



**TUGAS AKHIR - KS141501**

## **Pemonitoran Paket Jaringan menggunakan Raspberry Pi**

### **Network Package Monitoring Using Raspberry Pi**

**RIZANDI CHOIRUL ANAM**  
NRP 05211 44000 0163

Dosen Pembimbing  
Bekti Cahyo Hidayanto, S.Si., M.Kom

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

**TUGAS AKHIR - KS141501**

# **Pemonitoran Paket Jaringan menggunakan Raspberry Pi**

**RIZANDI CHOIRUL ANAM**  
NRP 05211 44000 0163

Dosen Pembimbing  
**Bekti Cahyo Hidayanto, S.Si., M.Kom**

**DEPARTEMEN SISTEM INFORMASI**  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

*Halaman ini sengaja dikosongkan*

**FINAL PROJECT - KS141501**

***NETWORK PACKAGE MONITORING USING  
RASPBERRY PI***

**RIZANDI CHOIRUL ANAM  
NRP 05211 44000 0163**

**Supervisor  
Bekti Cahyo Hidayanto, S.Si., M.Kom**

**INFORMATION SYSTEMS DEPARTMENT  
Information Technology and Communication Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**

*Halaman ini sengaja dikosongkan*

**LEMBAR PENGESAHAN**

**NETWORK PACKAGE MONITORING USING  
RASPBERRY PI**

**TUGAS AKHIR**

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Elektro Dan Informasi Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh :

**RIZANDI CHOIRUL ANAM**  
**NRP. 05211440000163**

Surabaya, 21 Januari 2020

**KEPALA  
DEPARTEMEN SISTEM INFORMASI**



**Dr. Mudjahidin, ST., MT.**  
**NIP. 19701010 200312 1 001**

*Halaman ini sengaja dikosongkan*

**LEMBAR PERSETUJUAN**

**NETWORK PACKAGE MONITORING USING  
RASPBERRY PI**

**TUGAS AKHIR**

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh :

**RIZANDI CHORUL ANAM**

**NRP 05211 44000 0163**

Disetujui Tim Penguji : Tanggal Ujian : 10 Oktober 2019  
Periode Wisuda : Maret 2020

**Bekti Cahyo Hidayanto, S.Si., M.Kom.**

**(Pembimbing I)**

**Febriyian Samopa S.Kom., M.Kom., Dr.Eng.** (Penguji I)

**Nisfu Asrul Sani S.Kom., M.Sc.**

**(Penguji II)**





## PEMONITORAN PAKET JARINGAN MENGUNAKAN RASPBERRY PI

**Nama Mahasiswa** : RIZANDI CHOIRUL ANAM  
**NRP** : 0521144100163  
**Departemen** : Sistem Informasi FTIK-ITS  
**Dosen Pembimbing1** : Bekti Cahyo Hidayanto, S.Si,  
M.Kom

### ABSTRAK

*pada era digital yang sedang berkembang sangat pesat ini, kebutuhan akan informasi yang cepat menjadi hal yang cukup krusial bagi setiap kalangan. Terutama pelajar atau mahasiswa dalam melakukan pembelajaran. Dengan adanya internet, informasi akan sangat mudah di dapat dengan cepat dan dapat membantu ekspektasi pembelajaran yang baik. Namun, tak dapat dipungkiri bahwa penyediaan informasi pada internet memang sangat luas dan beragam. Jadi, penyalahgunaan pada saat pembelajaran oleh mahasiswa atau pelajar justru cukup besar. Mereka dapat saja menggunakan internet yang ada untuk kebutuhan lain diluar pembelajaran. Maka dari itu, dengan adanya monitoring jaringan pada saat pembelajaran di kelas dapat menjadi suatu pilihan untuk melihat lalu lintas jaringan penggunaan informasi pada pembelajaran., dengan adanya raspberry pi yang dapat mengambil data jaringan maka dari itu dilakukan pembuatan sistem pada jaringan Sistem Informasi Institut Teknologi Sepuluh Nopember dengan mengambil data pada jurusan sistem informasi dan mengirimkan data tersebut menggunakan aplikasi dan data jaringan tersebut ditampilkan menggunakan dashboard pada server, dengan menggunakan sistem ini administrasi dapat mengawasi penggunaan internet dalam lab ketika pembelajaran berlangsung.*

**Kata Kunci : Wireshark, Raspberry Pi, dashboard,  
Monitoring, Jaringan**

## **NETWORK PACKAGE MONITORING USING RASPERRY PI**

**Student Name : RIZANDI CHOIRUL ANAM**  
**Student ID : 0521144100163**  
**Departement : Information System FTIK-ITS**  
**Supervisor I : Bekti Cahyo Hidayanto, S.Si,  
M.Kom**

### **ABSTRACT**

*In this rapidly growing digital era, the need for fast information is a crucial thing for every community. Especially students or college students in conducting learning. With the internet, information will be very easy to get and have an expectations of good learning. However, it cannot be denied that the provision of information on the internet is indeed very wide and diverse. So, abusing internet purpose during learning by students or college students is actually quite large. They can use internet for other needs other than learning. Therefore, with network monitoring when learning in the classroom can be an option to see the efficiency of the use of information in learning, with the rasperry pi that can retrieve network data, it is made a system on the Information Systems Network of the Ten November Institute of Technology retrieve data on the IKTI lab network and mobilize the data using application and network data displayed using the dashboard on the server, using this system the administration can monitor internet usage in the lab when learning takes place.*

**Keywords : Wireshark, Raspberry Pi, dashboard,  
Monitoring, Network**

## SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Rizandi Choirul Anam  
NRP : 05211440000163  
Tempat / Tanggal lahir : Malang / 26 Mei 1996  
Fakultas / Departemen : FTEIC / Sistem Informasi  
Nomor Telp/Hp/email : 081231918282/  
rizandica@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul

**PEMONITORAN PAKET JARINGAN MENGGUNAKAN RASPBERRY PI**

### **Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.**

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 20 Januari 2019



Rizandi Choirul Anam  
NRP.05211440000163

## **KATA PENGANTAR**

Alhamdulillah atas karunia , rahmat, berkah dan jalan yang diberikan oleh Allah SWT selama ini sehingga penulis dapat diberikan kelancaran dalam menyelesaikan tugas akhir dengan judul :

### **NETWORK PACKAGE MONITORING USING RASPBERRY PI**

Terima kasih atas pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, bantuan dan doa demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Mudjahidin, ST., MT. Selaku Ketua Departemen Sistem Informasi ITS Surabaya.
2. Bapak Bakti Cahyo Hidayanto, S.Si., M.Kom. selaku dosen pembimbing yang meluangkan waktu, memberikan ilmu, petunjuk dan motivasi pada pengerjaan tugas akhir ini.
3. Bapak Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom. dan Bapak Nisfu Asrul Sani S.Kom., M.Sc. selaku dosen penguji yang telah memberikan saran dan masukan untuk perbaikan tugas akhir ini.
4. Orang tua penulis yang telah mendoakan dan mendukung dalam pengerjaan tugas akhir ini.
5. Seluruh dosen Departemen Sistem Informasi yang telah memberikan ilmu yang sangat bermanfaat bagi penulis.
6. Rekan-rekan OSIRIS yang telah berjuang bersama dalam menjalani perkuliahan di Departemen Sistem Informasi ITS.
7. Berbagai pihak yang membantu dalam penyusunan Tugas Akhir ini dan belum dapat disebutkan satu per satu dengan dukungan, semangat dan kebersamaan.

Penyusunan laporan ini masih sangat jauh dari kata sempurna, untuk itu saya sangat terbuka apabila adanya kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, Juli 2019  
Penulis,

(Rizandi Choirul Anam)

## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	<b>xii</b>
<b>DAFTAR ISI</b> .....	<b>xiv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xviii</b>
<b>DAFTAR TABEL</b> .....	<b>xx</b>
<b>DAFTAR KODE</b> .....	<b>xxii</b>
<i>Halaman ini sengaja dikosongkan</i> .....	<b>1</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Manfaat Penelitian .....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Penelitian Sebelumnya .....	5
2.2 Dasar Teori.....	6
2.2.1 <i>Raspberry Pi</i> .....	6
2.2.2 <i>Wireshark</i> .....	7
2.2.3 <i>Tcpdump</i> .....	7
2.2.4 <i>Cron</i> .....	7
2.2.6 <i>Open ID</i> .....	8
2.2.7 <i>Windows Task Scheduler</i> .....	8
2.2.8 <i>Tableau</i> .....	8
<b>BAB III METODOLOGI</b> .....	<b>9</b>
3.1 Studi Literatur .....	10
3.2 Implementasi.....	10
3.2.1 <i>Data Monitoring</i> .....	11
3.2.2 <i>Monitoring Jaringan</i> .....	12
3.2.3 <i>Pengambilan Data Monitoring</i> .....	12
3.2.4 <i>Hipotesa Awal</i> .....	12
3.2.5 <i>Spesifikasi Hardware</i> .....	13
3.2.6 <i>Penyusunan Buku Tugas Akhir</i> .....	13
3.2.7 <i>Skema Sistem Monitoring</i> .....	13
<b>BAB IV PERANCANGAN SISTEM</b> .....	<b>17</b>
4.1 Konfigurasi server.....	17
4.2 Pengambilan Data .....	18

4.2.1	Pengambilan data pada jaringan .....	18
4.2.2	Automisasi Pengambilan data.....	19
4.3	Transfer Data .....	19
4.3.1	Pengiriman data dari Sensor ke dalam Server.....	19
4.3.2	Automasi Pengiriman Data .....	20
4.4	Perubahan format Data .....	20
4.5	Dashboard Model.....	20
4.6	Automasi Pengambilan dan Pengiriman Data .....	21
4.7	Desain Usecase .....	21
4.7.1	Masuk Sistem Dashboard .....	21
4.7.2	Tambah Data.....	22
4.8	Ekspektasi Data.....	24
<b>BAB V</b>	<b>IMPLEMENTASI .....</b>	<b>25</b>
5.1	Spesifikasi Perangkat Keras.....	25
5.2	Perangkat lunak.....	26
5.2.1	Pengambilan data jaringan dengan Wireshark... ..	26
5.2.2	Automasi pengambilan dan pengiriman data menggunakan Cron pada sensor .....	27
5.2.3	Automasi penerimaan data pada komputer menggunakan <i>Task Scheduler</i> .....	28
5.2.4	Konversi ekstensi data Paket Jaringan.....	29
5.2.4	Upload data pada aplikasi dashboard <i>Tableau</i> ..	30
<b>BAB VI</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>33</b>
6.1	Pengujian menggunakan sistem Transfer file via FTP. ..	33
6.2	Pengujian automasi sistem dengan menggunakan Tshark.....	34
6.3	Pengujian perintah melakukan <i>command</i> pada linux dengan menggunakan SSH.....	35
6.4	Pengujian dengan menggunakan webservice .....	36
6.5	Penggunaan Aplikasi Dashboard Kibana .....	37
6.6	Kode Pada Server.....	38
6.7	Skenario Implementasi .....	47
6.8	Hail Uji Pengambilan dan Konversi Data .....	52
6.9	Hasil Biaya Realisasi.....	54
6.10	Pengujian Sistem dan Sub-sistem monitoring.....	54
<b>BAB VII</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>59</b>
7.1	Kesimpulan .....	59

7.2 Saran.....	59
<b>DAFTAR PUSTAKA .....</b>	<b>61</b>
<b>LAMPIRAN .....</b>	<b>63</b>
Lampiran A – Source Code Python Web .....	63



*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 3.1 Metodologi Penelitian .....	9
Gambar 3.2 Simulasi Sistem Monitoring Jaringan.....	11
Gambar 3.3 Skema Sistem Monitoring Jaringan.....	14
Gambar 4.1 Skema Raspberry Pi dalam jaringan .....	18
Gambar 4.2 Skema Raspberry Pi transfer ke Server .....	19
Gambar 4.3 Rancangan dashboard data monitoring .....	21
Gambar 4.4 Use Case akses Sistem Monitoring .....	22
Gambar 4.5 Use Case penambahan data monitoring .....	23
Gambar 4.5 list data monitoring jaringan .....	24
Gambar 5.1 Penggunaan Software Wireshark.....	27
Gambar 5.2 Penjadwalan pengambilan data raspberry pi .....	28
Gambar 5.3 List command crontab.....	29
Gambar 5.4 Editor command line pada program crontab.....	29
Gambar 5.4 Persentase Penggunaan Protokol pada router .....	30
Gambar 5.5 Waktu rata – rata akses user .....	31
Gambar 5.6 Waktu rata – rata akses router.....	32
Gambar 6.1 Command line putty pada cmd server.....	33
Gambar 6.2 Command Line mendapatkan file dari sensor .....	34
Gambar 6.3 Tampilan konfigurasi SSH pada sensor.....	35
Gambar 6.4 Tampilan awal dari website berbasis Python.....	37
Gambar 6.5 Hasil Pengambilan data jaringan .....	53

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya .....	5
Tabel 5.1 Daftar penggunaan perangkat lunak pada sistem....	26
Tabel 6.1 Daftar skenario implementasi sistem.....	48
Tabel 6.2 Realisasi Raspberry Pi .....	54

*Halaman ini sengaja dikosongkan*

## DAFTAR KODE

Kode 1. 1 .....	38
Kode 1. 2 .....	41
kode 1. 3 .....	42
kode 1. 4 .....	43
kode 1. 5 .....	44
kode 1. 6 .....	44
kode 1. 7 .....	45
kode 1. 8 .....	46
kode 1. 9 .....	46

*Halaman ini sengaja dikosongkan*





# **BAB I**

## **PENDAHULUAN**

Pada bagian pendahuluan ini, akan dijelaskan mengenai latar belakang masalah yang akan diselesaikan, Batasan masalah, tujuan serta manfaat yang dihasilkan dari Tugas Akhir ini.

