TUGAS AKHIR – KI141502

SISTEM AUTENTIKASI PROVISIONING JARINGAN WIRELESS MELALUI DHCP SERVER DENGAN MENGGUNAKAN LAYANAN PESAN

I GUSTI KETUT ANOM
NRP 5111100084

Dosen Pembimbing I
ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom., PhD.

Dosen Pembimbing II
HUDAN STUDIAWAN, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

UNDERGRADUATE THESES – KI141502

WIRELESS NETWORK PROVISIONING AUTHENTICATION SYSTEM VIA DHCP SERVER BY USING MESSAGE SERVICES

I GUSTI KETUT ANOM
NRP 5111100084

Supervisor I
ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom., Ph.D.

Supervisor II
HUDAN STUDIAWAN, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

LEMBAR PENGESAHAN

SISTEM AUTENTIKASI PROVISIONING JARINGAN WIRELESS MELALUI DHCP SERVER DENGAN MENGGUNAKAN LAYANAN PESAN

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Arsitektur Jaringan Komputer
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
I GUSTI KETUT ANOM
NRP : 5111 100 084

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Royyana Muslim Ijtihadi, S.Kom., M.Kom., Ph.D. .....
   NIP: 197708242006041001 (Pembimbing 1)

2. Hudan Studiawan, S.Kom., M.Kom. .....
   NIP: 198705112012121003 (Pembimbing 2)

SURABAYA

JUNI, 2016

SISTEM AUTENTIKASI PROVISIONING JARINGAN WIRELESS MELALUI DHCP SERVER DENGAN MENGGUNAKAN LAYANAN PESAN

Nama Mahasiswa	**: I Gusti Ketut Anom**
NRP	**: 5111100084**
Jurusan	**: Teknik Informatika FTIF-ITS**
Dosen Pembimbing 1	**: Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.**
Dosen Pembimbing 2	**: Hudan Studiawan, S.Kom., M.Kom.**

Abstrak

Komunikasi data secara nirkabel pada suatu jaringan sudah menjadi hal umum pada saat ini. Dengan komunikasi tersebut dapat memberikan efisiensi dan flexibelitas bagi pemilik maupun pengguna pada jaringan tersebut, tetapi tidak memungkiri akan adanya penyalahgunaan lalu lintas jaringan secara nirkabel tersebut oleh penggunanya. Tantangan yang ada sekarang ini yaitu bagaimana mengatur hak akses dari pengguna jaringan sesuai dengan keputusan yang diberikan oleh pemilik jaringan melalui layanan pesan.

Pada tugas akhir ini dibuat sebuah perangkat berupa router nirkabel dengan sistem yang dapat mengirimkan pesan notifikasi jika terdapat perangkat yang terhubung, kemudian memberi kewenangan kepada pemilik jaringan untuk mengautentikasi dan mengatur hak akses pada setiap perangkat yang hendak terhubung tersebut melalui layanan pesan.

Hasil uji coba terhadap sistem autentikasi komunikasi secara nirkabel melalui layanan pesan menunjukkan bahwa pengelolaan sebuah jaringan nirkabel melalui layanan pesan oleh pemilik jaringan lebih meningkatkan efisiensi dan keamanan pada jaringan tersebut dibandingkan tanpa menggunakan sistem ini.

Kata kunci: Autentikasi, jaringan nirkabel, router nirkabel, layanan pesan.

WIRELESS NETWORK PROVISIONING AUTHENTICATION SYSTEM VIA DHCP SERVER BY USING MESSAGE SERVICES

Student Name	**: I Gusti Ketut Anom**
NRP	**: 5111100084**
Department	**: Informatics Department, FTIF-ITS**
Supervisor 1	**: Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.**
Supervisor 2	**: Hudan Studiawan, S.Kom., M.Kom.**

Abstract

Data communication using wireless in a network is a common thing nowadays. With this communication, owner or user is provided with efficiency and flexibility, but there always a possibility of misconduct regarding wireless network traffic by its user. The challenge is, how to governance accessibility for the user given by the network owner through a message services.

In this final project created a device such as a wireless router with a system that can send a notification message if there is a connected device, then gave authority to the owner of the network to authenticate and set permissions on each device to be connected via message service.

The result of the authentication system regarding communication by wireless through message services, showing that a management of network wireless by its owner boosting its efficiency and security level for this network compared than the one which not applied this system.

Keywords: Authentication, wireless network, wireless router, message service

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Pada bab ini akan dijelaskan mengenai beberapa hal dasar dalam Tugas Akhir ini yang meliputi latar belakang, rumusan masalah, tujuan dan manfaat pembuatan Tugas Akhir, serta metodologi dan sistematika penulisan buku Tugas Akhir ini.

1.1 Latar Belakang

Sistem komunikasi berbasis IT dengan media transmisi kabel sudah beralih ke media transmisi non kabel atau yang biasa disebut *wireless*. Teknologi *Wireless* memiliki keuntungan dalam hal efisiensi, jangkauan area, flexibelitas, dan biaya dibandingkan dengan teknologi *wired*. Seperti yang diketahui bahwa perkembangan *mobile* telekomunikasi sangat berkembang pada saat ini sebagai dampak dari penggunaan teknologi *wireless*. Dalam jaringan komputer, keuntungan penggunaan teknologi *wireless* yaitu meningkatkan mobilitas, lebih baik dalam mengakses informasi, perluasan jaringan menjadi lebih mudah. Contohnya *public hotspot*, *wireless remote control*, jaringan Ad Hoc, dan jaringan infrastruktur *server* [1].

Dalam hal ini sebuah jaringan *wireless* yang terhubung pada suatu jaringan infrastruktur *server* membutuhkan suatu perangkat yang berfungsi sebagai router ataupun *gateway*, sehingga perangkat pengguna lain dalam jaringan *wireless* tersebut dapat melakukan komunikasi secara online, dimana dituntut untuk dapat memberikan layanan *real time* secara 24 jam setiap harinya. Oleh karena itu perkembangan penggunaan teknologi *wireless* berdampak pada sistem autentikasi yang dibangun. Seperti diketahui bahwa pada umumnya *wireless* router sendiri memiliki kelemahan dalam proses autentikasi. Salah satu contoh adalah dalam hal autentikasi *client* baru pada jaringan *wireless* yang diproteksi dengan proses autentikasi. Umumnya proses autentikasi diproses hanya sebatas antara *client* baru dengan perangkat router

untuk mendapatkan akses ke dalam jaringan tersebut, jika proses autentikasi berhasil maka diberikan layanan DHCP atau pemberian IP *address* sehingga *client* baru dapat terhubung dan dengan bebas menggunakan aturan-aturan lalu lintas jaringan yang ada pada teknologi *networking* seperti internet, SSH, FTP dll. Namun proses ini tidak diketahui oleh seorang *network administrator* yang dimana seorang *network administrator* bertanggung jawab pada jaringan yang dikelolanya. Terkadang karena dalam penggunaan dari proses autentikasi ini dapat menimbulkan hal yang tidak diinginkan seperti terdapat adanya *client* baru yang mengetahui *password* dalam proses autentikasi ini dan mempunyai niat buruk dalam menggunakan jaringan tersebut [2].

Demi menghindari masalah yang muncul dari proses autentikasi tersebut, diharapkan dalam setiap proses autentikasi diketahui oleh *network administrator* untuk menjamin keamanan jaringan yang dikelolanya [3]. Oleh karena itu seorang *network administrator* memiliki kewenangan untuk memberikan atau tidak hak akses dalam penggunaan lalu lintas jaringan yang dikelolanya setelah proses autentikasi pada *client* baru yang ingin masuk ke dalam jaringan tersebut. Tetapi kadang kala seorang *network administrator* tidak selalu dapat memberikan hak akses penggunaan jaringan kepada *client* baru tersebut secara tatap muka langsung ataupun secara fisik dengan perangkat router dikarenakan sedang tidak berada di tempat. Memang sebelumnya sudah terdapat teknologi yang dapat memantau sebuah router dari jarak jauh, tetapi tekonologi ini memiliki kelemahan yaitu seorang *network administrator* diharuskan berada dalam jaringan lokal tersebut atau harus memiliki koneksi internet jika berada di luar jaringan lokal tersebut. Sehingga dibutuhkan teknologi alternatif untuk membantu *network administrator* untuk mendapatkan informasi mengenai proses autentikasi *client* baru pada jaringannya secara cepat dan akurat walaupun sedang berada jauh dari lingkungan jaringan yang di kelolanya. Teknologi yang mungkin dapat digunakan adalah layanan pesan yaitu *email* dan SMS *Gateway*. Dibanding dengan teknologi *message sevice* baru lainya,

teknologi *email* dan SMS *Gateway* dapat digunakan untuk berkomunikasi secara mudah, cepat dan murah tanpa perlu melakukan kerja sama dengan developer pembuatnya karena teknologi ini berisifat *open source*. Salah satunya manfaat dan keuntungan implementasi teknologi ini pada sistem yaitu sebagai jaringan alternatif penghantar pesan informasi dalam proses autentikasi dan perintah otorisasi antara router dengan *network administrator* jika terdapat *client* baru. Serta menggunakan teknologi Iptables (*firewall*) yang berperan sebagai pelaksana aturan dalam fitur-fitur yang ada pada sebuah lalu lintas jaringan tersebut sesuai dengan aturan yang telah ditetapkan oleh *network administrator* kepada setiap *client* pada jaringan yang dikelolanya.

Berdasarkan hal ini, maka perlu adanya sebuah sistem autentikasi yang memperingan pekerjaan dan juga melibatkan *network administrator* dalam proses autentikasi *client* baru serta mengatur aturan lalu lintas jaringan yang dapat digunakan oleh *client* pada jaringan yang dikelolanya. Maka dari itu dibangunlah sebuah perangkat router dan sistem dengan judul "Sistem Autentikasi Provisioning Jaringan *Wireless* melalui DHCP *Server* dengan menggunakan Layanan Pesan" dengan tujuan bahwa setiap permintaan akses oleh *client* baru dalam sebuah jaringan harus seijin dan sepengetahuan *network administrator* melalui layanan pesan *email* dan SMS *Gateway*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membangun sebuah perangkat dan sistem *wireless* router dengan *firewall* menggunakan DHCP *server* dan Iptables.
2. Bagaimana membangun sebuah perangkat dan sistem pada *wireless* router DHCP *server* yang dapat memberitahu *network administrator* jika terdapat perangkat baru yang terhubung ke dalam jaringan *wireless* yang dikelolanya?

3. Bagaimana membangun sebuah sistem dan perangkat *wireless* router DHCP *server* yang memberi kewenangan seorang *network administrator* sebagai penentu keputusan dalam pemberian akses pada jaringan *wireless* yang dikelolanya?

1.3 Batasan Masalah

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

1. Sistem terhubung pada jaringan laboratorium AJK Teknik Informatika ITS.
2. Perangkat keras yang digunakan adalah mini PC Raspberry pi 2 model B, TP-Link WN322G *wireless adapter*, Wavecom SMS *gateway* m1306b.
3. *Wireless* router menggunakan Dnsmasq sebagai DHCP *server*, SMS dan *email* sebagai *message service* dan Iptables sebagai *firewall* pada sistem operasi Unix (Raspian).
4. Fungsi dari perangkat *wireless adapter* yang digunakan pada sistem ini hanya berfungsi sebagai *access point*.
5. Aturan lalu lintas jaringan yang akan berlaku untuk *client* yang terhubung pada jaringan *wireless* router ini adalah Internet, SSH, ICMP dan FTP saja.
6. Setiap pengguna dalam melakukan proses verifikasi perangkat diharuskan mengakses alamat 192.168.2.1 terlebih dahulu.

1.4 Tujuan

Tujuan pembuatan perangkat dan sistem pada tugas akhir ini adalah membangun sebuah sistem autentikasi pada sebuah jaringan *wireless* pada router DHCP *server* yang melibatkan peranan *network administrator* sebagai pemegang kewenangan proses pemberian akses aturan lalu lintas jaringan pada sebuah jaringan infrastruktur *server* jika terdapat *client* baru yang ingin mendapat akses ke dalam jaringan tersebut melalui respon *email* dan SMS.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah sebagai berikut:

1. Memberikan sebuah sistem autentikasi yang dapat mengirimkan notifikasi jika ada pengguna baru kepada *network administrator* terhadap sebuah jaringan *wireless* router DHCP *server* yang terhubung pada sebuah jaringan private atau jaringan infastruktur *server*.
2. Memberi kewenangan kepada *network administrator* sebagai penentu keputusan dalam proses autentikasi dan pemberian aturan lalu lintas jaringan terhadap *client* baru pada jaringan yang dikelolanya.
3. Memberikan kemudahan dalam proses autentikasi dan pemberian akses aturan lalu lintas jaringan melalui *email* dan SMS sehingga *network administrator* tidak perlu berinteraksi fisik pada lingkungan jaringan yang dikelolanya.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir

   Proposal tugas akhir ini berisi tentang deskripsi pendahuluan untuk pembuatan tugas akhir. Pendahuluan ini terdiri atas latar belakang diajukannya tugas akhir, permasalahan yang diangkat, batasan masalah, tujuan dan manfaat dibuatnya tugas akhir ini. Selain itu, dijabarkan pulau tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Pada Sub Bab Metodologi menjelaskan tentang mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Jadwal kegiatan yang dilakukan juga dilampirkan dalam proposal tugas akhir ini agar tepat waktu.
2. Studi literatur

Tahap ini merupakan tahap merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan Tugas sekaligus

mempelajarinya. Mulai dari pengumpulan literatur, diskusi, serta pemahaman topik Tugas Akhr diantaranya tentang:

1. Perancangan alat atau perangkat keras dan aplikasi jaringan yang akan diintegrasikan pada DHCP *server* agar mampu melakukan komunikasi pada setiap perangkat.
2. Membuat sebuah *web interface* untuk me-*monitoring* akses fitur pada *client* yang sudah diintegrasikan dengan DHCP *server*.

3. Perancangan perangkat lunak dan perangkat keras

   Tahap ini merupakan implementasi rancangan sistem yang telah dibuat. Tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya sehingga menjadi sebuah sistem yang sesuai dengan apa yang telah direncanakan.

4. Implementasi

   Implementasi merupakan tahap membangun rancangan perangkat keras dan perangkat lunak yang telah dibuat. Pada tahap ini akan direalisasikan mengenai rancangan apa saja yang telah didefiniskan pada tahap sebelumnya. Fungsi yang ada pada tahap ini merupakan fungsi hasil implementasi dari tahap analisis dan perancangan perangkat keras dan perancangan perangkat lunak.

5. Pengujian dan evaluasi

   Sistem akan diuji setelah selesai diimplementasikan menggunakan scenario yang sudah dipersiapkan. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Dengan melakukan pengujian dan evaluasi dimaksudkan juga mengevaluasi jalannya program, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan buku Tugas Akhir

   Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi

perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
    a. Latar Belakang
    b. Rumusan Masalah
    c. Batasan Tugas Akhir
    d. Tujuan
    e. Metodologi
    f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

1.7 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 yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu perumusan masalah, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar teori penunjang yang digunakan untuk mendukung penyelesaian Tugas Akhir..

Bab III Desain dan Perancangan

Bab ini berisi tentang desain dan rancangan sistem yang akan dibangun berupa perancangan alat dan perancangan perangkat lunak.

Bab IV Implementasi

Bab ini membahas implementasi dari desain sistem yang akan dilakukan pada tahap desain, meliputi potongan *pseudocode* yang terdapat dalam perangkat lunak dan perangkat keras yang digunakan.

Bab V Uji Coba Dan Evaluasi

Bab ini menjelaskan mengenai kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari perangkat lunak yang telah dibuat sesuai dengan data yang diujikan.

Bab VI Kesimpulan Dan Saran

Bab ini membahas uji coba dari perangkat lunak yang dibuat dengan melihat keluaran yang dihasilkan oleh perangkat lunak, analisa dan evaluasi untuk mengetahui kemampuan perangkat lunak.

