



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - K141502

PENERAPAN CLOUD COMPUTING UNTUK MENINGKATKAN KETERSEDIAAN SERVER DI ITS MENGUNAKAN TEKNOLOGI KONTAINER

MOHAMMAD WAHYU HIDAYAT
NRP 5112 100 125

Dosen Pembimbing
Ir. Muchammad Husni, M.Kom.
Ridho Rahman H., S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - K141502

PENERAPAN CLOUD COMPUTING UNTUK MENINGKATKAN KETERSEDIAAN SERVER DI ITS MENGGUNAKAN TEKNOLOGI KONTAINER

MOHAMMAD WAHYU HIDAYAT
NRP 5112 100 125

Dosen Pembimbing
Ir. Muchammad Husni, M.Kom.
Ridho Rahman H., S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - K141502

CLOUD COMPUTING IMPLEMENTATION FOR IMPROVING AVAILABILITY OF ITS SERVER USING CONTAINER TECHNOLOGY

MOHAMMAD WAHYU HIDAYAT
NRP 5112 100 125

Advisor

Ir. Muchammad Husni, M.Kom.
Ridho Rahman H., S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENERAPAN CLOUD COMPUTING UNTUK MENINGKATKAN KETERSEDIAAN SERVER DI ITS MENGUNAKAN TEKNOLOGI KONTAINER

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur dan Jaringan Komputer
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
MOHAMMAD WAHYU HIDAYAT
NRP : 5112 100 125

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Ir. Muchammad Husni, M.Kom.
NIP: 19600221 198403 1 001

(pembimbing 1)

Ridho Rahman H., S.Kom., M.Sc
NIP: 19870213 201404 1 001

DEPARTEMEN
TEKNIK INFORMATIKA
(pembimbing 2)

SURABAYA
JUNI 2017

[Halaman ini sengaja dikosongkan]

PENERAPAN CLOUD COMPUTING UNTUK MENINGKATKAN KETERSEDIAAN SERVER DI ITS MENGUNAKAN TEKNOLOGI KONTAINER

Nama Mahasiswa : Mohammad Wahyu Hidayat
NRP : 5112 100 125
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Ir. Muchammad Husni, M.Kom.
Dosen Pembimbing 2 : Ridho Rahman H., S.Kom., M.Sc.

ABSTRAK

Semakin berkembangnya teknologi pada era digital ini menuntut untuk penerapan teknologi informasi di segala sektor. Sehingga penggunaan infrastruktur penunjang IT juga besar seperti komputer dan server. Untuk membuat sebuah server dibutuhkan tempat serta biaya yang cukup besar dari biaya pengadaan server, instalasi server, dan biaya untuk merawat server itu sendiri. Untuk itu perlu sebuah teknologi yang bisa digunakan sebagai pengganti server. Salah teknologi yang bisa digunakan adalah menerapkan konsep cloud computing

Kontainer adalah salah satu teknologi cloud computing dimana kita membuat ruang isolasi pada server untuk menjalankan sebuah aplikasi khususnya aplikasi web. Teknologi kontainer sama seperti virtualisasi hanya saja jika virtualisasi kita seolah-olah membuat mesin terpisah didalam komputer sedangkan kontainer hanya membuat ruang isolasi pada kernel yang sama. Selain itu kelebihan kontainer adalah apabila Pada satu server pengguna bisa menjalankan beberapa kontainer secara bersamaan tanpa saling mengganggu satu sama lain.

Hasil membuktikan dengan menerapkan konsep kontainer bisa memenuhi ketersediaan server khususnya di ITS. Dengan resource yang ada dapat menjalankan beberapa aplikasi secara

bersamaan tanpa mengganggu aplikasi yang lain dan setiap aplikasi dapat diatur berapa resource yang akan digunakannya.

Kata kunci: server, cloud computing, kontainer.

CLOUD COMPUTING IMPLEMENTATION FOR IMPROVING AVAILABILITY OF ITS SERVER USING CONTAINER TECHNOLOGY

Name : Mohammad Wahyu Hidayat
NRP : 5112 100 125
Major : Teknik Informatika FTIf-ITS
Supervisor I : Ir. Muchammad Husni, M.Kom.
Supervisor II : Ridho Rahman H., S.Kom., M.Sc.

ABSTRACT

The development of technology in the digital era is demanding for the application of information technology in all sectors. So that the use of IT support infrastructure is also large such as computers and servers. To make a server takes place as well as a considerable cost of server procurement costs, server installation, and the cost to maintain the server itself. For that need a technology that can be used instead of server . One technology that can be used is to apply the concept of cloud computing

Container is one of cloud computing technology where we make isolation space on the server to run an application especially web application. Container technology is the same as virtualization only if our virtualization seems to make a separate machine inside the computer while the container only creates isolation space in the same kernel. In addition, the excess container is if on one server the user can run multiple containers simultaneously without interrupting each other.

The results prove by applying the concept of container can meet the availability of the server especially in ITS. With existing resources can run multiple applications simultaneously without disturbing other applications and each application can be set how many resources will be used.

Keywords: server, cloud computing, container.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

PENERAPAN CLOUD COMPUTING UNTUK MENINGKATKAN KETERSEDIAAN SERVER DI ITS MENGUNAKAN TEKNOLOGI KONTAINER

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, adik, kakak dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Ir. Muchammad Husni, M.Kom. selaku dosen pembimbing tugas akhir pertama yang telah membimbing dan memberikan banyak masukan dalam pengerjaan tugas akhir ini.
3. Bapak Ridho Rahman Hariadi, S.Kom.,M.Sc. selaku dosen pembimbing tugas akhir kedua yang selalu memberikan koreksi serta banyak masukan-masukan yang dapat penulis kembangkan dari tugas akhir ini
4. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak menyampaikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
5. Teman-teman yang selalu menemani, mengingatkan dan menyemangati dalam pengerjaan tugas akhir ini.
6. Administrator Laboratorium Arsitektur dan Jaringan Komputer dan administrator Laboratorium Desain Terapan dan Komputasi yang memberikan kesempatan untuk mengerjakan tugas akhir di laboratorium.
7. Teman-teman angkatan 2012 Jurusan Teknik Informatika ITS, khususnya C1C yang telah menjadi teman

seperjuangan dalam suka dan duka selama 5 tahun penulis menjalani kuliah.

8. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2017

Mohammad Wahyu Hidayat

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	1
1.3. Rumusan Permasalahan.....	2
1.4. Batasan Permasalahan	2
1.5. Metodologi	2
1.6. Sistematika Penulisan.....	3
BAB II DASAR TEORI.....	7
2.1. Cloud Computing	7
2.2. Docker	8
2.3. Laravel.....	9
2.4. MySQL.....	10
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	13
3.1. Analisis Metode Secara Umum	13
3.1.1. Analisis Permasalahan.....	13
3.1.2. Deskripsi Umum Sistem.....	14
3.1.3. Spesifikasi Kebutuhan Perangkat Lunak.....	14
3.1.3.1. Kebutuhan Fungsional.....	14
3.1.3.2. Diagram Kasus Penggunaan.....	15
3.1.3.3. Kasus Penggunaan Mengelola User	17
3.1.3.4. Kasus Penggunaan Membuat Image	19
3.1.3.5. Kasus Penggunaan Menghapus Image	19

3.1.3.6.	Kasus Penggunaan Membuat Kontainer	20
3.1.3.7.	Kasus Penggunaan Memperbaharui Kontainer	21
3.1.3.8.	Kasus Penggunaan Mendeploy Aplikasi	22
3.1.3.9.	Kasus Penggunaan Menghapus Kontainer	23
3.2.	Perancangan Sistem.....	24
3.2.1.	Perancangan Arsitektur.....	24
3.2.2.	Desain Umum Sistem	25
3.2.3.	Desain Docker	25
3.2.4.	ITSConcloud.....	27
3.2.5.	Penyimpanan Data.....	28
	BAB IV IMPLEMENTASI.....	29
4.1.	Lingkungan Implementasi	29
4.2.	Implementasi Docker.....	29
4.2.1.	Instalasi Docker	30
4.2.2.	Konfigurasi Pada Docker.....	31
4.2.2.1.	Membuat Docker Sebagai Non-Root User	31
4.2.2.2.	Mengatur Jaringan Bridge Docker	32
4.2.2.3.	Mengatur API Docker	32
4.2.3.	Perintah API Docker.....	32
4.3.	Implementasi ITSConcloud.....	33
4.3.1.	Mengelola User.....	34
4.3.2.	Membuat Image	35
4.3.3.	Menghapus Image.....	36
4.3.4.	Membuat Kontainer	36
4.3.5.	Mendeploy Aplikasi	37
4.3.6.	Menghapus Kontainer.....	37
4.3.7.	Memperbaharui Kontainer.....	38
4.4.	Penyimpanan Data.....	38
	BAB V PENGUJIAN DAN EVALUASI.....	43
5.1.	Lingkungan Pengujian.....	43
5.2.	Skenario Uji Coba	43

5.2.1.	Skenario Uji Fungsionalitas Aplikasi ITSConcloud	43
5.2.2.	Skenario Uji Performa Aplikasi ITSConcloud...	45
5.2.2.1.	Uji Perfoma Waktu membuat Image.....	45
5.2.2.2.	Uji Perfoma Jumlah Kontainer yang Berjalan	45
5.3.	Hasil Uji Coba dan Evaluasi.....	45
5.3.1.	Uji Fungsionalitas Aplikasi ITS Concloud.....	45
5.3.1.1.	Skenario pengujian 1: Membuat Data User	46
5.3.1.2.	Skenario pengujian 2: Memperbaharui Data User	47
5.3.1.3.	Skenario pengujian 3 : Menghapus Data User ..	49
5.3.1.4.	Skenario pengujian 4 : Membuat Image.....	49
5.3.1.5.	Skenario pengujian 5 : Melihat Logs	50
5.3.1.6.	Skenario pengujian 6 : Melihat Image	51
5.3.1.7.	Skenario pengujian 7 : Menghapus Image	52
5.3.1.8.	Skenario pengujian 8 : Membuat Kontainer.....	52
5.3.1.9.	Skenario pengujian 9 : Memperbaharui Kontainer	53
5.3.1.10.	Skenario pengujian 10 : Menghentikan Kontainer	54
5.3.1.11.	Skenario pengujian 11 : Menjalankan Kontainer	55
5.3.1.12.	Skenario pengujian 12 : Menghapus Kontainer	56
5.3.2.	Uji Peforma	58
5.3.2.1.	Waktu Membuat Image.....	58
5.3.2.2.	Jumlah Kontainer yang Berjalan	58
6 BAB VI	KESIMPULAN DAN SARAN.....	61

6.1.	Kesimpulan.....	61
6.2.	Saran.....	61
DAFTAR PUSTAKA.....		63
Lampiran A.....		65
BIODATA PENULIS.....		71

DAFTAR GAMBAR

Gambar 2.1 Perbandingan docker dan virtual machine.....	9
Gambar 3.1 Gambar Diagram Kasus Penggunaan	17
Gambar 3.2 Desain Sistem Secara Umum	24
Gambar 4.1 Antarmuka menambah user	34
Gambar 4.2 Antarmuka Mengubah Data User	35
Gambar 4.3 Antarmuka List User	35
Gambar 4.4 Antarmuka Membuat Image	36
Gambar 4.5 Antarmuka Menghapus Image.....	36
Gambar 4.6 Membuat Kontainer	37
Gambar 4.7 Mendeploy Aplikasi dan Menghapus Kontainer	37
Gambar 4.8	38
Gambar 5.1 Data Awal User	46
Gambar 5.2 Menambah Data User	47
Gambar 5.3 Data Akhir User.....	47
Gambar 5.4 Data Awal User	48
Gambar 5.5 Mengubah Data User	48
Gambar 5.6 Data Akhir User.....	48
Gambar 5.7 Data Awal User	49
Gambar 5.8 Data Akhir User.....	49
Gambar 5.9 Data Awal Image	50
Gambar 5.10 Membuat Image	50
Gambar 5.11 Data Akhir Image	50
Gambar 5.12 Data Pembuatan Image	51
Gambar 5.13 Logs Pembuatan Image	51
Gambar 5.14 Data Image.....	52
Gambar 5.15 Data Awal Image	52
Gambar 5.16 Data Akhir Image	52
Gambar 5.17 Data Awal Kontainer	53
Gambar 5.18 Membuat Kontainer	53
Gambar 5.19 Data Akhir Kontainer	53
Gambar 5.20 Data Awal Kontainer	54
Gambar 5.21 Mengubah Konfigurasi Kontainer	54