### **1.1 Latar Belakang**

Pada Jaman Modern ini, kebutuhan atas perangkat lunak dan keras tidak terpisahkan dari kehidupan sehari – hari, dengan perkembangan Teknologi Informasi yang begitu pesat maka dunia internet ikut berperan dalam pertumbuhan Teknologi Informasi. Banyak aplikasi yang beredar di internet begitu pula dengan informasi – informasi yang dapat diakses dengan mudah di internet, namun di samping kemudahan tersebut ada dampak negatif yang dihasilkan salah satunya dari sisi keamanan.

Jaringan dapat didefinisikan sebagai sebuah kelompok, dimana kelompok tersebut terdiri atas 2 komputer atau lebih. Dimana jaringan-jaringan tersebut dapat mengakses jaringan yang lebih besar atau sama besarnya yang disebut dengan internet. Jaringan-jaringan tersebut dapat memudahkan pengguna dalam melakukan banyak dan beraktivitas, namun di tengah kemudahan tersebut adanya keamanan-keamanan yang memiliki celah.

Oleh karena itu diperlukan sebuah software yang digunakan dalam memonitoring sebuah jaringan dimana aktivitas pada jaringan perlu diketahui oleh administrator jaringan dan hasil dari monitoring dapat ditampilkan dan dapat dianalisa oleh administrator jaringan.

## 1.2 Perumusan Masalah

Beberapa masalah yang diselesaikan pada tugas akhir ini adalah :

1. Bagaimana data monitoring jaringan didapatkan.
2. Bagaimana data yang didapatkan raspberry pi dikirimkan ke server.
3. Bagaimana visualisasi data jaringan yang didapatkan oleh raspberry pi pada server.

## 1.3 Batasan Masalah

Dari rumusan masalah dalam pengerjaan Tugas Akhir ini adalah :

1. Penelitian ini akan sebatas konsep yang direpresentasikan
2. Penelitian ini akan menggunakan tools seperti Wireshark (untuk pengambilan data pada jaringan lokal) dan Aplikasi *tableau* (untuk ilustrasi dashboard yang dihasilkan).
3. Dapat menjadi acuan apabila model ini diterapkan menjadi sistem utuh yang dapat didayagunakan.

## 1.4 Tujuan Penelitian

Tujuan dari Tugas Akhir ini adalah :

1. Membuat *prototype* monitoring jaringan lokal menggunakan Raspberry Pi.
2. Mengambil data jaringan menggunakan *wireshark* aplikasi pengambilan aktivitas pada jaringan lokal.
3. Membuat sistem transfer data dari raspberry pi ke dalam server menggunakan web service *base*.
4. Memberikan log penggunaan jaringan lokal kepada administrator dan menampilkannya dalam bentuk dashboard.

## **1.5 Manfaat Penelitian**

Tugas Akhir ini diharapkan dapat memberikan manfaat antara lain :

1. Dengan adanya model ini, diharapkan dapat menjadi acuan dalam pembangunan / pengembangan dari sistem ini.
2. Memberikan arahan untuk ide-ide baru yang dapat dimanfaatkan di masa depan beriringan dengan adanya sistem ini nanti ataupun pengembangan dari penelitian ini.

*Halaman ini sengaja dikosongkan*

## BAB II

### TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan apa saja yang dilakukan pada tugas akhir ini, berikut ini akan dipaparkan apa saja konsep serta *tools* yang digunakan ataupun diterapkan. Adapun penerapan konsep dan *tools* yang akan dilakukan pada tugas akhir ini adalah sebagai berikut.

#### 2.1 Penelitian Sebelumnya

Berikut ini adalah penelitian lain yang sudah dilakukan oleh beberapa peneliti terkait dengan judul pemodelan sistem ini. Pada contoh studi literature ini, penulias akan mengacu pada beberapa aspek dari penelitian ini.

*Tabel 2.1 Penelitian Sebelumnya*

No.	Peneliti	Judul Penelitian	Keterkaitan dengan Penelitian
1	Pallavi Asrodia and Hemlata Patel	Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis	Penggunaan software <i>wireshark</i> dan metode pengambilan dengan menggunakan <i>tcpdump</i>
2	Yusuf Turk, Onur Demir, Sezer Goren	Real Time Wireless Packet Monitoring with Raspberry Pi Sniffer	Penyusunan sistem pengambilan paket data berawal dari sistem yang digunakan pada Paper ini

## 2.2 Dasar Teori

Pada bagian ini akan dijelaskan mengenai teori-teori yang digunakan untuk mendukung penelitian tugas akhir ini.

### 2.2.1 *Raspberry Pi*

Raspberry Pi adalah mini-komputer yang dimana Raspberry Pi memiliki kemampuan hampir sama dengan komputer, Raspberry Pi bersifat Open Source dikarenakan target dari Raspberry Pi sebagai alat untuk meningkatkan kemampuan programming seseorang dan juga dapat digunakan sebagai alat yang menggunakan beberapa sensor seperti pada Arduino dan kelebihan Raspberry Pi dapat memberikan display output tidak seperti Arduino. Raspberry Pi memiliki keterbatasan juga seperti kecilnya ram dan processor yang tidak sebagus komputer tapi untuk menjalankan program dan membuat program Raspberry Pi dapat melakukannya. Pembuat dari Raspberry Pi bernama Eben Upton dimana tujuan dia dalam membuat Raspberry Pi adalah bagaimana semua orang dapat membuat program atau menjalankan hal-hal sederhana yang komputer dapat lakukan dan alat tersebut sedikit menggunakan energi dan murah. Raspberry Pi juga dilengkapi sensor-sensor seperti pada alat micro-controller yaitu Arduino.

Pada Tugas Akhir ini Raspberry Pi digunakan untuk pengambilan data, dimana data tersebut didapatkan pada jaringan dimana aktivitas-aktivitas yang terjadi pada jaringan direkam kemudian data tersebut dibuat real-time dashboard agar dapat dilihat perkembangannya secara real time. Raspberry Pi digunakan sebagai alat untuk memonitoring jaringan dikarenakan subjek jaringan yang akan diimplementasikan adalah jaringan sederhana sehingga single-purpose machine dapat melakukan monitoring

jaringan, dan dikarenakan jaringan dapat di monitoring dengan single purpose machine, maka Raspberry Pi merupakan pilihan yang tepat terutama dalam efisiensi biaya.

### 2.2.2 Wireshark

Wireshark adalah *packet analyzer*. Wireshark digunakan untuk memecahkan masalah pada jaringan dan digunakan untuk menganalisis [1], Nama mula dari Wireshark adalah Ethereal pada tahun 2006 nama Ethereal berganti menjadi Wireshark dikarenakan masalah merk dagang. Wireshark adalah software lintas platform, dengan menggunakan pcap untuk menangkap paket jaringan. Wireshark adalah perangkat lunak yang dapat mengetahui struktur protokol dari jaringan yang berbeda – beda, dan dengan mengetahui struktur Wireshark dapat menampilkan enkapsulasi dan *fields* beserta artinya dari paket yang berbeda yang telah ditentukan dari beberapa jaringan protokol.

### 2.2.3 Tcpdump

Tcpdump adalah *packet analyzer* yang digunakan melalui basis perintah dari *command line* [2], Tcpdump memperbolehkan pengguna untuk *intercept* dan menampilkan TCP/IP dan paket lainnya yang dikirim dan diterima dalam jaringan dimana komputer terpasang pada jaringan tersebut.

### 2.2.4 Cron

Cron adalah perangkat lunak utilitas, Cron mempunyai fungsi untuk mengotomasi pekerjaan agar dapat bekerja pada waktu yang telah ditentukan oleh pengguna, Cron adalah perangkat lunak yang berjalan diatas sistem operasi berbasis UNIX, Pengguna dapat menentukan jadwal pekerjaan yang akan dijalankan

oleh cron. Cron dapat menjadwalkan pekerjaan untuk dikerjakan pada waktu, tanggal, atau interval waktu yang telah ditentukan oleh pengguna. Selain program Cron juga dapat menjadwalkan dan melakukan perintah yang berbasis dari *command line* yang biasa dijalankan melalui Terminal.

### 2.2.6 Open ID

Open ID adalah sistem yang dapat mensentralisasikan otorisasi user. Open ID memberikan akses kepada pengguna untuk otorisasi dengan menggunakan 1 database saja, sehingga penggunaan pada banyak website dapat hanya menggunakan satu akun yang terdaftar saja. Open ID sekarang banyak digunakan oleh website besar maupun kecil untuk otorisasinya dikarenakan kemudahan pengguna dalam melakukan transaksi pada website dan website mendapatkan cukup informasi mengenai identitas pengguna dengan menggunakan Open ID.

### 2.2.7 Windows Task Scheduler

Windows Task Scheduler adalah perangkat lunak utilitas untuk sistem operasi windows. Windows Task Scheduler mempunyai fungsi yang sama dengan Cron perbedaan terletak pada tampilan yang diberikan dan sistem operasi.

### 2.2.8 Tableau

Tableau adalah perangkat lunak yang berfungsi menghasilkan visualisasi data interaktif. Tableau dapat disebut juga sebagai perangkat lunak desktop dalam pembuatan Dashboard, Penggunaan Tableau seringkali difokuskan pada penggunaan *business intelligence*.



### BAB III

## METODOLOGI

Pada bab ini dijelaskan mengenai metodologi yang digunakan dalam penyelesaian tugas akhir yang akan dilakukan. Metodologi ini dibutuhkan sebagai panduan pengerjaan Tugas Akhir yang sistematis.



*Gambar 3.1 Metodologi Penelitian*

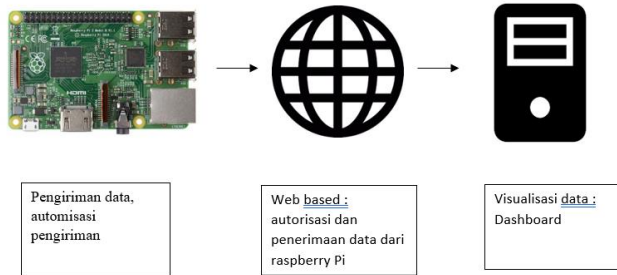
### 3.1 Studi Literatur

Pada tahap ini dilakukan pencarian serta pemahaman informasi serta literatur yang berkaitan dengan penelitian yang dilakukan. Pemahaman mengenai monitoring jaringan, teori dan konsep *wireshark* yang menjadi salah satu *tool* utama dalam pemantauan jaringan.

### 3.2 Implementasi

Pada tahap ini dilakukan Analisa sistem jaringan yang akan dibuat, yaitu bagaimana struktur jaringan dalam subjek, bagaimana data monitoring diambil menggunakan Raspberry Pi, Bagaimana adanya komunikasi antara Raspberry Pi dengan komputer. Ada tahap analisa desain sistem ini yang harus dilakukan adalah melakukan analisa dan mendaftar kebutuhan fungsional apa saja yang akan di buat, kebutuhan fungsional tersebut terbagi menjadi empat yaitu :

1. Bagian Desain Jaringan dan Tempat Aplikasi
  - Desain Jaringan Subjek
  - Menganalisa penempatan Raspberry Pi
2. Bagian Pengolahan Data
  - Mengambil data hasil monitoring jaringan
3. Bagian Penyimpanan Data
  - Membuat tempat penyimpanan data hasil monitoring
4. Bagian Visualisasi Data
  - Mengambil data monitoring dari Raspberry Pi dikirim ke Komputer
  - Melakukan visualisasi data monitoring



**Gambar 3.2 Simulasi Sistem Monitoring Jaringan**  
 (<https://images.app.goo.gl/IznnRiPmcL44Ai3G8>)

### 3.2.1 Data Monitoring

Pada tahap ini akan dimulai untuk melakukan perancangan sistem sehingga raspberry pi dapat melakukan monitoring jaringan dan mengambil data dan memasukkan dalam database berupa log dari hasil monitoring. Pada Gambar 3.2 adalah gambaran rancangan arsitektur sistem yang akan diterapkan.

