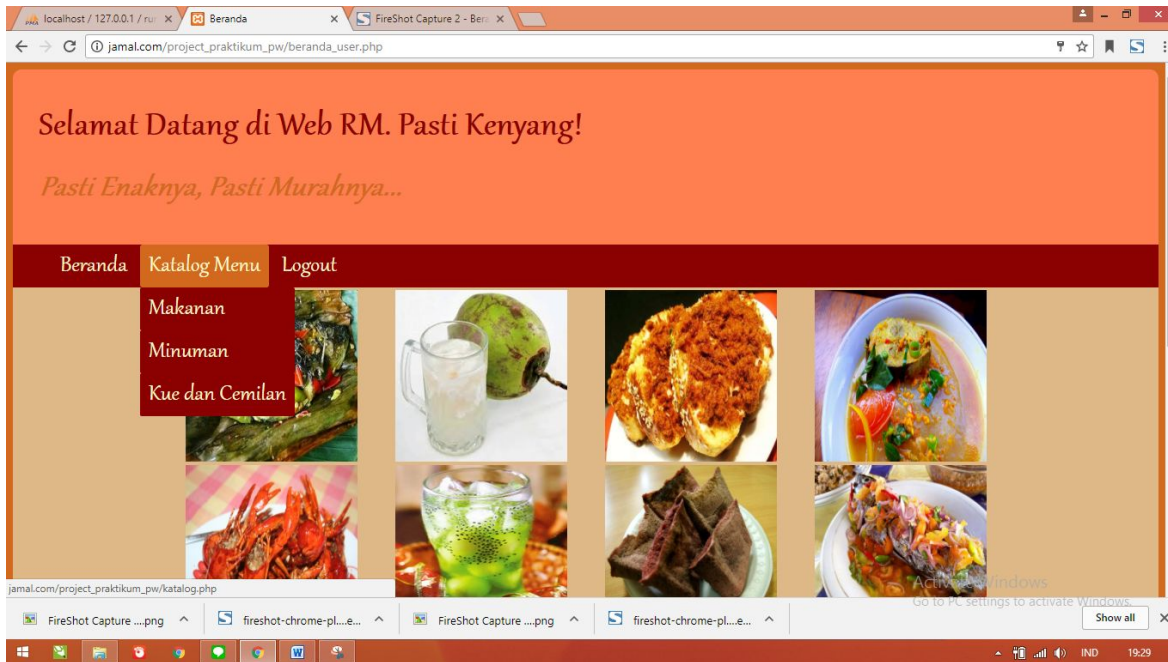
```
docker build -t ode .
```

```
EDs-MacBook-Pro:src ed$ docker build -t ode .
Sending build context to Docker daemon 3.072kB
Step 1/4 : FROM php:7.4-cli
7.4-cli: Pulling from library/php
54fec2fa59d0: Already exists
cc4504446071: Pull complete
f8d5b89827d: Pull complete
6bf7f8bea876: Pull complete
9a178eabfa4a: Pull complete
99b83d0b7ff2: Pull complete
79c3ed7f9118: Pull complete
cc3f66def185: Pull complete
4f837fda4b52: Pull complete
Digest: sha256:aa449b3b778a3549a89dde21b35261516ecabd84891081b72000a8f9aa457f3c
Status: Downloaded newer image for php:7.4-cli
--> 6340e7a80170
Step 2/4 : COPY . /usr/src/myapp
--> 5ffa9f23b23c
Step 3/4 : WORKDIR /usr/src/myapp
--> Running in dd90c19bdedc
Removing intermediate container dd95c19bdedc
--> 02db77538e8e
Step 4/4 : CMD [ "php", "./index.php" ]
--> Running in 4b152824a6e2
Removing intermediate container 4b152824a6e2
--> 284e5534ee9c
Successfully built 284e5534ee9c
Successfully tagged ode:latest
EDs-MacBook-Pro:src ed$ docker run -p 80:80 ode
Hello worldEDs-MacBook-Pro:src ed$ docker image
```

Jika menggunakan Dockerfile di atas, maka file yang dibuat muncul dalam bentuk CMD

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 - 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.	Pembagian Tugas								
3.	Persiapan komponen pendukung								
4.	Pengerjaan 1 - Proyek 1 dan konsultasi								
5.	Pengerjaan 2 - Proyek 2 dna konsultasi								
6.	Pengerjaan 3 - Laporan								
7.	Pengerjaan 4 - <i>finishing</i> Proyek 1 dan 2								
8.	Pengerjaan 8 - <i>Finishing</i> laporan								
9.	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.	Persiapan proyek, dan konsultasi	Edwina
2.	Bab I - Latar Belakang Masalah	Maruf
3.	Bab II - Komponen yang digunakan - Proyek 1	Maruf
4.	Bab II - Komponen yang digunakan - Proyek 2	Edwina
5.	Bab II - Rancangan Arsitektur Cloud Computing	Edwina
6.	Bab II - Parameter Konfigurasi	Maruf
7.	Bab II - Proyek 1	Maruf
8.	Bab II - Proyek 2	Edwina
9.	Pengujian Singkat	Maruf
10.	Bab III - Pembagian Tugas	Edwina

11.	Finishing laporan	Edwina
12.	Submisi Proyek	Edwina

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

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.