Gambar 5.22 Data Akhir Kontainer	54
Gambar 5.23 Data Awal Kontainer	55
Gambar 5.24 Data Akhir Kontainer	55
Gambar 5.25 Data Awal Kontainer	56
Gambar 5.26 Data Akhir Kontainer	56
Gambar 5.27 Data Awal Kontainer	56
Gambar 5.28 Data Akhir Kontainer	57
Gambar 7.1 Halaman Utama ITSConcloud.....	67
Gambar 7.2 Halaman Pendaftaran Pengguna	67
Gambar 7.3 Halaman Setelah Login.....	68
Gambar 7.4 Halaman Upload Image	69
Gambar 7.5 Halaman List Image.....	69
Gambar 7.6 Halaman Membuat Kontainer.....	70
Gambar 7.7 Halaman List Kontainer.....	70

DAFTAR TABEL

Tabel 3.1 Daftar Kode Kebutuhan Fungsional Aplikasi	15
Tabel 3.2 Daftar Kode Kasus Penggunaan	15
Tabel 3.3 Skenario Kasus Penggunaan Mengelola User	17
Tabel 3.4 Skenario kasus Penggunaan Membuat Image	19
Tabel 3.5 Skenario Kasus Penggunaan Menghapus Image	20
Tabel 3.6 Skenario Kasus Penggunaan membuat kontainer.....	20
Tabel 3.7 Skenario Kasus Penggunaan Memperbaharui Kontainer	21
Tabel 3.8 Skenario Kasus Penggunaan Mendeploy Aplikasi.....	22
Tabel 3.9 Skenario Kasus Penggunaan Menghapus Kontainer ...	23
Tabel 3.10 Rute pada API Docker.....	26
Tabel 3.11 Tabel yang digunakan pada aplikasi	28
Tabel 4.1 Rute API Docker Bagian Pengelolaan Image	33
Tabel 4.2 Rute API Docker Bagian Pengelolaan Kontainer	33
Tabel 4.3 Tabel User	38
Tabel 4.4 Tabel Image.....	39
Tabel 4.5 Tabel Container	40
Tabel 4.6 Tabel file	40
Tabel 5.1 Skenario Uji Fungsionalitas Aplikasi ITSConcloud	44
Tabel 5.2 Hasil uji Fungsionalitas Aplikasi ITSConcloud	57
Tabel 5.3 Hasil uji Peforma Waktu Membuat Image	58
Tabel 5.4 Hasil uji Peforma Kontainer yang Dapat Berjalan	58

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 1 Send API Method Get	65
Kode Sumber 2 Send API Method POST	65
Kode Sumber 3 Send API Method DELETE	66

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Semakin berkembangnya teknologi informasi di masyarakat membuat penggunaan komputer semakin banyak untuk membantu pekerjaan sehari-hari. Penggunaan komputer dapat dijumpai diberbagai aktivitas harian masyarakat, diantaranya bank, instansi pendidikan, perkantoran, instansi pemerintahan, dan lain lain. Hal ini membuat kebutuhan akan penggunaan komputer meningkat lalu bagaimana dengan infrastruktur yang kurang memadai, instalasi jaringan, dan perawatan. Dari hal tersebut terpikirlah konsep cloud computing, dimana pengguna dapat menggunakan infrakstruktur tanpa perlu pengguna menyediakan hanya cukup dengan menyewa saja.

Kontainer adalah salah satu teknologi cloud computing dimana kita membuat ruang isolasi pada server untuk menjalankan sebuah aplikasi. Pada satu server pengguna bisa menjalankan beberapa kontainer secara bersamaan tanpa saling mengganggu satu sama lain. Teknologi kontainer sama seperti virtualisasi hanya saja jika virtualisasi kita seolah-olah membuat mesin terpisah didalam komputer sedangkan kontainer hanya membuat ruang isolasi pada kernel yang sama. Sehingga kontainer lebih ringan dijalankan tidak membutuhkan resource yang tinggi.

1.2. Tujuan

Tugas akhir dibuat dengan beberapa tujuan. Berikut beberapa tujuan dari pembuatan tugas akhir:

1. Mampu menerapkan cloud computing pada jaringan ITS.

2. Meningkatkan ketersediaan server di lingkup ITS.
3. Memudahkan pengelolaan setiap aplikasi pada server tersebut.
4. Memudahkan perawatan server tersebut.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini antara lain:

1. Bagaimana menjalankan banyak aplikasi pada server yang terbatas.
2. Bagaimana mendesain *hosting* aplikasi berbasis kontainer.
3. Bagaimana mendekomposisi aplikasi kedalam bentuk yang sederhana.

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, antara lain:

1. Server hanya dapat diakses melalui jaringan ITS.
2. Format *file* untuk pembuatan *Docker Image* hanya bias dalam format *tar archive* yang didalamnya terdapat *Dockerfile*.
3. Server yang digunakan dalam pengimplementasian tugas akhir hanya satu.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Ide tugas akhir ini muncul berawal ingin menerapkan cloud computing di ITS serta ketersediaan server di ITS masih kurang. Dari ide tersebut beserta artikel dan paper dari internet yang menjadi inspirasi dalam pengerjaan tugas akhir ini.

2. Implementasi

Hasil analisis dan desain kemudian diimplementasikan melalui komponen – komponen pendukung. Mengimplementasikan kontainer pada sebuah server dan mengatur jaringan yang digunakan sehingga bisa dan mudah dijalankan.

3. Pengujian dan evaluasi

Pengujian dari penerapan cloud computing untuk meningkatkan ketersediaan server di ITS menggunakan teknologi kontainer akan diujikan pada Laboratorium Arsitektur dan Jaringan Komputer pada Teknik Informatika ITS dan yang akan diujikan antara lain sebagai berikut :

1. Melakukan ujicoba dengan pengguna akan mendeploy aplikasi pada cloud computing dan bisa dijalankan.
2. Membuat image secara bersamaan serta menguji berapa aplikasi yang bisa dijalankan secara bersamaan pada server yang digunakan.

4. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.6. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Metode Pemecahan Masalah

Bab ini membahas mengenai Metode yang digunakan untuk memecahkan masalah yang dipaparkan pada rumusan permasalahan.

Bab IV Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab V Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab VI Pengujian dan Evaluasi

Pengujian dari penerapan cloud computing untuk meningkatkan ketersediaan server di ITS menggunakan teknologi kontainer akan diujikan pada Laboratorium Arsitektur dan Jaringan Komputer pada Teknik Informatika ITS.

Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Teori-teori tersebut meliputi pengertian dan beberapa analisis terkait penggunaan kontainer dan pembuatan aplikasi berbasis *web* meningkatkan ketersediaan server.

2.1. Cloud Computing

Cloud computing adalah teknologi dimana pengguna menggunakan jasa komputasi tanpa perlu tahu dimana letak server itu berada. Terdapat 5 karakteristik sehingga sistem tersebut disebut *Cloud Computing* [3][9], yaitu:

1. Resource Pooling

Sumber daya komputasi (*storage, CPU, memory, network bandwidth*, dan sebagainya) yang dikumpulkan oleh penyedia layanan (*service provider*) untuk memenuhi kebutuhan banyak pelanggan (*service consumers*) dengan model *multi-tenant*.

2. Broad Network Access

Kapabilitas layanan dari *cloud provider* tersedia lewat jaringan dan bisa diakses oleh berbagai jenis perangkat, seperti *smartphone, tablet, laptop, workstation*, dan sebagainya.

3. Measured Service

Tersedia layanan untuk mengoptimasi dan memonitor layanan yang dipakai secara otomatis. Dengan monitoring sistem ini, kita bisa melihat berapa sumber daya komputasi yang telah dipakai, seperti: *bandwidth, storage, processing*, jumlah pengguna aktif, dan sebagainya. Layanan monitoring ini sebagai bentuk transparansi antara *cloud provider* dan *cloud consumer*.

4. Rapid Elasticity

Kapabilitas dari layanan cloud provider bisa dipakai oleh cloud consumer secara dinamis berdasarkan kebutuhan. *Cloud consumer* bisa menaikkan atau menurunkan kapasitas layanan. Kapasitas layanan yang disediakan ini biasanya tidak terbatas, dan

service consumer bisa dengan bebas dan mudah memilih kapasitas yang diinginkan setiap saat.

5. *Self Service*

Cloud Consumer bisa mengkonfigurasi secara mandiri layanan yang ingin dipakai melalui sebuah sistem, tanpa perlu interaksi manusia dengan pihak *cloud provider*. Konfigurasi layanan yang dipilih ini harus tersedia segera dan saat itu juga secara otomatis.

2.2. Docker

Docker adalah virtualisasi menggunakan teknologi *linux container*. Docker berjalan pada lapisan kernel, sehingga memungkinkan untuk menjalankan lebih banyak virtualisasi dengan sumber daya yang terbatas. Semua virtualisasi docker berjalan pada sistem operasi yang sama dan saling berbagi sumber daya. Dengan menggunakan docker dapat mempermudah proses deployment. Docker sangat populer digunakan dalam layanan-layanan berbasis cloud [4][5].

Docker dapat dijalankan di berbagai sistem operasi, pengembang dapat dengan mudah menggunakan layanan docker melalui <https://hub.docker.com> untuk mengunduh *images* yang diinginkan. Kemudian *images* tersebut dapat dijalankan pada container docker. Terdapat 2 istilah penting dalam penggunaan docker yaitu :

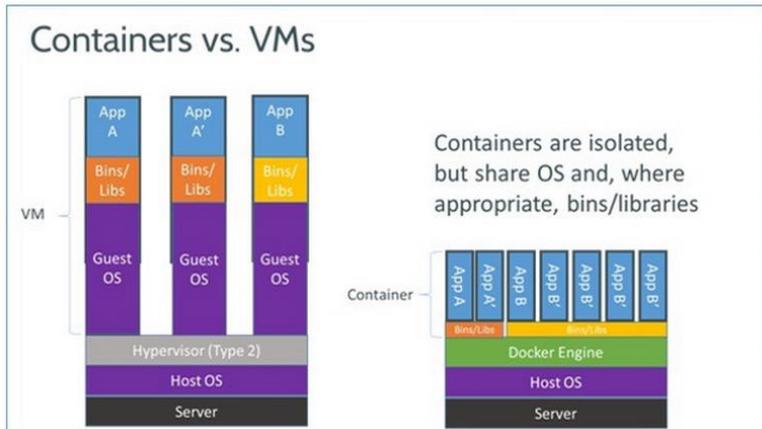
1. *Docker Daemon*

Docker daemon berfungsi untuk membangun, mendistribusikan dan menjalankan *container docker*. *User* tidak dapat langsung menggunakan *docker daemon*, akan tetapi untuk menggunakan *docker daemon* maka *user* menggunakan *docker client* sebagai perantara *CLI (Command Line Interface)*.

2. Kontainer

Kontainer merupakan alat untuk mempermudah mengemas dan mendistribusikan suatu hal dari satu tempat ke tempat lain. Sedangkan dalam konteks lingkungan *linux* (*linux environment*), kontainer dapat diartikan sebagai alat yang dapat

dipergunakan untuk memberikan sistem yang terisolasi (isolated environment) pada level sistem operasi yang dijalankan pada satu induk *linux kernel* (host).



Gambar 2.1 Perbandingan docker dan virtual machine

3. *Images*

Images adalah sebuah aplikasi beserta *library-library* pendukung yang dijadikan dalam satu wadah. Sehingga *Images* dapat berjalan pada kontainer tanpa diperlukan wadah pendukung lainnya.

2.3. **Laravel**

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. Sehingga memudahkan dalam pembuatan aplikasi ITSConcloud lebih baik.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view [11].

2.4. MySQL

MySQL adalah sistem manajemen *database SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database MySQL* mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database* manajemen sistem (*DBMS*). *Database* ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan.