Dalam pengimplementasian dari Raspberry Pi dalam jaringan adalah di setiap sub – sub jaringan diberikan Raspberry Pi sebagai sensor untuk menangkap aktivitas jaringan, kemudian data tersebut diberikan kepada komputer, dimana ada hubungan diantara computer dengan raspberry pi dan pada hubungan antara komputer dengan raspberry pi dalam memberikan data diperlukan adanya sekuritas atau *authenticate*, setelah dipastikan dari masing-masing raspberry pi data tersebut diolah untuk dijadikan dashboard

### 3.2.2 Monitoring Jaringan

Dalam tahap ini jaringan yang akan dimonitoring diaplikasikan Raspberry Pi kedalam jaringan dimana tugas dari Raspberry pi sebagai sensor untuk merekam aktivitas-aktivitas yang dilakukan dalam jaringan oleh pengguna jaringan dari yang terautentikasi dan yang tidak. Hasil dari monitoring tersebut dihasilkan menjadi sebuah data log dari monitoring aktivitas dan diberikan kepada komputer inti untuk diproses. Raspberry pi digunakan dari masing-masing sub-jaringan sehingga raspberry pi yang digunakan lebih dari 1 sensor, dan sensor-sensor tersebut berhubungan dengan 1 komputer inti yang dimiliki oleh administrasi jaringan.

### 3.2.3 Pengambilan Data Monitoring

Data yang diperoleh dari jaringan oleh sensor tersebut mengirimkan data-data hasil monitoring mereka, disini terjadi komunikasi antara sensor pada masing-masing jaringan dengan komputer inti dimana dalam komunikasi tersebut digunakan autentikasi untuk menghindari adanya kebocoran komunikasi dan adanya gangguan luar dari pihak terotorisasi. Kemudian setelah masing-masing terotorisasi data hasil monitoring diperoleh oleh komputer inti dan data tersebut dikumpulkan menjadi satu.

### 3.2.4 Hipotesa Awal

Pada tahapan ini direncanakan hasil akhir yang dianalisa sebelumnya oleh peneliti mengenai Tugas Akhir, Hipotesa awal yang didapatkan setelah Analisa awal adalah raspberry pi mengirimkan data pada masing – masing jaringan, kemudian adanya autentikasi antar komputer dengan raspberry pi dalam pemberian data dan data dapat ditampilkan dalam bentuk dashboard

### 3.2.5 Spesifikasi Hardware

hardware yang digunakan dalam pengimplementasian monitoring paket dalam jaringan single-purpose adalah Raspberry Pi 3 Model B dimana spesifikasi dari hardware tersebut seperti dibawah ini:

SoC	:Broadcom BCM2837
CPU	:4x ARM Cortex-A53, 1.2GHz
GPU	:Broadcom VideoCore IV
RAM	:1GB LPDDR2 (900 MHz)
Networking	:10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth	:Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage	:microSD
GPIO	:40-pin header, populated
Ports	:HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

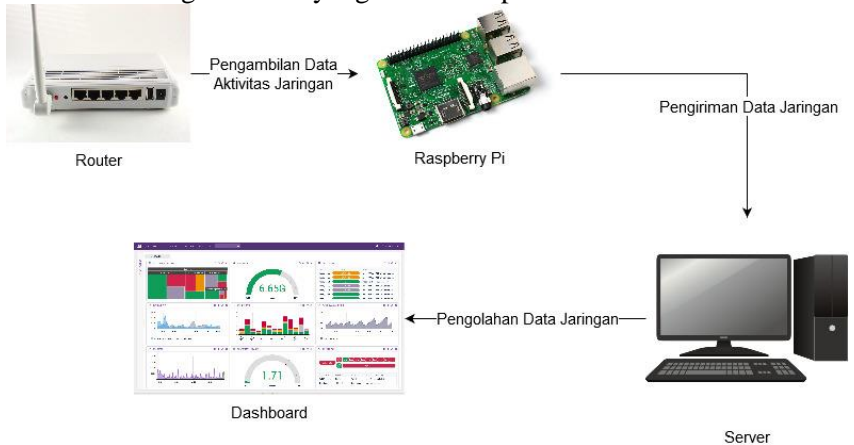
### 3.2.6 Penyusunan Buku Tugas Akhir

Pada tahap ini, akan disusun pembuatan buku tugas akhir sebagai dokumentasi dari pengerjaan tugas akhir yang akan dilakukan. Selain menjadi acuan sebagai penerapan model ke implementasi, juga dapat menjadi panduan apabila ada pengembangan model lebih lanjut ataupun penelitian serupa.

### 3.2.7 Skema Sistem Monitoring

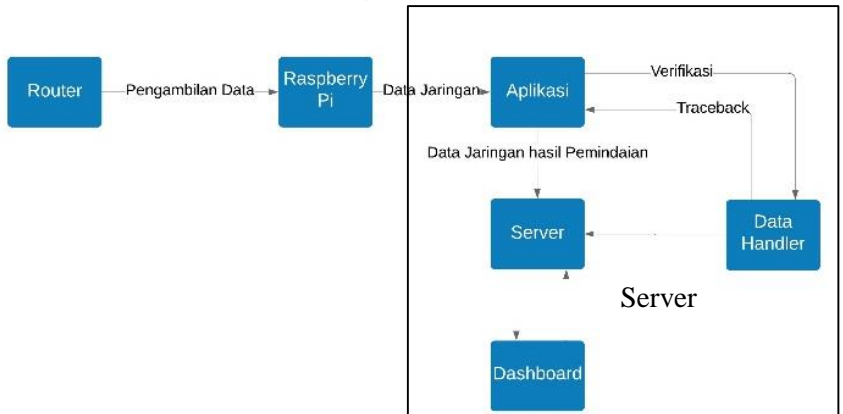
Pada tahap ini dilakukan penyusunan sistem yang akan digunakan sehingga raspberry pi dapat melakukan pengambilan data jaringan dan mengirimkan data

tersebut ke dalam server. Gambar 3.3 merupakan rancangan sistem yang akan di implementasikan.



**Gambar 3.3 Skema Sistem Monitoring Jaringan**

()



**Gambar 3.4 Skema Sistem dan sub Sistem Monitoring Jaringan**

Pada Gambar 3.3 terlihat bahwa Raspberry Pi mengambil data jaringan pada *router*, kemudian langsung mengirimkan data dan ditangkap oleh server, Kemudian data yang telah ditangkap tersebut di kirim ke server dan dijadikan 1 log dari hasil jaringan dan

setelah data didapat secara periodik aplikasi dashboard yang terdapat pada server yang akan menampilkan hasil data yang diperoleh dari Raspberry Pi pada router lokal. Fungsi dari data handler disini berguna untuk mengecek apakah data yang akan dikirimkan sudah terkirim apabila belum terkirim maka akan di *request* kan lagi untuk pengiriman data yang gagal terkirim tersebut.

Pengambilan data pada Raspberry Pi menggunakan *tshark* dilakukan sebanyak 15 detik sekali dan lama pemindaian jaringan 1 menit. Data yang telah diambil oleh Raspberry Pi dipindahkan ke dalam server melalui jaringan lan dengan menggunakan *port* 22, Kemudian data yang diambil akan di gabungkan menjadi satu dan file tersebut akan di konversikan menjadi file yang dapat dibaca oleh aplikasi dashboard sehingga data yang diambil dapat ditampilkan dalam bentuk visual dan mudah untuk dilihat admin jaringan.

*Halaman ini sengaja dikosongkan*



## **BAB IV**

### **PERANCANGAN SISTEM**

Bab ini akan membahas mengenai perancangan alur beberapa hal yang digunakan dalam proses pembuatan sistem monitoring network sesuai dengan alur yang telah dijelaskan pada bab 3. Dalam bab ini menjelaskan perancangan proses pengambilan data dan desain sistem.

#### **4.1 Konfigurasi server**

Pada sub bagian ini akan dijelaskan konfigurasi sistem server pada Raspberry Pi dengan klien yang menggunakan Sistem Operasi Window.

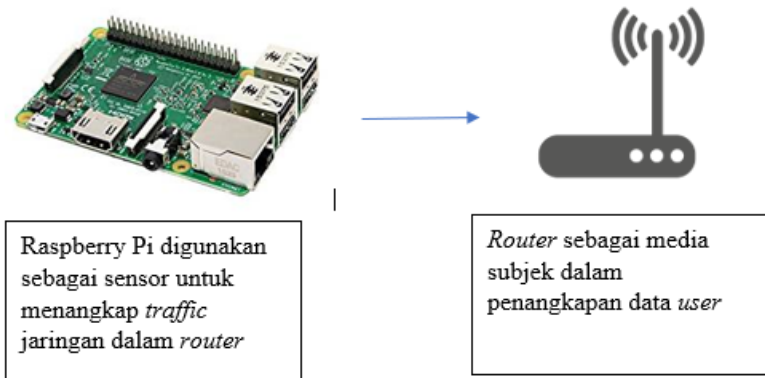
Dalam pembuatan server diputuskan bahwa penggunaan server pada Raspberry Pi dikarenakan sifat dari sistem operasi Raspberry Pi adalah raspbian dimana berbasis linux dan sifat dari linux adalah mudah dalam modifikasi. Window digunakan sebagai klien dikarenakan konfigurasi server pada window mengalami proses yang rumit dan perlu akses – akses khusus dalam konfigurasi server tersebut. Konfigurasi Server pada Raspberry Pi dapat diakses dengan mengaktifkan konfigurasi server yang terdapat pada raspberry pi yang terletak pada konfigurasi raspberry pi di menu.

## 4.2 Pengambilan Data

Pada sub bagian ini akan dijelaskan sistem pengambilan data dari jaringan melalui sensor

### 4.2.1 Pengambilan data pada jaringan

Langkah pertama dalam pengambilan data memasang raspberry pi dalam jaringan lokal yang akan digunakan dapat menggunakan Lan atau secara *wireless* pada gambar 4.1 dijelaskan skema jaringan lokal dengan raspberry pi :



**Gambar 4.1 Skema Raspberry Pi dalam jaringan**

(<https://images.app.goo.gl/kgReqRoBrcv3A67r6>)

Setelah Raspberry Pi terkoneksi dengan router lokal jaringan perlu diambil data *traffic* pengguna jaringan, pengambilan data jaringan menggunakan software *tcpdump* dimana software tersebut dapat mengambil data jaringan yang melalui hardware yang terkoneksi pada jaringan router tersebut. Setelah data pada jaringan router lokal didapatkan data tersebut di simpan dalam folder sementara pada raspberry pi

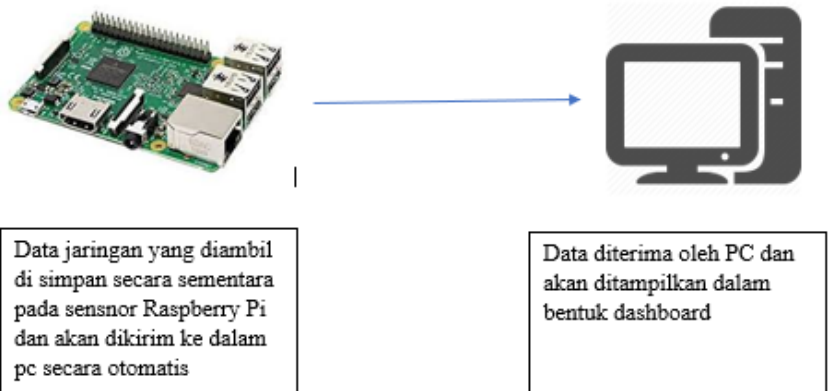
### 4.2.2 Automisasi Pengambilan data

Pada langkah ini pengambilan data yang dilakukan melalui otomasi program, program yang digunakan adalah *cron* dimana pengambilan data dari jaringan melalui sensor raspberry pi di automatisasi dalam waktu yang telah ditentukan

## 4.3 Transfer Data

Pada sub bagian ini akan dijelaskan proses transfer data dari sensor kedalam komputer

### 4.3.1 Pengiriman data dari Sensor ke dalam Server



**Gambar 4.2 Skema Raspberry Pi transfer ke Server**

(<https://images.app.goo.gl/yZev2kz4BefD6ARN9>)

Dalam transfer data digunakan software *putty* menjadi sarana transfer data antara sensor dengan PC (*Personal Computer*), port jaringan yang digunakan untuk mentransfer data adalah 22 dikarenakan default dari SSH port adalah 22.

