



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

RANCANG BANGUN MIDDLEWARE UNTUK MELACAK KONFIGURASI PERANGKAT JARINGAN MENGGUNAKAN GIT

MUHAMMAD FARIS DIDIN ANDIYAR
NRP 05111540000118

Dosen Pembimbing I
Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D

Dosen Pembimbing II
Bagus Jati Santoso, S.Kom., Ph.D

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2019

(Halaman ini sengaja dikosongkan)

TUGAS AKHIR - KI141502

**RANCANG BANGUN MIDDLEWARE UNTUK MELACAK
KONFIGURASI PERANGKAT JARINGAN MENGGUNAKAN
GIT**

MUHAMMAD FARIS DIDIN ANDIYAR
NRP 05111540000118

Dosen Pembimbing I
Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D

Dosen Pembimbing II
Bagus Jati Santoso, S.Kom., Ph.D

DEPARTEMENT INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2019

(Halaman ini sengaja dikosongkan)

UNDERGRADUATE THESIS - KI141502

**DESIGN OF MIDDLEWARE FOR TRACK NETWORK DEVICE
CONFIGURATION WITH GIT**

MUHAMMAD FARIS DIDIN ANDIYAR
NRP 05111540000118

Supervisor I

Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D

Supervisor II

Bagus Jati Santoso, S.Kom., Ph.D

DEPARTEMENT OF INFORMATICS

Faculty of Information Technology and Communication

Institut Teknologi Sepuluh Nopember

Surabaya, 2019

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN
RANCANG BANGUN MIDDLEWARE UNTUK
MELACAK KONFIGURASI PERANGKAT JARINGAN
MENGGUNAKAN GIT

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur Jaringan dan Komputer
Program Studi S1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

MUHAMMAD FARIS DIDIN ANDIYAR
NRP: 05111540000118

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D

.....

NIP: 197708242006041001

(Pembimbing 1)

Bagus Jati Santoso, S.Kom., Ph.D

NIP: 198611252018031001

.....

(Pembimbing 2)

SURABAYA
Juni 2019

(Halaman ini sengaja dikosongkan)

RANCANG BANGUN MIDDLEWARE UNTUK MELACAK KONFIGURASI PERANGKAT JARINGAN MENGUNAKAN GIT

Nama : MUHAMMAD FARIS DIDIN
ANDIYAR
NRP : 05111540000118
Departemen : Informatika FTIK
Pembimbing I : Royyana Muslim Ijtihadie, S.Kom.,
M.Kom., Ph.D
Pembimbing II : Bagus Jati Santoso, S.Kom., Ph.D

Abstrak

Saat ini infrastruktur jaringan semakin kompleks dan terdiri dari banyak perangkat. Seiring dengan perubahan kebutuhan maka pengaturan dari infrastruktur jaringan juga akan selalu berubah. Perangkat jaringan yang ada pada saat ini mampu menyimpan konfigurasi kedalam file dan dapat di simpan di server penyimpanan lain. Konfigurasi dari perangkat jaringan akan berubah-ubah sesuai dengan kebutuhan infrastruktur jaringan yang ada. Meskipun selalu berubah, tidak menutup kemungkinan kita ingin melihat atau menggunakan konfigurasi yang sudah lama.

(VCS) Version Control System merupakan cara yang saat ini umum digunakan untuk mencatat setiap perubahan yang ada pada file sehingga kita dapat melacak perubahan yang ada. VCS akan menyimpan setiap perubahan yang ada pada file dan mencatatnya di dalam database repositori dalam bentuk urutan perubahan dari waktu ke waktu. Salah satu VCS yang sekarang banyak digunakan adalah Git.

Dalam tugas akhir ini akan dibuat rancangan sebuah sistem yang memungkinkan untuk membuat versioning dari setiap

konfigurasi perangkat jaringan menggunakan Git. Sistem ini bisa menyimpan catatan perubahan dari file konfigurasi perangkat jaringan ke dalam server. Jika dibutuhkan versi konfigurasi yang lama, konfigurasi bisa diambil dari catatan yang disimpan di dalam server.

Kata-Kunci: *version control system, git*

DESIGN OF MIDDLEWARE FOR TRACK NETWORK DEVICE CONFIGURATION WITH GIT

**Name : MUHAMMAD FARIS DIDIN
ANDIYAR**
NRP : 05111540000118
Department : Informatics FTIK
**Supervisor I : Royyana Muslim Ijtihadie, S.Kom.,
M.Kom., Ph.D**
Supervisor II : Bagus Jati Santoso, S.Kom., Ph.D

Abstract

Kata-Kunci: *middleware, hypervisor, vmware, proxmox, ahp, virtual machine*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamini, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **Rancang Bangun *Middleware* untuk Melacak Konfigurasi Perangkat Jaringan Menggunakan Git**. Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Departemen Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari. Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT atas anugerahnya yang tidak terkira kepada penulis dan Nabi Muhammad SAW.
2. Keluarga penulis yang selalu menyemangati.
3. Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D selaku pembimbing I yang telah membantu, membimbing dan memotivasi penulis mulai dari pengerjaan proposal hingga terselesaikannya Tugas Akhir ini.
4. Bapak Bagus Jati Santoso, S.Kom., Ph.D selaku pembimbing II yang juga telah membantu, membimbing dan memotivasi penulis mulai dari pengerjaan proposal hingga terselesaikannya Tugas Akhir ini.
5. Teman-teman *Administrator* laboratorium AJK.
6. Darlis Herumurti, S.Kom., M.Kom., selaku Kepala Departemen Informatika ITS pada masa pengerjaan Tugas Akhir, Bapak Radityo Anggoro, S.Kom., M.Sc., selaku

koordinator TA dan segenap dosen Departemen Informatika yang telah memberikan ilmu dan pengalamannya.

7. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2019

Muhammad Faris Didin A.

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	3
1.6 Metodologi	3
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur	4
1.6.3 Analisis dan Desain Perangkat Lunak	4
1.6.4 Implementasi Perangkat Lunak	4
1.6.5 Pengujian dan Evaluasi	4
1.6.6 Penyusunan Buku Tugas Akhir	5
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Python	7
2.2 Flask	7
2.3 Gitpython	8
2.4 Python Watchdog	8

2.5	TFTP	8
2.6	FTP	9
BAB III DESAIN DAN PERANCANGAN		11
3.1	Deskripsi Umum Sistem	11
3.2	Kasus Penggunaan	11
3.3	Arsitektur Sistem	13
3.3.1	Desain Umum Sistem	14
3.3.2	Perancangan Repositori Perangkat	15
3.3.3	Perancangan <i>Repository Observer</i>	15
3.3.4	Perancangan <i>Web Service</i>	17
BAB IV IMPLEMENTASI		19
4.1	Lingkungan Implementasi	19
4.1.1	Perangkat Keras	19
4.1.2	Perangkat Lunak	19
4.2	Implementasi Repositori Perangkat	19
4.2.1	Implementasi Protokol TFTP	20
4.2.2	Implementasi Protokol FTP	20
4.3	Implementasi <i>Repository Observer</i>	22
4.4	Implementasi <i>Web Service</i>	23
4.4.1	Implementasi Gitea	23
4.4.2	Implementasi Flask API	24
BAB V PENGUJIAN DAN EVALUASI		25
5.1	Lingkungan Uji Coba	25
5.2	Skenario Uji Coba	25
5.2.1	Skenario Uji Coba Fungsionalitas	25
5.2.2	Skenario Uji Coba Performa	25
5.3	Hasil Uji Coba dan Evaluasi	25
5.3.1	Uji Fungsionalitas	25
5.3.2	Hasil Uji Performa	25

BAB VI PENUTUP	27
6.1 Kesimpulan	27
6.2 Saran	27
DAFTAR PUSTAKA	29
BAB A INSTALASI PERANGKAT LUNAK	31
BAB B KODE SUMBER	33
DAFTAR PUSTAKA	35
BIODATA PENULIS	37

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

3.1	Daftar Kode Kasus Penggunaan	12
4.1	Tabel rute <i>web service</i> pada <i>middleware</i>	24
4.2	Tabel rute <i>web service</i> pada <i>middleware</i>	24

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

3.1	Diagram Kasus Penggunaan	12
3.2	Desain Umum Sistem	14
3.3	Desain Repositori Perangkat	15
3.4	Alur Pengiriman File	16
3.5	Alur Pembuatan Branch	16
3.6	Perancangan Web <i>Service</i>	17

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

IV.1	Konfigurasi TFTP	20
IV.2	Konfigurasi direktori TFTP	20
IV.3	Aktivasi port FTP	21
IV.4	Konfigurasi file FTP	21
IV.5	Pengguna FTP	21
IV.6	Jalan ulang FTP	22
A.1	Instalasi Bahasa Go	31
A.2	Pengaturan Path	31

(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

Di dalam suatu instansi, arsitektur jaringan merupakan bagian yang sangat penting untuk menunjang kinerja dari instansi tersebut. Semakin besar suatu instansi maka arsitektur jaringan disana juga semakin kompleks. Oleh karena itu diperlukan suatu sistem yang mampu mengatur seluruh perangkat jaringan dengan mudah, sehingga administrator jaringan dapat bekerja secara maksimal.

Saat ini beberapa perangkat jaringan hanya memiliki *filesystem* untuk penyimpanan konfigurasi. Namun tidak memiliki mekanisme penyimpanan perubahan pada konfigurasi, sehingga kesulitan dalam melacak versi konfigurasi setelah melakukan banyak perubahan konfigurasi pada perangkat. Hal ini juga dialami oleh DPTSI ITS. Ketika administrator jaringan ingin melihat versi konfigurasi pada waktu tertentu maka akan kesulitan karena sulit untuk identifikasi versi perubahan yang ada.

Dalam perkembangan teknologi saat ini banyak terdapat alat untuk melacak perubahan konfigurasi yang disebut VCS (Version Control System) seperti Git, Subversion, dan Bazaar. Untuk menyelesaikan permasalahan administrator jaringan dalam melacak perubahan konfigurasi dibutuhkan VCS seperti Git untuk menyimpan perubahan konfigurasi perangkat jaringan. Git merupakan VCS yang umum digunakan oleh pengembang aplikasi[1]. Selain itu dibandingkan *file versioning* lain Git lebih cepat dalam proses penggunaannya[2].

Pada tugas akhir ini akan dibuat sebuah sistem untuk melacak perubahan file konfigurasi perangkat jaringan dalam bentuk aplikasi web yang memanfaatkan Git sebagai penyimpanan perubahan file konfigurasi.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana merancang *versioning* penyimpanan konfigurasi perangkat jaringan berbasis git?
2. Bagaimana merancang *middleware* protokol penyimpanan konfigurasi perangkat jaringan untuk *versioning* penyimpanan konfigurasi secara transparan?
3. Bagaimana merancang sistem informasi *backend* untuk administrator untuk pengelolaan versi konfigurasi?
4. Bagaimana mengimplementasi sistem *versioning* untuk perangkat jaringan di DPTSI ITS?

1.3 Batasan Masalah

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada tugas akhir ini, yaitu:

1. Perangkat jaringan yang digunakan adalah *router* dan *switch*.
2. Perangkat jaringan merupakan produk dari Cisco, Huawei, dan Mikrotik.