MySQL dan *PHP* merupakan sistem yang saling terintegrasi. Maksudnya adalah pembuatan *database* dengan menggunakan sintak *PHP* dapat di buat. Sedangkan *input* yang di masukkan melalui aplikasi web yang menggunakan *script* serverside seperti *PHP* dapat langsung dimasukkan ke *database MySQL* yang ada di server dan tentunya web tersebut berada di sebuah web server. Berikut ini beberapa kelebihan *MySQL* sebagai *database* server antara lain :

1. *Source MySQL* dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan *database* dapat dilakukan dengan mudah.
4. *MySQL* merupakan program yang *multithreaded*, sehingga dapat dipasang pada server yang memiliki *multiCPU*.
5. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dsb.

6. Bekerja pada berbagai *platform*. (tersedia berbagai versi untuk berbagai sistem operasi).
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
8. Memiliki sistem sekuriti yang cukup baik dengan verifikasi *host*.
9. Mendukung *ODBC* untuk sistem operasi Windows.
10. Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada Bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari tugas akhir. Perancangan ini meliputi perancangan data, perancangan proses, dan perancangan antar muka, serta juga akan dijelaskan tentang analisis implementasi metode secara umum pada sistem.

3.1. Analisis Metode Secara Umum

Pada subbab berikut akan dijelaskan analisis perangkat lunak untuk pengoperasian komputer menggunakan Bahasa Indonesia. Analisis yang dilakukan meliputi analisis permasalahan, kebutuhan umum perangkat lunak, deskripsi umum sistem, arsitektur dan kebutuhan fungsional sistem.

3.1.1. Analisis Permasalahan

Permasalahan utama yang diangkat pada pembuatan Tugas Akhir ini adalah bagaimana menerapkan *cloud computing* dengan teknologi kontainer serta mengatur interaksi antara pengguna dengan aplikasi pada komputer.

Untuk membuat server *cloud computing* dengan menggunakan docker. Docker adalah virtualisasi menggunakan teknologi *linux container*. Docker berjalan pada lapisan *kernel*, sehingga memungkinkan untuk menjalankan lebih banyak virtualisasi dengan sumber daya yang terbatas

Untuk mempermudah pengguna dalam menggunakan docker maka dibuatlah aplikasi pendukung yang berbasis web yang bernama ITSConcloud. Pada aplikasi ITSConcloud terdapat fitur – fitur yang ada pada docker. Sehingga dapat memudahkan pengguna dalam menggunakannya.

Untuk menghubungkan antara aplikasi ITSConcloud dengan docker, digunakan *API* docker. *API* docker nantinya digunakan sebagai jembatan untuk jalan komunikasi antara aplikasi ITSConcloud dengan docker. *API* docker yang dibutuhkan beragam, sesuai dengan kebutuhan fitur yang akan diimplementasi.

3.1.2. Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dibuat sebuah perangkat lunak bernama ITSConcloud. ITSConcloud adalah aplikasi berbasis Web yang memanfaatkan teknologi docker untuk membuat *cloud computing* dengan teknologi kontainer. Aplikasi ITSConcloud dibuat dengan menggunakan *framework Laravel*. Alasan menggunakan *framework Laravel* karena Laravel telah menggunakan konsep MVC (*Model, View, Controller*) Sehingga memudahkan dalam pengembangan aplikasi kedepannya.

Respon yang dihasilkan dari aplikasi ini setelah menerima penggunaan oleh pengguna adalah berupa berjalannya fitur - fitur yang ada pada aplikasi ITSConcloud. Untuk berkomunikasi antara tiap fitur yang ada pada ITSConcloud dengan docker menggunakan *API* docker.

3.1.3. Spesifikasi Kebutuhan Perangkat Lunak

Subbab ini membahas spesifikasi kebutuhan perangkat lunak dari hasil analisis yang telah dilakukan. Bagian ini berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk kebutuhan fungsional dan diagram kasus penggunaan.

3.1.3.1. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan yang harus dimiliki oleh perangkat lunak, reaksi dari perangkat lunak terhadap suatu masukan, hasil yang dilakukan perangkat lunak pada situasi khusus. Kebutuhan fungsional dari perangkat lunak untuk mengoperasikan docker menggunakan aplikasi ITSConcloud pada Tabel 3.1.3.1

Tabel 3.1 Daftar Kode Kebutuhan Fungsional Aplikasi

Kode Kebutuhan Fungsional	Kebutuhan Fungsional
FR01	Mengelola User
FR02	Membuat Image
FR03	Menghapus Image
FR04	Membuat Kontainer
FR05	Memperbaharui Kontainer
FR06	Mendeploy Aplikasi
FR07	Menghapus Kontainer

3.1.3.2. Diagram Kasus Penggunaan

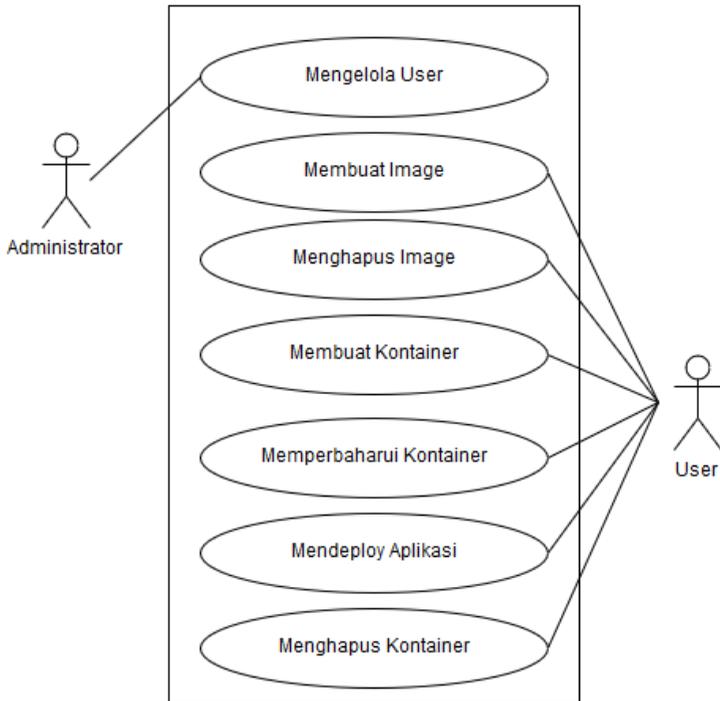
Kasus penggunaan perangkat lunak dijelaskan secara rinci dalam subbab ini. Kasus penggunaan didasarkan pada hasil analisis kebutuhan fungsional dari perangkat lunak. Terdapat tujuh kasus penggunaan pada perangkat lunak ini, sebagaimana ditunjukkan pada Gambar 3.1, yaitu kasus penggunaan untuk Mengelola User, Membuat Image, Menghapus Image, Membuat Kontainer, Memperbaharui Kontainer, Mendeploy Aplikasi, Menghapus Kontainer.

Digram kasus penggunaan pada Gambar 3.1 dideskripsikan masing-masing pada Tabel 3.2.

Tabel 3.2 Daftar Kode Kasus Penggunaan

Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
UC-0001	Mengelola <i>User</i>	<i>Administrator</i> dapat mengelola data <i>user</i> seperti menambah

		memperbaharui dan menghapus data <i>user</i>
UC-0002	Membuat <i>Image</i>	<i>User</i> dapat membuat <i>image</i> yang diinginkan
UC-0003	Menghapus <i>Image</i>	<i>User</i> dapat menghapus <i>Image</i> yang telah dibuat
UC-0004	Membuat Kontainer	<i>User</i> dapat membuat Kontainer untuk menjalankan <i>image</i> yang telah dibuat sebelumnya
UC-0005	Memperbaharui Kontainer	User dapat memperbaharui konfigurasi Kontainer.
UC-0006	Mendeploy Aplikasi	<i>User</i> dapat mengatur container seperti menjalankan, menghentikan aplikasi pada kontainer
UC-0007	Menghapus Kontainer	<i>User</i> dapat menghapus Kontainer yang telah dibuat.



Gambar 3.1 Gambar Diagram Kasus Penggunaan

3.1.3.3. Kasus Penggunaan Mengelola User

Pada kasus penggunaan mengelola *user*, administrator memilih menu list user. Kemudian administrator memilih mau menambahkan user, mengubah data user atau menghapus data user.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.3.

Tabel 3.3 Skenario Kasus Penggunaan Mengelola User

Nama Kasus	Mengelola User
-------------------	----------------

Nomor	UC-0001
Aktor	Administrator
Kondisi Awal	<ol style="list-style-type: none"> 1. Data <i>user</i> belum ada. 2. Data <i>user</i> belum diubah. 3. Data <i>user</i> belum dihapus.
Kondisi Akhir	<ol style="list-style-type: none"> 1. Data <i>user</i> berhasil dibuat. 2. Data <i>user</i> berhasil diubah. 3. Data <i>user</i> berhasil dihapus.
Alur Normal	<ol style="list-style-type: none"> 1. Administrator memilih menu list <i>user</i>. 2. Sistem menampilkan antarmuka tambah <i>user</i> dan list <i>user</i>. <ol style="list-style-type: none"> A. Tambah <i>user</i> <ol style="list-style-type: none"> 3. Administrator mengisi data pada form tambah <i>user</i>. 4. Administrator memilih tombol simpan data <i>user</i>. 5. Sistem menyimpan data <i>user</i>. B. Ubah data <i>user</i> <ol style="list-style-type: none"> 3. Administrator memilih tombol ubah data <i>user</i> pada data yang dipilih. 4. Sistem menampilkan data <i>user</i>. 5. Administrator mengubah data <i>user</i>. 6. Administrator memilih tombol simpan data <i>user</i>. 7. Sistem menyimpan data <i>user</i>. C. Hapus data <i>user</i> <ol style="list-style-type: none"> 3. Administrator memilih tombol hapus data <i>user</i> pada data yang dipilih. 4. Sistem menghapus data <i>user</i>.

3.1.3.4. Kasus Penggunaan Membuat Image

Pada kasus penggunaan membuat *image*, pengguna memilih menu upload. Kemudian pengguna mengisi form dan mengunggah file.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.4.

Tabel 3.4 Skenario kasus Penggunaan Membuat Image

Nama Kasus Penggunaan	Membuat <i>Image</i>
Nomor	UC-0002
Aktor	Pengguna
Kondisi Awal	Image belum dibuat
Kondisi Akhir	Image telah dibuat
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu upload. 2. Sistem menampilkan form tambah <i>image</i> dan <i>history upload file</i>. 3. Pengguna mengisi data form tambah <i>image</i> dan mengunggah <i>file</i>. 4. Pengguna memilih tombol unggah. 5. Sistem memproses permintaan dan menyimpan data.

3.1.3.5. Kasus Penggunaan Menghapus Image

Pada kasus penggunaan menghapus *image*, pengguna memilih menu image. Kemudian pengguna memilih tombol hapus *image* pada data yang dipilih.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.5.

Tabel 3.5 Skenario Kasus Penggunaan Menghapus Image

Nama Kasus Penggunaan	Menghapus <i>Image</i>
Nomor	UC-0003
Aktor	Pengguna
Kondisi Awal	<i>Image</i> belum dihapus.
Kondisi Akhir	<i>Image</i> berhasil dihapus
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu list image. 2. Sistem menampilkan data <i>image</i>. 3. Pengguna memilih tombol hapus <i>image</i> pada data yang dpilih. 4. Sistem memproses permintaan dan menghapus data <i>image</i>.

3.1.3.6. Kasus Penggunaan Membuat Kontainer

Pada kasus penggunaan membuat kontainer, pengguna memilih menu kontainer. Kemudian pengguna memilih *image* dan mengisi form tambah kontainer.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.6.

Tabel 3.6 Skenario Kasus Penggunaan membuat kontainer

Nama Kasus Penggunaan	Membuat Kontainer
Nomor	UC-0004
Aktor	Pengguna
Kondisi Awal	Kontainer belum dibuat

Kondisi Akhir	Kontainer berhasil dibuat
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu kontainer. 2. Sistem menampilkan form tambah kontainer dan list data kontainer. 3. Pengguna mengisi form tambah kontainer dan memilih <i>image</i>. 4. Pengguna memilih tombol simpan. 5. Sistem memproses permintaan dan menyimpan data kontainer.