BAB II
TINJAUAN PUSTAKA

2.1 DHCP

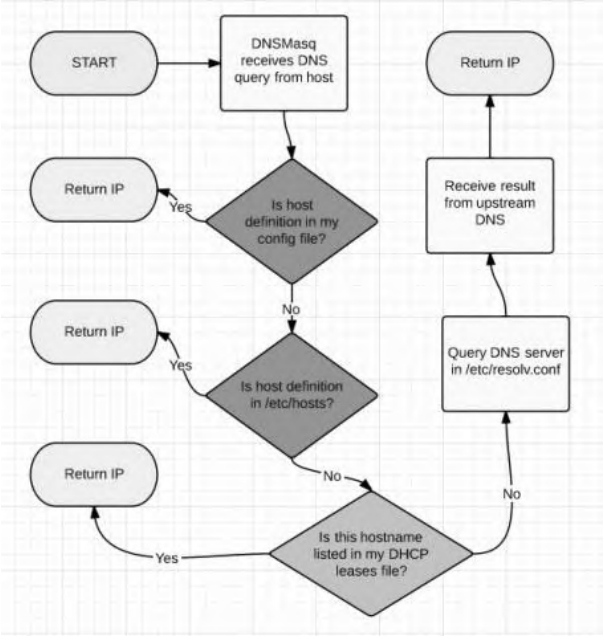
Dynamic Host Configuration Protocol (DHCP) adalah layanan yang secara otomatis memberikan alamat *Internet Protocol* (IP) kepada komputer atau *client* yang memintanya. Komputer yang memberikan alamat IP disebut sebagai DHCP *Server*, sedangkan komputer yang meminta alamat IP disebut sebagai DHCP *Client*. Dengan demikian *network administrator* tidak perlu lagi harus memberikan alamat IP secara manual pada saat konfigurasi TCP/IP, tapi cukup dengan memberikan referensi kepada DHCP *Server*.

Pada saat DHCP *Client* dinyalakan, maka computer *client* tersebut melakukan request ke DHCP *Server* untuk mendapatkan alamat IP. DHCP *Server* menjawab dengan memberikan alamat IP yang ada di *database* DHCP kepada *client*. Setelah memberikan alamat IP, maka DHCP *Server* meminjamkan (*lease*) alamat IP yang ada kepada DHCP *Client* dan mencoret alamat IP tersebut dari daftar *pool*. Alamat IP diberikan bersama dengan *subnet mask* dan *default gateway*. Jika tidak ada lagi alamat IP yang dapat diberikan, maka *client* tidak dapat menginisialisasi TCP/IP, dengan sendirinya tidak dapat tersambung pada jaringan tersebut. Secara umum ada empat struktur mekanismeantara DHCP *Server* dengan DHCP *Client* yaitu DHCPDiscover, DHCPOffer, DHCPRequest, DHCPAck [4].

Setelah periode waktu tertentu, masa pemakaian alamat IP oleh DHCP *Client* tersebut dapat dinyatakan habis. Oleh karena itu alamat IP tersebut dikembalikan kepada DHCP *Server*, dan DHCP *Server* dapat memberikan alamat IP tersebut kepada *Client* baru yang membutuhkan. Lama periode ini dapat ditentukan dalam menit, jam, bulan atau selamanya. Jangka waktu tersebut disebut *Leased Period*.

2.2 Dnsmasq

Dnsmasq adalah sebuah perangkat lunak *Domain Name System* (DNS) *forwarder* dan *Dynamic Host Configuration Protocol* (DHCP) *server* untuk jaringan komputer sederhana [5]. Dnsmasq memiliki ketentuan yang mudah untuk dijalankan pada suatu sistem. Dnsmasq dapat berjalan di Linux, BSD, Android, OS X dan sudah termasuk dalam komponen sistem operasi pada kebanyakan distribusi Linux. DHCP *Server* pada Dnsmasq mendukung *static* dan *dynamic* DHCP *leases, muliple networks* dan *IP address ranges*. DHCP *server* terintegrasi dengan DNS *server* dan memungkinkan *local machines* dengan DHCP mengalokasikan alamat untuk muncul di DNS. Pada Gambar 2.1 menunjukkan modifikasi metode yang digunakan ketika DNS *server* menerima sebuah query sekaligus menjadi sebuah DHCP *server*.



Gambar 2.1 Alur Kerja Dnsmasq

2.3 Hostapd

Hostpad adalah sebuah untuk daemon untuk membuat *access point* dan *server* autentikasi. Pada umumnya, hostapd memungkinkan kita untuk membuat perangkat lunak *wifi access point* dengan berbagai macam pilihan konfigurasi. Program ini mengimplementasikan pengelolaan *access point* IEEE 802.11, IEEE 802.1X/WPA/WPA2/EAP *Authenticators,* Radius *Client,* EAP *server,* dan *server* autentikasi Radius [6].

Hostapd di desain menjadi sebuah daemon, dimana program tersebut berjalan di belakang sistem dan bertindak sebagai komponen yang mengontrol autentikasi.

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* managemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan.

Pada MySQL terdapat *logical model* yang terdiri dari *databases, tables, views, rows,* dan *columns* untuk membantu penggunaan data yang terstruktur di lingkungan programer. MySQL juga mendukung sistem klien *server,* sehingga data dapat terpusat di *server* dan dapat diakses oleh banyak klien [7].

2.5 Raspberry

Raspberry Pi merupakan komputer kecil seukuran dengan sebuah kartu kredit dengan berbagai fungsi yang dapat dilakukannya [8]. Raspberry Pi menggunakan sistem operasi Raspbian. Gambar 2.2 merupakan Raspberry Pi 2 model B yang memiliki prosesor dengan spesifikasi 900 MHz quad-core ARM Cortex A7 dan memilik RAM sebesar 1 GB. Sistem operasi utama untuk Raspberry Pi adalah Raspbian OS dan didasari dari Debian

(based on debian). Raspberry Pi menggunakan SD *Card* sebagai media penyimpanannya. Selain itu Raspberry juga dilengkapi 2 buah port USB untuk tipe B, konektor HDMI untuk tipe B, serta dilengkapi dengan port Ethernet. Pada Raspberry Pi tidak disediakan *switch power*. Port micro USB pada Raspberry Pi digunakan sebagai *supply power*, penggunaan micro USB dikarenakan murah dan mudah didapatkan. Rasberry Pi membutuhkan *supply* sebesar 5V dengan arus minimal 700mA untuk tipe B. Pada Raspberry Pi disediakan pin-pin input/output (IO), diantaranya adalah :

- General Purpose Input dan Output (GPIO)
  Pin-pin tersebut dapat digunakan untuk membaca input dari tombol serta switches serta mengontrol actuator seperti LED.
- Display Serial Interface (DSI) connector
  Konektor ini dapat digunakan dengan menggunakan kabel pita tipis 15 pin sebagai penghubung antara LCD atau layar OLED.
- Camera Serial Interface (CSI) connector
  Port ini berfungi sebagai penghubung langsung antara Rapberry Pi dengan sebuah modul kamera.

Gambar 2.2 Raspberry Pi 2 Model B

Raspberry Pi dapat melakukan banyak hal yang tidak membutuhkan komputer mahal untuk membuatnya. Seperti berjalan sebagai NAS (*Network Attached Storage*), *web server*, router, media center, TorrentBox dan masih banyak lagi.

2.6 Wireless Adapter

Wireless adapter adalah perangkat keras yang dipakai oleh komputer untuk menerima dan mentransmisikan sinyal. *Wireless adapter* mempunyai prinsip kerja yang hampir sama dengan sebuah *access point*, tetapi lebih sederhana. Apabila dalam sebuah *access point* terdapat *memory* maupun prosesor, maka pada *wireless adapter* penggunaannya tidak sekompleks *access point*. Perangkat ini adalah perangkat sederhana yang digunakan untuk *access point*.

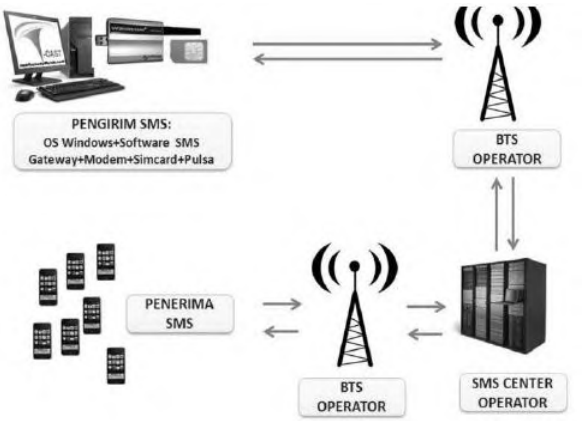
Gambar 2.3 TP-Link WN322G Wireless Adapter

Gambar 2.3 merupakan TP-Link WN322G yang dirancang untuk memberikan kecepatan tinggi dan kinerja nirkabel yang baik untuk komputer sebagai *wireless adapter* [9]. Selain itu *wireless adapter* ini mendukung fungsi QSS, yang dapat membantu penggunanya membuat koneksi nirkabel menjadi lebih aman.

2.7 SMS Gateway

Short Message Service (SMS) adalah kemampuan untuk mengirim dan menerima pesan dalam bentuk teks dari dan kepada ponsel. Teks tersebut bisa terdiri dari huruf, angka atau kombinasi alphanumeric. SMS Gateway adalah komunikasi mengandung informasi berupa nomor telepon selular pengirim, penerima, waktu dan pesan. Informasi tersebut dapat diolah dan bisa melakukan aktivasi transaksi tergantung kode-kode yang sudah disepakati. Untuk dapat mengelola semua transaksi yang masuk dibutuhkan sebuah sistem yang mampu menerima kode SMS dengan jumlah tertentu, mengolah informasi yang terkandung dalam pesan SMS dan melakukan transaksi yang dibutuhkan. Aplikasi SMS Gateway adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi selular yang diintegrasikan guna mendistribusikan pesan-pesan yang dipadukan lewat sistem informasi melalui media SMS yang ditangani oleh jaringan seluler. SMS Gateway biasanya mendukung untuk pesan berupa teks, Unicode, character, dan juga *smart messaging* [10].

Modem Wavecom Fastrack M1306B adalah modul komunikasi selular GSM yang menggunakan prinsip Plug and Play sehingga tidak memerlukan instalisasi yang rumit untuk menggunakannya. Wavecom Fastrack M1306B juga menyediakan komunikasi data dengan perangkat luar melalui antar muka serial serta yang dapat deprogram dengan menggunakan perintah-perintah AT Command.



Gambar 2.4 Alur Kerja SMS Gateway

Gambar 2.5 Modem Wavecom M1306B

2.8 Email SMTP

Email (Electronic Mail) adalah sebuah fasilitas komunikasi dalam internet yang berfungsi dalam mengirim surat atau pesan

secara elektronik yang dapat menjangkau ke seluruh dunia. DIbandingkan dengan surat biasa, *email* mempunyai keunggulan yang lebih aman serta tidak membedakan jarak dan waktu. Secara garis besar *email* dapat dibedakan menjadi dua, yaitu *email* berbasis SMTP/POP dan *email* berbasis *web*.

SMTP (*Simple Mail Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk pengiriman surat elektronik di internet. Protokol ini dipergunakan untuk mengirimkan data dari komputer pengirim surat elektronik ke *server* surat elektronik penerima. Protokol ini ada karena desain sistem surat elektronik yang mengharuskan adanya *server* surat elektronik yang menampiung sementara sampai surat elektronik diambil oleh penerima yang berhak.

SMTP adalah protokol yang cukup sederhana, berbasis teks dimana protokol ini menyebutkan satu atau lebih penerima *email* untuk kemudian diverivikasi. Jika penerima *email* valid, maka *email* akan segera dikirim. SMTP menggunakan port 25 dan dapat dihubungi melalui program telnet. Agar dapat menggunakan SMTP *server* lewat nama domain, maka record DNS (Domain Name *Server*) pada bagian MX (Mail Exchange) digunakan.
SMTP hanya protokol yang melakukan push, artinya hanya mengambil *email* dari client tetapi tidak bisa melakukan pull, yaitu melayani pengambilan *email* di erver oleh client [11].

2.9 Iptables

Iptables adalah suatu *tools* dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan filter terhadap lalu lintas data. Secara sederhana digambarkan sebagai pengatur lalu lintas data. Dengan Iptables inilah kita akan mengatur semua lalu lintas data komputer kita, baik yang masuk ke komputer, keluar komputer, ataupun lalu lintas yang sekedar melewati komputer kita.

Dengan kemampuan tools Iptables ini, kta bisa melakukan banyak hal dengan Iptables. Yang terpenting adalah bahwa Iptables ini kita dapat membuat aturan (rule), untuk arus lalu lintas data.

Aturan-aturan tersebut mencakup banyak hal, seperti besar data yang boleh lewat, jenis paket yang diterima, mengatur lalu lintas berdasarkan asal dan tujuan data, forwarding NAT, *redirecting*, pengelolaan *port*, dan *firewall* [12].
Iptables memiliki empat tabel aturan yaitu :

- Filter
  Untuk melakukan penyaringan paket data, apakah paket tersebut akan di DROP, LOG, ACCEPT atau REJECT
- NAT
  Melakukan *Network Address Translation* yang merupakan pengganti alamat asal tujuan dari paket data.
- Mangle
  Untuk melakukan penghalusan (mangle) paket data seperti TTL, TOS dan MARK.

Pada tabel terdapat chains yang berisi aturan yang berbeda-beda. Chains pada tabel filter yaitu :

- INPUT
  Paket yang disiapkan untuk soket lokal atau komputer kita sendiri atau untuk mengatasi paket data yang masuk.
- FORWARD
  Paket yang diarahkan ke *box* atau untuk mengalihkan paket yang datang.
- OUTPUT
  Paket yang dibuat sendiri atau untuk menghasilkan paket data yang akan diteruskan.

Paket-paket yang masuk akan diperiksa, apakah rusak, salah informasi atau tidak, kemudian diberikan ke chain INPUT.
Keputusan yang diambil untuk suatu paket dapat berupa:

- ACCEPT
  Menerima paket dan diproses lebih lanjut oleh kernel.
- DROP
  Menolak paket tanpa pemberitahuan terlebih dahulu.
- REJECT

Mengembalikan paket ke asalnya dengan pesan kesalahan ICMP.

- LOG
  Melakukan log (pencatatan) terhadap paket yang bersesuaian.
- RETURN
  Untuk chain *user defined* chain INPUT, OUTPUT, FORWARD akan dijalankan kebijakan default.

BAB III
PERANCANGAN PERANGKAT LUNAK DAN PERANGKAT KERAS

Pada bab ini akan dijelaskan mengenai dasar perancangan perangkat lunak yang akan dibuat dalam tugas akhir ini. Secara khusus akan dibahas mengenai deskripsi umum aplikasi, perancangan proses, alur, serta gambaran implementasi perangkat lunak.

3.1 Deskripsi Umum Sistem

Pada tugas akhir ini dibangun sebuah router dengan sistem autentikasi *provisioning* jaringan *wireless* melalui DHCP *server* dengan menggunakan layanan pesan yang dapat membantu seorang *network administrator* dalam mengelola jaringan infrastruktur servernya. *Wireless* router ini dibangun dengan beberapa perangkat keras yaitu menggunakan mini pc sebagai DHCP *server*, *wireless adapter* sebagai pemancar *access point* dan sebuah perangkat sms *gateway* sebagai penghantar layanan sms. Pada router ini setiap terdapat *client* baru yang terhubung ke dalam jaringan, maka sistem pada router akan mencatat data *client* berupa kode verifikasi, mac address, ip address dan *device name* ke dalam *database* yang ada pada router. Sistem mengirimkan notifikasi berisi informasi *client* tersebut kepada *network administrator* melalui *email* dan sms hanya sekali saja dan menerima perintah balasan untuk melakukan proses otorisasi aturan fitur yang telah ditentukan untuk setiap *client* yang diproses. Secara *default* setiap *client* yang masuk ke dalam jaringan dan belum di otorisasi oleh *network administrator* maka tersebut tidak dapat menggunakan fitur-fitur yang telah diatur dalam jaringan tersebut, fitur-fitur yang ada berupa aturan iptables berupa Internet, FTP dan SSH.