1.4 Tujuan

Tujuan pembuatan tugas akhir ini antara lain:

1. Membuat *versioning* konfigurasi perangkat jaringan berbasis git.

2. Membuat *middleware* untuk menjembatani penyimpanan konfigurasi perangkat jaringan.
3. Membuat sistem informasi *backend* untuk pengelolaan versi konfigurasi.
4. Membuat sistem untuk *versioning* konfigurasi perangkat jaringan di DPTSI ITS.

1.5 Manfaat

Manfaat dari pembuatan tugas akhir ini adalah mempermudah melacak versi konfigurasi perangkat jaringan.

1.6 Metodologi

Metodologi yang digunakan dalam pembuatan Tugas Akhir ini adalah sebagai berikut.

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan tugas akhir ini terdiri dari hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah pada tugas akhir, tujuan dari pembuatan tugas akhir dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.6.2 Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan referensi mengenai Git dan Python Watchdog untuk mendukung dan memastikan setiap tahap pembuatan tugas akhir sesuai dengan prosedur yang berlaku serta dapat diimplementasikan. Sumber informasi dan referensi bisa didapatkan melalui buku, jurnal, dan internet.

1.6.3 Analisis dan Desain Perangkat Lunak

Pada tahap ini dilakukan analisis dan perancangan terhadap arsitektur tugas akhir yang akan dibuat. Tahap ini merupakan tahap yang paling penting dimana segala bentuk implementasi dapat berjalan dengan baik ketika arsitektur sistem juga baik.

1.6.4 Implementasi Perangkat Lunak

Pada tahap ini dilakukan implementasi atau realisasi dari analisis dan perancangan arsitektur sistem yang sudah dibuat sebelumnya, sehingga menjadi infrastruktur yang sesuai dengan apa yang direncanakan.

1.6.5 Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian untuk mengukur performa dari sistem penyimpanan konfigurasi perangkat jaringan menggunakan arsitektur sistem yang telah dibuat. Beberapa performa yang diukur antara lain, kecepatan protokol pengiriman dan ketepatan versi dengan perubahan yang ada. Setelah dilakukan ujicoba, maka dilakukan evaluasi terhadap kinerja arsitektur sistem yang telah diimplementasikan dengan tujuan bisa diperbaiki jika ada pengembangan selanjutnya.

1.6.6 Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan buku tugas akhir yang berisi dokumentasi yang mencakup teori, konsep, implementasi dan hasil pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir secara garis besar adalah sebagai berikut :

1. Bab I. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi dan sistematika penulisan dari pembuatan tugas akhir.

2. Bab II. Tinjauan Pustaka

Bab ini berisi kajian teori atau penjelasan metode, algoritma, *library* dan *tools* yang digunakan dalam pembuatan tugas akhir ini. Kajian teori yang dimaksud berisi tentang penjelasan singkat mengenai *Python*, *Flask*, *Gitpython* dan *Python Watchdog*.

3. Bab III. Desain dan Perancangan

Bab ini berisi mengenai analisis dan perancangan arsitektur sistem yang akan diimplementasikan dalam pembuatan tugas akhir.

4. Bab IV. Implementasi

Bab ini berisi mengenai implementasi dari arsitektur sistem yang dibuat sebelumnya. Penjelasan berupa kode program dan pengaturan yang digunakan untuk implementasi arsitektur sistem.

5. Bab V. Pengujian dan Evaluasi

Bab ini berisi tentang tahapan ujicoba terhadap performa arsitektur sistem dan evaluasi terhadap sistem yang dibuat.

6. Bab VI. Penutup

Bab ini merupakan bab terakhir yang memaparkan

kesimpulan dari hasil pengujian dan evaluasi yang telah dilakukan. Pada bab ini juga terdapat saran yang ditujukan bagi pembaca yang berminat untuk melakukan pengembangan terhadap tugas akhir ini.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam tugas akhir.

8. Lampiran

Dalam lampiran terdapat kode sumber program secara keseluruhan.

BAB II

TINJAUAN PUSTAKA

2.1 Python

Python adalah bahasa pemrograman interpretatif, interaktif dan berorientasi objek. Python menggabungkan modul, pengecualian, penulisan secara dinamis, tipe data dinamis yang sangat tinggi dan kelas. Python memiliki antarmuka ke banyak *system call* dan pustaka diberbagai sistem dan dapat diperluas ke bahasa pemrograman C atau C++. Python dapat berjalan pada berbagai sistem operasi seperti Unix, Linux, Mac Os dan Windows.

Python adalah bahasa pemrograman tingkat tinggi yang dapat diterapkan pada berbagai masalah. Bahasa ini dilengkapi pustaka yang besar untuk melakukan pemrosesan *string*, protokol internet, rekayasa perangkat lunak dan antarmuka sistem operasi[3].

Dilihat dari kelebihan, bahasa pemrograman Python dapat digunakan dalam pengembangan aplikasi yang kompleks. Pada tugas akhir ini, bahasa pemrograman Python digunakan untuk pembuatan *middleware*.

2.2 Flask

Flask adalah kerangka aplikasi web Python yang ringan. Flask dirancang untuk memulai membuat web dengan cepat dan mudah, dengan kemampuan untuk membuat aplikasi web sampai tingkat yang rumit. Flask dibuat dengan terintegrasi dengan modul Werkzeug dan Jinja. Flask termasuk salah satu kerangka aplikasi web Python yang populer.

Flask didesain tidak memiliki depedensi dan tata letak kerangka aplikasi, dengan demikian pengembang memiliki kebebasan untuk mengatur kerangka aplikasinya sendiri serta menambahkan modul yang diperlukan sesuai kebutuhan. Flask

memiliki berbagai ekstensi yang dikembangkan oleh komunitas sehingga dapat menambahkan berbagai fungsi dengan mudah[4].