3.1.3.7. Kasus Penggunaan Memperbaharui Kontainer

Pada kasus penggunaan memperbaharui kontainer, pengguna memilih menu kontainer. Kemudian pengguna memilih tombol ubah kontainer pada data yang dipilih dan perbaharui data.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.7.

Tabel 3.7 Skenario Kasus Penggunaan Memperbaharui Kontainer

Nama Kasus Penggunaan	Memperbaharui Kontainer
Nomor	UC-0005
Aktor	Pengguna
Kondisi Awal	Kontainer belum diperbaharui
Kondisi Akhir	Kontainer telah diperbaharui

Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu kontainer. 2. Sistem menampilkan form tambah kontainer dan list data kontainer. 3. Pengguna memilih tombol ubah pada kontainer yang dipilih. 4. Sistem menampilkan form ubah kontainer. 5. Pengguna mengubah data form ubah kontainer. 6. Pengguna memilih tombol simpan. 7. Sistem memproses permintaan dan menyimpan data kontainer
--------------------	--

3.1.3.8. Kasus Penggunaan Mendeploy Aplikasi

Pada kasus penggunaan mendeploy aplikasi, pengguna memilih menu kontainer. Kemudian pengguna memilih tombol start atau stop pada kontainer yang dipilih.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.8.

Tabel 3.8 Skenario Kasus Penggunaan Mendeploy Aplikasi

Nama Kasus Penggunaan	Mendeploy Aplikasi
Nomor	UC-0006
Aktor	Pengguna
Kondisi Awal	<ol style="list-style-type: none"> 1. Kontainer Berhenti. 2. Kontainer Berjalan.
Kondisi Akhir	<ol style="list-style-type: none"> 1. Kontainer Berjalan . 2. Kontainer Berhenti.

Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu kontainer. 2. Sistem menampilkan form tambah kontainer dan list data kontainer. <ol style="list-style-type: none"> A. Menjalankan Kontainer 3. Pengguna memilih tombol start pada kontainer yang dipilih dengan status stop. 4. Sistem memproses permintaan. B. Menghentikan Kontainer 3. Pengguna memilih tombol stop pada kontainer yang dipilih dengan status running. 4. Sistem memproses permintaan.
--------------------	--

3.1.3.9. Kasus Penggunaan Menghapus Kontainer

Pada kasus penggunaan menghapus kontainer, pengguna memilih menu kontainer. Kemudian pengguna memilih tombol hapus kontainer pada data yang dipilih.

Penjelasan skenario yang lebih rinci dapat dilihat pada Tabel 3.9.

Tabel 3.9 Skenario Kasus Penggunaan Menghapus Kontainer

Nama Kasus Penggunaan	Menghapus Kontainer
Nomor	UC-0007
Aktor	Pengguna
Kondisi Awal	Kontainer belum dihapus
Kondisi Akhir	Kontainer berhasil dihapus

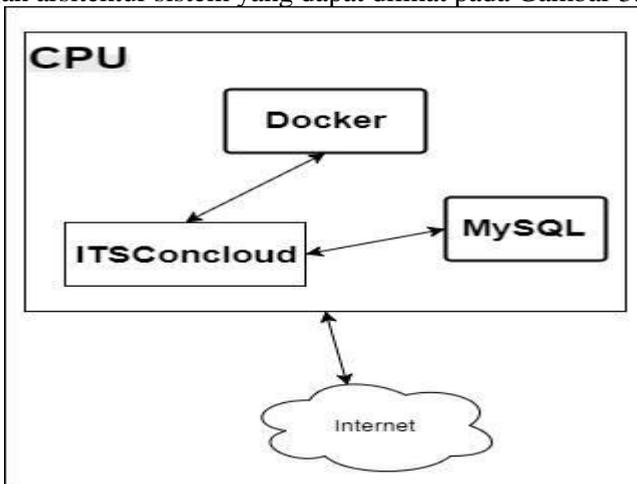
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu kontainer. 2. Sistem menampilkan data kontainer. 3. Pengguna memilih tombol hapus kontainer pada data yang dipilih. 4. Sistem memproses permintaan dan menghapus data kontainer.
--------------------	--

3.2. Perancangan Sistem

Pada subbab ini dijelaskan mengenai tahapan perancangan sistem. Perancangan sistem ini dibagi menjadi beberapa bagian yang meliputi perancangan data, perancangan arsitektur sistem, perancangan proses aplikasi, dan perancangan antarmuka pengguna.

3.2.1. Perancangan Arsitektur

Dalam tugas akhir ini akan dirancang dan diaplikasikan sebuah perangkat lunak berbasis web dengan memanfaatkan *framework laravel*. Secara garis besar aplikasi ini memiliki rancangan arsitektur sistem yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Desain Sistem Secara Umum

Aplikasi ITSConcloud mengakses data pada database untuk mengambil data. Kemudian aplikasi mengirimkan perintah kepada *docker*. Docker menjalankan perintah yang diterima dan kembali mengirimkan data kepada aplikasi.

3.2.2. Desain Umum Sistem

Sistem Cloud Computing berbasis kontainer yang akan dibangun terdiri beberapa komponen yang telah disusun. Komponen-komponen tersebut dibangun ke dalam satu sistem sehingga dapat saling terhubung. Komponen-komponen yang dibangun antara lain:

1. Docker

Aplikasi Docker adalah virtualisasi menggunakan teknologi *linux container*. Docker berjalan pada lapisan kernel, sehingga memungkinkan untuk menjalankan lebih banyak virtualisasi dengan sumber daya yang terbatas. Semua virtualisasi docker berjalan pada sistem operasi yang sama dan saling berbagi sumber daya. Dengan menggunakan docker dapat mempermudah proses *deployment*.

2. ITSConcloud

ITSConcloud adalah halaman antarmuka yang diunakan *user* untuk mengelola virtualisasi yang berjalan pada *docker*.

3. Penyimpanan Data

Penyimpanan data adalah sebuah lingkungan basis data yang berfungsi sebagai penyimpanan terpusat semua data aplikasi.

3.2.3. Desain Docker

Aplikasi *Docker* adalah virtualisasi menggunakan teknologi *linux container*. *Docker* berjalan pada lapisan kernel, sehingga memungkinkan untuk menjalankan lebih banyak virtualisasi dengan sumber daya yang terbatas. Aplikasi *docker* dipasang pada komputer server untuk mengelola kontainer yang ada. Dalam melakukan perintah dan pertukaran data dengan

Aplikasi ITSConcloud, *docker* mempunyai API sendiri sehingga aplikasi ITSConcloud mengirimkan perintah dan akan mendapatkan jawabannya dalam bentuk *json*. Daftar rute yang digunakan dalam REST API docker dijelaskan pada Tabel 3.2

Tabel 3.10 Rute pada API Docker

No	Rute	Metode	Aksi
1	/containers/json	GET	Menampilkan semua data container yang ada
2	/containers/create	POST	Membuat container dengan image yang dimaksud
3	/containers/{id}/json	GET	Menampilkan data secara detail pada salah satu container yang dimaksud
4	/containers/{id}/start	POST	Menjalankan servis container yang dimaksud
5	/containers/{id}/stop	POST	Menghentikan servis container yang dimaksud
6	/containers/{id}/attach	POST	Menjalankan proses <i>input/output</i> pada container yang dimaksud
7	/containers/{id}	DELETE	Menghapus kontainer yang dimaksud
8	/containers/{id}/update	POST	Memperbaharui kontainer yang dimaksud
9	/images/json	GET	Menampilkan semua data <i>image</i> yang ada
10	/build	POST	Membuat image dengan file <i>tar archive</i> dengan <i>Dockerfile</i> didalamnya

11	/images/{id}/json	GET	Menampilkan data secara detail pada salah satu <i>images</i> yang dimaksud
12	/images/{id}	DELETE	Menghapus <i>image</i> yang dimaksud

3.2.4. ITSConcloud

ITSConcloud adalah halaman yang digunakan sistem administrator dan user untuk mengelola dan menampilkan data docker. ITSConcloud adalah aplikasi berbasis web yang dibangun menggunakan Apache sebagai web server dan menggunakan framework Laravel 5.3 dalam pembuatannya. Material untuk mendapatkan tampilan yang sederhana dan nyaman digunakan. ITSConcloud digunakan oleh sistem administrator dan user untuk berinteraksi dengan sistem. ITSConcloud memiliki menu-menu yang memudahkan pengelolaan sistem. Menu-menu yang terdapat pada ITSConcloud antara lain:

- Kelola User

Menu yang digunakan sistem administrator untuk mengelola penyimpanan data user. Pengelolaan antara lain membuat, mengedit dan menghapus penyimpanan data user.

- Membuat *Image*

Menu yang digunakan *User* untuk dapat membuat *image* yang diinginkan Definisi yang dikelola terkait nama *image* dan *file upload*.

- Menghapus *Image*

Menu yang digunakan *User* untuk dapat menghapus *Image* yang telah dibuat. *Image* yang akan dihapus sesuai dengan data yang dipilih.

- Membuat Kontainer

Menu yang digunakan *User* untuk dapat membuat kontainer yang diinginkan Definisi yang dikelola terkait

nama container, image yang dipilih dan atribut pendukungnya.

- Memperbaharui Kontainer

Menu yang digunakan *User* dapat memperbaharui konfigurasi pada container.

- Mendeploy Aplikasi

Menu yang digunakan *User* dapat mengatur container seperti menjalankan, menghentikan aplikasi pada container.

- Menghapus Kontainer

Menu yang digunakan *User* untuk dapat menghapus kontainer yang telah dibuat. kontainer yang akan dihapus sesuai dengan data yang di pilih.

3.2.5. Penyimpanan Data

Penyimpanan data adalah sebuah lingkungan basis data yang berfungsi sebagai penyimpanan data aplikasi ITSConcloud. Daftar tabel penyimpanan data yang digunakan dalam sistem dijelaskan pada Tabel 3.3

Tabel 3.11 Tabel yang digunakan pada aplikasi

No	Nama Tabel	Keterangan
1	User	Tabel berisi data user
2	Image	Tabel berisis data <i>image</i> yang digunakan <i>user</i>
3	Kontainer	Tabel berisis data kontainer yang digunakan <i>user</i>
4	File	Tabel berisi data <i>file upload</i> yang diunggah <i>user</i>

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah bahasa pemrograman *PHP* dengan *framework laravel*. Pembahasan dilakukan secara rinci untuk setiap komponen yang ada yaitu: Docker, ITSConcloud.

4.1. Lingkungan Implementasi

Lingkungan implementasi dan pengembangan dilakukan menggunakan komputer dengan spesifikasi Host komputer adalah Intel® Core™ i5-2450M CPU @ 2.50GHz (4 CPUs) ~2.4GHz dengan memori 8 GB di Laboratorium Arsitektur dan Jaringan Komputer, Teknik Informatika ITS. Perangkat lunak yang digunakan dalam pengembangan adalah sebagai berikut :

- Sistem Operasi Linux Ubuntu 16.04 LTS 64bit
- Editor teks nano
- Editor teks Sublime Text 2
- git versi 1.9.1 untuk pengolahan versi program
- NodeJS versi 4.2.3 untuk pengembangan aplikasi
- Angular JS versi 1.5.0
- Microsoft Word untuk pembuatan buku tugas akhir
- Postman sebagai untuk tes mengirimkan Rest API
- Peramban web Mozilla Firefox

4.2. Implementasi Docker

Docker akan dipasang pada satu server dengan alamat IP 10.151.36.109. Untuk API docker menggunakan port 4243

4.2.1. Instalasi Docker

Instalasi docker diawali dengan mengatur repository docker terlebih dahulu. Berikut langkah-langkah :

1. Mempebaharui indeks paket *apt*

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

2. *Install* paket-paket yang dibutuhkan *apt*

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

3. Tambahkan Docker's official GPG key

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

4. Tambahkan repository Docker

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

Setelah menambahkan repository docker dilanjutkan dengan proses instalasi dengan perintah

1. Mempebaharui indeks paket apt

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

2. Install Docker

```
$ sudo apt-get install docker-ce
```

4.2.2. Konfigurasi Pada Docker

Pada tahapan ini akan dijelaskan konfigurasi konfigurasi pada docker sebelum digunakan aplikasi ITSConcloud.