DHCP *server* yang digunakan pada router ini adalah dnsmasq dan hostapd sebagai aplikasi yang mengintegrasikan *wireless adapter* dengan DHCP *server* sehingga menjadi sebuah

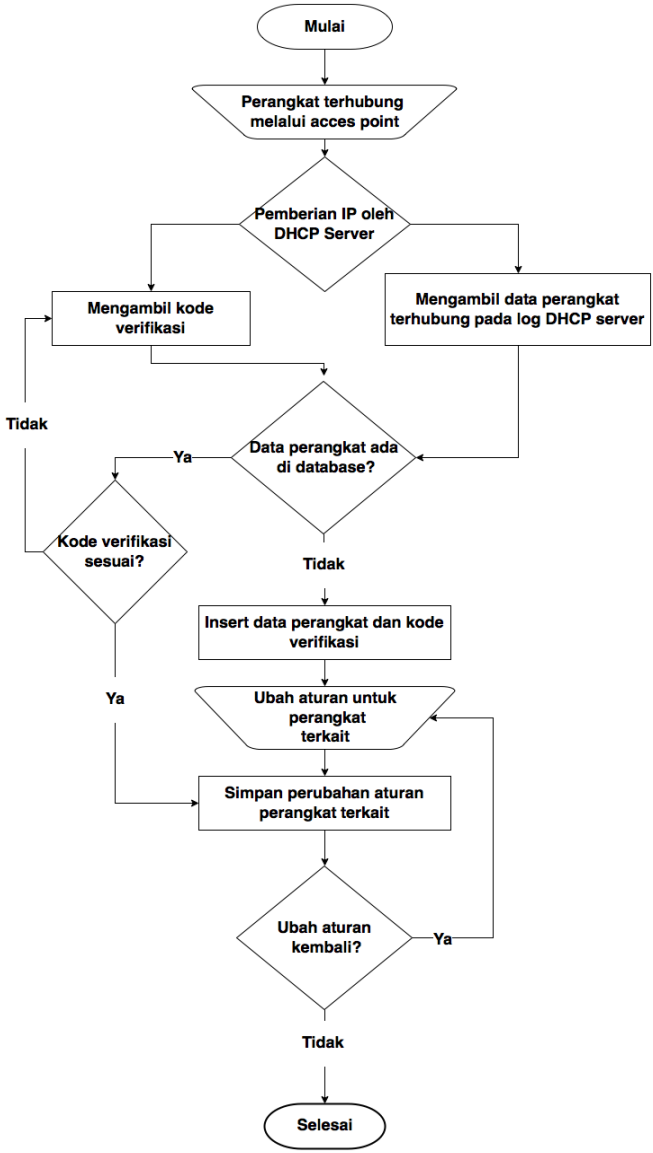
wireless router. Data informasi *client* yang diperlukan terdapat pada log DHCP *server* dan diambil dengan aplikasi berupa *script* yang kemudian disimpan ke dalam *database* MySql. Untuk mengirimkan notifikasi informasi *client* baru, sistem menggunakan sms dengan aplikasi gammu yang sudah diintegrasikan dengan perangkat keras SMS *gateway* dan menggunakan *email* dengan aplikasi php pada codeigniter melalui protokol SMTP. Dalam mengelola aturan fitur-fitur yang diterima *network administrator* melalui *email* akan di lanjutkan ke dalam sebuah *web interface* yang dibangun dengan semantic ui.

3.2 Arsitektur Umum Sistem

Agar dapat menjalankan fungsinya, maka alur kerja dari kesatuan aplikasi ini dirancang seperti pada Gambar 3.1.

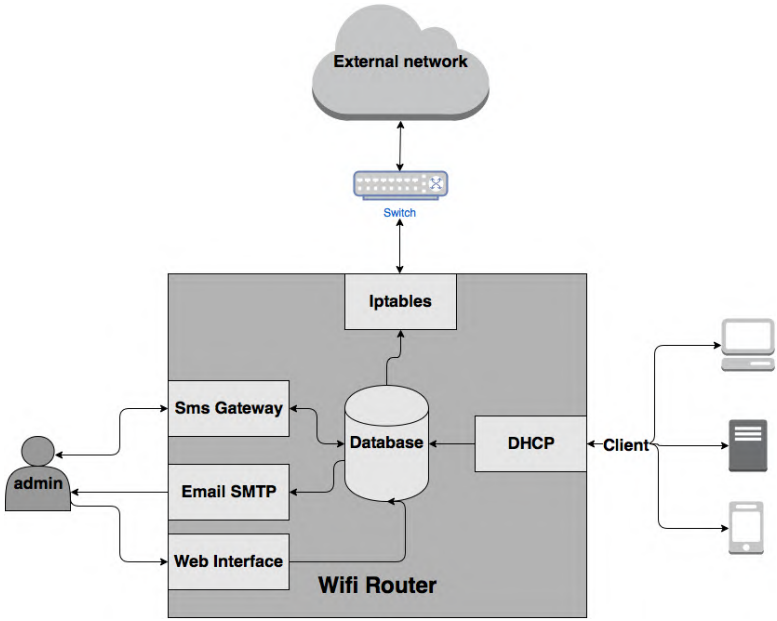
Berdasarkan Gambar 3.1, alur kerja sistem ini dijabarkan sebagai berikut:

1. Pengguna baru atau perangkat baru harus terhubung ke *wireless* router melalui *access point*.
2. Ketika perangkat pengguna telah terhubung, DHCP *server* melalui Dnsmasq memberikan IP kepada perangkat baru tersebut.
3. Sistem memeriksa log Dnsmasq dan kode verifikasi yang dimasukkan. Kemudian sistem mengambil data perangkat tersebut berupa, MAC Address, Device Name dan IP *address* serta kode verifikasi yang diberikan sebelumnya.
4. Setelah mengambil data perangkat dan kode verifikasi , sistem akan mencocokan data perangkat dengan data yang ada pada *database*, jika data perangkat sesuai dan terdapat dalam *database* maka tidak akan di masukkan kembali.
5. Jika data perangkat tidak terdapat pada *database* atau memiliki perbedaan dengan yang ada pada *database* maka data perangkat tersebut dimasukkan dan diperbarui, kemudian aturan yang diberikan sebelumnya akan di setel ulang.



Gambar 3.1 Alur Kerja Sistem

6. Kode verifikasi digunakan untuk memvalidasi kebenaran data perangkat yang digunakan pengguna dan sebagai acuan dalam mengaktifkan aturan yang diberikan.
7. Sistem mengirimkan pemberitahuan pada *network administrator* melalui sms dan *email* melalui SMS Gateway dan *email* SMTP.
8. *Network administrator* memberikan aturan pada perangkat sesuai dengan notifikasi yang diterima sebelumnya melalui SMS atau *Web interface* yang alamatnya dikirimkan melalui *email*.
9. Sistem memperbarui aturan perangkat tersebut.
10. Sistem dapat memperbarui aturan pada perangkat yang sebelumnya pernah diatur dengan menerima format perintah melalui SMS maupun dengan pemilihan aturan yang ada pada *Web Interface* oleh *network administrator*.



Gambar 3.2 Arsitektur Umum Sistem

Pada Gambar 3.2 merupakan gambaran arsitektur dari pembangunan sistem aplikasi pada perangkat *wireless* router. Sistem aplikasi terdiri sebuah *wireless* router yang dibangun menggunakan perangkat mini pc, *wireless adapter* dan perangkat sms *gateway*. Pada *wireless* router ini terdiri dari beberapa komponen pendukung yang telah diintegrasikan serta melibatkan *network* admininstrator dalam penggunaannya. Penjelasan komponen pendukung pada *wireless* router ini sebagai berikut.

- DHCP
  Pada *wireless* router ini DHCP berfungsi sebagai pemberi IP *Address* pada perangkat pengguna yang terhubung dengan menggunakan aplikasi Dnsmasq dan diintegrasikan dengan aplikasi Hostapd yang berperan sebagai *access point*.
- SMS Gateway
  SMS Gateway pada *wireless* router ini bekerja menggunakan aplikasi Gammu dan berperan sebagai pengirim informasi perangkat pengguna baru yang terhubung dan penerima perintah eksekusi pemilihan aturan melalui layanan SMS.
- Email SMTP
  Email pada *wireless* router ini bekerja menggunakan *framework* PHP Codeigniter melalui protokol SMTP. Fungsi dari *email* yaitu mengirimkan data informasi perangkat pengguna baru yang terhubung kepada *network administrator*.
- *Web Interface*
  Web Interface pada *wireless* router ini dibangun dengan *framework* Semantic UI yang berperan sebagai halaman penentuan aturan pada setiap perangkat pengguna yang terhubung oleh *network administrator* melalui alamat *Web* yang diberikan sebelumnya melalui *email*.
- *Database*
  Database pada *wireless* router ini menggunakan aplikasi MySql yang berperan sebagai tempat penyimpanan data

informasi perangkat pengguna dan skema data aturan yang akan diberlakukan pada perangkat tersebut.

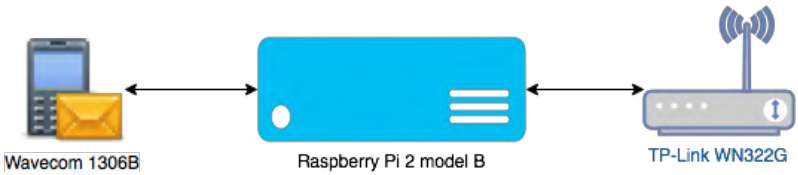
- Iptables
  Komponen ini berperan sebagai tools yang menjalankan aturan lalu lintas pada setiap perangkat pengguna yang terhubung dalam jaringan *wireless* router.

3.3 Rancangan Perangkat Keras

Pada sistem ini, perangkat keras yang digunakan terbagi menjadi 3 komponen yaitu satu buah mini PC sebagai router, satu buah *wireless adapter* yang menghubungkan router dengan perangkat pengguna melalui jaringan *wireless* dan satu buah perangkat SMS *Gateway* sebagai media penghantar layanan SMS. Komponen perangkat keras yang dibutuhkan untuk membangun sistem ini adalah sebagai berikut.

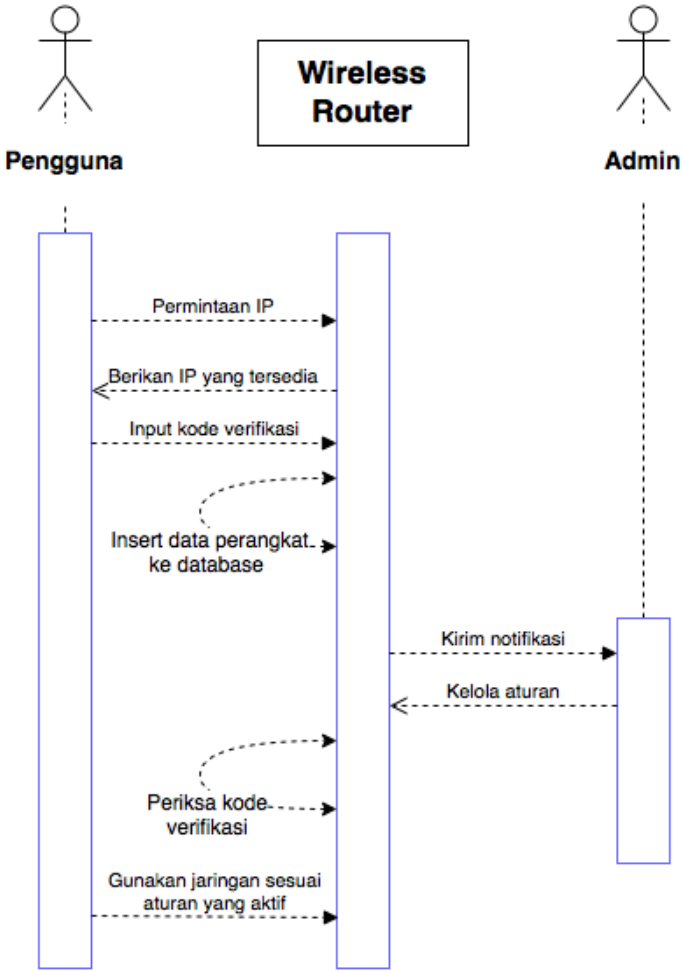
- 1 buah Raspberry Pi 2 Model B
- 1 buah TP-Link WN322G
- 1 buah Wavecom 1306B

Skema pada rangkaian sistem *wireless* router ini akan dihubungkan menjadi satu rangkaian yang akan terintegrasi. Skema *wireless* router yang dibangun akan ditunjukan pada gambar 3.3 dibawah ini.



Gambar 3.3 Skema Rangkaian Sistem

Diagram interaksi menggambarkan fungsionalitas sistem beserta aktor yang terlibat. Gambar 3.4 menunjukkan bahwa sistem memiliki 3 aktor utama yaitu pengguna (*client*), *wireless* router dan *network administrator*.



Gambar 3.4 Diagram Interaksi

Pada diagram interaksi diatas, pengguna (*client*) memiliki fasilitas untuk meminta IP *Address* untuk perangkat yang digunakannya serta dapat menggunakan jaringan sesuai dengan aturan aktif yang diberikan. *Wireless* router berfungsi memberikan IP *Address* yang tersedia kepada perangkat pengguna yang

meminta IP *Address* sebelumnya dan memasukkan data informasi perangkat baru yang digunakan oleh pengguna kedalam *database*, serta mengirimkan notifikasi kepada *network administrator* bahwa terdapat perangkat pengguna baru yang terhubung ke dalam jaringan. *Network administrator* berperan sebagai pengelola jaringan *wireless* router tersebut dengan memilih aturan yang telah tersedia sesuai dengan setiap perangkat pengguna yang terhubung.

3.4 Perancangan Diagram Interaksi

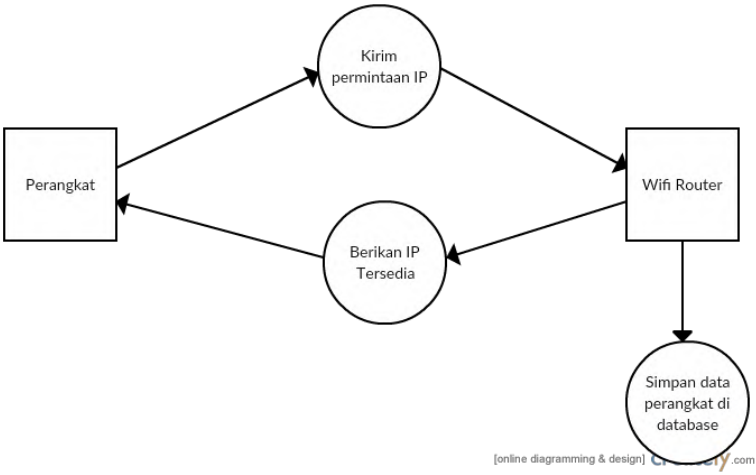
Diagram interaksi menggambarkan fungsionalitas sistem beserta aktor yang terlibat. Gambar 3.4 menunjukkan bahwa sistem memiliki 3 aktor utama yaitu pengguna (client), *wireless* router dan *network administrator*

Pada diagram interaksi diatas, pengguna (*client*) memiliki fasilitas untuk meminta IP *Address* untuk perangkat yang digunakannya serta dapat menggunakan jaringan sesuai dengan aturan aktif yang diberikan. *Wireless* router berfungsi memberikan IP *Address* yang tersedia kepada perangkat pengguna yang meminta IP *Address* sebelumnya dan memasukkan data informasi perangkat baru yang digunakan oleh pengguna kedalam *database*, serta mengirimkan notifikasi kepada *network administrator* bahwa terdapat perangkat pengguna baru yang terhubung ke dalam jaringan. *Network administrator* berperan sebagai pengelola jaringan *wireless* router tersebut dengan memilih aturan yang telah tersedia sesuai dengan setiap perangkat pengguna yang terhubung.