Flask memiliki kelebihan yaitu sangat ringan dan sangat sederhana dalam proses pengembangan. Sehingga Flask sangat cocok digunakan untuk pembuatan *HTTP Rest API*. Pada tugas akhir ini, Flask akan digunakan untuk pembuatan *HTTP Rest API*.

2.3 Gitpython

Gitpython adalah library python untuk berinteraksi dengan repositori git, dalam interaksi level tinggi seperti git-porcelain maupun level rendah seperti git-plumbing.

Gitpython menyediakan konsep dari obyek git untuk mempermudah mengakses data repositori dan juga mampu untuk mengakses repositori git secara langsung baik dengan implementasi python atau dengan menggunakan command git.[5]

2.4 Python Watchdog

Python watchdog adalah modul dari python untuk mengawasi *event* dari suatu file system .Watchdog dapat menangkap semua operasi yang terjadi di dalam direktori yang di tentukan.Watchdog bekerja secara real time dengan menggunakan thread.[6]

2.5 TFTP

Trivial File Transfer Protocol (TFTP) adalah protokol pengiriman file yang sederhana tanpa menggunakan autentikasi

pengguna. TFTP menggunakan protokol UDP dalam pengiriman data. TFTP tidak bisa digunakan untuk melihat isi dari suatu direktori, TFTP hanya bisa digunakan untuk mengirim dan menerima data.[7]

2.6 FTP

File Transfer Protocol (FTP) adalah protokol untuk mengirim file dari *server* ke *client* dengan menggunakan model *client-server*. FTP menggunakan protokol TCP dalam pengiriman *file*. FTP memiliki dua mode yaitu mode pasif dan mode aktif yang menentukan bagaimana client dan server terhubung.[8]

(Halaman ini sengaja dikosongkan)

BAB III

DESAIN DAN PERANCANGAN

Pada bab ini dibahas mengenai analisis dan perancangan sistem.

3.1 Deskripsi Umum Sistem

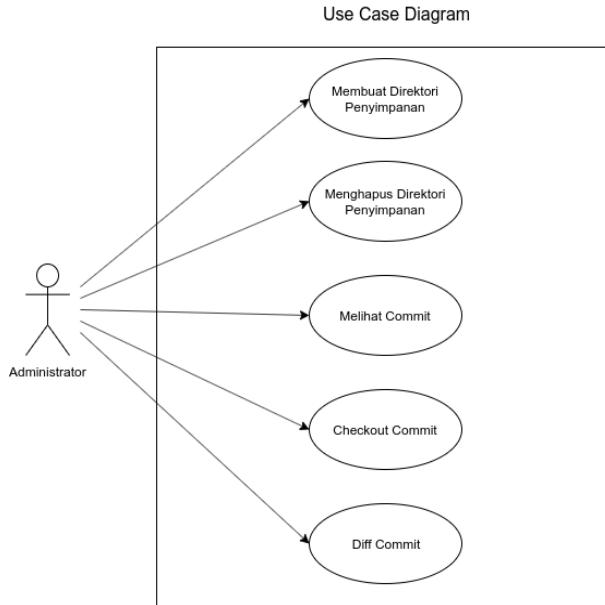
Sistem yang akan dibuat dalam tugas akhir ini adalah sistem yang digunakan untuk melacak perubahan konfigurasi perangkat jaringan. Sistem terhubung dengan perangkat jaringan dan menyimpan semua versi perubahan dari file konfigurasi perangkat jaringan. Sistem bisa mengatur versi konfigurasi yang dibutuhkan oleh perangkat jaringan untuk dipasang pada perangkat jaringan.

Sistem memiliki server repositori untuk menyimpan file konfigurasi dari perangkat jaringan yang dikirim melalui protokol TFTP dan FTP menyesuaikan protokol yang didukung oleh perangkat jaringan. Di dalam sistem terdapat *Repository Observer* yang berfungsi untuk melihat perubahan di dalam repositori. Ketika ada perubahan, sistem otomatis melakukan commit terhadap Git untuk mencatat perubahan dari file.

Di dalam sistem terdapat dua *web service* yang digunakan yaitu Gitea dan Flask. Web service tersebut digunakan untuk menerjemahkan instruksi dari administrator kepada sistem sesuai dengan diagram penggunaan pada Gambar 3.1.

3.2 Kasus Penggunaan

Dalam sistem ini hanya ada satu aktor yaitu *administrator* jaringan yang akan mengatur penyimpanan konfigurasi. Diagram kasus penggunaan digambarkan pada Gambar 3.1.



Gambar 3.1: Diagram Kasus Penggunaan

Diagram kasus penggunaan pada Gambar 3.1 dideskripsikan masing-masing pada Tabel 3.1.