4.2.2.1. Membuat Docker Sebagai Non-Root User

Docker daemon menggunakan Unix socket untuk menggunakan TCP port. Unix socket dimiliki user root sehingga aplikasi lain jika menggunakan harus memakai perintah sudo. Untuk menghilangkan penggunaan sudo maka membuat Unix group dan tambahkan users kedalamnya. Berikut perintah membuatnya:

1. Membuat *docker* group

```
$ sudo groupadd docker
```

2. Menambahkan user kedalam grup *docker*

```
$ sudo usermod -aG docker $USER
```

4.2.2.2. Mengatur Jaringan Bridge Docker

Untuk mengatur jaringan container, secara pengaturan awal docker menyiapkan bridge network dengan range ip default. Jika ingin membuat bridge network sendiri dalam jaringan docker maka ikuti perintah seperti dibawah ini

```
$ docker network create --driver bridge --subnet \ 10.151.36.80/28
nama_newbridge
```

Penjelasan perintah tersebut --driver untuk menentukan jenis driver bridge network, --subnet untuk menentukan subnet yang diinginkan, dan nama_newbridge adalah nama bridge ang baru dibuat dan bias diganti dengan yang lain. Agar container yang dibuat jalan pada bridge network maka saat pembuatan container perlu dimasukkan pegaturan tambahan untuk terkoneksi.

4.2.2.3. Mengatur API Docker

Sebelum menggunakan API Docker, perlu mengatur beberapa hal sampai bisa digunakan. Hal pertama ang diperlukan adalh mengubah file `/etc/init/docker.conf` dan ubahlah pada variable `DOCKER_OPTS` dengan tulisan seperti:

```
DOCKER_OPTS='-H tcp://0.0.0.0:4243 -H
unix:///var/run/docker.sock'
```

Kemudian simpan file tersebut dan kemudia *restart* docker sehingga bias digunakan.

```
service docker restart
```

API docker bias diakses dengan alamat IP server docker berada disertai port 4243.

4.2.3. Perintah API Docker

API docker digunakan berkomunikasi antara aplikasi ITSConcloud dengan docker. Berikut API yang digunakan dalam aplikasi:

1. API Pengelolaan Image Docker

Tabel 4.1 Rute API Docker Bagian Pengelolaan Image

No	Rute	Metode	Parameter
1	/images/json	GET	-
2	/build	POST	Nama <i>image</i> <i>File upload</i>
3	/images/{id}/json	GET	-
4	/images/{id}	DELETE	-

2. API Pengelolaan Kontainer Docker

Tabel 4.2 Rute API Docker Bagian Pengelolaan Kontainer

No	Rute	Metode	Parameter
1	/containers/json	GET	-
2	/containers/create	POST	<i>Image</i> Nama kontainer <i>Port host</i> <i>Port expose</i> <i>Memory ram</i> <i>environment</i>
3	/containers/{id}/json	GET	-
4	/containers/{id}/start	POST	-
5	/containers/{id}/stop	POST	-
6	/containers/{id}/attach	POST	-
7	/containers/{id}	DELETE	-
8	/containers/{id}/update	POST	<i>Memory ram</i>

4.3. Implementasi ITSConcloud

ITSConcloud diimplementasikan menggunakan PHP dan *framework* Laravel. ITSConcloud memiliki menu-menu sebagai berikut:

- Mengelola *User*
- Membuat *Image*
- Menghapus *Image*
- Membuat Kontainer

- Memperbaharui Kontainer
- Mendeploy Aplikasi
- Menghapus Kontainer

Masing-masing selanjutnya akan dijelaskan secara rinci.

4.3.1. Mengelola User

Mengelola *User* adalah menu yang digunakan untuk mengelola data user. Fitur yang didukung antara lain adalah membuat *user* baru, mengubah data *user*, dan menghapus data *user*. Antarmuka Mengelola *User* ditunjukkan pada gambar 4.1, 4.2 dan 4.3.

Tambah User

Nama *	<input type="text"/>
Tanggal Lahir *	<input type="text" value="YYYY-MM-DD"/>
Tempat Lahir *	<input type="text"/>
Alamat *	<input type="text"/>
Nomor Handphone *	<input type="text"/>
Jabatan *	<input type="text" value="admin"/>
Username *	<input type="text"/>
Password *	<input type="password"/>
Confirm Password *	<input type="password"/>

Gambar 4.1 Antarmuka menambah user

Pada gambar 4.1 ditunjukkan antarmuka menambah data *user*. Setiap data yang diisi akan disimpan ke dalam *database*

Edit Data User

Nama *	<input type="text" value="wahyu"/>
Tanggal lahir *	<input type="text" value="2017-07-12"/>
Tempat lahir *	<input type="text" value="Jakarta"/>
Alamat *	<input type="text" value="Jakaerta"/>
Nomor handphone *	<input type="text" value="0819"/>
Jabatan *	<input type="text" value="petugas"/>
Username *	<input type="text" value="wahyu"/>
	<input type="button" value="Cancel"/> <input type="button" value="Submit"/>

Gambar 4.2 Antarmuka Mengubah Data User

User									
No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan	
1	Admin	2017-06-18	Admin	Admin	0	admin	admin		
2	wahyu	1999-03-17	Jakarta	Jakaerta	2	petugas	wahyu		
3	tata	2017-06-13	Jakarta	Jakaerta	33332	petugas	tata		

Showing 1 to 3 of 3 entries

Previous **1** Next

Gambar 4.3 Antarmuka List User

4.3.2. Membuat Image

Membuat *Image* adalah menu yang digunakan untuk membuat *image*. Fitur yang didukung antara lain adalah membuat *Image* dan melihat *log* pembuatan *image*. Antarmuka Membuat *Image* ditunjukkan pada gambar 4.4.

Gambar 4.4 Antarmuka Membuat Image

4.3.3. Menghapus Image

Menghapus *Image* adalah menu yang digunakan untuk menghapus image. Fitur yang didukung antara lain adalah melihat list *Image* dan menghapus *image*. Antarmuka Menghapus *Image* ditunjukkan pada gambar 4.5.

No	Nama Image	Status	Created	Size	Action
1	hello	Success	2017-06-21 05:26:04	278.98 MB	

Gambar 4.5 Antarmuka Menghapus Image

4.3.4. Membuat Kontainer

Membuat kontainer adalah menu yang digunakan untuk membuat kontainer. Fitur yang didukung antara lain adalah membuat kontainer lalu menjalankannya. Antarmuka Membuat kontainer ditunjukkan pada gambar 4.6.

Tambah Container

Nama Image *

Nama Aplikasi *

Port *

Memory *

Environment

Gambar 4.6 Membuat Kontainer

4.3.5. Mendeploy Aplikasi

Mendeploy Aplikasi adalah menu yang digunakan untuk membuat kontainer berjalan atau berhenti. Fitur yang didukung antara lain adalah membuat container berhenti atau berjalan. Antarmuka Mendeploy aplikasi kontainer ditunjukkan pada gambar 4.7.

Container

CSV Excel Print Search:

No ^A	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b	172.17.0.15	65536 KB	Running	64 MB	<input type="button" value="stop"/> <input type="button" value="refresh"/>
2	aplikasi_a		65536 KB	Stop	64 MB	<input type="button" value="start"/> <input type="button" value="refresh"/> <input type="button" value="delete"/>

Showing 1 to 2 of 2 entries

Gambar 4.7 Mendeploy Aplikasi dan Menghapus Kontainer

4.3.6. Menghapus Kontainer

Menghapus kontainer adalah menu yang digunakan untuk menghapus kontainer yang berhenti. Fitur yang didukung antara lain adalah menghapus kontainer yang berhenti. Antarmuka menghapus aplikasi kontainer ditunjukkan pada gambar 4.7

4.3.7. Memperbaharui Kontainer

Memperbaharui kontainer adalah menu yang digunakan untuk memperbaharui kontainer berjalan atau berhenti. Fitur yang didukung antara lain adalah memperbaharui hapus kontainer yang berhenti. Antarmuka menghapus aplikasi kontainer ditunjukkan pada gambar 4.8

Ubah Container

Gambar 4.8

4.4. Penyimpanan Data

Untuk melakukan proses aplikasi ITSConcloud memerlukan media penyimpanan. Media penyimpanan yang paling tepat adalah basis data relasional salah satunya yaitu Mysql. Mysql server yang digunakan adalah versi 5.5. Mysql server memiliki definisi tabel user, image, Kontainer dan file. Tabel user digunakan untuk proses autentikasi. Berikut definisi tabel user pada tabel 4.3

Tabel 4.3 Tabel User

No	Kolom	Tipe	Keterangan
1	id	Int	Sebagai primary key pada tabel, nilai awal adalah AUTO_INCREMENT.
2	username	varchar(255)	username user
3	password	varchar(255)	password user dalam bentuk hash
4	nama	varchar(255)	Nama user
5	alamat	varchar(255)	Alamat user
6	telepon	varchar(255)	No telepon user

7	Tanggal lahir	Datetime	Tanggal lahir user
8	Tempat lahir	varchar(255)	Tempat lahir user
9	jabatan	varchar(255)	Hak akses user
10	Remember_to ken	varchar(255)	Session user saat login
11	Created_at	Datetime	Waktu pembuatan penyimpanan data.
12	Updated_at	Datetime	waktu penyimpanan data di perbarui.
13	Deleted_at	Datetime	waktu penyimpanan data dihapus, nilai awal adalah null.

Tabel image digunakan untuk proses penyimpanan data. Berikut definisi tabel image pada tabel 4.4

Tabel 4.4 Tabel Image

No	Kolom	Tipe	Keterangan
1	id	Int	Sebagai primary key pada tabel, nilai awal adalah AUTO_INCREMENT.
2	Id_image	varchar(255)	Id Image
3	Id_user	varchar(255)	Id user
4	Created_at	Datetime	Waktu pembuatan penyimpanan data.
5	Updated_at	Datetime	waktu penyimpanan data di perbarui.
6	Deleted_at	Datetime	waktu penyimpanan data dihapus, nilai awal adalah null.

Tabel container digunakan untuk proses penyimpanan data. Berikut definisi tabel container pada tabel 4.5

Tabel 4.5 Tabel Container

No	Kolom	Tipe	Keterangan
1	id	Int	Sebagai primary key pada tabel, nilai awal adalah AUTO_INCREMENT.
2	Id_con	varchar(255)	Id containers
3	Id_user	varchar(255)	Id user
4	Created_at	Datetime	Waktu pembuatan penyimpanan data.
5	Updated_at	Datetime	waktu penyimpanan data di perbarui.
6	Deleted_at	Datetime	waktu penyimpanan data dihapus, nilai awal adalah null.

Tabel file digunakan untuk proses penyimpanan data. Berikut definisi tabel file pada tabel 4.6

Tabel 4.6 Tabel file

No	Kolom	Tipe	Keterangan
1	id	Int	Sebagai primary key pada tabel, nilai awal adalah AUTO_INCREMENT.
2	Nama_file	varchar(255)	Id containers
3	Path_file	varchar(255)	Id user
4	Path_log	Datetime	Waktu pembuatan penyimpanan data.
5	Nama_Image	Datetime	waktu penyimpanan data di perbarui.
6	Id_user	varchar(255)	Id user
7	Size	Int	Ukuran file
8	Jenis_file	varchar(255)	Format file yang di unggah
9	Created_at	Datetime	Waktu pembuatan penyimpanan data.

10	Updated_at	Datetime	waktu penyimpanan data di perbarui.
11	Deleted_at	Datetime	waktu penyimpanan data dihapus, nilai awal adalah null.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi sistem Deteksi Mobil

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kaku sebagai berikut:

Prosesor : Prosesor: Intel® Core™ i5-2450M CPU
@ 2.50GHz (4 CPUs) ~2.4GHz
RAM : 8192 MB
Jenis *Device* : *Computer*
Sistem Operasi : Ubuntu 16.04 64bit

5.2. Skenario Uji Coba

Uji coba akan dilakukan untuk mengetahui keberhasilan sistem yang telah dibangun. Skenario pengujian dibedakan menjadi 2 bagian yaitu :

- Uji Fungsionalitas.