3.5 Perancangan Diagram Alir Melayani Perangkat Baru

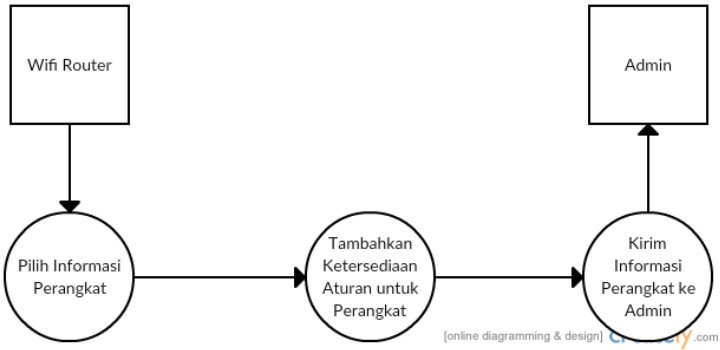
Gambar 3.5 merupakan diagram alir melayani perangkat baru secara umum. Ketika setiap perangkat baru yang melakukan permintaan IP *Address* kedalam jaringan *wireless* router melalui *access point* maka *wireless* router melalui DHCP *server* akan memilih dan memberikan IP *Address* yang tersedia serta menyimpan data-data informasi perangkat baru tersebut kedalam

database berupa Mac *Address*, IP *Address* yang diberikan dan *device name*.



Gambar 3.5 Diagram Alir Melayani Perangkat Baru

3.6 Perancangan Diagram Alir Kirim Notifikasi



Gambar 3.6 Diagram Kirim Notifikasi

Gambar 3.6 merupakan diagram alir kirim notifikasi pada sistem *wireless* router. Dimana pada sistem, sebelum melakukan pengiriman notifikasi, sistem akan memilih data informasi

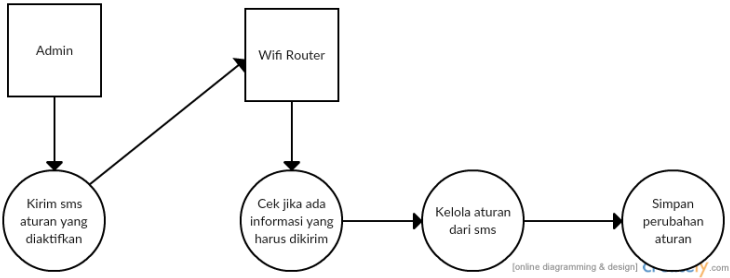
perangkat yang terdapat pada *database* serta menambahkan ketersediaan aturan yang akan diberikan kepada perangkat baru tersebut, kemudian mengirimkan notifikasi kepada *network administrator* melalui sms serta *email* yang berisi alamat *web interface* olah aturan.

3.7 Perancangan Diagram Alir Olah Aturan Melalui SMS

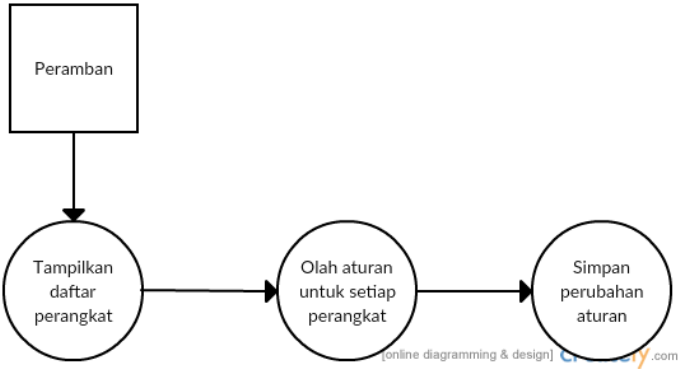
Gambar 3.7 merupakan diagram alir olah aturan melalui sms, dimana sistem pada *wireless* router menerima sms perintah dari *network administrator* untuk mengolah aturan yang akan diberlakukan pada perangkat yang terhubung kedalam jaringan, sesuai dengan data informasi perangkat yang disebutkan pada notifikasi yang diterima sebelumnya oleh *network administrator* melalui sms. Pada diagram ini setelah *network administrator* mengirimkan perintah untuk mengolah aturan, sistem akan memeriksa terlebih dahulu apakah terdapat notifikasi yang harus dikirimkan atau tidak, jika tidak ada maka sistem akan mengolah aturan untuk perangkat baru tersebut sesuai dengan perintah yang dikirimkan sebelumnya oleh *network administrator* melalui sms.

3.8 Perancangan Diagram Alir Olah Aturan Melalui Halaman *Web*

Gambar 3.8 merupakan diagram alir olah aturan melalui *web* interface. Pada saat sistem mengirimkan notifikasi kepada *network administrator* melalui *email*, *email* yang diterima yaitu berisi data informasi perangkat baru yang terhubung ke dalam jaringan serta sebuah link alamat *web interface* yang didalamnya terdapat fitur-fitur yang berfungsi untuk memilih aturan yang akan diberikan kepada perangkat baru tersebut dan juga dapat mengatur ulang aturan yang telah diberikan pada perangkat lainnya yang pernah tercatat kedalam *database*.



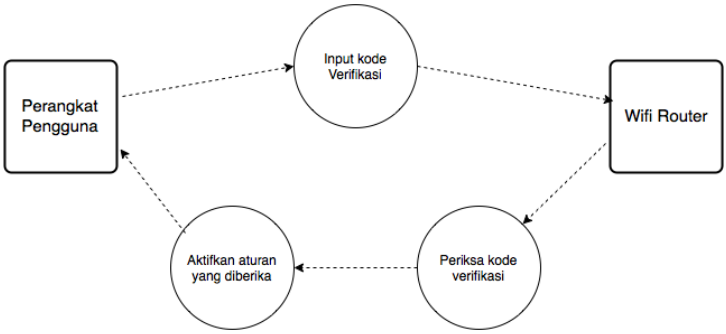
Gambar 3.7 Diagram Alir Olah Aturan Melalui SMS



Gambar 3.8 Diagram Alir Olah Aturan Melalui halaman *Web*

3.9 Perancangan Diagram Alir Periksa Kode Verifikasi

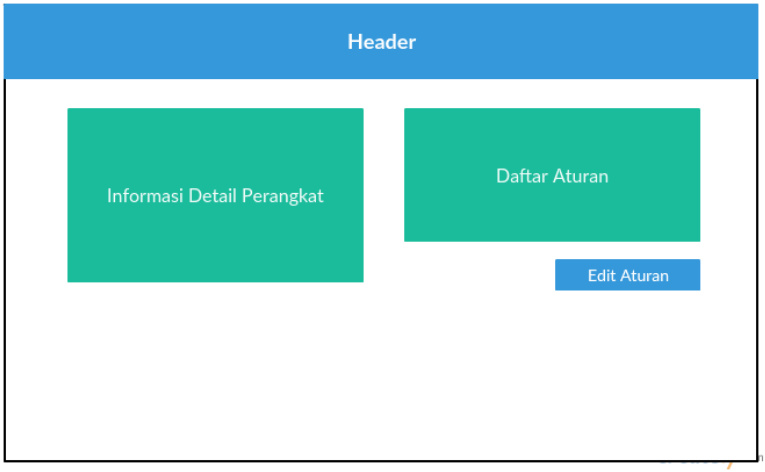
Pada Gambar 3.9 merupakan diagram alir periksa kode verifikasi dimana pada proses ini penguna diwajibkan untuk memasukkan kode verifikasi berupa no telepon selular yang menjadi acuan sistem dalam mengaktifkan aturan yang diberikan oleh *network administrator*. Bagi pengguna baru untuk pertama kali harus mendaftarkan nomor telepon selular sesuai yang diinginkan, untuk pengguna lama harus memasukkan nomor telepon selular sesuai dengan nomor yang didaftarkan pertama kali.



Gambar 3.9 Diagram Alir Periksa Kode Verifikasi

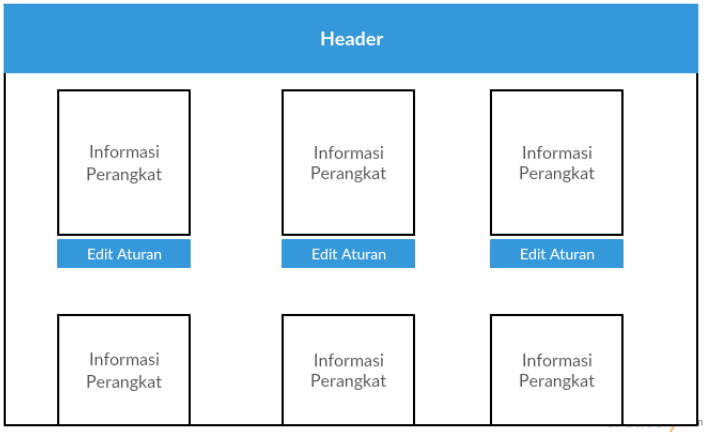
3.10 Perancangan Desain Antarmuka Sistem

Pada Tugas Akhir ini antarmuka sistem untuk olah aturan melalui peramban ialah berupa tampilan *web* . Untuk itu diperlukan perancangan desain antarmuka untuk *web* ini. Rancangan desain *web* ada tiga yaitu antarmuka halaman olah aturan, antarmuka halaman daftar perangkat dan antarmuka halaman verifikasi pernggunа. Rancangan antarmuka dapat dilihat pada Gambar 3.10 dan Gambar 3.11.



Gambar 3.10 Halaman Atur Perangkat

Pada Gambar 3.10 menunjukkan halaman *web* menampilkan informasi detail perangkat dan daftar aturan yang tersedia. Tampilan ini berfungsi untuk mengetahui informasi detail perangkat yang akan diotorisasi oleh *network administrator* berikut dengan pilihan aturan yang akan diberikan kepada perangkat tersebut.



Gambar 3.11 Halaman Daftar Perangkat

Gambar 3.12 Antarmuka Kode Verifikasi Pengguna

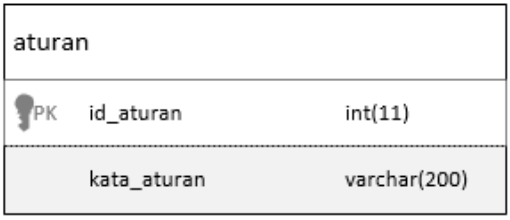
Pada Gambar 3.11 menunjukkan halaman *web* menampilkan daftar perangkat-perangkat yang terhubung kedalam jaringan *wireless* router. *Network administrator* dapat memantau dan mengelola perangkat-perangkat yang terhubung ke dalam jaringan dan dengan menekan tombol *edit* aturan pada salah satu daftar perangkat tersebut, maka akan diteruskan ke halaman *web* olah aturan untuk setiap perangkat.

Pada Gambar 3.12 menunjukkan halaman *web* menampilkan *form* kode verifikasi yaitu dengan memasukkan no telepon selular. Dengan menekan tombol *submit,* sistem akan mecocokkan apakah kode verifikasi yang berupa nomor telepon selular sama dengan yang ada pada database, yaitu no telepeon selular setiap perangkat pengguna yang pernah terhubung. Untuk pengguna baru, antar muka ini adalah sebagai pengisian pertama kode verifikasi agar akses perangkatnya tidak digunakan oleh orang lain yang tidak bertanggung jawab.

3.11 Perancangan Basis Data

Perancangan *database* diperlukan untuk melakukan penyimpanan data informasi perangkat pengguna dan aturan yang tersedia berikut dengan parameter serta pengolahan aturan tersebut. Pada Tugas Akhir ini dalam *database* dibuat empat tabel yaitu tabel `pengguna`, tabel `aturan`, tabel `parameter`, tabel `olah_aturan`.

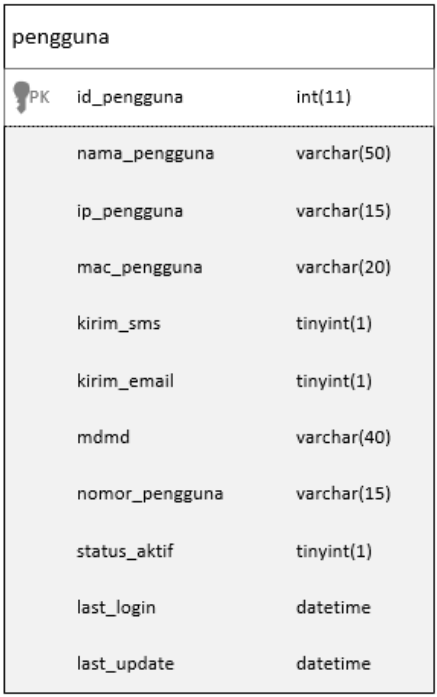
- Tabel Aturan



Gambar 3.13 Tabel Aturan

Pada Gambar 3.13 adalah tabel `aturan` yang digunakan untuk menyimpan informasi jenis-jenis aturan yang telah ditentukan. Informasi yang disimpan pada tabel ini adalah `id_aturan` untuk menyimpan id dari aturan yang nantinya akan menjadi *reference* dari tabel `olah_aturan`. `Kata_aturan` untuk menyimpan nama dari aturan yang telah ditentukan.

- Tabel Pengguna

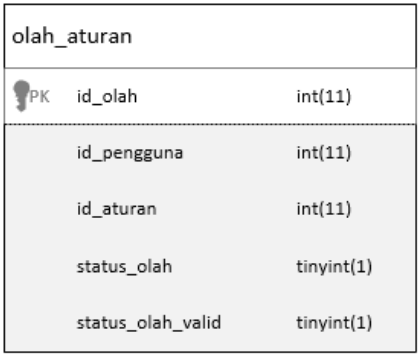


Gambar 3.14 Tabel Pengguna

Pada Gambar 3.14 adalah tabel `pengguna` yang digunakan untuk menyimpan data identitas dan informasi dari

setiap perangkat pengguna *wireless* router yang terhubung. Data yang disimpan adalah `id_pengguna` yang merupakan id perangkat yang terhubung ke dalam *wireless* router serta yang nantinya menjadi *reference* dari tabel `olah_aturan`, `nama_pengguna` untuk menyimpan informasi *device name* perangkat yang terhubung ke dalam *wireless* router, `ip_pengguna` untuk menyimpan informasi IP *Address* perangkat yang diberikan oleh *wireless* router, `mac_pengguna` untuk menyimpan informasi Mac *Address* perangkat yang terhubung ke dalam *wireless* router, `kirim_sms` untuk menyimpan informasi status notifikasi melalui sms telah terkirim atau tidak, `kirim_email` untuk menyimpan informasi status notifikasi melalui *email* telah terkirim atau tidak serta `mdmd` untuk meyimpan informasi hasil hash antara IP *Address* dan Mac *Address* perangkat, nomor_pengguna untuk menyimpan nomor telepon selular yang mejadi nomor verikasi pengguna perangkat, status_aktif menyimpan status aturan yang diberikan *network administrator*, `last_login` dan `last_update` yaitu untuk menyimpan waktu terakhir masuk kedalam *wireless router* dan waktu terakhir aturan diberikan oleh *network administrator*.

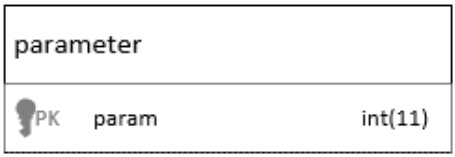
- Tabel Olah_aturan



Gambar 3.15 Tabel Olah_aturan

Pada Gambar 3.15 adalah tabel `olah_aturan` yang digunakan untuk menyimpan informasi id *reference* dari tabel `pengguna` dan `aturan` serta menyimpan informasi status aturan sudah diberikan atau tidak. Data informasi yang disimpan adalah `id_olah` untuk menyimpan id dari keseluruhan informasi id *reference* yang akan diintegrasikan, `id_pengguna` untuk menyimpan informasi id *reference* dari tabel `pengguna`, `id_aturan` untuk menyimpan informasi id reference dari tabel `aturan`, `status_olah` untuk menyimpan status pemberian aturan pada setiap perangkat dengan setiap jenis aturan.

- Tabel Parameter



Gambar 3.16 Tabel Parameter

Pada Gambar 3.16 adalah tabel `parameter` yang digunakan untuk menyimpan informasi parameter. Informasi yang disimpan adalah `param` yang berfungsi sebagai parameter dalam mengiintegrasikan seluruh data informasi yang telah tersimpan, yang nantinya berperan dalam proses bekerjanya iptables dalam memproses aturan lalu lintas jaringan kepada perangkat yang terhubung.