Tabel 3.1: Daftar Kode Kasus Penggunaan

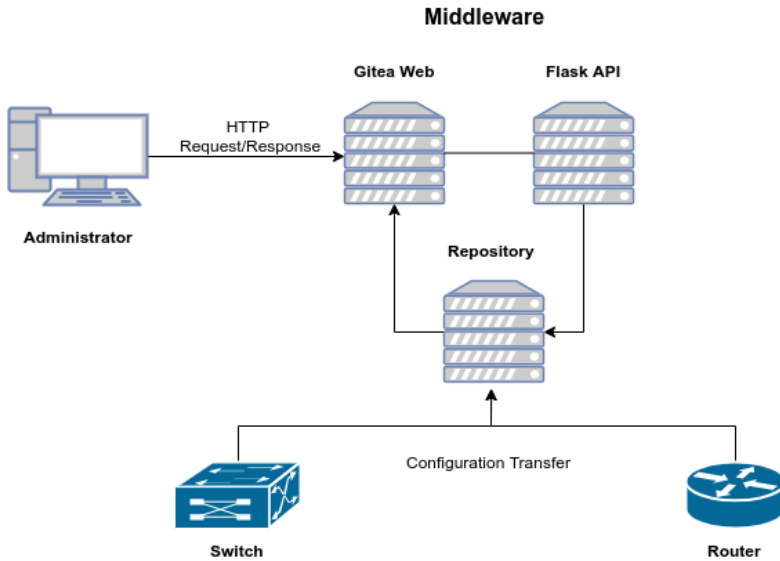
Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
UC-0001	Membuat Direktori Penyimpanan.	<i>Administrator</i> dapat membuat direktori untuk menyimpan konfigurasi dari perangkat jaringan.

Tabel 3.1: Daftar Kode Kasus Penggunaan

Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
UC-0002	Menghapus Direktori Penyimpanan.	<i>Administrator</i> dapat menghapus direktori penyimpanan jika sudah tidak digunakan.
UC-0003	Melihat Commit.	<i>Administrator</i> dapat melihat riwayat commit dalam repositori perangkat.
UC-0004	Checkout Commit.	Administrator dapat berpindah commit (checkout) sesuai dengan versi commit yang diinginkan.
UC-0005	Diff Commit.	Administrator dapat melihat perbedaan antara commit satu dengan lainnya.

3.3 Arsitektur Sistem

Pada sub-bab ini, dibahas mengenai tahap analisis dan kebutuhan bisnis dan desain dari sistem yang akan dibangun. Arsitektur sistem secara umum ditunjukkan pada Gambar 3.2.



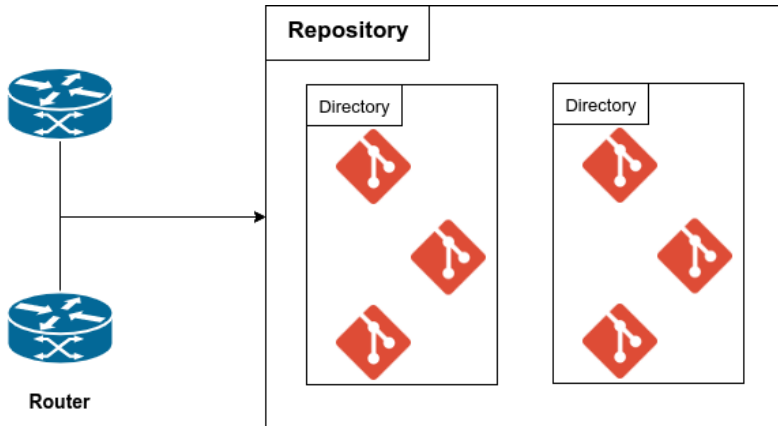
Gambar 3.2: Desain Umum Sistem

3.3.1 Desain Umum Sistem

Berdasarkan yang dijelaskan pada deskripsi umum sistem, dapat diperoleh kebutuhan sistem sebagai berikut:

1. *Repository Server* untuk menyimpan file konfigurasi dari perangkat jaringan.
2. *Repository Observer* untuk melihat perubahan file yang disimpan di dalam repository server.
3. *Web Service* untuk menerjemahkan intruksi dari admin kepada sistem.

3.3.2 Perancangan Repositori Perangkat



Gambar 3.3: Desain Repositori Perangkat

Repositori perangkat adalah komponen untuk menyimpan file konfigurasi perangkat jaringan. Repositori perangkat merupakan direktori yang menjadi tujuan pengiriman file konfigurasi perangkat jaringan. Pengiriman file konfigurasi menggunakan TFTP dan FTP. Di dalam repositori perangkat, direktori akan dibedakan berdasarkan protokol pengiriman dan nama perangkat. Setiap perangkat memiliki direktori masing-masing dan setiap direktori merupakan *Git Repository*.

3.3.3 Perancangan *Repository Observer*

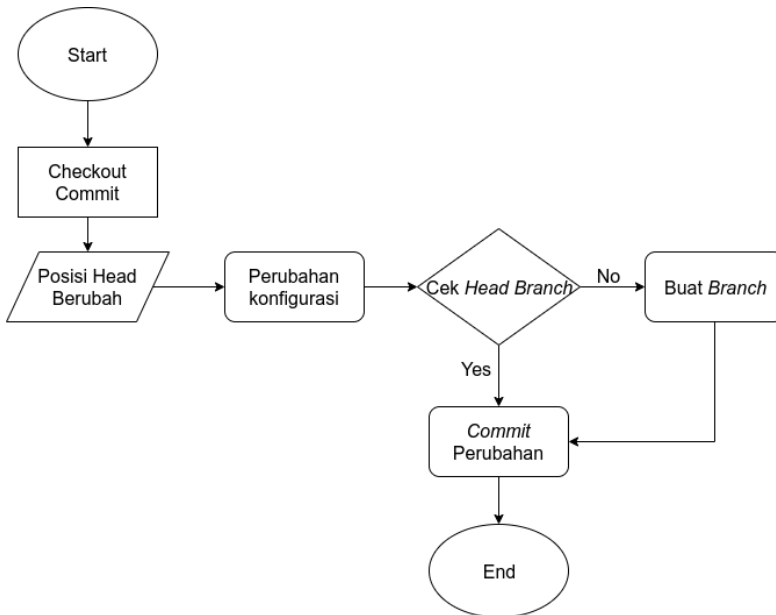
Pada sistem ini Middleware harus bisa mengamati repositori perangkat jaringan secara berkelanjutan dan melakukan update pada history commit pada repositori. Untuk melakukan hal tersebut modul watchdog di dalam middleware yang akan melihat setiap perubahan pada repositori perangkat jaringan.