Pengujian ini didasarkan pada fungsionalitas yang disajikan sistem.

- Uji Performa.

Pengujian ini untuk menguji ketahanan sistem terhadap sejumlah permintaan yang masuk.

5.2.1. Skenario Uji Fungsionalitas Aplikasi ITSConcloud

Aplikasi ITSConcloud digunakan untuk mengelola container pada docker. Aplikasi ITSConcloud memiliki 6 menu yaitu Mengelola User, Membuat Image Menghapus Image, Membuat Kontainer, Memperbaharui Kontainer, Mendeploy

Aplikasi, Menghapus Kontainer. Rancangan pengujian dan hasil yang diharapkan ditunjukkan dengan tabel 5.1

Tabel 5.1 Skenario Uji Fungsionaitas Aplikasi ITSConcloud

No	Menu	Uji Coba	Hasil Harapan
1	Mengelola <i>User</i>	Membuat Data <i>User</i>	Membuat data user baru berhasil dibuat pada mysql.
		Memperbaharui Data <i>User</i>	Data user pada mysql berhasil diperbarui
		Menghapus Data <i>User</i>	Data user pada mysql berhasil dihapus
2	Membuat <i>Image</i>	Membuat Image	Image berhasil ditambahkan pada mysql
		Melihat Logs	Dapat melihat logs pembuatan Image
3	Menghapus Image	Melihat Image	Dapat menampilkan data image
		Menghapus Image	Data Image pada mysql berhasil dihapus
4	Membuat Kontainer	Membuat Kontainer	Kontainer berhasil ditambahkan pada mysql
5	Memperbaharui Kontainer	Memperbaharui Kontainer	Kontainer dapat diperbaharui
6	Mendeploy Aplikasi	Menghentikan Kontainer	Dapat Menghentikan proses Kontainer
		Menjalankan Kontainer	Dapat menjalankan proses container
7	Menghapus Kontainer	Menghapus Kontainer	Data Container pada mysql berhasil dihapus

5.2.2. Skenario Uji Performa Aplikasi ITSConcloud

Pada pengujian ini dilakukan untuk menguji ketahanan sistem terhadap sejumlah permintaan yang masuk pada aplikasi ITSConcloud. Terdapat 3 hal yang akan dibandingkan yaitu waktu membuat *image* dan jumlah kontainer yang dapat dibuat pada server implementasi.

5.2.2.1. Uji Performa Waktu membuat Image

Pada pengujian ini dilakukan dengan menghitung waktu yang dibutuhkan aplikasi ITSConcloud dalam melakukan permintaan pembuatan *image*. Uji coba dilakukan dengan membuat 1 *image* dan 10 *image* secara bersamaan. Uji coba pengambilan data dilakukan masing-masing sebanyak lima kali dan diambil rata-ratanya.

5.2.2.2. Uji Performa Jumlah Kontainer yang Berjalan

Pada pengujian ini dilakukan dengan menghitung jumlah kontainer yang dapat berjalan pada server aplikasi ITSConcloud. Uji coba dilakukan dengan membuat 1 kontainer yang berjalan dengan RAM masing 512 MB. Uji coba pembuatan kontainer dilakukan sampai tidak bisa menjalankan kontainer yang baru dibuat.

5.3. Hasil Uji Coba dan Evaluasi

Berikut dijelaskan hasil uji coba dan evaluasi berdasarkan skenario yang sudah dijelaskan pada bab 5.2.

5.3.1. Uji Fungsionalitas Aplikasi ITS Concloud

Sesuai dengan skenario pengujian yang akan dilakukan pada aplikasi ITSConcloud. Pengujian dilakukan dengan menguji setiap menu pada aplikasi ITSConcloud. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu:

1. Skenario pengujian 1: Membuat Data *User*.

2. Skenario pengujian 2: Memperbaharui Data *User*.
3. Skenario pengujian 3 : Menghapus Data *User*.
4. Skenario pengujian 4 : Membuat *Image*.
5. Skenario pengujian 5 : Melihat *Logs*.
6. Skenario pengujian 6 : Melihat *Image*.
7. Skenario pengujian 7 : Menghapus *Image*.
8. Skenario pengujian 8 : Membuat Kontainer.
9. Skenario pengujian 9 : Memperbaharui Kontainer.
10. Skenario pengujian 10 : Menghentikan Kontainer.
11. Skenario pengujian 11 : Menjalankan Kontainer.
12. Skenario pengujian 12 : Menghapus Kontainer.

5.3.1.1. Skenario pengujian 1: Membuat Data *User*

Pengujian membuat data *user* merupakan pengujian terhadap kemampuan aplikasi untuk menambah data *user* dan menyimpan kedalam sistem yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan menambah data *user*.

No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan
1	Admin	2017-07-20	Admin	Admin	0	admin	admin	 
2	wahyu	2017-07-12	Jakarta	Jakaerta	0819	petugas	wahyu	 

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.1 Data Awal User

Pada gambar 5.1 terlihat ada 2 data *user*. Kemudian pengguna ingin menambah data *user* maka pengguna mengisi data pada form tambah *user* gambar 5.2. Setelah selesai pengguna memilih tombol submit, maka list *user* akan bertambah sesuai dengan gambar 5.3. Dengan ini uji fungsionalitas menambah *user* berhasil diuji.

Tambah User

Nama *	<input type="text" value="UjiFungsi"/>
Tanggal Lahir *	<input type="text" value="2017-07-02"/>
Tempat Lahir *	<input type="text" value="Jakarta"/>
Alamat *	<input type="text" value="Jakarta"/>
Nomor Handphone *	<input type="text" value="0"/>
Jabatan *	<input type="text" value="petugas"/>
Username *	<input type="text" value="ujifungsi"/>
Password *	<input type="password" value="..."/>
Confirm Password *	<input type="password" value="..."/>
<input type="button" value="Submit"/>	

Gambar 5.2 Menambah Data User

User

No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan
1	Admin	2017-07-20	Admin	Admin	0	admin	admin	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	wahyu	2017-07-12	Jakarta	Jakarta	0819	petugas	wahyu	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	UjiFungsi	2017-07-02	Jakarta	Jakarta	0	petugas	ujifungsi	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 3 of 3 entries

Previous **1** Next

Gambar 5.3 Data Akhir User

5.3.1.2. Skenario pengujian 2: Memperbaharui Data User

Pengujian memperbaharui data *user* merupakan pengujian terhadap kemampuan aplikasi untuk mengubah data *user* dan menyimpan kedalam sistem yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan mengubah data *user*.

Pada gambar 5.4 terlihat ada data *user*. Kemudian pengguna ingin mengubah data *user* ketiga maka pengguna memilih tombol ubah user warna biru. Kemudian Pengguna

mengubah data pada form ubah data *user* gambar 5.5. Setelah selesai pengguna memilih tombol submit, maka data *user* akan berubah sesuai dengan gambar 5.6. Dengan ini uji fungsionalitas mengubah data *user* berhasil diuji.

User

No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan
1	Admin	2017-07-20	Admin	Admin	0	admin	admin	 
2	wahyu	2017-07-12	Jakarta	Jakaerta	0819	petugas	wahyu	 
3	UjiFungsi	2017-07-02	Jakarta	Jakarta	0	petugas	ujifungsi	 

Showing 1 to 3 of 3 entries

Previous 1 Next

Gambar 5.4 Data Awal User

Edit Data User

Nama *	<input type="text" value="UjiFungsi ubah"/>
Tanggal lahir *	<input type="text" value="2017-07-02"/>
Tempat lahir *	<input type="text" value="Medan"/>
Alamat *	<input type="text" value="Medan"/>
Nomor handphone *	<input type="text" value="9"/>
Jabatan *	<input type="text" value="admin"/>
Username *	<input type="text" value="ujifungsi"/>
	<input type="button" value="Cancel"/> <input type="button" value="Submit"/>

Gambar 5.5 Mengubah Data User

User

No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan
1	Admin	2017-07-20	Admin	Admin	0	admin	admin	 
2	wahyu	2017-07-12	Jakarta	Jakaerta	0819	petugas	wahyu	 
3	UjiFungsi ubah	2017-07-02	Medan	Medan	9	admin	ujifungsi	 

Showing 1 to 3 of 3 entries

Previous 1 Next

Gambar 5.6 Data Akhir User

5.3.1.3. Skenario pengujian 3 : Menghapus Data *User*

Pengujian menghapus data *user* merupakan pengujian terhadap kemampuan aplikasi untuk menghapus data *user* yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan menghapus data *user*.

Pada gambar 5.7 terlihat ada 3 data *user*. Kemudian pengguna ingin menghapus data *user* ketiga maka pengguna memilih tombol hapus data *user* warna merah. Setelah selesai maka list *user* akan berkurang sesuai dengan gambar 5.8. Dengan ini uji fungsionalitas menghapus *user* berhasil diuji.

User

No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan
1	Admin	2017-07-20	Admin	Admin	0	admin	admin	 
2	wahyu	2017-07-12	Jakarta	Jakaerta	0819	petugas	wahyu	 
3	UjiFungsi ubah	2017-07-02	Medan	Medan	9	admin	ujifungsi	 

Showing 1 to 3 of 3 entries

Previous 1 Next

Gambar 5.7 Data Awal User

User

No	Nama	Tanggal Lahir	Tempat Lahir	Alamat	No Handphone	Jabatan	Username	Keterangan
1	Admin	2017-07-20	Admin	Admin	0	admin	admin	 
2	wahyu	2017-07-12	Jakarta	Jakaerta	0819	petugas	wahyu	 

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.8 Data Akhir User

5.3.1.4. Skenario pengujian 4 : Membuat *Image*

Pengujian membuat *image* merupakan pengujian terhadap kemampuan aplikasi untuk membuat *image* yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan membuat *image*

Pada gambar 5.9 terlihat tidak ada data *image*. Kemudian pengguna ingin menambah data *image* maka pengguna mengisi data pada form *Upload Image* gambar 5.10. Setelah selesai

pengguna memilih tombol upload, maka data *image* akan bertambah sesuai dengan gambar 5.11. Dengan ini uji fungsionalitas menambah *image* berhasil diuji.

Image Aplikasi

No	Nama Image	Status	Created	Size	Action
----	------------	--------	---------	------	--------

Gambar 5.9 Data Awal Image

Upload Image

Nama Image *

Jenis File *

Browse *

Gambar 5.10 Membuat Image

Image Aplikasi

No	Nama Image	Status	Created	Size	Action
1	aplikasi	Success	2017-07-20 17:33:34	389.25 MB	<input type="button" value="🗑"/>

Gambar 5.11 Data Akhir Image

5.3.1.5. Skenario pengujian 5 : Melihat Logs

Pengujian melihat *logs* merupakan pengujian terhadap kemampuan aplikasi untuk melihat *logs* proses pembuatan *image* yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan melihat logs.

Pada gambar 5.12 terlihat ada data pembuatan *image*. Kemudian pengguna ingin melihat *logs* pembuatan *image* maka pengguna memilih tombol detail pada data pembuatan *image* untuk melihat *logs* dari pembuatan image pengguna tersebut gambar 5.13. Dengan ini uji fungsionalitas melihat *logs* berhasil diuji.

Image

No	Nama Image	Nama File	Tipe File	Tanggal	Ukuran	Action
1	aplikasi	t10.tar.gz	dockerfile	2017-07-21 00:33:33	0.38 KB	Q
2	Aplikasi	t10.tar.gz	dockerfile	2017-07-21 00:31:52	0.38 KB	Q
3	Aplikasi A	t10.tar.gz	dockerfile	2017-07-21 00:27:16	0.38 KB	Q

Showing 1 to 3 of 3 entries

Previous 1 Next

Gambar 5.12 Data Pembuatan Image

Log

LineLog

```

1 Note: Unnecessary use of -X or --request, POST is already inferred.
2 % Total % Received % Xferd Average Speed Time Time Time Current
3 Dload Upload Total Spent Left Speed
4 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0* Trying 127.0.0.1...
5 * Connected to localhost (127.0.0.1) port 4243 (#0)
6 > POST /build?t=aplikasi HTTP/1.1
7 > Host: localhost:4243
8 > User-Agent: curl/7.47.0
9 > Accept: */*
10 > Content-Type:application/tar
11 > Content-Length: 390
12 >
13 } [390 bytes data]
14 * upload completely sent off: 390 out of 390 bytes

```

Gambar 5.13 Logs Pembuatan Image

5.3.1.6. Skenario pengujian 6 : Melihat *Image*

Pengujian membuat image merupakan pengujian terhadap kemampuan aplikasi untuk membuat image yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan membuat image

Pada gambar 5.14 terlihat ada data *image*. Dengan ini uji fungsionalitas melihat *image* berhasil diuji.