### 4.3.2 Automasi Pengiriman Data

Pada tahap ini data yang telah diambil dari sensor Raspberry Pi disimpan dan akan dikirimkan kedalam server secara otomatis, pada tahap ini menggunakan software dari server windows bernama *Task Scheduler* dengan program ini dapat dilakukan automatisasi pengambilan data secara otomatis dari sensor ke dalam tujuan folder pada server.

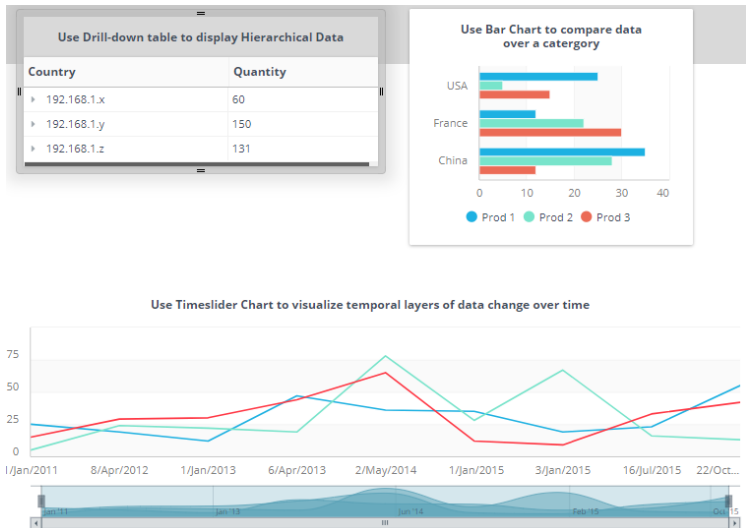
## 4.4 Perubahan format Data

Pada tahap ini dijelaskan data yang didapatkan dari sensor akan di ubah formatnya dari *pcap* menjadi *csv*. Perubahan format data jaringan dapat menggunakan *wireshark* dalam pengubahan format data yang telah diterima

Pada file *pcap* yang diterima oleh server dibuka dengan menggunakan software *wireshark*, kemudian di rubah file monitoring jaringan tersebut menjadi format *csv*, tahap berikutnya adalah *export* data menjadi *format* yang diinginkan disini menggunakan *csv* dikarenakan *tableau* sebagai program yang menjadikan rekaman data menjadi visual dalam bentuk *dashboard* tidak dapat membaca file yang berkstensi *pcap*, namun *tableau* dapat membaca file yang memiliki ekstensi *csv* sehingga proses perubahan format data diperlukan untuk pembuatan visualisasi data dalam bentuk *dashboard*.

## 4.5 Dashboard Model

Pada sub bagian ini akan dijelaskan visualisasi data monitoring jaringan yang dilakukan oleh sensor yang ditampilkan dalam bentuk dashboard pada gambar 4.3 adalah prototype visualisasi dashboard yang direncanakan :



**Gambar 4.3 Rancangan dashboard data monitoring**

## 4.6 Automasi Pengambilan dan Pengiriman Data

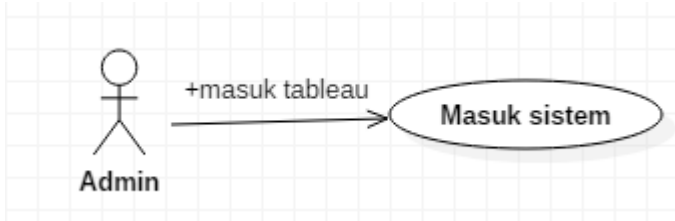
Pada sub-bab ini akan dijelaskan bagaimana sistem automisasi yang digunakan untuk pengiriman data paket jaringan dari sensor ke server. Pengambilan dan Pengiriman Data pada sistem ini menggunakan Web Service sebagai mediator dalam automasi pengambilan data dari server dan disimpan pada database server.

## 4.7 Desain Usecase

Pada desain usecase pada sistem akan dijelaskan bagaimana sistem digunakan oleh administrator jaringan dalam menampilkan data yang akan divisualisasi.

### 4.7.1 Masuk Sistem Dashboard

Pada sub section berikut akan dijelaskan bagaimana admin dapat mengakses dashboard hasil monitoring data dari sensor akan dijelaskan dalam gambar 4.4 :

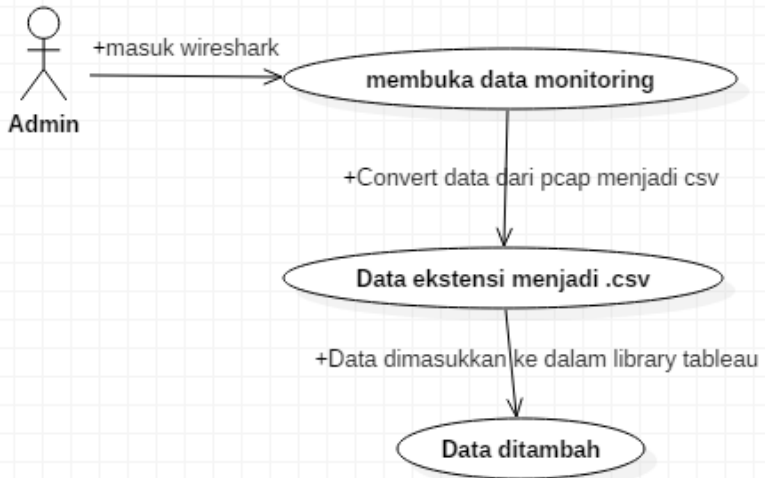


**Gambar 4.4 Use Case akses Sistem Monitoring**

Pada gambar 4.4 Admin masuk ke dalam server komputer dan membuka aplikasi *tableau* untuk mengakses dashboard yang telah dikonfigurasi. Penggunaan admin untuk masuk tableau memerlukan data untuk dimasukkan ke dalam tableau. Pada Gambar 4.5 dijelaskan konversi data yang diambil oleh raspberry pi ke dalam server untuk dibaca ke dalam tableau.

#### 4.7.2 Tambah Data

Pada sub bab berikut akan dijelaskan bagaimana admin dapat menambahkan data terbaru ke dalam dashboard monitoring akan dijelaskan pada Gambar 4.5 :



**Gambar 4.5 Use Case penambahan data monitoring**

Pada Gambar 4.5 menjelaskan konversi dari file yang berkecstensi *pcapng* menjadi file yang berbentuk *csv*, penggunaan ini perlu dilakukan dikarenakan penggunaan tableau dari file berekstensi *pcapng* tidak dapat dilakukan sehingga perlu dilakukan konversi file yang didapatkan oleh raspberry pi dari jaringan router lokal. Dengan cara mengakses file yang didapatkan oleh raspberry pi melalui aplikasi *wireshark* kemudian mengekspor file yang dibuka oleh software *wireshark* menjadi file yang berekstensi *csv* maka data yang sebelumnya tidak dapat terbaca oleh tableau sekarang dapat diproses oleh tableau.

## 4.8 Ekspektasi Data

```

1  "No.", "Time", "Source", "Destination", "Protocol", "Length", "Info"
2  "1", "0.000000", "54.230.156.218", "192.168.1.102", "TLSv1.2", "435", "Application Data"
3  "2", "0.000001", "54.230.156.218", "192.168.1.102", "TLSv1.2", "92", "Application Data"
4  "3", "0.000043", "192.168.1.102", "54.230.156.218", "TCP", "54", "53581 > 443 [ACK] Seq=1 Ack=420 Win=1019 Len=0"
5  "4", "0.004088", "192.168.1.102", "204.79.197.222", "TCP", "66", "53582 > 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_
6  "5", "0.036308", "204.79.197.222", "192.168.1.102", "TCP", "66", "443 > 53582 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 W
7  "6", "0.036375", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0"
8  "7", "0.036597", "192.168.1.102", "204.79.197.222", "TLSv1.2", "289", "Client Hello"
9  "8", "0.079510", "204.79.197.222", "192.168.1.102", "TCP", "60", "443 > 53582 [ACK] Seq=1 Ack=235 Win=262144 Len=0"
10 "9", "0.079915", "204.79.197.222", "192.168.1.102", "TCP", "1466", "443 > 53582 [ACK] Seq=1 Ack=235 Win=262144 Len=1412 [TCP se
11 "10", "0.079948", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=235 Ack=1413 Win=262144 Len=0"
12 "11", "0.080037", "204.79.197.222", "192.168.1.102", "TCP", "1466", "443 > 53582 [ACK] Seq=1413 Ack=235 Win=262144 Len=1412 [TC
13 "12", "0.080050", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=235 Ack=2825 Win=262144 Len=0"
14 "13", "0.080177", "204.79.197.222", "192.168.1.102", "TCP", "1466", "443 > 53582 [ACK] Seq=2825 Ack=235 Win=262144 Len=1412 [TC
15 "14", "0.080191", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=235 Ack=4237 Win=262144 Len=0"
16 "15", "0.080274", "204.79.197.222", "192.168.1.102", "TCP", "1466", "443 > 53582 [ACK] Seq=4237 Ack=235 Win=262144 Len=1412 [TC
17 "16", "0.080284", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=235 Ack=5649 Win=262144 Len=0"
18 "17", "0.080298", "204.79.197.222", "192.168.1.102", "TLSv1.2", "140", "Server Hello, Certificate, Certificate Status, Server Key
19 "18", "0.080308", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=235 Ack=5735 Win=261888 Len=0"
20 "19", "0.082197", "192.168.1.102", "204.79.197.222", "TLSv1.2", "147", "Client Key Exchange, Change Cipher Spec, Encrypted Handsh
21 "20", "0.082333", "192.168.1.102", "204.79.197.222", "TLSv1.2", "141", "Application Data"
22 "21", "0.082471", "192.168.1.102", "204.79.197.222", "TLSv1.2", "670", "Application Data"
23 "22", "0.113228", "204.79.197.222", "192.168.1.102", "TCP", "60", "443 > 53582 [ACK] Seq=5735 Ack=328 Win=262144 Len=0"
24 "23", "0.113441", "204.79.197.222", "192.168.1.102", "TCP", "60", "443 > 53582 [ACK] Seq=5735 Ack=415 Win=262144 Len=0"
25 "24", "0.113869", "204.79.197.222", "192.168.1.102", "TLSv1.2", "105", "Change Cipher Spec, Encrypted Handshake Message"
26 "25", "0.113899", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=1031 Ack=5786 Win=261888 Len=0"
27 "26", "0.113940", "204.79.197.222", "192.168.1.102", "TLSv1.2", "123", "Application Data"
28 "27", "0.113949", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=1031 Ack=5855 Win=261888 Len=0"
29 "28", "0.114186", "204.79.197.222", "192.168.1.102", "TLSv1.2", "92", "Application Data"
30 "29", "0.114189", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=1031 Ack=5893 Win=261888 Len=0"
31 "30", "0.114252", "192.168.1.102", "204.79.197.222", "TLSv1.2", "92", "Application Data"
32 "31", "0.117035", "204.79.197.222", "192.168.1.102", "TLSv1.2", "262", "Application Data"
33 "32", "0.117035", "204.79.197.222", "192.168.1.102", "TLSv1.2", "92", "Application Data"
34 "33", "0.117062", "192.168.1.102", "204.79.197.222", "TCP", "54", "53582 > 443 [ACK] Seq=1069 Ack=6139 Win=261632 Len=0"
35 "34", "0.145374", "204.79.197.222", "192.168.1.102", "TCP", "60", "443 > 53582 [ACK] Seq=6139 Ack=1069 Win=261376 Len=0"
36 "35", "0.124236", "192.168.1.102", "192.168.1.255", "UDP", "385", "54915 > 54915 Len=263"
37 "36", "1.1.244564", "192.168.1.102", "192.168.1.255", "UDP", "385", "54915 > 54915 Len=263"
38 "37", "1.1.672688", "192.168.1.102", "204.79.197.200", "TLSv1.2", "2517", "Application Data"
39 "38", "1.1.672749", "192.168.1.102", "204.79.197.200", "TLSv1.2", "876", "Application Data"
40 "39", "1.1.672772", "192.168.1.102", "204.79.197.200", "TLSv1.2", "92", "Application Data"
41 "40", "1.1.703332", "204.79.197.200", "192.168.1.102", "TCP", "60", "443 > 53574 [ACK] Seq=1 Ack=2464 Win=1025 Len=0"
42 "41", "1.1.703391", "204.79.197.200", "192.168.1.102", "TCP", "60", "443 > 53574 [ACK] Seq=1 Ack=3286 Win=1022 Len=0"
43 "42", "1.1.703740", "204.79.197.200", "192.168.1.102", "TCP", "60", "443 > 53574 [ACK] Seq=1 Ack=3324 Win=1022 Len=0"
44 "43", "1.1.703820", "204.79.197.200", "192.168.1.102", "TLSv1.2", "109", "Application Data"
45 "44", "1.1.703844", "192.168.1.102", "204.79.197.200", "TCP", "54", "53574 > 443 [ACK] Seq=3324 Ack=56 Win=1021 Len=0"
46 "45", "1.1.744426", "204.79.197.200", "192.168.1.102", "TLSv1.2", "194", "Application Data"
47 "46", "1.1.744462", "192.168.1.102", "204.79.197.200", "TCP", "54", "53574 > 443 [ACK] Seq=3324 Ack=196 Win=1020 Len=0"
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF

```

*Gambar 4.5 list data monitoring jaringan*

Desain basis data yang akan digunakan seperti pada gambar 4.5, Pada gambar tersebut terlihat detail dari hasil monitoring dari raspberry pi yang berisi data – data waktu, IP address sumber, IP address tujuan, protokol jaringan, dan panjang jaringan.