Modul watchdog berjalan sebagai thread yang menunggu perubahan kondisi di dalam repositori. Ketika thread mengidentifikasi ada perubahan di dalam repositori maka thread akan menjalankan perintah commit menggunakan modul gitPython yang terintegrasi dengan middleware.



Gambar 3.4: Alur Pengiriman File

Repository Observer juga mengatur pembentukan cabang dari repositori penyimpanan konfigurasi perangkat jaringan. Alur pembuatan cabang dari repositori seperti pada gambar 3.5.

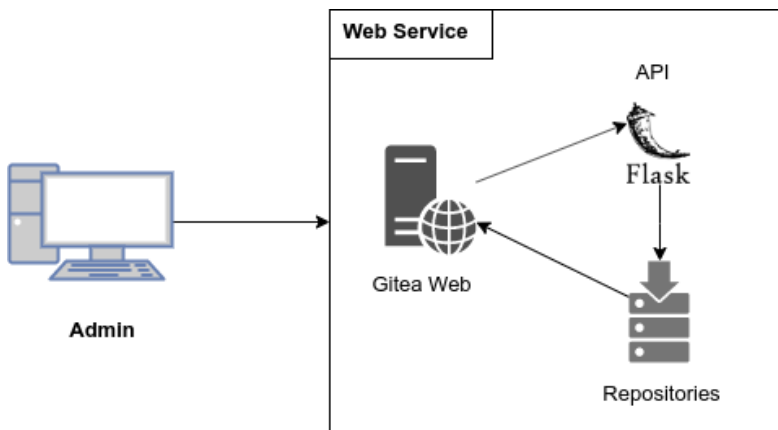


Gambar 3.5: Alur Pembuatan Branch

3.3.4 Perancangan Web Service

Dalam sistem yang dibangun, web service digunakan untuk menerjemahkan permintaan dari administrator jaringan. Web service memiliki antarmuka dan rute dengan parameter nama repositori dan permintaan fitur yang diinginkan. Setiap rute akan diproses oleh *Middleware* dan kemudian mengirimkan respon kepada administrator.

Terdapat dua web service yang digunakan dalam sistem pelacakan konfigurasi perangkat jaringan yakni Gitea webapp dan flask API. Gitea webapp digunakan untuk menampilkan user interface dari sistem tetapi dalam sistem yang akan dibuat ada beberapa fitur yang tidak disediakan oleh Gitea yaitu fitur untuk letak head commit dan fitur untuk checkout commit. Untuk mengatasi hal tersebut maka digunakan Flask API untuk untuk melihat letak head commit dan checkout commit pada repositori lokal.



Gambar 3.6: Perancangan Web Service

(Halaman ini sengaja dikosongkan)

BAB IV

IMPLEMENTASI

Pada bab ini akan dibahas implementasi dari perancangan setiap komponen sistem pada bab sebelumnya. Setiap komponen akan dibahas proses pembuatan dilengkapi dengan konfigurasi dan pseudocode dari sistem.

4.1 Lingkungan Implementasi

Dalam mengimplementasikan sistem, digunakan beberapa perangkat pendukung sebagai berikut.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah sebagai berikut:

1. Komputer dengan processor Intel Core i5-8250U @ 8x 3.4GHz dan RAM 8GB.

4.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan adalah sebagai berikut:

1. Sistem Operasi Ubuntu 18.04 LTS 64 Bit.
2. Flask versi 1.0.3 untuk pengembangan *Web Service*.
3. Git versi 2.17.1 untuk *versioning* file konfigurasi.
4. TFTP untuk protokol pengiriman file konfigurasi.
5. FTP untuk protokol pengiriman file konfigurasi.
6. GNS3 untuk emulator perangkat jaringan.

4.2 Implementasi Repositori Perangkat

Repositori perangkat mendukung dua protokol pengiriman file yaitu File Transfer Protocol (FTP) dan Trivial File Transfer Protocol (TFTP). Ada beberapa tahap agar protokol-protokol

tersebut bisa digunakan yakni pemasangan dan konfigurasi. Untuk implementasi Reposi Perangkat terdapat dua tahap yakni :

1. Implementasi direktori untuk protokol TFTP.
2. Implementasi direktori untuk protokol FTP.

4.2.1 Implementasi Protokol TFTP

Untuk melakukan pemasangan TFTP bisa dilihat di lampiran kode sumber . Setelah selesai melakukan pemasangan maka kita perlu melakukan konfigurasi TFTP pada file `/etc/default/tftpd-hpa`.

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/home/didin/repo-local/tftp"
TFTP_ADDRESS=":69"
TFTP_OPTIONS="--secure --create"
```

Kode Sumber IV.1: Konfigurasi TFTP

Setelah melakukan konfigurasi TFTP selanjutnya adalah melakukan konfigurasi pada direktori yang digunakan sebagai *root* TFTP dengan menjalankan perintah berikut.

```
chown -R didin:tftp /home/didin/repo-local/
tftp
```

Kode Sumber IV.2: Konfigurasi direktori TFTP

4.2.2 Implementasi Protokol FTP

Selanjutnya dalam implementasi Repositori perangkat adalah pemasangan dan konfigurasi FTP. Untuk melakukan pemasangan FTP jalankan dapat dilihat di lampiran kode sumber. Setelah melakukan pemasangan langkah selanjutnya adalah mengaktifkan port yang digunakan dalam FTP yakni port 20 dan 21 dengan menjalankan perintah.