Image Aplikasi

No	Nama Image	Status	Created	Size	Action
1	aplikasi	Success	2017-07-20 17:33:34	389.25 MB	

Gambar 5.14 Data Image

5.3.1.7. Skenario pengujian 7 : Menghapus *Image*

Pengujian menghapus *image* merupakan pengujian terhadap kemampuan aplikasi untuk menghapus *image* yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan menghapus *image*.

Pada gambar 5.15 terlihat ada data *image*. Kemudian pengguna ingin menghapus *image* maka pengguna memilih tombol hapus pada data *image*. Setelah selesai maka data *image* tersebut terhapus gambar 5.16. Dengan ini uji fungsionalitas menghapus *image* berhasil diuji.

Image Aplikasi

No	Nama Image	Status	Created	Size	Action
1	aplikasi	Success	2017-07-20 17:33:34	389.25 MB	

Gambar 5.15 Data Awal Image

Image Aplikasi

No	Nama Image	Status	Created	Size	Action
----	------------	--------	---------	------	--------

Gambar 5.16 Data Akhir Image

5.3.1.8. Skenario pengujian 8 : Membuat Kontainer

Pengujian membuat kontainer merupakan pengujian terhadap kemampuan aplikasi untuk membuat kontainer yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan kontainer image

Pada gambar 5.17 terlihat data kontainer. Kemudian pengguna ingin menambah kontainer maka pengguna mengisi data pada form tambah kontainer gambar 5.18. Setelah selesai pengguna memilih tombol submit, maka data kontainer akan

bertambah sesuai dengan gambar 5.19. Dengan ini uji fungsionalitas menambah kontainer berhasil diuji.

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_a	172.17.0.14	131072 KB	Running	64 MB	stop refresh

Showing 1 to 1 of 1 entries

[Previous](#)
[1](#)
[Next](#)

Gambar 5.17 Data Awal Kontainer

Tambah Container

Nama Image *

Nama Aplikasi *

Port *

Memory *

Environment

[Submit](#)

Gambar 5.18 Membuat Kontainer

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b	172.17.0.15	65536 KB	Running	64 MB	stop refresh
2	aplikasi_a		65536 KB	Stop	64 MB	start refresh delete

Showing 1 to 2 of 2 entries

[Previous](#)
[1](#)
[Next](#)

Gambar 5.19 Data Akhir Kontainer

5.3.1.9. Skenario pengujian 9 : Memperbaharui Kontainer

Pengujian memperbaharui Kontainer merupakan pengujian terhadap kemampuan aplikasi untuk mengubah konfigurasi kontainer dan menyimpan kedalam sistem yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan mengubah konfigurasi kontainer.

Pada gambar 5.20 terlihat ada data kontainer. Kemudian pengguna ingin mengubah data kontainer kedua maka pengguna memilih tombol ubah kontainer warna biru. Kemudian Pengguna mengubah data pada form ubah kontainer gambar 5.21. Setelah selesai pengguna memilih tombol submit, maka konfigurasi kontainer akan berubah sesuai dengan gambar 5.22. Dengan ini uji fungsionalitas memperbaharui kontainer berhasil diuji.

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b	172.17.0.15	65536 KB	Running	64 MB	stop ↗
2	aplikasi_a		65536 KB	Stop	64 MB	start ↗ 🗑️

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.20 Data Awal Kontainer

Ubah Container

Nama Image *

Nama Aplikasi *

Memory *

Cancel Submit

Gambar 5.21 Mengubah Konfigurasi Kontainer

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b	172.17.0.15	65536 KB	Running	64 MB	stop ↗
2	aplikasi_a		131072 KB	Stop	64 MB	start ↗ 🗑️

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.22 Data Akhir Kontainer

5.3.1.10.Skenario pengujian 10 : Menghentikan Kontainer

Pengujian menghentikan kontainer merupakan pengujian terhadap kemampuan aplikasi untuk menghentikan kontainer

yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan menghentikan kontainer.

Pada gambar 5.23 terlihat ada data kontainer. Kemudian pengguna ingin menghentikan kontainer maka pengguna memilih tombol stop pada data kontainer. Setelah selesai maka kontainer tersebut berhenti gambar 5.24. Dengan ini uji fungsionalitas menghentikan kontainer berhasil diuji.

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b	172.17.0.15	65536 KB	Running	64 MB	stop 🔗
2	aplikasi_a		131072 KB	Stop	64 MB	start 🔗 🗑️

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.23 Data Awal Kontainer

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b		65536 KB	Stop	64 MB	start 🔗 🗑️
2	aplikasi_a		131072 KB	Stop	64 MB	start 🔗 🗑️

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.24 Data Akhir Kontainer

5.3.1.11. Skenario pengujian 11 : Menjalankan Kontainer

Pengujian menjalankan kontainer merupakan pengujian terhadap kemampuan aplikasi untuk menjalankan kontainer yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan menjalankan kontainer.

Pada gambar 5.25 terlihat ada data kontainer. Kemudian pengguna ingin menjalankan kontainer maka pengguna memilih tombol start pada data kontainer. Setelah selesai maka kontainer tersebut berjalan gambar 5.26. Dengan ini uji fungsionalitas

menghentikan kontainer berhasil diuji.

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b		65536 KB	Stop	64 MB	start refresh delete
2	aplikasi_a		131072 KB	Stop	64 MB	start refresh delete

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.25 Data Awal Kontainer

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b		65536 KB	Stop	64 MB	start refresh delete
2	aplikasi_a	172.17.0.14	131072 KB	Running	64 MB	stop refresh

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.26 Data Akhir Kontainer

5.3.1.12. Skenario pengujian 12 : Menghapus Kontainer

Pengujian menghapus kontainer merupakan pengujian terhadap kemampuan aplikasi untuk menghapus kontainer yang diminta oleh pengguna. Pengujian ini dilakukan saat pengguna akan menghapus kontainer.

Pada gambar 5.27 terlihat ada data kontainer. Kemudian pengguna ingin menghapus kontainer maka pengguna memilih tombol hapus pada data kontainer. Setelah selesai maka data kontainer tersebut terhapus gambar 5.28. Dengan ini uji fungsionalitas menghapus kontainer berhasil diuji.

Container

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_b		65536 KB	Stop	64 MB	start refresh delete
2	aplikasi_a	172.17.0.14	131072 KB	Running	64 MB	stop refresh

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar 5.27 Data Awal Kontainer

Container

CSV Excel Print Search:

No	Nama Container	IP Address	Memory Limit	Status	Size	Action
1	aplikasi_a	172.17.0.14	131072 KB	Running	64 MB	stop ↗

Showing 1 to 1 of 1 entries

Previous **1** Next

Gambar 5.28 Data Akhir Kontainer

Hasil uji coba tertera pada tabel 5.2.

Tabel 5.2 Hasil uji Fungsionalitas Aplikasi ITSConcloud

No	Menu	Uji Coba	Hasil
1	Mengelola User	Membuat Data User	Membuat data user baru berhasil dibuat pada mysql.
		Memperbaharui Data User	Data user pada mysql berhasil diperbarui
		Menghapus Data User	Data user pada mysql berhasil dihapus
2	Membuat Image	Membuat Image	Image berhasil ditambahkan pada mysql
		Melihat Logs	Dapat melihat <i>logs</i> pembuatan Image
3	Menghapus Image	Melihat Image	Dapat menampilkan data <i>images</i>
		Menghapus Image	Data <i>image</i> pada mysql berhasil dihapus
4	Membuat Kontainer	Membuat Kontainer	Kontainer berhasil ditambahkan pada mysql
5	Memperbaharui Kontainer	Memperbaharui Kontainer	Kontainer berhasil diperbaharui

6	Mendeploy Aplikasi	Menghentikan Kontainer	Dapat Menghentikan proses Kontainer
		Menjalankan Kontainer	Dapat menjalankan proses container
7	Menghapus Kontainer	Menghapus Kontainer	Data Container pada mysql berhasil dihapus

5.3.2. Uji Peforma

Seperti yang sudah dijelaskan pada bab 5.2 pengujian performa dilakukan bertahap. Beberapa hal yang dibandingkan adalah waktu menampilkan data, waktu menulis banyak data dan waktu menulis dengan banyak perangkat.

5.3.2.1. Waktu Membuat Image

Sesuai dengan skenario pengujian yang akan dilakukan pada aplikasi ITSConcloud. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan membuat *image* pada aplikasi ITSConcloud. Hasil uji coba tertera pada tabel 5.3.

Tabel 5.3 Hasil uji Peforma Waktu Membuat Image

No	Jumlah Permintaan	Hasil
1	1 Permintaan	2 Menit 43 Detik
2	10 Permintaan	4 Menit 52 Detik

5.3.2.2. Jumlah Kontainer yang Berjalan

Sesuai dengan skenario pengujian yang akan dilakukan pada aplikasi ITSConcloud. Pengujian dilakukan dengan menjalankan kontainer hingga mencapai batas maksimum yang bias dijalankan pada server ITSConcloud. Hasil uji coba tertera pada tabel 5.4.

Tabel 5.4 Hasil uji Peforma Kontainer yang Dapat Berjalan

No	Jumlah Kontainer	Kontainer yang dapat berjalan	RAM Server (7.7 GB)
----	------------------	-------------------------------	---------------------

1	-	-	4.4 - 4.7 GB
2	1 Kontainer	1 Kontainer	4.5 – 4.8 GB
3	5 Kontainer	5 Kontainer	5.1 – 5.5 GB
4	10 Kontainer	8-9 Kontainer	5.7 – 6.4 GB

Masing - Masing kontainer membutuhkan sumber daya *ram* sekitar 200-300 MB. Server mulai mengalami gangguan saat menjalankan kontainer yang kesembilan. Hal ini disebabkan oleh penggunaan *ram* yang mulai mencapai batas dari server tersebut.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang dapat diambil dari tujuan pembuatan sistem dan hubungannya dengan hasil uji coba dan evaluasi yang telah dilakukan. Selain itu, terdapat beberapa saran yang bisa dijadikan acuan untuk melakukan pengembangan dan penelitian lebih lanjut.

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Penggunaan Docker dalam tugas akhir ini serta dari hasil uji peforma dapat menjalankan banyak aplikasi yang berjalan pada container server.
2. Pembuatan aplikasi ITSConcloud sebagai antarmuka dari penggunaan docker serta manajemen system menjadi implementasi dari desain *hosting* aplikasi.
3. Aplikasi yang akan dijalankan pada server ITSConcloud dibuat dalam bentuk *image* terlebih dahulu. *Images* adalah sebuah aplikasi beserta *library-library* pendukung yang dijadikan dalam satu wadah.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Dari segi Aplikasi masih perlu diperbaiki dalam hal validasi data masukan oleh *user*.
2. Belum adanya sistem penagihan biaya user.
3. Penambahan server ITSConcloud sehingga dapat meningkatkan layanan *cloud computing*.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] S. P. A. W. a. E. W. R. Fisher, "Image Analysys - Connected Component Labelling," School of Informatics The University of Edinburg, 2003. [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm>. [Accessed 4 Juni 2016].
- [2] A. Firmansyah, "CONTAINER DAN DOCKER: EKNIK VIRTUALISASI DALAM PENGELOLAAN BANYAK APLIKASI WEB," *Jurnal Ilmiah SimanteC*, vol. IV, pp. 167-176, 2015.
- [3] C. George, "Distributed System, Concepts and Design 3rd Selection," 2001.
- [4] Docker, "Docker Docs," 2017. [Online]. Available: <http://docs.docker.com>. [Accessed 13 2 2017].
- [5] S. Goasguen, "Docker Cookbook," 2015.
- [6] A. P. J. M. B. Laud Charles Ochei, "Optimizing the Deployment of Cloud-hosted Application Components for Guaranteeing Multitenancy Isolation," *Information Society (i-Society)*, pp. 77-83, 2016.
- [7] Lxc, "Linux Container," 2017. [Online]. Available: <http://linuxcontainers.org>. [Accessed 14 2 2017].
- [8] A. V. M. R. Pamidi, "Impact of Containers on Data Center Virtualization," 2017. [Online]. Available: <http://www.itnewswire.us/Container%20s.pdf>. [Accessed 14 2 2017].
- [9] R. P. Z. M. Thomas Erl, "Cloud Computing: Concepts, Technology & Architecture," p. 528.