## BAB V

### IMPLEMENTASI

Bab ini akan menjelaskan skenario implementasi sensor pada jaringan lokal dan transfer data ke dalam komputer admin jaringan dan menampilkannya dalam bentuk dashboard.

#### 5.1 Spesifikasi Perangkat Keras

Pada bagian ini akan dibahas lingkungan pengujian yang digunakan dalam implementasi tugas akhir terkait perangkat yang digunakan baik lunak dan keras. Berikut akan dijelaskan spesifikasi perangkat raspberry pi sebagai sensor dan komputer yang digunakan dalam pengujian. Tabel berisikan spesifikasi perangkat keras yang akan digunakan dan pada tabel terdapat spesifikasi perangkat keras hardware yang digunakan sebagai server dan menampilkan data. Berikut adalah spesifikasi komputer yang berperan sebagai server untuk menampung data dan menampilkan data:

Processor	:1.4GHz dual-core Intel Core i5 (Turbo Boost up to 2.7GHz) with 3MB shared L3 cache
Storage	:128GB PCIe-based flash storage
Memory	:8GB of 1600MHz LPDDR3 onboard memory
Graphics	:Intel HD Graphics 5000
Wireless	:Wi-Fi 802.11ac Wi-Fi networking;4 IEEE 802.11a/b/g/n compatible
Operating System	:Windows 10
Ports	:Headphone, Dual mics, Two USB 3 ports (up to 5 Gbps),

## 5.2 Perangkat lunak

Pada tahap ini akan dijelaskan perangkat lunak yang dibutuhkan dalam menjalankan sistem, dipastikan bahwa perangkat lunak sudah ter *install* dalam komputer sebelum melakukan pengerjaan.

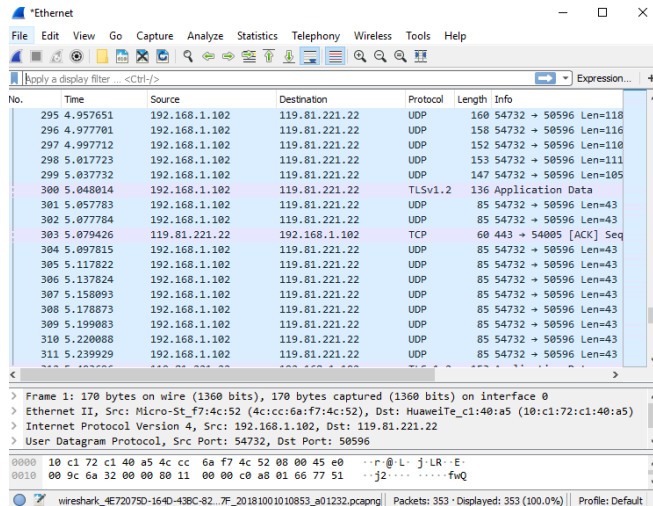
**Tabel 5.1 Daftar penggunaan perangkat lunak pada sistem**

Nama Perangkat Lunak	Kegunaan dalam Implementasi
Wireshark / Tshark / Dumpcap	Mengambil data jaringan
Cron dan Windows Task Scheduler	Automisasi pengambilan dan pengiriman data
Tableau	Dashboard
Web Server	Sistem berbasis web menggunakan Session untuk verifikasi identitas upload data

Setelah perangkat terpasang pastikan opsi dalam openssh server dinyalakan untuk membuka akses pemberian data dari sensor kepada komputer.

### 5.2.1 Pengambilan data jaringan dengan Wireshark

Pada tahapan ini dilkakukan pengambilan jaringan oleh sensor Raspberry Pi menggunakan aplikasi *wireshark*, berikut gambar pengambilan aktivitas jaringan menggunakan wireshark.



**Gambar 5.1 Penggunaan Software Wireshark**

Pada gambar diatas menjelaskan pengambilan jaringan menggunakan software wireshark pada router, dan pada hasil monitoring ditemukan beberapa tipe jaringan yang digunakan dan beberapa lalu lintas jaringan yang sering dalam penggunaannya

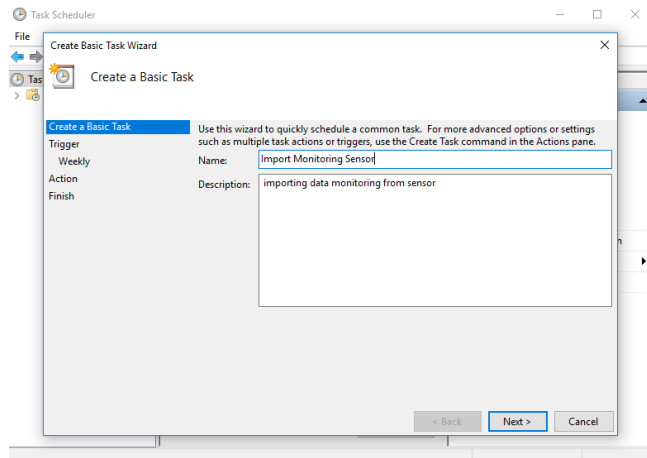
### 5.2.2 Automasi pengambilan dan pengiriman data menggunakan Cron pada sensor

Pada tahapan ini dilakukan automatisasi perintah pada terminal di jam yang telah ditentukan untuk pengambilan data, Dengan menggunakan *command line* TcpDump pada terminal hasil data dari monitoring network pada router dapat di ambil dan di simpan pada Raspberry Pi berbentuk file yang mempunyai ekstensi *pcapng*.

Penggunaan automasi jaringan dapat menggunakan *command line* crontab pada terminal, crontab yang digunakan di program terlebih dahulu sebelum digunakan dalam pengambilan data pada jaringan.

### 5.2.3 Automasi penerimaan data pada komputer menggunakan *Task Scheduler*

Pada bagian ini akan dijelaskan mengenai cara automasi penerimaan data pada sisi user yang menggunakan *operating system windows*, pada tahap ini dilakukan pengaturan penjadwalan otomasi *command line* dari windows menggunakan software resmi dari windows yaitu *Task Scheduler*.



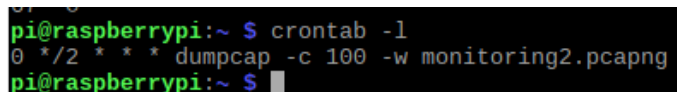
***Gambar 5.2 Penjadwalan pengambilan data raspberry pi***

Pada Gambar diatas dijelaskan setting dari pekerjaan yang perlu dilakukan oleh komputer pada waktu yang telah ditentukan, disini settingan saya adalah pengambilan data menggunakan *putty* melalui software command prompt.

### 5.2.4 Konversi ekstensi data Paket Jaringan

Pada Bagian ini dilakukan konversi ekstensi data yang diambil dari sensor yang berupa *pcapng* dirubah pada komputer user :

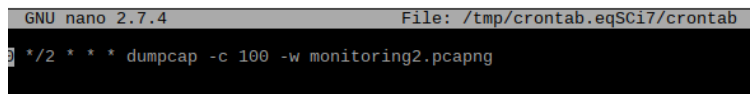
Konversi data dapat menggunakan *wireshark*, setelah data dikonversi menjadi berbasis excel data dapat digunakan oleh *tableau* untuk dimasukkan dalam bentuk dashboard.



```
pi@raspberrypi:~ $ crontab -l
0 */2 * * * dumpcap -c 100 -w monitoring2.pcapng
pi@raspberrypi:~ $
```

**Gambar 5.3 List command crontab**

Pada Gambar 5.3 adalah list dari automatisasi dengan menggunakan crontab, dengan command *crontab -l* list yang akan dijalankan oleh sistem dengan kondisi – kondisi tertentu. Untuk mengganti *command line* yang akan dijalankan oleh sistem dilakukan dengan command *crontab -e* dimana akan muncul seperti pada gambar 5.4



```
GNU nano 2.7.4 File: /tmp/crontab.eqSCi7/crontab
0 */2 * * * dumpcap -c 100 -w monitoring2.pcapng
```

**Gambar 5.4 Editor command line pada program crontab**

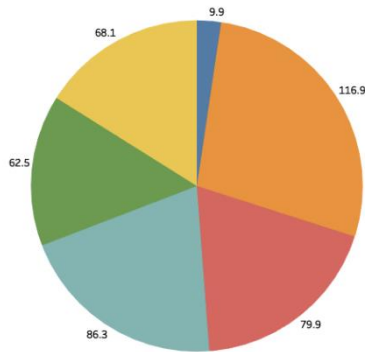
Pada Gambar 5.4 adalah cara modifikasi *command line* yang akan dijalankan oleh program *crontab* dimana untuk command line edit pada crontab akan diberikan 5 pilihan dan disini penulis memilih menggunakan modifikasi command crontab dengan menggunakan GNU nano.

### 5.2.4 Upload data pada aplikasi dashboard *Tableau*

Pada Bagian ini dilakukan upload data dan hasil pada dashboard pada software *tableau*, data yang telah didapatkan dari raspberry pi menggunakan web server ditransfer kedalam *tableau* untuk ditampilkan untuk hasil visualisasi.

Pada gambar diatas adalah dashboard utama dari tableau yang menampilkan semua parameter dari hasil data yang didapatkan oleh Raspberry Pi. Untuk data detail dari dashboard tersebut dapat dilihat dari masing – masing sheet dari software dashboard tersebut.

Average Time on accessing protocol



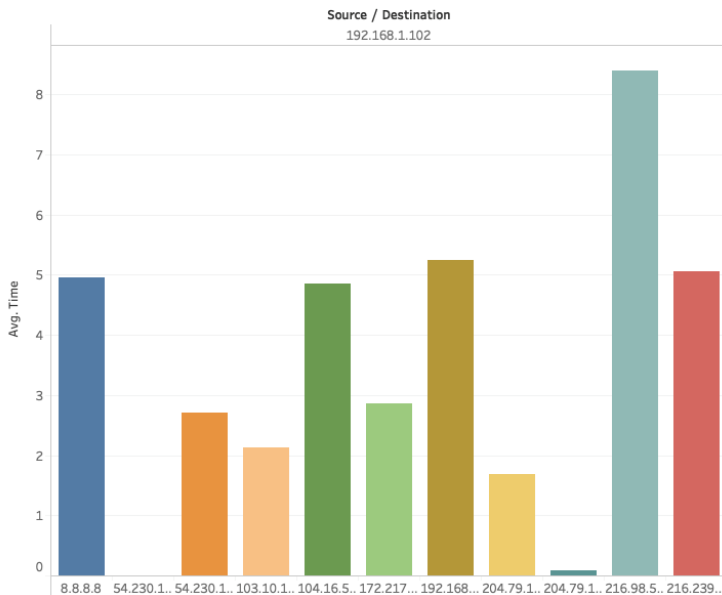
***Gambar 5.5 Persentase Penggunaan Protokol pada router***

Gambar diatas menjelaskan penggunaan protokol yang digunakan user pada jaringan router yang di monitoring oleh Raspberry Pi, fungsi dari dashboard diatas untuk mengetahui adakah penggunaan port yang berbahaya seperti ARP dalam lingkungan laboratorium oleh user. Pada gambar diatas terdapat 6 warna pada diagram pie, menunjukan 6 protokol yang dilewati pada router. 6 warna tersebut masing – masing adalah:

- DNS = Biru
- SSDP = Merah

- TCP = Abu – abu
- TLSv1.2 = Hijau
- UDP = Kuning
- GQUIC = Oranye

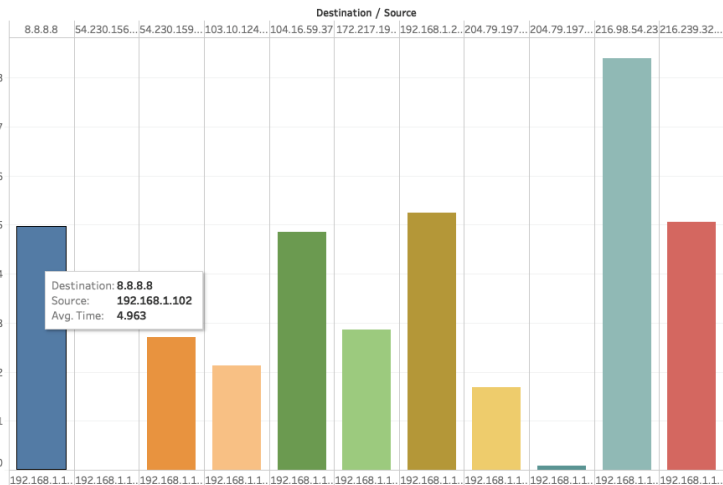
Avg Time User to Router



***Gambar 5.6 Waktu rata – rata akses user***

Gambar diatas menjelaskan rata - rata dari penggunaan data dari user network pada router. Fungsi dari dashboard tersebut digunakan untuk melihat frekuensi penggunaan user dalam mengakses IP Address tersebut dan dapat digunakan sebagai parameter apakah IP Address tersebut berhubungan dengan kegiatan di kampus atau tidak.