```
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw status
```

Kode Sumber IV.3: Aktivasi port FTP

Tahap selanjutnya adalah mengatur konfigurasi dari FTP pada file `/etc/vsftpd.conf` dengan menulis konfigurasi berikut.

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
chroot_local_user=YES
allow_writeable_chroot=YES
local_root=/home/$USER/repo-local/ftp
```

Kode Sumber IV.4: Konfigurasi file FTP

Kemudian tambahkan nama pengguna yang punya otoritas untuk FTP di dalam file `/etc/vsftpd.userlist` dengan menjalankan perintah.

```
echo "didin" | sudo tee -a /etc/vsftpd.  
userlist
```

Kode Sumber IV.5: Pengguna FTP

Untuk menerapkan konfigurasi jalankan ulang FTP dengan menjalankan perintah.

```
systemctl restart vsftpd
```

Kode Sumber IV.6: Jalan ulang FTP

4.3 Implementasi *Repository Observer*

Middleware memiliki tugas untuk mencatat setiap perubahan yang terjadi pada file konfigurasi. Perubahan file konfigurasi terjadi ketika perangkat jaringan mengirim file konfigurasi menuju *middleware*. Untuk mengamati perubahan dalam direktori terdapat *Repository Observer* dalam bentuk thread. Untuk membuat repository observer ada beberapa tahap yang diperlukan yakni pemasangan bahasa python dan modul-modul yang diperlukan kemudian pembuatan program untuk menjalankan *thread repository observer*. Perangkat lunak yang diperlukan untuk dipasang adalah:

1. Python.
2. Python Watchdog.
3. Git Python.

Berikut pseudocode yang berjalan dalam *Repository Observer*.

Pseudocode 1: Repository observer

```

if file modified then
  if head not a branch head then
    | create new branch;
  else
    | reference head to branch;
  end
  git add;
  git commit;
  git push;
end
if checkout then
  | pause observer 2 second
end

```

4.4 Implementasi *Web Service*

Web service pada *middleware* berfungsi untuk menjembatani antara pengguna dengan *middleware*. Pengguna mengirimkan permintaan melalui rute-rute yang dimiliki *web service* kemudian permintaan diproses oleh *middleware*. Dalam sistem yang dibuat ini *web service* yang digunakan adalah Gitea yang merupakan self-hosted git dan juga Flask API.

4.4.1 Implementasi Gitea

Untuk menggunakan Gitea ada beberapa tahap yang harus dilakukan terlebih dahulu yakni :

1. Instalasi bahasa Go.
2. Instalasi database Mysql.
3. Pembuatan database gitea.
4. Clone kode sumber Gitea.

Instalasi bahasa Go dapat dilihat di A.1.

Untuk menggunakan fitur-fitur dalam sistem yang dibuat pengguna perlu mengakses rute-rute dalam *Web Service*. Berikut rute yang disediakan Gitea webapp pada Tabel 4.1.

Tabel 4.1: Tabel rute *web service* pada *middleware*

No	Rute	Method	Keterangan
1	/user/sign_up	Post	Membuat user untuk administrator.
2	/user/login	Post	Login user administrator.
3	/ {username} / {reponame}	Get	Menampilkan repositori dari user.
4	/ {username} / {reponame} / commits / branch / {namabranch}	Get	Menampilkan commit pada repositori.

4.4.2 Implementasi Flask API

Untuk melengkapi fitur yang tidak disediakan oleh Gitea maka dibuat Flask API. Berikut rute yang disediakan oleh Flask API4.2.

Tabel 4.2: Tabel rute *web service* pada *middleware*

No	Rute	Method	Keterangan
1	/create/ {protocol} / {name}	Get	Membuat repositori lokal untuk perangkat jaringan.
2	/remove/ {reponame}	Get	Menghapus repositori lokal.
3	/checkout/ {reponame} / {commit}	Get	Checkout pada commit.
4	/head/ {reponame}	Get	Melihat head saat ini.

BAB V

PENGUJIAN DAN EVALUASI

5.1 Lingkungan Uji Coba

5.2 Skenario Uji Coba

5.2.1 Skenario Uji Coba Fungsionalitas

5.2.2 Skenario Uji Coba Performa

5.3 Hasil Uji Coba dan Evaluasi

5.3.1 Uji Fungsionalitas

5.3.2 Hasil Uji Performa

(Halaman ini sengaja dikosongkan)

BAB VI

PENUTUP

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 proses perancangan, implementasi dan pengujian terhadap sistem, dapat diambil beberapa kesimpulan berikut:

1. Sistem dapat melakukan manajemen alokasi *virtual machine* pada lingkungan *hypervisor* yang heterogen. *Hypervisor* yang didukung oleh sistem adalah *Vmware Vsphere* dan *Proxmox*.
2. Sistem dapat membagi distribusi alokasi *virtual machine* baru pada *server* yang tersedia dengan algoritma *Analytical Hierarchy Process*.
3. Sistem dapat melakukan *Provisioning* sampai proses pengaturan IP berdasarkan *hypervisor* dan sistem operasi.
4. Sistem dapat diakses oleh pengguna melalui *interface web* dan *command line interface*.
5. Dari hasil pengujian performa, semakin banyak *worker* yang digunakan sangat rawan terjadinya kegagalan alokasi *virtual machine* pada *hypervisor Proxmox*.

6.2 Saran

Berikut beberapa saran yang diberikan untuk pengembangan lebih lanjut:

1. Untuk mempercepat waktu alokasi pada *hypervisor Proxmox*, diperlukan *storage area network* sebagai tempat menyimpan *file template* sistem operasi sehingga saat

alokasi *virtual machine* baru, *middleware* tidak perlu mengirimkan file *template* terlebih dahulu.