- [10] Vagrant, "Docker," 2017. [Online]. Available: <https://www.vagrantup.com/docs/docker/>. [Accessed 15 02 2017].
- [11] Laravel, "Laravel Documentation," [Online]. Available: <https://laravel.com/docs/5.3/>. [Accessed 14 06 2017].

LAMPIRAN A

KODE SUMBER

A.1 Kode Sumber Guzzle Function

A.1.1 Send API Method GET

Kode Sumber 1 Send API Method Get

```
function apiGET($url){
    $client = new Client();
    $data = 0;
    try{
        $result = $client->request('GET', $url);
        $body = $result->getBody();
        $data = json_decode($body->getContents());
    }
    catch(RequestException $e){

    }
    return $data;
}
```

A.1.2 Send API Method POST

Kode Sumber 2 Send API Method POST

```
function apiPOSTbody($url,$image){
    $client = new Client();
    $result = $client->post($url, [
        'json' => [
            'Hostname'=> "",
            'Domainname'=> "",
            'User'=> "",
            'AttachStdin'=> false,
            'AttachStdout'=> true,
            'AttachStderr'=> true,
            'Tty'=> false,
            'OpenStdin'=> false,
```

```

'StdinOnce'=> false,
'Env'=> [ 'FOO=bar', 'BAZ=quux' ],
'Entrypoint'=> "",
'Cmd'=> [ 'date' ],
'Image'=> $image,
'Volumes'=> [
    '/volumes/data' => new stdClass()
],
'WorkingDir'=> "",
'NetworkDisabled'=> false,
'MacAddress'=> '12:34:56:78:9a:bc',
'ExposedPorts'=>
[
    '22/tcp'=> new stdClass()
],
'StopSignal'=> 'SIGTERM',
'StopTimeout'=> 10
]
]);
$body = $result->getBody();
$data = json_decode($body->getContents());
return $data;
}

```

A.1.3 Send API Method DELETE

Kode Sumber 3 Send API Method DELETE

```

function apiDELETE($url){
    $client = new Client();
    $data = 0;
    try{
        $result = $client->request('DELETE', $url);
        $body = $result->getBody();
        $data = json_decode($body->getContents());
    }
    catch(RequestException $e){
    }
    return $data; }

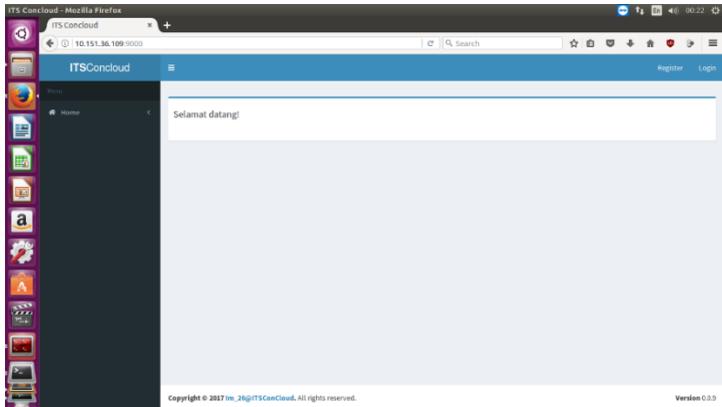
```

LAMPIRAN B

TUTORIAL PENGGUNAAN APLIKASI

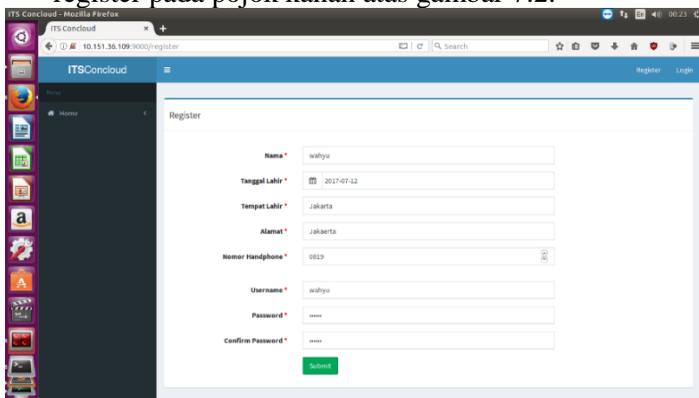
1. ITSConcloud

- Pengguna Mengakses halaman ITSConcloud gambar 7.1.



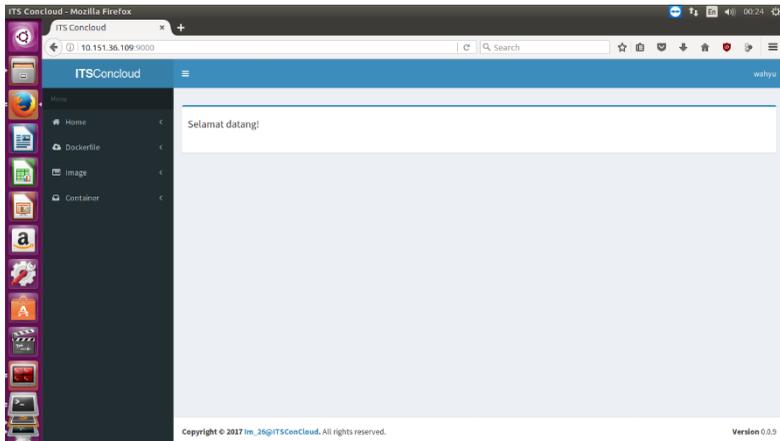
Gambar 7.1 Halaman Utama ITSConcloud

- Jika pengguna sudah mempunyai akun maka silahkan pilih menu masuk, jika tidak pilih menu register pada pojok kanan atas gambar 7.2.



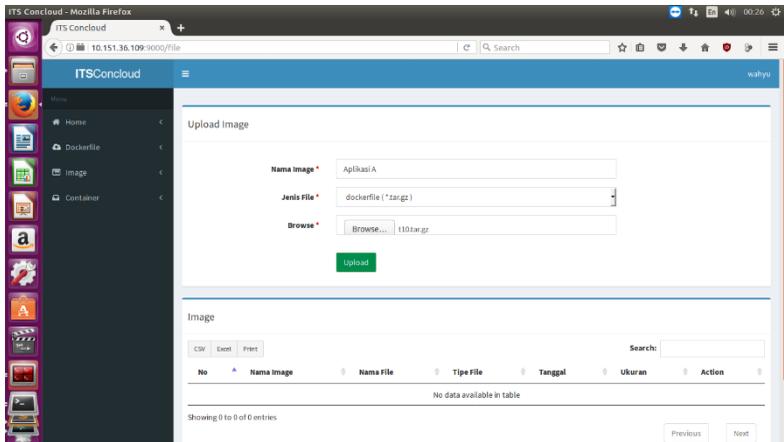
Gambar 7.2 Halaman Pendaftaran Pengguna

- Setelah pengguna login atau register maka pengguna sudah masuk ke aplikasi ITSConcloud gambar 7.3.



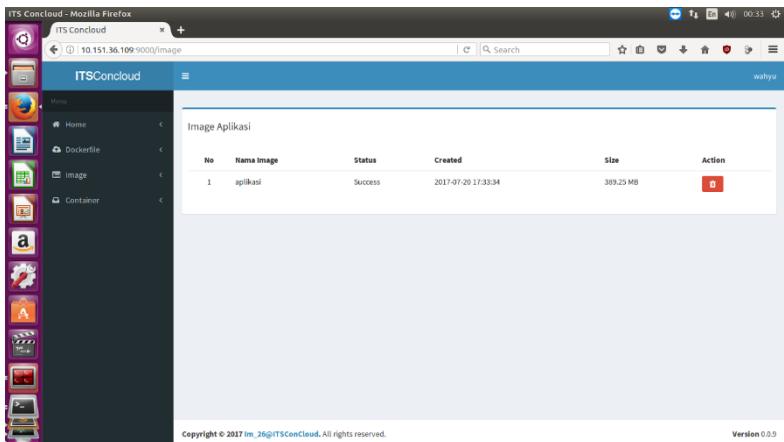
Gambar 7.3 Halaman Setelah Login

- Langkah selanjutnya pengguna membuat *images*. *Images* berisi sebuah aplikasi beserta *library-library* pendukung yang dijadikan dalam satu wadah.
- Syarat membuat *Images* adalah membuat file dengan format **.tar.gz* yang berisi *Dockerfile*, aplikasi dan berkas – berkas pendukung. *Dockerfile* adalah sebuah berkas konfigurasi docker untuk membuat aplikasi menjadi *image*.
- Setelah file siap maka bias diunggah untuk membuat images gambar 7.4.



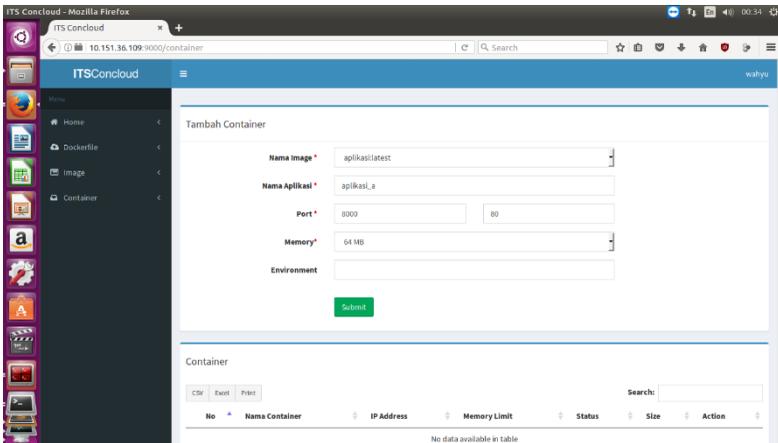
Gambar 7.4 Halaman Upload Image

- Pengguna dapat melihat *image* yang dimiliki pengguna pada menu list image gambar 7.5.



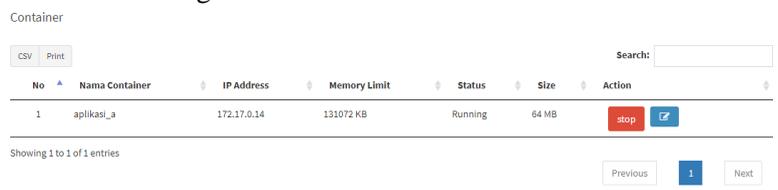
Gambar 7.5 Halaman List Image

- Pengguna menjalankan aplikasi pada menu kontainer dengan *image* yang telah dibuat gambar 7.6.



Gambar 7.6 Halaman Membuat Kontainer

- Setelah itu pengguna dapat memonitor kontainer yang telah dibuat gambar 7.7.



Gambar 7.7 Halaman List Kontainer

BIODATA PENULIS



Mohammad Wahyu Hidayat, lahir di Jakarta pada tanggal 03 April 1994. Lulusan MAS Ma'had Al-Zaytun Indramayu pada tahun 2012 dan sedang menempuh pendidikan di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya angkatan tahun 2012. Pernah menjadi asisten mata kuliah Sistem Operasi dan Jaringan Komputer. Terlibat aktif dalam organisasi kemahasiswaan diantaranya Staff Hubungan Dalam Negeri Himpunan Mahasiswa Teknik Computer-Informatika (HMTC).

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Arsitekyur dan Jaringan Komputer (AJK). Penulis dapat dihubungi melalui surel: hmwahyu@yahoo.com.