Avg Time Router to Destination



***Gambar 5.7 Waktu rata – rata akses router***

Pada Gambar di atas menjelaskan waktu rata – rata akses router dalam jaringan router, fungsi dari histogram berikut untuk mengetahui lama penggunaan user dalam mengakses IP tujuan dan dapat dianalisa apakah penggunaan jaringan dalam kampus tidak ada indikasi penyerangan dan lalu lintas jaringan tidak ada permasalahan dan apabila ada dapat dilihat seberapa besar permasalahan pada jaringan kampus.



## BAB VI

### HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dan pembahasan proses dari proses pengujian raspberry pi dan sistem yang telah dibuat.

#### 6.1 Pengujian menggunakan sistem Transfer file via FTP

Pada pengujian ini akan dijelaskan metode automisasi monitoring raspberry pi dengan server dengan mediasi FTP untuk pemberian paket data dari raspberry pi kepada server.

Setelah putty terinstall. Transfer file dapat menggunakan command line untuk server windows atau mac os, untuk konsep penggunaan windows dan mac sedikit berbeda namun untuk penggunaan command line pada sensor dengan mac os menggunakan command yang sama. Kemudian penggunaan *copy* file dari sensor ke dalam server dapat menggunakan *pscp* (PuTTY Secure Copy Client), pada Gambar 6.1 adalah penggunaan *pscp* dalam bentuk command line dalam windows

```
pscp [options] [user@]host:source target
```

**Gambar 6.1 Command line putty pada cmd server**

Sebelum file transfer bisa dilakukan dengan menggunakan command line dari windows, server dari SSH pada Raspberry Pi harus dinyalakan dikarenakan apabila akan menggunakan SSH dalam transfer file jika SSH sistem pada Raspberry Pi belum dinyalakan maka akses pada sensor menjadi restricted dan sensor tidak bisa memberikan data kepada server.

Untuk Automasi Transfer Data dari Putty SSH ke dalam Server pada sistem ini menggunakan windows task scheduler pada *Operating System Windows* atau Cron Job pada *Operating System linux*, Automasi tersebut memudahkan dalam pengiriman data pada waktu yang telah ditentukan untuk

pengambilan data jaringan pada periodik tertentu yang sudah direncanakan.

Untuk Penggunaan pada Gambar 6.1 setelah menyiapkan server pada raspberry pi dapat dilakukan dengan memasukkan command yang akan digunakan, dikarenakan penggunaan transfer data menggunakan user tertentu dengan adanya *password* dan port maka digunakan user@host source target dimana user adalah user yang digunakan adalah user pada raspberry pi nya dan host *source* target adalah ip address raspberry pi tersebut.

```
pscp -pw raspberry pi@192.168.1.143:/home/pi/Desktop/lel.txt* e:\
```

### ***Gambar 6.2 Command Line mendapatkan file dari sensor***

Pada gambar diatas dijelaskan *command line* automisasi pengambilan data dengan menggunakan Putty SSH.

## **6.2 Pengujian automasi sistem dengan menggunakan Tshark**

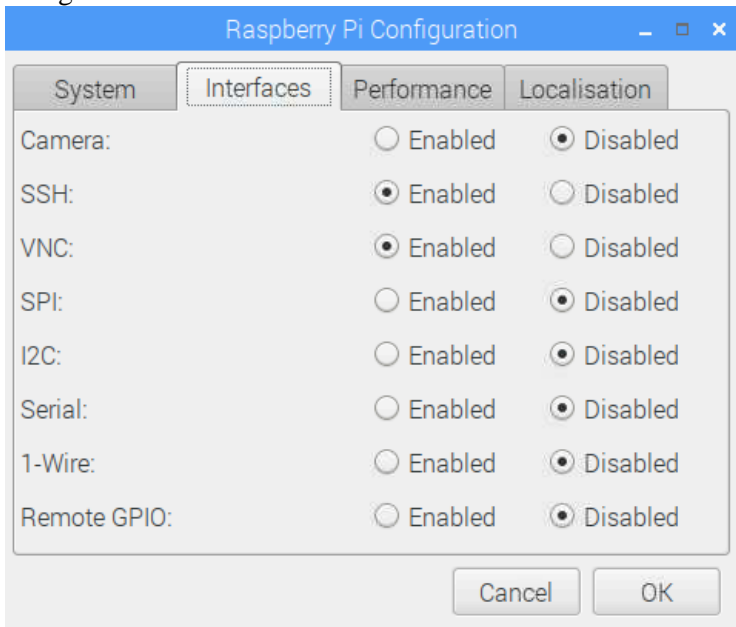
Pada pengujian ini akan dijelaskan metode automisasi monitoring dengan konversi dari data paket data yang berekstensi pcap dari pengambilan jaringan menjadi file yang berekstensi csv untuk diolah menjadi dashboard oleh program dashboard *tableau*.

Penggunaan program Tshark digunakan dalam pengambilan paket jaringan dalam raspberry pi, Program Tshark juga digunakan oleh sistem operasi Windows dalam penggabungan data yang diterima dari sensor. Penggunaan Tshark ini diikuti juga dengan hasil pengujian file transfer yang dilakukan pada bab sebelumnya. proses ini dilakukan secara otomatis dengan menggunakan program buatan yang menggunakan bahasa *python*. Dengan pemrograman tersebut pengendalian sistem secara otomatis dapat dilakukan dimana pemograman tersebut dapat menjadwalkan aktivasi *command* yang perlu dijalankan pada rentang waktu tertentu.

### 6.3 Pengujian perintah melakukan *command* pada linux dengan menggunakan SSH

Pada pengujian ini dijelaskan mengenai perintah yang perlu dilakukan oleh raspberry pi oleh windows server melalui port SSH.

Penggunaan SSH dilakukan dengan aktivasi fitur SSH pada Raspberry Pi dan juga fitur SSH pada Windows, pada pengaktifan fitur SSH pada Raspberry Pi dapat dilakukan sebagai berikut



**Gambar 6.3** Tampilan konfigurasi SSH pada sensor

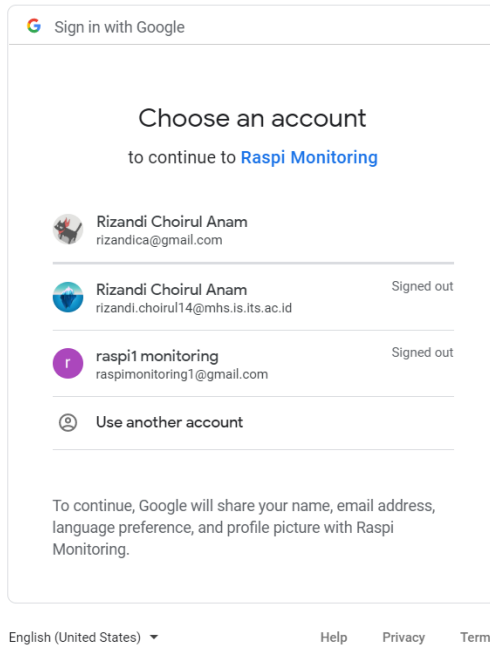
Pada Gambar 6.3 dilakukan pengaktifan port SSH dengan menggunakan GUI pada sistem preferensi Raspberry Pi, untuk aktifasi SSH pada windows memerlukan instalasi khusus yang sudah disediakan dalam *built in*. cara instalasi SSH pada windows dapat mengakses pada menu *setting*, kemudian masuk

ke *manage optional features* kemudian add features OpenSSH client.

Penggunaan SSH dalam Server ke Raspberry Pi dapat digunakan sebagai *command* dimana dengan memanfaatkan port 22 dapat mengontrol *command* yang akan digunakan oleh Raspberry Pi dengan bantuan program berbasis *python*. *Python* memiliki peranan penting dalam pengontrolan SSH yang dilakukan server kepada sensor dimana pengambilan data jaringan, transfer data, dan merging data dilakukan dengan pemrograman dan dilakukan secara otomatis.

#### **6.4 Pengujian dengan menggunakan webservice**

Transfer data paket dari sensor ke dalam server dengan menggunakan aplikasi berbasis web service, dalam pengaplikasian webservice yang digunakan untuk mentransfer data menggunakan Python dan menggunakan OpenID sebagai autentikasi transfer data antara server dengan sensor. Pada Gambar 6.4 dijelaskan interface dari *webservice* berbasis Python dengan menggunakan Oauth2 dari Google untuk masuk dengan menggunakan gmail.



***Gambar 6.4 Tampilan awal dari website berbasis Python***

Pada gambar diatas adalah halaman depan dari website yang digunakan sebagai website untuk transfer data menggunakan bahasa pemograman python. Pada halama depan admin login menggunakan email google mereka untuk mengupload data hasil monitoring pada sensor untuk diberikan pada server.

## **6.5 Penggunaan Aplikasi Dashboard Kibana**

Pada pengujian ini dijelaskan rencana pengaplikasian dashboard *kibana*, Kibana adalah sebuah software dashboard berbasis open-source dari elasticsearch. Penggunaan Kibana tidak dapat dilakukan dikarenakan perlu adanya integrasi sistem yang telah disediakan dari developer software yaitu *elasticsearch* dimana penggunaan sistem perlu mempelajari Bahasa pemograman baru.

## 6.6 Kode Pada Server

Pada bagian ini dijelaskan mengenai kode yang ada pada server, kode yang ada pada server berguna untuk memberikan perintah pengambilan data pada raspberry pi yang di kontrol melalui port SSH, pengiriman data dari raspberry pi ke dalam server dengan menggunakan FTP melalui port 22, menggabungkan data hasil penangkapan dari raspberry pi.

Kode 1. 1

```
import paramiko
import os
import time

hostname = "192.168.1.143"
password = "raspberrypi"

username = "pi"
port = 22

# try:
touch = 0
counter = 0
while(True):
    # print(counter)
    counter+=1
    client = paramiko.SSHClient()
    client.load_system_host_keys()

    client.set_missing_host_key_policy(paramiko.WarningPolicy)

    client.connect(hostname, port=port,
username=username, password=password)
    # while(True):
```

```

#     print(counter)
#     counter+=1
    stdin, stdout, stderr =
client.exec_command("tshark -a duration:15 -w
monitoring1.pcap -T fields -E separator=, -E
quote=d -e frame.number -e frame.time -
e_ws.col.Source -e_ws.col.Destination -
e_ws.col.Protocol -e_ws.col.info")
    exit_status =
stdout.channel.recv_exit_status()
    checker_pengambilan = os.system("pscp -pw
raspberry
pi@192.168.1.143:/home/pi/monitoring1.pcap D:/")
    print(str(checker_pengambilan))
    while(checker_pengambilan):
        checker_pengambilan = os.system("pscp -
pw raspberry
pi@192.168.1.143:/home/pi/monitoring1.pcap D:/")
        stdin_cek1, stdout_cek1, stderr_cek1 =
client.exec_command("python accesdate.py")
        a_cek1 = stdout_cek1.read()
        a_cek1 = str(a_cek1)
        b_cek1 = a_cek1.split(" ")
        c_cek1 = b_cek1[3]
        stdin, stdout, stderr =
client.exec_command("touch -a monitoring1.pcap")
        stdin_cek2, stdout_cek2, stderr_cek2 =
client.exec_command("python accesdate.py")
        a_cek2 = stdout_cek2.read()
        a_cek2 = str(a_cek2)
        b_cek2 = a_cek2.split(" ")
        c_cek2 = b_cek2[3]
        print(c_cek1, c_cek2)
        if(c_cek1 != c_cek2):

```

```

        print(touch)
        touch = 1
    else:
        print(touch)
        touch = 0
    while(not touch):
        print(c_cek1, c_cek2)
        stdin_cek1, stdout_cek1, stderr_cek1 =
client.exec_command("python accesdate.py")
        print(c_cek1, c_cek2)
        a_cek1 = stdout_cek1.read()
        a_cek1 = str(a_cek1)
        b_cek1 = a_cek1.split(" ")
        c_cek1 = b_cek1[3]
        stdin, stdout, stderr =
client.exec_command("touch -a monitoring1.pcap")
        print(c_cek1, c_cek2)
        stdin_cek2, stdout_cek2, stderr_cek2 =
client.exec_command("python accesdate.py")
        print(c_cek1, c_cek2)
        a_cek2 = stdout_cek2.read()
        a_cek2 = str(a_cek2)
        b_cek2 = a_cek2.split(" ")
        c_cek2 = b_cek2[3]
        print(c_cek1, c_cek2)
        if(c_cek1 != c_cek2):
            touch = 1
        else:
            touch = 0
    client.close()

```

Pada server telah dilakukan setting untuk melakukan pengiriman data melalui jaringan SSH dimana pada kode diatas dengan menggunakan library dari python yang bernama



*paramiko* dimana dia dapat mengkomunikasikan penggunaan perjalanan aktivitas dalam raspberry pi dengan menggunakan ssh sehingga aktivitas tersebut dapat di kontrol oleh server. Setelah data tersebut diambil oleh raspberry pi, server mengambil data jaringan pada raspberry pi untuk di proses penggabungan dan konversi ekstensi file pada server. Dalam kode terdapat pengecekan data pada raspberry pi yang bertujuan untuk memastikan bahwa data jaringan pada raspberry pi telah diterima oleh server baru melanjutkan langkah berikutnya dikarenakan menghindari data jaringan tidak terkirim ke dalam server.