BAB IV
IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Pembangunan perangkat lunak dilakukan pada lingkungan pengembangan sebagai berikut.

4.1.1 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Linux Raspberry Pi 4.1.19-v7+ sebagai sistem operasi *server*.
- Dnsmasq version 2.72 sebagai DHCP *server*.
- Hostapd v2.3 sebagai penyedia *access point.*
- Shell Script sebagai *file teks* yang diisi perintah yang dieksekusi oleh sistem.
- Sublime Text versi 2.02 sebagai IDE untuk pembangunan aplikasi *web* sistem.
- Iptables sebagai tools *firewall* yang mengatur lalu lintas jaringan.
- Apache 2.4.10 (Raspbian) sebagai platform untuk menjalankan aplikasi *web* sistem.
- MySQL Ver 14.14 Distrib 5.5.44 sebagai *database*
- Codeigniter versi 3.0.6 sebagai *framework* php dalam membangun sistem.

4.1.2 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah mini PC Raspberry. Spesifikasi dari perangkat-perangkat tersebut adalah sebagai berikut:

1. Raspberry Pi 2 Model B ARMv7 Processor rev 5 (v7l) dengan memori 1 GB.
2. TP-Link TL-WN322G v3 *Wireless Adapter*.
3. Modem Wavecom M1306B.

4.2 Implementasi Perangkat Keras

Pembuatan implementasi sistem perangkat keras ini diawali dengan menggabungkan komponen-komponen perangkat keras yang ada menjadi sebuah *wireless* router terlebih dahulu. Pada sistem ini, komponen perangkat keras yang digunakan sebagai berikut:

1. Raspberry Pi 2 Model B
2. TP-Link TL-WN322G v3 *Wireless Adapter*.
3. Modem Wavecom M1306B.

Pengujian fungsi perangkat keras melibatkan semua komponen perangkat. Perangkat keras pada tugas akhir ini menggunakan sebuah mini PC Raspberry Pi. Mini PC Raspberry Pi berfungsi sebagai otak dari perangkat pendukung lainnya, tugasnya untuk mengintegrasikan fungsi komponen perangkat pendukung, memproses data dan mengolah aturan.

Mini PC Rasberry Pi dihubungkan dengan *wireless card* TP-Link dan modem Wavecom melalui *port* USB agar dapat memberikan layanan *access point* dan dapat menggunakan layanan SMS untuk komunikasi dengan *network administrator*. Kemudian menghubungkan mini PC Raspberry Pi dengan jaringan eksternal yang merupakan jaringan utama infrastruktur *server* pada port LAN melalui switch.

Pada saat pengujian, alat diletakkan di ruang *server* laboratorium AJK Teknik Informatika ITS dan menggunakan jaringan pada laboratorium AJK Teknik Informatika ITS.

4.3 Implementasi Perangkat Lunak

Pembuatan implementasi pada perangkat lunak terbagi menjadi dua bagian. Bagian pertama adalah implementasi perangkat lunak pada *wireless* router, bagian kedua implementasi pada aplikasi pendukung sistem.

4.3.1 Implementasi *Wireless* Router

Implementasi *wireless* router dimulai dari mengaktifkan DHCP *server*, mengaktifkan *access point*, mengaktifkan SMS *gateway*, mengolah data, mengirim notifikasi mengolah SMS masuk, dan mengolah aturan Iptables.

4.3.1.1 Mengaktifkan DHCP *Server*

Implementasi mengaktifkan DHCP *server* diawali dengan mengatur konfigurasi jaringan sesuai dengan lingkungan dimana *wireless* router akan terhubung. Konfigurasi dilakukan pada *file network interface* yang terdapat pada router dengan direktori sebagai berikut.

```
/etc/network/interfaces
```

Gambar 4.1 Direktori Network Interface

```
auto lo
iface lo inet loopback

auto eth0
iface ethx inet static
address 10.151.36.88
netmask 255.255.255.0
gateway 10.151.36.1

auto wlan0
iface wlanx inet static
address 192.168.2.1
```

```
netmask 255.255.255.0
```

Gambar 4.2 Konfigurasi Network Interface

Kata *auto* yang mendahului nama suatu *interface* menandakan bahwa *interface* tersebut akan dinyalakan secara otomatis pada saat *booting*. *Interface* lo tidak memiliki konfigurasi IP karena lo digunakan sebagai *loopback* sehingga memiliki IP yang pasti yaitu 127.0.0.1 . Alamat IP ini digunakan oleh komputer untuk berkomunikasi dengan dirinya sendiri. Konfigurasi untuk eth0 harus diberikan karena *interface* ini dikonfigurasi menggunakan IP statis. Parameter yang harus disebutkan untuk jenis *interface* statis adalah sebagai berikut.

1. *address* yaitu menentukan IP *address* yang digunakan komputer
2. *network* yaitu menentukan *Network address* komputer
3. netmask yaitu menentukan subnet mask *network* komputer
4. *broadcast* yaitu menentukan alamat broadcast yang digunakan komputer untuk memperkenalkan diri pada jaringan
5. *gateway* yaitu menentukan *default gateway* yang digunakan apabila komputer tersebut mengirimkan paket data ke luar jaringan anggotanya.

Setelah melakukan konfigurasi *network interface wireless* router, dilanjutkan dengan instalisasi aplikasi Dnsmasq dengan perintah sebagai berikut.

```
apt-get install dnsmasq
```

Gambar 4.3 Perintah Instalisasi Dnsmasq

Setelah Dnsmasq terpasang, kemudian mulai untuk konfigurasi pada *file* dnsmasq.conf sesuai yang dikehendaki pada direktori sebagai berikut.

```
/etc/dnsmasq.conf
```

Gambar 4.4 Direktori Konfigurasi Dnsmasq

```
domain-needed
interface=wlan0
dhcp-range=192.168.2.50,192.168.2.150
dhcp-option=3,192.168.2.1

#log dhcp
log-facility=/destination directory/dhcp.log
log-async
log-dhcp
```

Gambar 4.5 *File* Konfigurasi Dnsmasq

Konfigurasi pada dnsmasq.conf merupakan konfigurasi yang akan menentukan aturan dhcp *server* sebagai berikut.

1. *interface* yaitu menentukan *interface* dari perangkat *wireless* yang dikehendaki.
2. dhcp-*range* yaitu menentukan retang IP *address* yang dapat digunakan
3. dhcp-*option* yaitu dengan angka 3 memberikan default *gateway* yang dikehendaki.
4. log-facility yaitu digunakan untuk merekam log dhcp dan menentukan direktori untuk menyimpannya.

4.3.1.2 Mengaktifkan *Access Point*

Implementasi mengaktifkan *access point* merupakan proses memberikan router dhcp *server* fasilitas akses nirkabel pada *client*. Proses ini dimulai dengan inisialisasi dari perangkat *wireless card* yang terhubung pada router dengan perintah sebagai berikut.

```
lsusb
```

Gambar 4.6 Perintah Insialisasi Perangkat

Jika perangkat *wireless* terdeteksi, kemudian *install driver* yang sesuai untuk *wireless* card tersebut. Setelah itu install aplikasi Hostapd pada router dengan perintah sebagai berikut.

```
apt-get install hostapd hostapd-utils iw
```

Gambar 4.7 Perintah Instalisasi Hostapd

Setelah instalisasi aplikasi Hostapd, kemudian periksa nama *interface wireless card* dengan perintah sebagai berikut.

```
ip link show
```

Gambar 4.8 Perintah Inisialisasi Nama Interface Wireless Card

Kemudian konfigurasi aplikasi Hostapd pada file hostpad.conf yang terletak pada direktori sebagai berikut.

```
/etc/hostapd/hostapd.conf
```

Gambar 4.9 Direktori File Konfigurasi Hostapd

```
interface=wlan0
country_code=ID
ctrl_interface=0
channel=1
hw_mode=g
wmm_enabled=1
ssid=TitikAkses
wpa_passphrase=password
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
```

Gambar 4.10 File Konfigurasi Hostapd

Konfigurasi Hostapd tersebut menentukan *profil* dari *access point* dan kesesuaian dari perangkat keras yang digunakan. Penjelasan konfigurasi sebagai berikut.

1. *interface* yaitu *interface wireless card* yang akan digunakan.
2. *country_kode* yaitu kode negara yang digunakan.
3. ctrl_*interface* yaitu nama path direktori dimana untuk membuat *Unix domain socket*.
4. *channel* yaitu channel sinyal *wireless* yang digunakan.
5. hw_mode=g yaitu tipe sinyal yang digunakan 2.4GHz
6. wmm_*enabled* yaitu mendukung QoS.

7. ssid yaitu nama *access point* yang digunakan.
8. wpa_passphrase yaitu kata sandi yang digunakan.
9. auth_algs yaitu algoritma enkripsi autentikasi yang digunakan.
10. wpa yaitu tipe enkripsi yang digunakan.
11. rsn_pairwise yaitu menggunakan AES.

Agar dapat melakukan paket *forwarding*, maka tambahkan perintah pada konfigurasi file sysctl.conf pada direktori sysctl.conf sebagai berikut.

```
sudo nano /etc/sysctl.conf
```

Gambar 4.11 File Konfigurasi Sysctl

```
net.ipv4.ip_forward=1
```

Gambar 4.12 Perintah paket Forwarding

Agar *access point* dapat aktif pada saat *booting* tambahkan perintah pada direktori hostapd sebagai berikut

```
/etc/default/hostapd
```

Gambar 4.13 Direktori Hostapd

```
DAEMON_CONF="/etc/hostapd/hostpad.conf"
```

Gambar 4.14 Perintah Otomasi Hostapd

Untuk memastikan aplikasi Hostapd maka harus melakukan *restart service* agar layanan *access point* berjalan.

4.3.1.3 Mengaktifkan SMS *Gateway*

Pada implementasi mengaktifkan SMS *gateway* yang pertama dilakukan adalah install aplikasi gammu dengan perintah sebagai berikut.

```
sudo apt-get install gammu
```

Gambar 4.15 Perintah Instalisasi Gammu

Kemudian pastikan bahwa modem yang dipakai telah terkoneksi dengan perintah sebagai berikut.

```
lsusb
```

Gambar 4.16 Perintah Inisialisasi Perangkat

Jika modem yang dipakai telah terkoneksi, maka selanjutnya periksa port modem yang digunakan dengan perintah sebagai berikut.

```
dmesg | grep ttyUSB
```

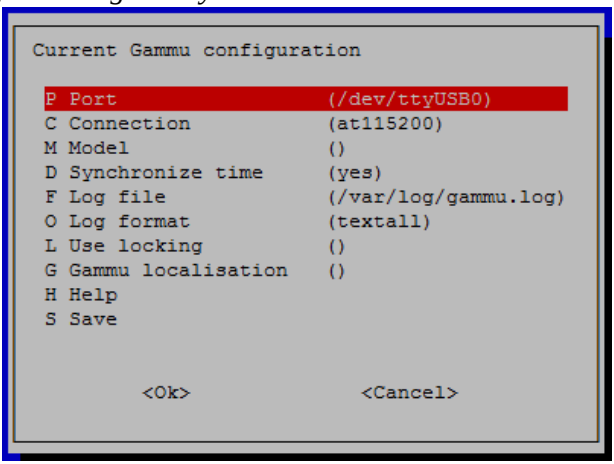
Gambar 4.17 Perintah Insialisasi Port

Kemudian ubah konfigurasi aplikasi gamu dengan perintah sebagai berikut.

```
gammu-config
```

Gambar 4.18 Perintah Konfigurasi Gammu

Maka akan keluar menu pilihan seperti pada Gambar 4.20. Ubah port yang sesuai dengan keterangan hasil dmesg dan sesuaikan *connection* dengan daftar koneksi yang dipakai pada perangkat SMS *gateway*.



Gambar 4.19 Menu Konfigurasi Gammu

4.3.1.4 Implementasi Mengolah Data dan Mengirim Notifikasi

Pada implementasi ini dijelaskan tentang proses dimana sistem mengambil data informasi perangkat pengguna yang terhubung pada log DHCP *server* yaitu pada baris log proses DHCPACK. Pada baris log tersebut sistem dapat mengambil data informasi berupa MAC *address*, IP *address* dan *device name*. Setelah mengambil data informasi perangkat pengguna tersebut sistem akan membandingkan dengan data yang ada pada *database*, jika data tidak terdapat pada *database* maka data perangkat tersebut akan ditambahkan ke dalam *database* atau jika terdapat ketidaksesuaian data informasi MAC *address* maupun IP *address* perangkat, maka data informasi perangkat tersebut akan diperbarui. Kemudian sistem akan melakukan proses pengiriman notifikasi, proses pengiriman notifikasi dilakukan dengan dua layanan yaitu melalui layanan SMS dan *email*.

```
N <- 0
Aturan[] <- Daftar aturan
WHILE true
      READ DHCP Log
       baris <- COUNT kemunculan DHCPACK
       IF baris > N
                 Beda <- baris – N
                Log <- GET DHCPACK terakhir sebanyak
Beda baris
```

Gambar 4.20 *Pseudocode* Mengambil Data Informasi Perangkat

Pada Gambar 4.20 menunjukkan *pseudocode* implementasi mengambil data informasi log DHCP *server* perangkat yang terhubung pada *wireless* router. Pada log DHCP *server* terdapat baris DHCPACK yang merupakan sebuah proses dimana perangkat pengguna yang terhubung kedalam DHCP *server* telah mendapatkan IP *address* yang berarti perangkat tersebut telah terhubung dengan *wireless router*. Pada saat sistem *booting*, nilai *default* variabel N adalah 0, kemudian sistem menghitung terlebih dahulu banyaknya total baris DHCPACK yang ada pada log. Jika

jumlah total baris DHCPACK lebih besar dari nilai variabel N, maka dihitung berapa jumlah perbedaannya yaitu jumlah baris DHCPACK dikurangi dengan nilai variabel N. Baris log DHCPACK yang diambil adalah baris log terakhir yang sesuai dengan jumlah perbedaan baris yang telah dihitung sebelumnya.

```
WHILE all Log not READ
       line <- GET satu baris Log
       pecah <- SPLIT line
       cari <- select count(*) where ip = pecah[6]
and mac = pecah[7]
       IF cari = 0
              INSERT pecah[6], pecah[7], pecah[8] to
database
              id <- select id_pengguna where ip =
pecah[6] and mac = pecah[7]
              FOR all Aturan[] as atur
                     INSERT atur for id to database
```

Gambar 4.21 *Pseudocode* Memasukkan Data Informasi Perangkat

Pada Gambar 4.21 menunjukkan *pseudocode* implementasi memasukkan data informasi perangkat yang terhubung kedalam *database*, yang sebelumnya telah diambil dari log DHCP *server*. Dapat dijelaskan ketika baris DHCPACK baru yang muncul telah diambil, maka setiap satu baris DHCPACK akan dipecah-pecah menurut suku kata, kemudian dibandingkan dengan yang ada pada *database*. Suku kata yang diambil yaitu yang berkaitan dengan data informasi perangkat yaitu MAC *address*, IP *address* dan *device name*. Jika data yang diambil belum terdapat di dalam *database*, maka data informasi perangkat tersebut akan dimasukkan berikut dengan data aturan yang tersedia.

```
ceklagi <- select count(*) where ip_pengguna !=
pecah[6] and mac_pengguna = pecah[7]
       IF ceklagi = 1
              UPDATE ip_pengguna to pecah[6]
             id <- select id_pengguna where ip =
pecah[6] and mac = pecah[7]
               SET all aturan for id to 0
```