2. Untuk memperbanyak dukungan terhadap sistem operasi, untuk pengaturan IP dapat dilakukan dengan mekanisme *IP Floating*.

DAFTAR PUSTAKA

- [1] “Compare repositories,” 5 Agustus 2019. [Daring]. Tersedia pada: <https://www.openhub.net/repositories/compare>. [Diakses: 5 Agustus 2019].
- [2] “About - git,” 5 Agustus 2019. [Daring]. Tersedia pada: <https://git-scm.com/about/small-and-fast>. [Diakses: 5 Agustus 2019].
- [3] “General Python FAQ,” 3 Mei 2018. [Daring]. Tersedia pada: <https://docs.python.org/3/faq/general.html#what-is-python>. [Diakses: 3 Mei 2018].
- [4] “A simple framework for building complex web applications,” 3 Mei 2018. [Daring]. Tersedia pada: <https://pypi.org/project/Flask/>. [Diakses: 3 Mei 2018].
- [5] “GitPython overview,” 3 Mei 2019. [Daring]. Tersedia pada: <https://gitpython.readthedocs.io/en/stable/intro.html>. [Diakses: 3 Mei 2019].
- [6] “Watchdog,” 29 Agustus 2019. [Daring]. Tersedia pada: <https://pypi.org/project/watchdog/>. [Diakses: 29 Agustus 2019].
- [7] N. F. S. Pauzi, M. A. M. Isa, H. Hashim, S. F. S. Adnan, dan L. Mazalan, “Performance measurement of secure TFTP protocol for smart embedded devices,” in *2014 IEEE Asia Pacific Conference on Wireless and Mobile*, Aug. 2014.
- [8] N. Udayakumar, A. Khera, L. Suri, C. Gupta, dan T. Subbulakshmi, “Bandwidth analysis of file transfer protocol,” in *2018 International Conference on Communication and Signal Processing (ICCSP)*, April 2018.

(Halaman ini sengaja dikosongkan)

LAMPIRAN A

INSTALASI PERANGKAT LUNAK

Instalasi Protokol Pengiriman File

- TFTP
\$ sudo apt install tftp-hpa tftpd-hpa
- FTP
\$ sudo apt-get install vsftpd

Instalasi Bahasa Go

```
cd ~  
curl -O https://dl.google.com/go/go1.13.5.linux-  
amd64.tar.gz  
  
tar xvf go1.13.5.linux-amd64.tar.gz  
sudo chown -R root:root ./go  
sudo mv go /usr/local
```

Kode Sumber A.1: Instalasi Bahasa Go

Selanjutnya export path untuk bahasa Go di dalam ~/.profile

```
export GOPATH=$HOME/go  
export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin
```

Kode Sumber A.2: Pengaturan Path

Instalasi Pustaka Python

Dalam pengembangan sistem ini, digunakan berbagai pustaka pendukung. Pustaka pendukung yang digunakan merupakan pustaka untuk bahasa pemrograman Python. Berikut adalah daftar pustaka yang digunakan dan cara pemasangannya:

- Python
\$ sudo apt-get install python3

- **Flask**
`$ sudo pip3 install Flask`
- **Watchdog**
`$ sudo pip3 install watchdog`
- **Gitpython**
`$ sudo pip3 install gitpython`

LAMPIRAN B
KODE SUMBER

File Environment Middleware

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] “Compare repositories,” 5 Agustus 2019. [Daring]. Tersedia pada: <https://www.openhub.net/repositories/compare>. [Diakses: 5 Agustus 2019].
- [2] “About - git,” 5 Agustus 2019. [Daring]. Tersedia pada: <https://git-scm.com/about/small-and-fast>. [Diakses: 5 Agustus 2019].
- [3] “General Python FAQ,” 3 Mei 2018. [Daring]. Tersedia pada: <https://docs.python.org/3/faq/general.html#what-is-python>. [Diakses: 3 Mei 2018].
- [4] “A simple framework for building complex web applications,” 3 Mei 2018. [Daring]. Tersedia pada: <https://pypi.org/project/Flask/>. [Diakses: 3 Mei 2018].
- [5] “GitPython overview,” 3 Mei 2019. [Daring]. Tersedia pada: <https://gitpython.readthedocs.io/en/stable/intro.html>. [Diakses: 3 Mei 2019].
- [6] “Watchdog,” 29 Agustus 2019. [Daring]. Tersedia pada: <https://pypi.org/project/watchdog/>. [Diakses: 29 Agustus 2019].
- [7] N. F. S. Pauzi, M. A. M. Isa, H. Hashim, S. F. S. Adnan, dan L. Mazalan, “Performance measurement of secure TFTP protocol for smart embedded devices,” in *2014 IEEE Asia Pacific Conference on Wireless and Mobile*, Aug. 2014.
- [8] N. Udayakumar, A. Khera, L. Suri, C. Gupta, dan T. Subbulakshmi, “Bandwidth analysis of file transfer protocol,” in *2018 International Conference on Communication and Signal Processing (ICCSP)*, April 2018.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Muhammad



Faris Didin Andiyar, biasa lahir pada 23 April 1997 di Mojokerto. Penulis adalah mahasiswa yang sedang menjalani studi di Departemen Informatika Institut Teknologi Sepuluh Nopember. Selama menempuh studi penulis aktif di Organisasi sebagai Ketua Departemen Kaderisasi dan Pemetaan (KDPM) Himpunan Mahasiswa Teknik Computer-Informatika pada tahun ke-3. Pernah menjadi staff di kepanitiaan

Schematics. Penulis juga merupakan administrator lab Arsitektur dan Jaringan Komputer (AJK) dan juga pernah menjadi asisten mata kuliah Sistem Operasi dan Jaringan Komputer.