## Kode Penggabungan Data

### Kode 1. 2

```
import os

# from shutil import copyfile

# copyfile(D:\\networkmonitoring1.pcap,
D:\\networkmonitoringcopy1.pcap)
os.path.isfile('D:\\networkmonitoring1.pcap')
# copy pcap untuk di merge
os.system('copy D:\\networkmonitoring1.pcap
D:\\networkmonitoringcopy1.pcap')
# merger pcap menggunakan merg pcap
os.system("mergpcap -w
D:\\networkmonitoring1.pcap
D:\\networkmonitoringcopy1.pcap
D:\\monitoring1.pcap")
#jadikan csv untuk dibaca
os.system("tshark -r D:\\networkmonitoring1.pcap
-T fields -e frame.number -e frame.time -E
separator > D:\\output.csv")
```

Setelah memastikan bahwa data pada raspberry pi sudah dikirimkan ke dalam server maka dilakukan langkah untuk penggabungan data dalam server. Setelah penggabungan data berhasil maka dilakukan konversi data jaringan, dimana program dashboard yang digunakan tidak mendukung pembacaan ekstensi file dari data jaringan yaitu ekstensi *pcap* oleh karena itu perlu adanya konversi file dari server agar dapat di olah oleh program dashboard. Ekstensi file yang dipilih adalah *csv* dikarenakan file berekstensi *csv* mempunyai banyak kompatibel dengan program dashboard dan dapat menampung jumlah data yang banyak.

Kode Pengaksesan informasi modifikasi file

kode 1.3

```
import os
import stat
import time

fs = os.stat("/home/pi/monitoring1.pcap")
at = time.ctime(fs[stat.ST_ATIME])
print(at)
```

Pada kode diatas adalah penggunaan program *touch* pada raspberry pi dimana kegunaan dari program tersebut adalah untuk membantu pengaksesan file untuk pengecekan apakah file tersebut telah diambil oleh server atau belum, apabila belum maka file tersebut tidak di *touch* dikarenakan pengambilan file terjadi masalah sehingga pengambilan data perlu diulang. Dalam kode *accessdate* ini berguna untuk membantu memberikan informasi ke dalam program python yang ada pada server untuk memberikan waktu terakhir di modifikasi dikarenakan perlu adanya informasi tambahan dari raspberry pi

tersebut maka kode *accessdate* perlu dipisah untuk memberikan informasi kedalam program server.

### Kode Akses SSH menggunakan Paramiko

kode 1.4

```
import paramiko
import os
import time

hostname = "192.168.43.226"
password = "raspberry"

username = "pi"
port = 22

# try:
touch = 0
counter = 0
while(True):
    print(counter)
    counter+=1
    client = paramiko.SSHClient()
    client.load_system_host_keys()

    client.set_missing_host_key_policy(paramiko.WarningPolicy)

    client.connect(hostname, port=port,
username=username, password=password)
```

pada kode diatas adalah penggunaan library *paramiko* dimana penggunaan *paramiko* digunakan dalam koneksi dari *port* SSH untuk dimasukkan dalam program python. Penggunaan

paramiko selain untuk mengambil data jaringan pada raspberry pi juga menggunakannya untuk memberikan perintah aktivitas kepada raspberry pi melalui SSH sehingga penggunaan Raspberry Pi dikhususkan untuk mengambil data jaringan.

## Kode Perintah Pengambilan Data Pada Raspberry Pi

### *kode 1.5*

```
stdin, stdout, stderr =
client.exec_command("tshark -a duration:60 -w
monitoring1.pcap -T fields -E separator=, -E
quote=d -e frame.number -e frame.time -
e_ws.col.Source -e_ws.col.Destination -
e_ws.col.Protocol -e_ws.col.info")
    exit_status =
stdout.channel.recv_exit_status()
```

pada kode 1.5 mempunyai fungsi untuk memberikan perintah aktivitas kepada raspberry pi untuk mengambil data jaringan pada waktu tertentu dan berapa lama pengambilan berlangsung. Pada kode diatas berisi perintah yang perlu dilakukan pada terminal raspberry pi untuk mengambil data jaringan. Penggunaan perintah diatas menggunakan pengontrolan aktivitas pada raspberry pi oleh server.

## Kode Transfer Data kedalam Server

### *kode 1.6*

```
checker_pengambilan = os.system("pscp -pw
raspberry
pi@192.168.1.143:/home/pi/monitoring1.pcap D:/")
    print(str(checker_pengambilan))
```

pada kode 1.6 mempunyai fungsi untuk penagambilan data jaringan dengan menggunakan program dalam server yaitu *putty* dimana fungsi dari *putty* untuk interaksi antara raspberry pi dengan server dimana komunikasi tidak hanya dalam transfer file saja namun dapat digunakan juga untuk pengiriman informasi aktivitas seperti *paramiko* pada python.

### Kode Pengecakan Pengambilan Data

kode 1.7

```
while(checker_pengambilan):
    checker_pengambilan = os.system("pscp -
pw raspberry
pi@192.168.1.143:/home/pi/monitoring1.pcap D:/")
    stdin_cek1, stdout_cek1, stderr_cek1 =
client.exec_command("python accesdate.py")
    a_cek1 = stdout_cek1.read()
    a_cek1 = str(a_cek1)
    b_cek1 = a_cek1.split(" ")
    c_cek1 = b_cek1[3]
    stdin, stdout, stderr =
client.exec_command("touch -a monitoring1.pcap")
    stdin_cek2, stdout_cek2, stderr_cek2 =
client.exec_command("python accesdate.py")
    a_cek2 = stdout_cek2.read()
    a_cek2 = str(a_cek2)
    b_cek2 = a_cek2.split(" ")
    c_cek2 = b_cek2[3]
    print(c_cek1, c_cek2)
```

Pada kode 1.7 mempunyai fungsi untuk mengecek apakah data jaringan yang ada pada raspberry pi sudah terambil dan dikirim kepada server untuk dilanjutkan pada tahap berikutnya atau masih ada masalah dalam pengiriman datanya, dengan adanya pengecekan dapat memastikan bahwa data yang

diambil dari raspberry pi sudah terkirim kedalam server apabila data jaringan pada raspberry pi mengalami kegagalan dalam pengiriman maka bagian pengecekan akan memberi tahu kepada program *putty* tersebut untuk mengambil kembali data jaringan yang gagal diambil oleh server.

Kode Pengecekan pengambilan data

kode 1. 8

```
if(c_cek1 != c_cek2):
    print(touch)
    touch = 1
else:
    print(touch)
    touch = 0
```

Pada Kode 1.8 adalah penggantian value dari hasil pengambilan dengan menggunakan *putty*, dimana apabila pengambilan dalam raspberry pi kedalam server berhasil maka nilai keluaran yang dihasilkan oleh *putty* adalah 0 sementara apabila *putty* gagal dalam mengambil data jaringan yang ada pada raspberry pi untuk dikirimkan kedalam server maka nilai keluaran yang dihasilkan oleh *putty* adalah 1. Disini sebagai salah satu bagian dari pengecekan dalam pengambilan data jaringan dari raspberry pi kedalam server.

Kode Pengecekan setelah gagal pengambilan

kode 1. 9

```
while(not touch):
    print(c_cek1, c_cek2)
    stdin_cek1, stdout_cek1, stderr_cek1 =
client.exec_command("python accessdate.py")
    print(c_cek1, c_cek2)
    a_cek1 = stdout_cek1.read()
```

```

a_cek1 = str(a_cek1)
b_cek1 = a_cek1.split(" ")
c_cek1 = b_cek1[3]
stdin, stdout, stderr =
client.exec_command("touch -a monitoring1.pcap")
print(c_cek1, c_cek2)
stdin_cek2, stdout_cek2, stderr_cek2 =
client.exec_command("python accessdate.py")
print(c_cek1, c_cek2)
a_cek2 = stdout_cek2.read()
a_cek2 = str(a_cek2)
b_cek2 = a_cek2.split(" ")
c_cek2 = b_cek2[3]
print(c_cek1, c_cek2)
if(c_cek1 != c_cek2):
    touch = 1
else:
    touch = 0

```

Pada kode 1.9 adalah aktivitas yang dilakukan oleh program apabila pengambilan data jaringan dalam raspberry pi gagal dan disini program akan mengecek lagi apabila pengambilan gagal dan apabila gagal dan kedua kali dapat diambil maka program bias lanjut ke tahap berikutnya. Disini ada *accessdate* dimana fungsi dari program tersebut untuk memberikan informasi dalam pengecekan tanggal terakhir diubah oleh program apabila gagal maka file tidak diubah dan pengambilan data terus dilakukan.

## 6.7 Skenario Implementasi

Pada bagian ini dijelaskan skenario – skenario yang bisa terjadi kepada Raspberry Pi dan server dan untuk masalah yang dicakup pada skenario Robustness ini antara lain apakah pemograman yang gagal dalam mendapatkan atau mengirim

data atau hardware dari perangkat tersebut yang memiliki masalah. Berikut adalah skenario yang dapat terjadi pada Sistem Monitoring Jaringan

*Tabel 6.1 Daftar skenario implementasi sistem*

No	Problem	Kondisi	Kondisi Raspberry Pi	Status dashboard	Update Dashboard	Pengendalian
1	Pengambilan Paket data banyak	Cuaca diluar 45 derajat dan suhu ruangan 30	Pada bagian processor panas dan kinerja mulai lambat sehingga ada kemungkinan paket data tidak terkirim	Memuat dashboard lama namun status update tidak ada	Tidak	Memambahkan baris kode untuk mengecek apakah ada status update dari file yang baru dibuat
2	Listrik mati pada saat pengambilan	Listrik Mati sehingga server dan raspberry pi tidak bias	Tidak dapat mengambil data jaringan	Memuat data dashboard ketika listrik menyala	Tidak	Untuk listrik mati diluar dari pengendalian, namun ketika listrik



		dipakai		kembali		menyala akan ditambahkan cron job untuk menjalankan program setelah menyala kembali
3	Raspberry pi kehabisan penyimpanan data	Paket data pada jaringan banyak dan raspberry pi mengambil banyak paket data untuk dikirimkan pada dashboard server	Tidak dapat mengambil data jaringan dikarenakan storage penuh	Memuat dashboard namun update dari dashboard tidak dilakukan	Tidak	Dengan cara replace file yang sudah dilakukan oleh raspberry pi, sehingga data lengkap ada pada server
4	Pengiriman data pada	Paket data ada pada	Tidak dapat mengambil	Memuat dashboard	Tidak	Dengan adanya double checkin

	raspberry pi tidak didapatkan server	raspberry pi namun pada server tidak dikirmkan atau gagal atau tidak diterima server	data atau data tidak terkirim dikarenakan ada kemungkinan bertabrakan dengan perintah kode yang lain	namun untuk update dashboard tidak dapat dilakukan		g yang pertama dengan menggunakan return value dari program pemindahan file untuk mengecek apakah file yang telah didapatkan berhasil di transfer atau tidak, yang kedua dengan menggunakan program touch dimana apabila file yang dibuat sama
--	--------------------------------------	--	--	--	--	--

						dengan file awal maka dilakukan pengiriman data kembali dan kemudian di touch untuk menandai bahwa file sudah berhasil ditransfer
--	--	--	--	--	--	---

- Pada pengujian pertama dilakukan pada suhu kamar 30 derajat celcius dimana suhu diluar adalah 45 derajat celcius. Pada jaringan dilakukan video viewer sebagai representasi data banyak digunakan, hasilnya adalah pengecekan bisa gagal dilakukan dikarenakan suhu prosesor panas sehingga terjadi penurunan kecepatan prosesor, oleh karena itu ada baris kode yang tidak dapat diproses.
- Pada pengujian kedua dilakukan pemutusan arus listrik sehingga raspberry pi dan server tidak dapat menyala, kendali yang dilakukan dengan menjalankan program auto start program monitoring ketika mesin menyala kembali
- Pada pengujian ketiga dilakukan penggantian penyimpanan dari micro sd raspberry pi ke dalam

server untuk menghindari habisnya penyimpanan data, pada raspberry pi dilakukan penggantian file yang baru apabila file monitoring berhasil terkirim

- Pada pengujian keempat ditambahkan kode pada pemograman monitoring untuk pengecekan ulang untuk memastikan data yang telah diambil dari jaringan sudah berhasil diberikan kepada server.