Gambar 4.22 *Pseudocode* Memperbarui Data Informasi Perangkat

Pada Gambar 4.22 menunjukkan *pseudocode* implementasi memperbarui data informasi perangkat berupa IP *address*. Implementasi ini dijalankan pada kondisi dimana perangkat yang terhubung memiliki perbedaan data informasi yang terdapat pada *database*, dalam kasus ini MAC *addrees* yang terdaftar menggunakan IP *address* yang berbeda dengan yang terdaftar sebelumnya pada *database*. Maka pada implementasi ini data informasi IP *address* yang digunakan salah satu MAC *address* tersebut akan diperbarui, serta secara otomatis aturan aktif akan kembali ke aturan awal dan sistem akan mengirimkan notifikasi ulang kepada *network administrator*.

```
ceklagi <- select count(*) where ip_pengguna =
pecah[6] and mac_pengguna != pecah[7]
IF ceklagi = 1
         UPDATE mac_pengguna to pecah[7]
         id <- select id_pengguna where ip =
pecah[6] and mac = pecah[7]
        SET all aturan for id to 0
```

Gambar 4.23 *Pseudocode* Memperbarui Data Informasi Perangkat

Pada Gambar 4.23 menunjukkan *pseudocode* implementasi memperbarui data informasi perangkat berupa MAC *address*. Implementasi ini dijalankan pada kondisi dimana perangkat yang terhubung memiliki perbedaan data informasi yang terdapat pada *database*, dalam kasus ini IP *address* yang terdaftar digunakan MAC *address* yang berbeda dengan yang terdaftar sebelumnya pada *database*. Maka pada implementasi ini data informasi MAC address yang menggunakan salah satu IP address tersebut akan diperbarui, serta secara otomatis aturan aktif akan kembali ke aturan awal dan sistem akan mengirimkan notifikasi ulang kepada *network administrator*.

```
SEND sms to admin
UPDATE kirim_sms to 1
SEND email to admin
```

```
UPDATE kirim_email to 1
UPDATE param to 1
```

Gambar 4.24 *Pseudocode* Kirim Notifikasi

Pada Gambar 4.24 menunjukkan *pseudocode* implementasi kirim notifikasi. Ketika proses implementasi mengambil, memasukkan, dan memperbarui data informasi telah selesai, maka dilanjutkan dengan implementasi ini. Dapat dijelaskan bahwa setelah megirim sms kepada admin, sistem memperbarui status kirim_sms pada *database* untuk setiap perangkat yang berarti data informasi perangkat tersebut telah dikirimkan melalui SMS. Begitu juga sebaliknya , setelah sistem mengirim notifikasi email kepada admin, sistem memperbarui status `kirim_email` pada *database* untuk setiap perangkat yang berarti data informasi perangkat tersebut telah dikirim melalui email. Untuk mengirim notifikasi sms, sistem menggunakan perintah yang sesuai dengan format perintah yang dipakai aplikasi gammu. Sedangkan untuk mengirim *email*, sistem menggunakan *framework* PHP Codeingniter. pada salah satu controller untuk meletakkan sebuah fungsi kirim *email* melalui SMTP *server*. Karena implementasi ini dilakukan di lingkungan Teknik Informatika ITS dan dengan pembatasan akses ketika menggunakan SMTP *server* milik penyedia layanan email di luar lingkungan ITS, maka digunakan SMTP *server* ITS pada smtp.its.ac.id dengan port 587.

```
perangkat <- GET device where kirim_email = 0
IF not null
       link <- CREATE link by id_perangkat
       LOAD library email
       SET email config
       SET pengirim, penerima, subject email
       SET pesan + link
       SEND email
       UPDATE parameter to 0
ELSE
       PRINT "kosong"
ENDIF
```

Gambar 4.25 *Pseudocode* Fungsi Kirim Email

Pada Gambar 4.25 menunjukkan *pseudocode* dari fungsi kirim *email*. Dapat dijelaskan jika pada *database* terdapat status `kirim_email` sama dengan 0 maka sistem akan mengirimkan *email* kepada admin yang berisi informasi perangkat terhubung berikut dengan link halaman olah aturannya.

```
UPDATE status_aktif to 0
UPDATE all status_olah for id to 0
UPDATE param to 1
```

Gambar 4.26 *Pseudocode* Perangkat Lama Terhubung

Pada Gambar 4.26 menunjukkan *pseudocode* dimana jika terdapat perangkat yang terhubung tidak menjalani proses implementasi memasukkan data informasi baru maupun memperbarui data, maka sistem akan memperbarui status aktif menjadi 0, status olah menjadi 0 dan parameter menjadi 1 dimana yang berarti perangkat pengguna tidak dapat menggunakan aturan jaringan sebelum melakukan proses verifikasi yang benar terlebih dahulu.

4.3.1.5 Implementasi Mengolah SMS Masuk

Pada implementasi ini akan dijelaskan bagaimana proses sistem melaksanakan perintah yang diterima melalui SMS yang dikirimkan oleh *network administrator*.

```
num <- COUNT sms in Inbox
IF num = 0
       Print "kosong"
ELSE
       sms <- GET sms in Inbox
       nomor <- "nomor hp"
       IF phone number in sms = nomor
              IF sms contain ALLOW word
                     word <- GET sms text
                     pecah2 <- SPLIT word by ' '
                     atur[] <- SPLIT pecah2[2] by
','
```

```
                    IF all atur[] in Aturan[]
                                id <- select *
from pengguna where mdmd like '%pecah2[1]%'
                                FOR all atur[] as
aturatur
                                     IF
status_aktif = 0
                                        UPDATE
status_olah_valid to 1

                                 ELSE
                                       UPDATE
status_olah and status_olah_valid to 1
                                    ENDIF
                                 ENDFOR
                                    UPDATE
last_update
                                    SAVE sms to
file
                                    CLEAR inbox
                                    UPDATE
parameter to 1
                            ELSE
                                    SEND sms
warning to admin
                            ENDIF
                    ELSEIF sms contain RESET
word
                             word <- GET sms text
                              pecah2 <- SPLIT
word by ' '
                              id <- select * from
pengguna where mdmd like '%pecah2[1]%'
                              UPDATE last_update
                              UPDATE status_olah
and status_olah_valid for id to 0
                              SAVE sms to file
                              CLEAR inbox
                              UPDATE parameter to
1
                    ENDIF
             ELSE
                    PRINT "tidak sesuai nomor
admin"
```

```
            ENDIF
            DELETE sms in Inbox
```

Gambar 4.27 *Pseudocode* Mengolah SMS Perintah

Pada Gambar 4.27 menunjukan *pseudocode* implementasi mengolah sms masuk, dimana proses ini adalah membaca pesan SMS yang masuk pada aplikasi gammu. Awal mulai proses implementasi ini yaitu sistem mengambil pesan yang ada pada kotak masuk gammu, kemudian sistem mencari pesan yang hanya dikirimkan oleh nomor telepon yang sesuai dengan milik admin, jika ada maka sistem akan mencari pesan yang berisi kata 'ALLOW' atau 'RESET' dan mulai memecah kalimat pada pesan tersebut. Kemudian sistem akan memeriksa apakah kriteria format pesan tersebut sesuai atau tidak, jika sesuai maka sistem akan memperbarui status data aturan yang diberikan oleh admin pada *database*.

4.3.1.6 Implementasi Mengolah Aturan Iptables

Implementasi mengolah aturan Iptables merupakan proses menerapkan aturan-aturan lalu lintas jaringan yang telah ditentukan sebelumnya oleh admin.

```
CLEAR rule for IPTABLES
SET FORWARD chain to DROP
SET MASQURADE rule
SET block rule for ssh and ping
WHILE true
       temp <- COUNT grep mysql
       WHILE temp = 1
              SLEEP 3 seconds
              temp <- COUNT grep mysql
       ENDWHILE
       temp <- SELECT parameter from database
       WHILE temp = 0
              SLEEP 3 seconds
              temp <- SELECT parameter from database
       ENDWHILE
```

```
CLEAR rule for IPTABLES
SET FORWARD chain to DROP
SET MASQURADE rule
SET block rule for ssh and ping
ips <- SELECT ip where status_olah = 1
FOR all ips as ip
            SET rule for ip
      ENDFOR
      UPDATE parameter to 0
```

Gambar 4.28 *Pseudocode* Mengolah Aturan Iptables

Pada Gambar 4.28 menunjukkan *pseudocode* dari implementasi mengolah aturan Iptables. Pada setiap berjalannya implementasi ini sistem akan menghapus terlebih dahulu aturan-aturan yang ada sebelumnya. Kemudian melihat data dan status-status aturan pada *database* yang menjadi acuan untuk menerapkan aturan pada suatu perangkat. Implementasi ini akan bekerja ketika pada *database* parameter sama dengan 1.

4.3.2 Implementasi Antarmuka

Pada subbab ini menampilkan secara umum antar muka pengguna. Tampilan antarmuka pengguna aplikasi ini berupa halaman *web*. Pengguna dalam sistem ini adalah *network administrator* dan pengguna perangkat yang terhubung dalam jaringan *wireless* router. Interaksi antara pengguna dan perangkat lunak dilakukan pada halaman *web* tersebut. Antarmuka yang diimplementasikan yaitu halaman daftar perangkat, halaman olah aturan perangkat, autentikasi antarmuka admin, verifikasi perangkat pengguna.

4.3.2.1 Implementasi Halaman Daftar Perangkat

Implementaasi halaman daftar perangkat merupakan sebuah antarmuka melalui *web* antara *network administrator* dengan perangkat lunak *wireless* router, dimana pada antarmuka menyajikan informasi detail perangkat yang terhubung kedalam

jaringan *wireless* router. Antarmuka ini dibangun pada *framework* PHP Codeigniter.

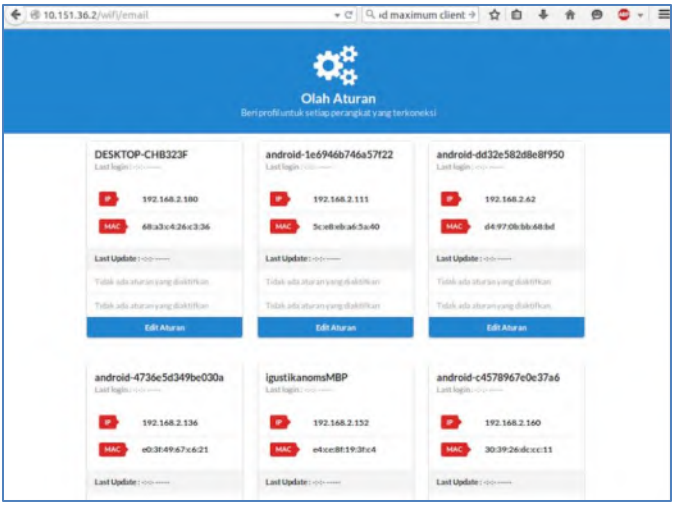
```
data[pengguna] <- CALL getAllPerangkat()
       FOREACH data[pengguna] as perangkat
              daftar[perangkat->id_pengguna] <- CALL
getAturanAktifbyID(perangkat->id_pengguna)
              valid[perangkat->id_pengguna] <- CALL
getAturanAktifValidbyID(perangkat->id_pengguna)
       ENDFOREACH
       data[daftar] <- daftar
       data[valid] <- valid
       temp <- CALL getAturan()
       FOREACH temp as atur
              aturan[atur->id_aturan] <- atur-
>kata_aturan
       ENDFOREACH
       data[aturan] <- aturan
       LOAD view daftar
```

Gambar 4.29 *Pseudocode* Fungsi Index

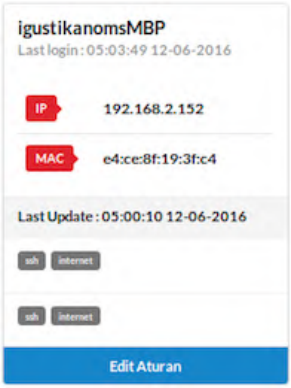
Pada Gambar 4.30 merupakan implementasi halaman daftar perangkat. Pada halaman ini admin dapat melihat daftar perangkat yang terhubung ke dalam jaringan *wireless* router yang dikelolanya, serta dapat melihat informasi detail berupa nama *device*, MAC a*ddress*, IP *address* dan waktu masuk maupun waktu aturan terakir diberikan. Serta teradapat tombol untuk merubah aturan untuk setiap perangkat, dengan menekan tombol edit aturan maka admin akan dilanjutkan ke halaman olah aturan perangkat.

4.3.2.2 Implementasi halaman Olah Aturan Perangkat

Pada Gambar 4.34 merupakan implementasi halaman olah aturan perangkat sebuah antarmuka antara *network administrator* dengan perangkat lunak *wireless* router, dimana pada antarmuka ini *network administrator* dapat memberi atau tidak aturan lalu lintas jaringan yang tersedia. Berikut tampilan halaman olah aturan perangkat.



Gambar 4.30 Tampilan Antarmuka Halaman Olah Aturan



Gambar 4.31 Tampilan Antarmuka Halaman Olah Aturan Per Perangkat

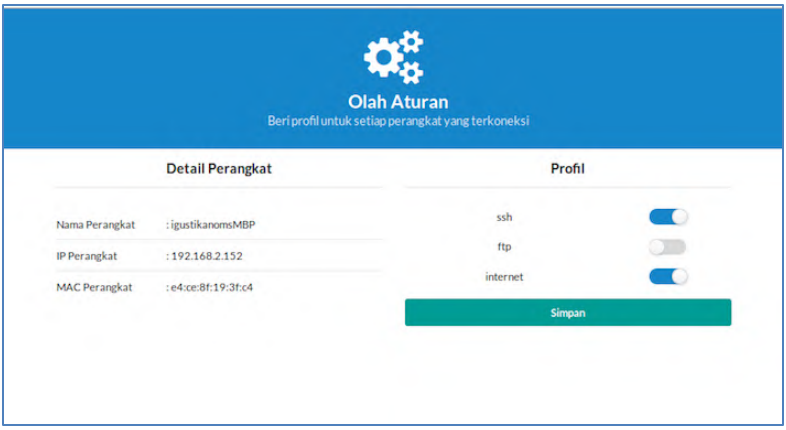
```
data[perangkat] <- CALL getPerangkat(id)
       data[aturan] <- CALL getAturan()
       temp <- CALL getAllAturanbyID(id)
       FOR all temp as tmp
              olah[tmp->id_aturan] <- tmp-
>status_olah_valid
```

```
        ENDFOR
        data[olah] <- olah
        LOAD view olah
```

Gambar 4.32 *Pseudocode* Fungsi Olah Aturan

```
        data[id] <- id
        jumlah <- CALL countAturan()
        perangkat <- CALL getPerangkat(id)
        FOR i=1 to jumlah
               data[aturan] <- i
               IF input->get(i) = 'on'
                      data[status] <- 1
               ELSE
                      data[status] <- 0
               ENDFOR
               INCREMENT i
               IF perangkat->status_aktif = 0
                      CALL updateAturanValid(data)
               ELSE
                      CALL updateAturan(data)
               ENDIF
               UPDATE last_update
        ENFOR
                CALL updateParam(1)
```

Gambar 4.33 *Pseudocode* Fungsi Simpan Aturan



Gambar 4.34 Antarmuka Halaman Olah Aturan

4.3.2.3 Implementasi Autentikasi Antarmuka Admin

Untuk memberikan keamanan dalam mengakses halaman antarmuka pada sisi *network administrator* maka ditambahkan sebuah autentikasi pada *web server* melalui apache. Pertama insall utility apache2 dengan perintah sebagai berikut.

```
apt-get install apache2-utils
```

Gambar 4.35 Perintah Instalisasi Utility Apache

Untuk menambahkan user yang valid dalam melakukan autentikasi maka menggunakan perintah sebagai berikut.

```
sudo htpasswd -c /etc/apache2/.htpasswd anom
```

Gambar 4.36 Perintah Konfigurasi Autentikasi Halaman Admin

Kemudian akan muncul permintaan *password* . Untuk melihat *password* yang telah dimasukkan bisa dilihat pada direktori sebagai berikut.

```
/etc/apache2/.htpasswd
```

Gambar 4.37 Direktori *Password* Autentikasi Halaman Admin

Maka jika isi file dilihat berikut hasilnya.

```
root@raspberrypi:/home/anom/script-wifi# cat /etc/apache2/.htpasswd
anom:$apr1$9R8FV4Fc$09q2Z0uwPCwF6C79DBG15/
```

Kemudian ubah file berikut.

```
/etc/apache2/sites-enabled/000-default.conf
```

Gambar 4.38 File Konfigurasi Autentikasi Halaman Admin

Di dalam tag direktori yang aktif tambahkan konfigurasi untuk melakukan autentikasi seperti Gambar 4.40 berikut.

```
<Directory /var/www/html>
        Options Indexes FollowSymLinks Includes
Multiviews
        AllowOverride All
        Order allow,deny
         allow from all

         AuthType Basic
         AuthName "Area terlarang, hanya
admin yang boleh akses"
         AuthUserFile /etc/apache2/.httpasswd
         Require valid-user
</Directory>
```

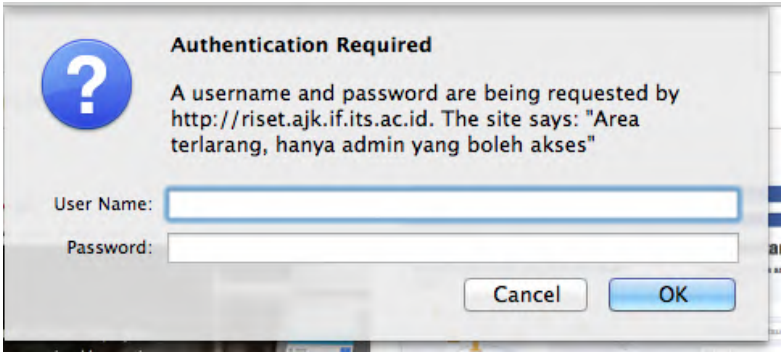
Gambar 4.39 Konfigurasi Autentikasi Halaman Admin

Kemudian *restart* apache sebagai berikut

```
service apache2 restart
```

Gambar 4.40 Perintah Restart Apache

Pada Gambar 4.41 berikut menampilkan permintaan autentikasi.