## **6.8 Hasil Uji Pengambilan dan Konversi Data**

Pada hasil uji pengambilan data berikut akan dijelaskan mengenai data yang akan ditampilkan kedalam dashboard merupakan data yang terpercaya dari raspberry pi bukan dari sumber lain, berikut merupakan hasil dari pengambilan dan konversi data

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.143	192.168.1.138	VNC	561	
2	0.024135436	192.168.1.138	192.168.1.143	VNC	86	
3	0.024228043	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
4	0.229344482	192.168.1.108	192.168.1.255	UDP	86	57621 > 57621 Len=86
5	1.500808633	192.168.1.143	192.168.1.138	VNC	526	
6	1.508042830	192.168.1.138	192.168.1.143	VNC	86	
7	1.508132833	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
8	3.001135977	192.168.1.143	192.168.1.138	VNC	316	
9	3.098873728	192.168.1.138	192.168.1.143	VNC	86	
10	3.098964825	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
11	4.502485848	192.168.1.143	192.168.1.138	VNC	321	
12	4.529180577	192.168.1.138	192.168.1.143	VNC	86	
13	4.529271622	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
14	6.002821812	192.168.1.143	192.168.1.138	VNC	320	
15	6.010242419	192.168.1.138	192.168.1.143	VNC	86	
16	6.010332682	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
17	7.503992847	192.168.1.143	192.168.1.138	VNC	317	
18	7.604578882	192.168.1.138	192.168.1.143	VNC	86	
19	7.604674979	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
20	8.114176934	Shanghai_10:bb:8a	Broadcast	ARP	42	Who has 192.168.1.143? length=28
21	8.858975521	192.168.1.138	192.168.1.143	VNC	81	
22	8.859076774	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
23	8.859250582	192.168.1.143	192.168.1.138	VNC	81	
24	8.909038293	192.168.1.138	192.168.1.143	TCP	54	49997 > 5900 [ACK]
25	9.004356442	192.168.1.143	192.168.1.138	VNC	314	
26	9.034354968	192.168.1.138	192.168.1.143	VNC	86	
27	9.078480933	192.168.1.143	192.168.1.138	TCP	54	5900 > 49997 [ACK]
28	10.505643810	192.168.1.143	192.168.1.138	VNC	333	
29	10.573790232	192.168.1.138	192.168.1.143	VNC	86	

**Gambar 6.5 Hasil Pengambilan data jaringan**

## 6.9 Hasil Biaya Realisasi

Hasil dari biaya dari realisasi yang digunakan pada raspberry pi pengirim data, dalam realisasi implementasi monitoring data jaringan dengan menggunakan raspberry pi memerlukan beberapa program dan hardware yang berincikan sebagai berikut:

*Tabel 6.2 Realisasi Raspberry Pi*

No	Nama	Harga
1	Raspberry Pi	Rp. 631.000
2	Tableau	Rp. 993,632

Tabel 6.1 merupakan tabel yang berisikan rincian dana untuk dapat merealisasikan sebuah raspberry pi mengambil data jaringan dalam router.

## 6.10 Pengujian Sistem dan Sub-sistem monitoring

Metode pengujian yang dilakukan oleh pengembang sistem pengambilan jaringan menggunakan metode *black box testing*. Skenario pada *black box testing* dibuat berdasarkan masing – masing unit yang termasuk dalam sistem pengambilan jaringan. Metode *black box testing* dilakukan dengan cara melakukan *check list* terhadap fungsionalitas masing – masing unit, kemudian dilihat apakah *output* dari masing – masing unit sesuai dengan hasil yang diharapkan.

Untuk setiap pengujian yang berhasil diberikan tanda berupa ‘OK’ untuk menandakan bahwa pengujian terhadap *test case* tersebut sudah sesuai dengan

fungisinya. Untuk setiap pengujian yang belum berhasil maka akan ditandai dengan 'NOK' untuk menandakan bahwa pengujian menunjukkan bahwa hasil yang diekuarkan belum sesuai dan perlu dilakukan perbaikan. Berikut hasil pengujian *test case* yang dilakukan oleh pengembang

Test Case	Skenario	Ekspektasi	OK/Not OK
Koneksi WiFi Raspberry Pi	Raspberry Pi disambungkan dengan WiFi	Terknokesi dengan jaringan WiFi, Mendapatkan IP Address	OK
Pengambilan File Jaringan	Pengambilan data pcap pada jaringan dilakukan dengan menggunakan Command Line	Pengambilan data dapat dilakukan dan terbuat file baru berekstensi <i>pcap</i>	OK
Pengambilan file jaringan dari raspberry pi kedalam server	Pengambilan file pcap hasil monitoring jaringan pada raspberry pi kedalam	Transfer data dapat dilakukan dan muncul nama file hasil monitoring	OK

	server dengan menggunakan command line	dari raspberry pi pada server	
Penggabungan file hasil monitoring	Penggabungan hasil file dari monitoring jaringan menjadi 1 file akhir	Semua data pada file hasil monitoring yang telah terkumpul dapat disatukan menjadi 1 file utama	OK
Konversi File Hasil Monitoring	Hasil file akhir monitoring dikonversikan ekstensi dari <i>pcap</i> menjadi <i>csv</i>	Data hasil monitoring akhir sesuai dengan hasil konversi, dan hasil konversi dapat diakses	OK
Pembacaan Hasil Monitoring kedalam dashboard	Hasil konversi diakses oleh aplikasi	Data dapat tertampil pada aplikasi dashboard	OK



	dashboard pada server		
--	-----------------------	--	--

Metode pengujian *system integration testing* dilakukan untuk mengetahui integrasi dari sistem. Skenario *system integration testing* berdasarkan pada *test case* pada sistem monitoring jaringan. Berikut adalah hasil pengujian *system integration testing*

Use Case	Kategori	Skenario	Ekspektasi	OK/NOK
Melihat Dashboard	Normal	Membuka Aplikasi Dashboard	Melihat dashboard hasil dari monitoring jaringan	OK
Mengganti Jaringan Raspberry Pi	Normal	Mengganti jaringan dan IP Address Raspberry Pi yang baru	Jaringan dapat dilakukan pemantauan	OK
	Alternatif I	Mengganti jaringan WiFi tanpa	Program tidak dapat melakukan	OK

		mengga nti IP Address	pemonito ran	
--	--	-----------------------------	-----------------	--

## BAB VII

### KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari keseluruhan penelitian yang dilakukan pada tugas akhir ini serta beberapa saran yang dapat dipertimbangkan untuk pengembangan tugas akhir ini.

#### 7.1 Kesimpulan

Berdasarkan hasil penelitian, berikut ini merupakan beberapa kesimpulan yang dapat diambil.

1. Raspberry Pi sebagai sensor untuk mengambil data telah dibuat, Sehingga aktivitas pada jaringan dapat terekam dan diolah dalam bentuk dashboard.
2. Perekaman jaringan masih terbatas apabila tidak menggunakan poisoning dalam jaringan lokal.
3. Pengambilan data efektif dalam penggunaan jaringan yang memiliki banyak router dikarenakan raspberry pi tidak membutuhkan *resource* yang banyak dalam proses pengambilannya dan dapat dilakukan dengan waktu yang bertepatan.

#### 7.2 Saran

Berdasarkan kesimpulan hasil penelitian, adapun saran yang dapat digunakan dalam perbaikan atau penelitian selanjutnya, yaitu :

1. Penelitian selanjutnya dapat mengoptimalkan dengan memberikan fitur *real time* pada proyek pengambilan data melalui sensor.
2. Pada penelitian ini dilakukan oleh penulis baru sehingga dapat menambahkan fitur online dengan website.
3. Adanya pengembangan dalam proses data lebih lanjut, pada penelitian ini tidak dibahas mengenai proses data pada tahap lebih lanjut oleh karena itu diperlukan proses data lebih lanjut pada penelitian berikutnya yang mencakup

filterisasi tanggal, *count* tipe paket yang diakses, dan data penggunaan ip address komputer.

**DAFTAR PUSTAKA**

- [1] Saxena, P. (2017). Analysis of Network Traffic by using Packet Sniffing Tool : Wireshark. *International Journal of Advanced Research, Ideas and Innovations In Technology*, 3(6), 804–808.
  
- [2] Kaur, I., Kaur, H., & Singh, E. G. (2014). Analysing Various Packet Sniffing Tools. *International Journal of Electrical Electronics & Computer Science Engineering*, 1(5), 65–69. Retrieved from <http://www.ijeece.com/V1N5-014.pdf>
  
- [3] Asrodia, P., & Patel, H. (2012). Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis. *International Journal of Electrical*, 1, 4. Retrieved from <http://oaji.net/articles/2014/660-1395767550.pdf>

*Halaman ini sengaja dikosongkan*

## LAMPIRAN

### Lampiran A – Source Code Python Web

```
from flask import Flask, redirect, url_for,
session, request, jsonify
from flask_oauthlib.client import OAuth

app = Flask(__name__)
app.config['GOOGLE_ID'] = "899324780450-
7mke22rvm4064rfgkd95okqa62q2m8ic.apps.googleuser
content.com"
app.config['GOOGLE_SECRET'] =
"X94DkxEEzjCzMuXHRq7fZUE"
app.debug = True
app.secret_key = 'development'
oauth = OAuth(app)

google = oauth.remote_app(
    'google',
    consumer_key=app.config.get('GOOGLE_ID'),
    consumer_secret=app.config.get('GOOGLE_SECRET'),
    request_token_params={
        'scope': 'email'
    },
    base_url='https://www.googleapis.com/oauth2/v1/'
    ,
    request_token_url=None,
    access_token_method='POST',
```

```
access_token_url='https://accounts.google.com/o/oauth2/token',

authorize_url='https://accounts.google.com/o/oauth2/auth',
)

@app.route('/')
def index():
    if 'google_token' in session:
        me = google.get('userinfo')
        return jsonify({"data": me.data})
    return redirect(url_for('login'))

@app.route('/login')
def login():
    return google.authorize(callback=url_for('authorized',
_externa=True))

@app.route('/logout')
def logout():
    session.pop('google_token', None)
    return redirect(url_for('index'))

@app.route('/login/authorized')
def authorized():
    resp = google.authorized_response()
    if resp is None:
```



```
        return 'Access denied: reason=%s
error=%s' % (
            request.args['error_reason'],
            request.args['error_description']
        )
    session['google_token'] =
(resp['access_token'], '')
    me = google.get('userinfo')
    return jsonify({"data": me.data})

@google.tokengetter
def get_google_oauth_token():
    return session.get('google_token')

if __name__ == '__main__':
    app.run()
```

*Halaman ini sengaja dikosongkan*

## BIODATA PENULIS



Penulis bernama Rizandi Choirul Anam, lahir di Malang pada tanggal 26 Mei 1996, merupakan anak kedua dari tiga bersaudara. Penulis telah menempuh jenjang pendidikan formal di beberapa sekolah, yaitu : SD Al-Hikmah (2002 - 2008), SMP Al-Hikmah(2008 - 2011), dan SMA Negeri 15 Surabaya (2011 - 2014). Penulis melanjutkan pendidikan sarjana di Departemen Sistem Informasi Fakultas Teknologi Elektro dan Informasi Cerdas

(FTEIC) Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2014 yang terdaftar sebagai mahasiswa dengan NRP 05211440000163.

Selama menjadi mahasiswa, penulis aktif mengikuti unit kegiatan mahasiswa, seperti IFLS ITS pada tahun 2014-2015, serta kegiatan kepanitiaan, seperti Information Systems Expo 2014. Penulis menjadi asisten praktikum mata kuliah “Keamanan Aset Informas” selama satu semensater pada tahun 2016 dan “Forensika Digital” selama satu semester pada tahun 2018.

Penulis dapat dihubungi melalui email [rizandi.choirul14@mhs.is.its.ac.id](mailto:rizandi.choirul14@mhs.is.its.ac.id).

*Halaman ini sengaja dikosongkan*