Gambar 4.41 Menampilkan Permintaan Auntetikasi

4.3.2.4 Implementasi Verifikasi Perangkat Pengguna

Implementasi verifikasi perangkat pengguna merupakan sebuah proses dimana untuk setiap perangkat pengguna yang terhubung diwajibkan untuk memasukkan nomor telepon selular

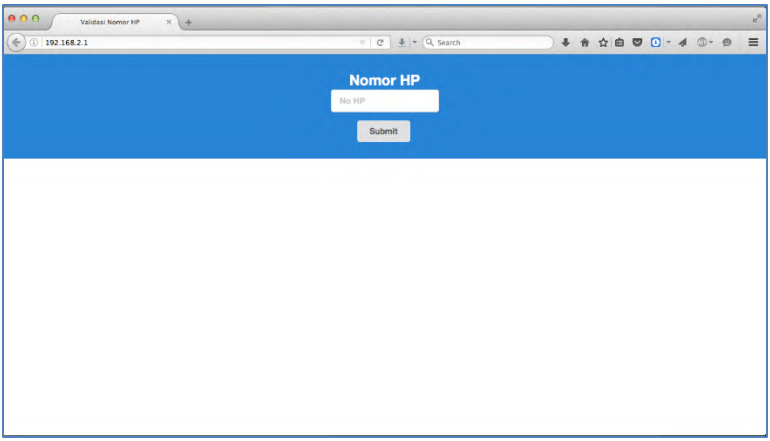
sebagai acuan dalam verifikasi perangkat yang digunakan sesuai dengan yang digunakan perangkat pengguna yang telah terdaftar. Jika nomor yang dimasukkan sesuai dengan nomor yang pertama kali didaftarkan maka aturan lalu lintas jaringan perangkat pengguna dapat langsung digunakan. Tetapi jika nomor yang dimasukkan salah maka pengguna akan diarahkan ke halaman pengisian nomor verifikasi kembali dan Pada Gambar 4.42 merupakan *pseudocode* verifikasi perangkat pengguna.

```
nomor ← GET nomor
       ip ← GET ip
       CREATE database connection
       result ← SELECT all from pengguna where
ip_pengguna = ip
       IF result→num_rows > 0
              row ← result→fetch()
              IF row[“nomor_pengguna”] == null
                     UPDATE nomor_pengguna to nomor,
status_aktif to 1 where ip_pengguna = ip
                     hasil ← GET all aturan where
ip_pengguna = ip
                     WHILE baris = hasil→fetch()
                            UPDATE status_olah to
baris[“status_olah_valid”] where id_olah =
baris[“id_olah”]
                     ENDWHILE
                     UPDATE param to 1
              ELSE IF row[“nomor_pengguna”] = nomor
                     UPDATE last_login
                     UPDATE status_aktif to 1 where
ip_pengguna = ip
                     hasil ← GET all aturan where
ip_pengguna = ip
                     WHILE baris = hasil→fetch()
                            UPDATE status_olah to
baris[“status_olah_valid”] where id_olah =
baris[“id_olah”]
                     ENDWHILE
                     UPDATE param to 1
              ELSE
```

```
                    SET session error to Kesalahan
nomor, validasi lagi ya
                    UPDATE status_aktif to 0 where
ip_pengguna = ip
                    UPDATE status_olah to 0 where
ip_pengguna = ip
                    UPDATE param to 1
              ENDIF
              REDIRECT to http://192.168.2.1
       ELSE
              PRINT "0 results"
       ENDIF
```

Gambar 4.42 *Pseudocode* Verifikasi Perangkat Pengguna

Pada Gambar 4.43merupakan tampilan antarmuka halaman isi nomor verifikasi pada *web browser.*



Gambar 4.43 Halaman Antarmuka Verifikasi Pengguna

LAMPIRAN

Bagian ini merupakan lapiran sebagai dokumen pelengkap dari buku Tugas Akhir, pada bagian ini diberikan informasi mengenai kode sumber dari sistem yang dibuat.

A. Kode Sumber

A.1. Kode Implementasi Mengirim Notifikasi Melalui Layanan Pesan dan Mengolah Aturan Melalui SMS

```
#!/bin/bash

n=0
database="mysql -uroot -padmin123! ta"
aturan=`echo "select id_aturan from aturan" | ${database}`
atur=( $aturan )
aturanaturan=`echo "select kata_aturan from aturan" | ${database}`
aturatur=( $aturanaturan )
aa=`echo "select id_aturan, kata_aturan from aturan" | ${database}`
echo $aa
aturanlengkap=( $aa )
declare -A daftaraturan
for (( i=2; i<${#aturanlengkap[@]}; i+=2 ))
do
        echo "ini aturan"
        echo ${aturanlengkap[i+1]}
        daftaraturan[${aturanlengkap[i+1]}]=${aturanlengkap[i]}
```

```
done


while :
do
        baris=$(grep DHCPACK /var/log/dhcp.log | wc -l)
        if [ $baris -gt $n ]
        then
                echo $baris
                beda=`expr $baris - $n`
                grep DHCPACK /var/log/dhcp.log | tail -n $beda | while read -r line ; do
                        pecah=( $line )
                        echo ${pecah[6]}
                        date +"%T %d-%m-%Y"
                        cari=`echo "select count(*) from pengguna where ip_pengguna='"${pecah[6]}"'
and mac_pengguna='"${pecah[7]}"'" | ${database}`
                        temp=( $cari )
                        if [ ${temp[1]} -eq 0 ]
                        then
                                echo "insert"
                                echo "insert into pengguna(nama_pengguna, ip_pengguna, mac_pengguna)
values ('"${pecah[8]}"', '"${pecah[6]}"', '"${pecah[7]}"')" | ${database}
                                idpengguna=`echo   "select   id_pengguna   from   pengguna   where
ip_pengguna='"${pecah[6]}"' and mac_pengguna='"${pecah[7]}"'" | ${database}`
                                id=( $idpengguna )
                                for (( i=1; i<${#atur[@]}; i+=1 ))
                                do
                                        echo                        "insert                        into
olah_aturan(id_pengguna,id_aturan,status_olah,status_olah_valid)                                values
("${id[1]}","${atur[i]}",0,0)" | ${database}
```

```
                                done
                                gabung=${pecah[6]}${pecah[7]}
                                mdmd=`echo -n $gabung | md5sum`
                                mdbenar=( $mdmd )
                                echo ${mdbenar[0]}

                                if echo -e "Nama : "${pecah[8]}"\nIP : "${pecah[6]}"\nMAC :
"${pecah[7]}"\nMD5 : "${mdbenar[0]} | gammu sendsms text 081231585252 | grep OK; then
                                        date +"%T %d-%m-%Y"
                                        echo "update pengguna set kirim_sms = 1,mdmd =
'"${mdbenar[0]}"' where ip_pengguna = '"${pecah[6]}"' and mac_pengguna = '"${pecah[7]}"'" | $$
                                fi
                                php /var/www/html/wifi/index.php email kirim
                                date +"%T %d-%m-%Y"
                                echo "update parameter set param = 1" | ${database}
                        else
                                ceklagi=`echo "select count(*) from pengguna where
mac_pengguna='"${pecah[7]}"' and ip_pengguna<>'"${pecah[6]}"'" | ${database}`
                                tmp=( $ceklagi )
                                if [ ${tmp[1]} -eq 1 ]
                                then
                                        echo "update"
                                        echo "update pengguna set
ip_pengguna='"${pecah[6]}"',kirim_email=0,kirim_sms=0 where mac_pengguna='"${pecah[7]}"'" |
${database}
                                        idpengguna=`echo "select id_pengguna from pengguna where
ip_pengguna='"${pecah[6]}"' and mac_pengguna='"${pecah[7]}"'" | ${database}`
                                        id=( $idpengguna )
                                        echo "update olah_aturan set
status_olah=0,status_olah_valid=0 where id_pengguna="${id[1]} | ${database}
                                        gabung=${pecah[6]}${pecah[7]}
```

```
                                        mdmd=`echo -n $gabung | md5sum`
                                                                        mdbenar=( $mdmd )
                                        echo ${mdbenar[0]}
                                        if echo -e "Nama : "${pecah[8]}"\nIP : "${pecah[6]}"\nMAC :
"${pecah[7]}"\nMD5 : "${mdbenar[0]} | gammu sendsms text 081231585252 | grep OK;$
                                                echo "update pengguna set kirim_sms = 1,mdmd =
'"${mdbenar[0]}"' where ip_pengguna = '"${pecah[6]}"' and mac_pengguna = '"${pecah[7]}$
                                        fi
                                        php /var/www/html/wifi/index.php email kirim
                                        echo "update parameter set param = 1" | ${database}
                                else

                                        ceklagi=`echo "select count(*) from pengguna where
mac_pengguna<>'"${pecah[7]}"' and ip_pengguna='"${pecah[6]}"'" | ${database}`
                                        tmp=( $ceklagi )
                                        if [ ${tmp[1]} -eq 1 ]
                                        then
                                                echo "update"
                                                echo "update pengguna set
nama_pengguna='"${pecah[8]}"',mac_pengguna='"${pecah[7]}"',kirim_email=0,kirim_sms=0 where
ip_pengguna='"$$
                                                idpengguna=`echo "select id_pengguna from pengguna
where ip_pengguna='"${pecah[6]}"' and mac_pengguna='"${pecah[7]}"'" | ${database}`
                                                id=( $idpengguna )
                                                echo "update olah_aturan set
status_olah=0,status_olah_valid where id_pengguna="${id[1]} | ${database}
                                                gabung=${pecah[6]}${pecah[7]}
                                                mdmd=`echo -n $gabung | md5sum`
                                                mdbenar=( $mdmd )
                                                echo ${mdbenar[0]}
```

```
                                                                                if echo -e
"Nama : "${pecah[8]}"\nIP : "${pecah[6]}"\nMAC : "${pecah[7]}"\nMD5 : "${mdbenar[0]} | gammu sendsms
text 081231585252 | grep OK;$
                                        echo "update pengguna set kirim_sms = 1,mdmd =
'"${mdbenar[0]}"' where ip_pengguna = '"${pecah[6]}"' and mac_pengguna = '"${pecah[7]$
                                        fi
                                        php /var/www/html/wifi/index.php email kirim
                                        echo "update parameter set param = 1" | ${database}
                                else

                                        ceklagi=`echo "select count(*) from pengguna where
mac_pengguna<>'"${pecah[7]}"' and ip_pengguna='"${pecah[6]}"'" | ${database}`
                                        tmp=( $ceklagi )
                                        if [ ${tmp[1]} -eq 1 ]
                                        then
                                                echo "update"
                                                echo "update pengguna set
nama_pengguna='"${pecah[8]}"',mac_pengguna='"${pecah[7]}"',kirim_email=0,kirim_sms=0 where
ip_pengguna='"$$
                                                idpengguna=`echo "select id_pengguna from pengguna
where ip_pengguna='"${pecah[6]}"' and mac_pengguna='"${pecah[7]}"'" | ${database}`
                                                id=( $idpengguna )
                                                echo "update olah_aturan set
status_olah=0,status_olah_valid where id_pengguna="${id[1]} | ${database}
                                                gabung=${pecah[6]}${pecah[7]}
                                                mdmd=`echo -n $gabung | md5sum`
                                                mdbenar=( $mdmd )
                                                echo ${mdbenar[0]}
                                                if echo -e "Nama : "${pecah[8]}"\nIP :
"${pecah[6]}"\nMAC : "${pecah[7]}"\nMD5 : "${mdbenar[0]} | gammu sendsms text 081231585252 | $
```

```
                                                                echo "update pengguna set kirim_sms = 1,mdmd
= '"${mdbenar[0]}"' where ip_pengguna = '"${pecah[6]}"' and mac_pengguna = '"${$
                                                        fi
                                                        php /var/www/html/wifi/index.php email kirim
                                                        echo "update parameter set param = 1" | ${database}
                                                else
                                                        echo "tidak insert atau update tapi konek lagi"
                                                        echo "update pengguna set status_aktif=0 where
ip_pengguna='"${pecah[6]}"' and mac_pengguna='"${pecah[7]}"'" | ${database}
                                                        idpengguna=`echo "select id_pengguna from pengguna
where ip_pengguna='"${pecah[6]}"' and mac_pengguna='"${pecah[7]}"'" | ${database}`
                                                        id=( $idpengguna )
                                                        echo "update olah_aturan set status_olah=0 where
id_pengguna="${id[1]} | ${database}
                                                        echo "update parameter set param = 1" | ${database}
                                                fi
                                        fi
                                fi
                                #echo ${pecah[5]}
                                #sleep 5
                        done
                        n=$baris
                else
                        echo "aman lalu baca sms"
                        tmp=`gammu getallsms | tail -n 1`
                        pecah=( $tmp )
                        if [ ${pecah[0]} -eq 0 ]
                        then
                                echo "kosong"
                        else
```

```
admin="\"+6281231585252\""
ambilnomor=`gammu getallsms | grep Remote`
nomor=( $ambilnomor )
if [ "${nomor[3]}" == "$admin" ];
then
        if gammu getallsms | grep ALLOW; then
                tmp2=`gammu getallsms | grep ALLOW`
                pecah2=( $tmp2 )
                IFS=', ' read -r -a sms <<< "${pecah2[2]}"
                i=0
                for element in "${sms[@]}"
                do
                        echo "$element"
                        if echo ${aturatur[@]} | grep -q -w $element; then
                                i=`expr $i + 1`
                                echo "ada"
                        fi
                done
                if [ ${#sms[@]} -eq $i ]
                then
                        echo "lanjut"
                        tmp=`echo  "select  id_pengguna,status_aktif  from
pengguna where mdmd like '%"${pecah2[1]}"'" | ${database}`
                        dapetid=( $tmp )
                        echo $dapetid
                        for element in "${sms[@]}"
                        do
                                if [ ${dapetid[3]} -eq 0 ]
                                then
                                        echo  "update  olah_aturan  set
status_olah_valid=1 where id_pengguna="${dapetid[2]}" and id_aturan="${daftaraturan[$el$
```

```
                                                        else
                                                                echo    "update    olah_aturan    set
status_olah=1,status_olah_valid=1 where id_pengguna="${dapetid[2]}" and id_aturan="${da$

                                                        fi
                                                done
                                                waktu=`date +"%Y-%m-%d %T"`
                                                echo "update pengguna set last_update = '"$waktu"'
where id_pengguna="${dapetid[2]}";" | ${database}

                                                now=`date +'%Y%m%d-%H:%M:%S'`
                                                gammu getallsms > /home/anom/backup_sms/sms_$now
                                                echo "update parameter set param = 1" | ${database}
                                        else
                                                echo "tidak sama lho"
                                                echo -e "Profil yang dikirimkan salah, kirim lagi
dengan profil yang sesuai" | gammu sendsms text 081231585252
                                        fi

                                #now=`date +'%Y%m%d-%H:%M:%S'`
                                #gammu getallsms > /home/anom/backup_sms/sms_$now
                                #echo "update parameter set param = 1" | ${database}
                                elif gammu getallsms | grep RESET; then
                                        tmp2=`gammu getallsms | grep RESET`
                                        pecah2=( $tmp2 )
                                        tmp=`echo "select id_pengguna from pengguna where mdmd like
'%"${pecah2[1]}"'" | ${database}`
                                        dapetid=( $tmp )
                                        waktu=`date +"%Y-%m-%d %T"`
```

```
                                        echo "update pengguna set last_update = '"$waktu"' where
id_pengguna="${dapetid[1]}";" | ${database}
                                        echo           "update           olah_aturan           set
status_olah=0,status_olah_valid=0 where id_pengguna="${dapetid[1]}";" | ${database}
                                        now=`date +'%Y%m%d-%H:%M:%S'`
                                        gammu getallsms > /home/anom/backup_sms/sms_$now
                                        echo "update parameter set param = 1" | ${database}
                                fi
                        else
                                echo "nomornya beda dari admin"
                        fi
                        gammu deleteallsms 1
                fi
        fi
        #sleep 5
done
```

A.2. Data Uji Coba Respon Mengirim Notifikasi SMS Saat Sistem Mulai Mengolah Data Informasi Hingga Pesan Diterima Admin

JUMLAH CLIENT		SISTEM MENGOLAH	ADMIN	SELISIH WAKTU	WAKTU RATA2
1	anom	0:26:52	0:27:01	9	9
5	igustikanomsMBP	0:48:02	0:48:13	11	10.6
	I-Gustis-iPhone	0:48:24	0:48:37	13	
	DESKTOP-CHB323F	0:47:40	0:47:50	10	
	Mi4i-MiPhone	0:47:51	0:48:01	10	
	android-1e6946b746a57f22	0:48:14	0:48:23	9	
10	DESKTOP-CHB323F	2:03:31	2:03:46	15	10.75
	android-1e6946b746a57f22	2:03:47	2:03:57	10	
	android-dd32e582d8e8f950	2:03:58	2:04:08	10	
	android-4736e5d349be030a	2:04:10	2:04:20	10	
	igustikanomsMBP	2:04:21	2:04:31	10	
	android-c4578967e0e37a6	2:04:32	2:04:42	10	
	I-Gustis-iPhone	2:04:43	2:04:53	10	
	android-32f304f560e7cb16	2:04:54	2:05:05	11	
10	Mi4i-pisanggoreng Mi4i-MiPhone	GAGAL TERHUBUNG			

A.3. Data Uji Coba Respon Mengirim Notifikasi SMS Saat Sistem Mulai Mengolah Data Informasi Hingga Pesan Dikirim Sistem

JUMLAH CLIENT	Nama Klien	SISTEM MENGOLAH	PROVIDER	SELISIH WAKTU	WAKTU RATA2
1	anom	0:26:52	0:27:01	9	9
5	igustikanomsMBP	0:48:02	0:48:13	11	10.6
	I-Gustis-iPhone	0:48:24	0:48:37	13	
	DESKTOP-CHB323F	0:47:40	0:47:50	10	
	Mi4i-MiPhone	0:47:51	0:48:01	10	
	android-1e6946b746a57f22	0:48:14	0:48:23	9	
10	DESKTOP-CHB323F	2:03:31	2:03:46	15	10.75
	android-1e6946b746a57f22	2:03:47	2:03:57	10	
	android-dd32e582d8e8f950	2:03:58	2:04:08	10	
	android-4736e5d349be030a	2:04:10	2:04:20	10	
	igustikanomsMBP	2:04:21	2:04:31	10	
	android-c4578967e0e37a6	2:04:32	2:04:42	10	
	I-Gustis-iPhone	2:04:43	2:04:53	10	
	android-32f304f560e7cb16	2:04:54	2:05:05	11	
10	Mi4i-pisanggoreng Mi4I-MiPhone	GAGAL TERHUBUNG			

A.4. Data Uji Coba Respon Mengirim Notifikasi SMS Saat Pesan Dikirim Sistem Hingga Pesan Diterima Admin

JUMLAH CLIENT		PROVIDER	ADMIN	SELISIH WAKTU	WAKTU RATA2
1	anom	0:27:01	0:27:01	0	0
5	igustikanomsMBP	0:48:13	0:48:13	0	0
	I-Gustis-iPhone	0:48:37	0:48:37	0	
	DESKTOP-CHB323F	0:47:50	0:47:50	0	
	Mi4i-MiPhone	0:48:01	0:48:01	0	
	android-1e6946b746a57f22	0:48:23	0:48:23	0	
10	DESKTOP-CHB323F	2:03:46	2:03:46	0	0
	android-1e6946b746a57f22	2:03:57	2:03:57	0	
	android-dd32e582d8e8f950	2:04:08	2:04:08	0	
	android-4736e5d349be030a	2:04:20	2:04:20	0	
	igustikanomsMBP	2:04:31	2:04:31	0	
	android-c4578967e0e37a6	2:04:42	2:04:42	0	
	I-Gustis-iPhone	2:04:53	2:04:53	0	
	android-32f304f560e7cb16	2:05:05	2:05:05	0	
10	Mi4i-pisanggoreng Mi4I-MiPhone	GAGAL TERHUBUNG			

A.5. Data Uji Coba Respon Mengirim Notifikasi Email Saat Sistem Mulai Mengolah Data Informasi Hingga Pesan Diterima Admin

JUMLAH CLIENT		SISTEM MENGOLAH	ADMIN	SELISIH WAKTU	WAKTU RATA2
1	anom	0:26:52	0:27:11	19	19
5	igustikanomsMBP	0:48:02	0:48:16	14	14.4
	I-Gustis-iPhone	0:48:24	0:48:40	16	
	DESKTOP-CHB323F	0:47:40	0:47:55	15	
	Mi4i-MiPhone	0:47:51	0:48:05	14	
	android-1e6946b746a57f22	0:48:14	0:48:27	13	
10	DESKTOP-CHB323F	2:03:31	2:03:52	21	13.375
	android-1e6946b746a57f22	2:03:47	2:04:01	4	
	android-dd32e582d8e8f950	2:03:58	2:04:12	14	
	android-4736e5d349be030a	2:04:10	2:04:24	14	
	igustikanomsMBP	2:04:21	2:04:34	13	
	android-c4578967e0e37a6	2:04:32	2:04:45	13	
	I-Gustis-iPhone	2:04:43	2:04:56	13	
	android-32f304f560e7cb16	2:04:54	2:05:09	15	
10	MI4I-pisanggoreng Mi4i-MiPhone	GAGAL TERHUBUNG			

A.6. Data Uji Coba Respon Mengirim Notifikasi Email Saat Sistem Mulai Mengolah Data Informasi Hingga Pesan Dikirim Sistem

JUMLAH CLIENT		SISTEM MENGOLAH	PROVIDER	SELISIH WAKTU	WAKTU RATA2
1	anom	0:26:52	0:27:04	12	12
5	igustikanomsMBP	0:48:02	0:48:14	12	11.4
	I-Gustis-iPhone	0:48:24	0:48:37	13	
	DESKTOP-CHB323F	0:47:40	0:47:51	11	
	Mi4i-MiPhone	0:47:51	0:48:02	11	
	android-1e6946b746a57f22	0:48:14	0:48:24	10	
10	DESKTOP-CHB323F	2:03:31	2:03:47	16	10.375
	android-1e6946b746a57f22	2:03:47	2:03:58	11	
	android-dd32e582d8e8f950	2:03:58	2:04:09	11	
	android-4736e5d349be030a	2:04:10	2:04:21	11	
	igustikanomsMBP	2:04:21	2:04:32	11	
	android-c4578967e0e37a6	2:04:32	2:04:43	11	
	I-Gustis-iPhone	2:04:43	2:04:54	11	
	android-32f304f560e7cb16	2:04:54	2:05:06	12	
10	Mi4i-pisanggoreng Mi4i-MiPhone	GAGAL TERHUBUNG			

A.7. Data Uji Coba Respon Mengirim Notifikasi Email Saat Pesan Dikirim Sistem Hingga Pesan Diterima Admin

JUMLAH CLIENT		PROVIDER	ADMIN	SELISIH WAKTU	WAKTU RATA2
1	anom	0:27:04	0:27:11	7	7
5	igustikanomsMBP	0:48:14	0:48:16	2	3
	I-Gustis-iPhone	0:48:37	0:48:40	3	
	DESKTOP-CHB323F	0:47:51	0:47:55	4	
	Mi4i-MiPhone	0:48:02	0:48:05	3	
	android-1e6946b746a57f22	0:48:24	0:48:27	3	
10	DESKTOP-CHB323F	2:03:47	2:03:52	5	2.875
	android-1e6946b746a57f22	2:03:58	2:04:01	3	
	android-dd32e582d8e8f950	2:04:09	2:04:12	3	
	android-4736e5d349be030a	2:04:21	2:04:24	3	
	igustikanomsMBP	2:04:32	2:04:34	2	
	android-c4578967e0e37a6	2:04:43	2:04:45	2	
	I-Gustis-iPhone	2:04:54	2:04:56	2	
	android-32f304f560e7cb16	2:05:06	2:05:09	3	
10	Mi4i-pisanggoreng Mi4i-MiPhone	GAGAL TERHUBUNG			

A.8. Data Uji Coba Respon Sistem Mengolah Aturan Dari Perintah SMS Yang Diterima

JUMLAH ATURAN	ADMIN	SISTEM MENGOLAH	SELISIH WAKTU
1	6:17:56	6:18:31	35
2	6:22:21	6:22:54	33
3	6:25:14	6:25:49	35

A.9. Data Uji Coba Respon Sistem Mengolah Aturan Dari *Web* Antarmuka

JUMLAH ATURAN	ADMIN	SISTEM MENGOLAH	SELISIH WAKTU
1	6:31:28	6:31:28	0
2	6:39:57	6:39:57	0
3	6:42:27	6:42:27	0

BAB VI
PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari tujuan rancang bangun sistem aplikasi serta hasil uji coba yang telah dilakukan pda Tugas Akhir ini. Selain itu juga terdapat beberapa saran untuk pengembangan aplikasi lebih lanjut.

6.1 Kesimpulan

Berdasarkan hasil pengamatan, perancangan, implementasi dan uji coba sistem, maka dapat diambil beberapa kesimpulan dari hasil pembuatan Tugas Akhir ini, yaitu:

1. Sistem autentikasi *provisioning* pada *wireless* router mampu saling terkoordinasi antara *access point*, DHCP *server*, dan layanan pesan untuk mengatur aturan setiap perangkat pengguna yang terhubung.
2. Sistem mampu memberikan layanan autentikasi *provisioning* melalui DHCP *server* pada *wireless* router dengan baik.
3. Proses pengiriman pesan notifikasi perangkat pengguna yang terhubung melalui layanan SMS lebih cepat dibandingkan menggunakan layanan *email* untuk setiap perbedaan jumlah perangkat pengguna yang terhubung dalam waktu bersamaan.
4. Proses mengatur aturan lalu lintas jaringan untuk perangkat pengguna lebih cepat melalui *web* antarmuka dibandingkan melalui layanan SMS.
5. Spesifikasi perangkat *wireless adapter* yang digunakan paling banyak dapat melayani 8 *client*.

6.2 Saran

Berdasarkan hasil pengamatan, perancangan, implementasi, serta hasil uji coba sistem ini, maka diperlukan beberapa saran untuk pengembangan aplikasi lebih lanjut, yaitu:

1. Penggunaan perangkat keras dengan spesifikasi lebih tinggi dapat meningkatkan kinerja sistem ini.
2. Penambahan jumlah perangkat pengguna dan aturan dapat meningkatkan keakuratan hasil uji coba.
3. Penambahan data *profil* pengguna pada *database* serta penyajian *web* antarmuka untuk *sign in* dan *login* pengguna dapat meningkatkan keamanan pada sistem ini.

DAFTAR PUSTAKA

[1] G. K.Sandhu, G. S. Mann dan R. Kaur, "Benefit and security issues in wireless technologies: Wi-fi and WiMax," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 1, no. 4, pp. 976-982, 2013.

[2] M.-k. Choi, R. J. Robles, C.-h. Hong dan T.-h. Kim, "Wireless Network Security: Vulnerabilities, Threats and Countermeasures," *International Journal of Multimedia and Ubiquitous Engineering,* vol. 8, pp. 77-86, 2008.

[3] J. Vollbrecht, "802.11b wireless networking and why it needs authentication," dalam *Wireless LAN Access Control and Authentication*, Ann Arbor, InterlinkNetworks,Inc., 2002, pp. 7-29.

[4] Americas Headquarters, IP Addressing: DHCP Configuration Guide,, San Jose: Cisco Systems, Inc, 2012.

[5] S. Powers, "DNSMasq, the Pint-Sized Super Dæmon!," *Linux Jurnal,* pp. 1-2, 24 February 2015.

[6] J. Malinen, "hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator," p. http://w1.fi/hostapd/, 12 January 2013.

[7] MySQL AB, MySQL Reference Manual, Boston: Free Software Foundation, Inc, 2002.

[8] M. S. V. Gawande dan D. P. R. Deshmukh, "Raspberry Pi Technology," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, no. 4, pp. 37-40, 2015.

[9] TP-LINK TECHNOLOGIES CO., LTD. , User Guide TL-WN322G Wireless G USB Adapter, Shenzhen.

[10] V. K.Katankar dan Dr.V.M.Thakare, "Short Message Service using SMS Gateway," *Veena K.Katankar et. al. /*

(IJCSE) International Journal on Computer Science and Engineering, vol. 2, pp. 1487-1491, 2010.

[11] MIT Lincoln Laboratory, "MIT Lincoln Laboratory: Cyber system & technolog: DARPA Intrusion Detection," MIT Lincoln Laboratory, [Online]. Available: https://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/docs/index.html. [Diakses 23 Mei 2016].

[12] V. Riabov, "Simple Mail Transfer Protocol (SMTP)," pp. 1-23, January 2006.

[13] B. Sharma dan K. Bajaj, "Packet Filtering using IP Tables in Linux," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 4, pp. 320-325, 2011.

BIODATA PENULIS

Penulis dilahirkan di Surabaya, 31 Oktober 1992 merupakan anak keempat dari empat bersaudara. Penulis menempuh pendidikan formal di SD Negeri Kertajaya XII Surabaya, SMP Negeri 19 Surabaya, dan SMA Negeri 5 Surabaya. Pada tahun 2011, penulis melanjutkan S1 di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Selama menempuh pendidikan S1 di Teknik Informatika ITS, penulis menekuni bidang minat Arsitektur Jaringan Komputer.

Penulis aktif sebagai anggota organisasi mahasiswa jurusan yaitu Himpunan Mahasiswa Teknik Computer (HMTC) sebagai Staf Ahli Departemen Kewirausahaan dan Minat Bakat (KMB) dan selama menempuh kuliah, Penulis juga aktif sebagai staf Departemen Pengembangan Sumber Daya Manusia Tim Pembina Kerohanian Hindu (TPKH-ITS). Pada beberapa acara kampus, Penulis juga beberapa sering aktif menjadi panitia, baik sebagai anggota maupun koordinator. Diantaranya sebagai wakil kedua REEVA pada kegiatan Schematics 2013. Penulis dapat dihubungi melalui alamat *email* igustiketutanom@gmail.com.