PERANCANGAN SISTEM MONITORING *TRAFFIC* PADA JARINGAN *WIRELESS* MENGGUNAKAN SISTEM EMBEDDED

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun Oleh:

KAUTSARANI PERMATA ALAM

NIM. 0910683057

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

MALANG

2014

LEMBAR PERSETUJUAN

PERANCANGAN SISTEM MONITORING *TRAFFIC* PADA JARINGAN *WIRELESS* MENGGUNAKAN SISTEM EMBEDDED

Disusun Oleh :

Kautsarani Permata Alam

0910683057

Telah diperiksa dan disetujui:

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Heru Nurwarsito, M.Kom.	Gembong Edhi Setyawan, S.T,. M.T.
NIP. 19650402 199002 1 001	NIK. 761201 16 1 1 0373

LEMBAR PENGESAHAN

PERANCANGAN SISTEM MONITORING *TRAFFIC* PADA JARINGAN *WIRELESS* MENGGUNAKAN SISTEM EMBEDDED

Disusun oleh :

Kautsarani Permata Alam
NIM. 0910683057

Skripsi ini diuji dan dinyatakan lulus pada
tanggal 11 Juli 2014

Penguji I

Sabriansyah R.A, S.T., M.Eng.
NIK. 820809 06 1 1 0084

Penguji II

Agung Setia Budi, S.T., M.T.

Penguji III

Wijaya Kurniawan, S.T., M.T.
NIK. 820125 16 1 1 0481

Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINILITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh pihak lain dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebut dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (Sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan pasal 70).

Malang, Juli 2014

Mahasiswa,

Kautsarani Permata Alam
0910683057

KATA PENGANTAR

Alhamdulillah senantiasa penulis ucapkan atas limpahan serta rahmat dan hidayah ALLAH SWT kepada kita semua sehingga hanya dengan petunjuk dan"Perancangan Sistem Monitoring Traffic pada Jaringan Wireless Menggunakan Sistem Embedded". Shalawat serta Salam kepada Nabi Muhammad SAW yang menjadi tauladan umat manusia, khususnya kita sekalian.

Adapun tujuan dari penulisan skripsi ini adalah untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya Malang.

Dalam kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesarnya atas saran, bimbingan, motivasi dan petunjuk yang diberikan oleh beberapa pihak yang membantu menyelesaikan skripsi ini kepada:

1. Bapak Ir. Heru Nurwarsito, M.Kom. dan Bapak Gembong Edhi Setyawan, S.T., M.T. selaku Dosen Pembimbing yang telah memberikan nasehat, arahan, masukan serta saran sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Bapak Sugiharso, Ibu Retno Suryaningrum, saudara-saudaraku tercinta Abang Reggi, Adik Risa, Adik Raka dan seluruh keluarga besar yang senantiasa mendoakan, mendukung dan mencurahkan cintanya. Semoga Allah SWT senantiasa memberikan perlindungan, kasih sayang dan ridho-Nya.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwarsito, M.Kom, Bapak Himawat Aryadita, S.T., M.Sc, dan Bapak Eddy Santoso, S.Kom, selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom,. M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
5. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.

6. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama menyelesaikan skripsi.
7. Rezky Wahyu Satrio yang selalu memberikan semangat, doa serta kritik dan saran kepada penulis selama penulisan skripsi ini.
8. Melynda Aula Rahmawati, S.Kom, Alfina Zulfa, S.Kom, Dara Nisa‘ Ulum Bilgies, S.Kom, Ria Agustina, S.Kom dan sahabat-sahabat yang selalu setia dan memberi dukungan kepada penulis.
9. Teman-teman Teknik Informatika 2009.
10. Semua pihak yang telah memberikan motivasi dan dorongan yang tidak ternilai hingga terselesaikannya skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, baik dari segi materi, sistematika pembahasan, maupun susunan bahasa. Oleh karena itu, saran dan masukan yang konstruktif senantiasa penulis harapkan. Semoga ke depannya skripsi ini bisa bermanfaat dalam pengembangan pendidikan bangsa Indonesia.

Malang, 18 Juni 2014

Penulis

ABSTRAK

Kautsarani Permata Alam. 2014. Perancangan Sistem Monitoring Traffic pada Jaringan Wireless Menggunakan Sistem Embedded.
Dosen Pembimbing: Ir. Heru Nurwarsito, M.Kom. dan Gembong Edhi Setyawan, S.T,. M.T.

Dengan meningkatnya perkembangan teknologi di era informasi dan globalisasi, penggunaan teknologi *wireless* juga semakin meningkat. Peningkatan penggunaan *wireless* pada suatu area menyebabkan terjadinya kepadatan lalu lintas data, sehingga dibutuhkan media yang mampu melakukan *monitoring* jaringan. Dalam penelitian ini, dibangun sistem monitoring *traffic* menggunakan perangkat *raspberry pi*, yang berguna untuk mengumpulkan informasi kapasitas *traffic* jaringan. Perancangan sistem memanfaatkan protokol SNMP (*Simple Network Management Protocol*) sebagai media penghubung dan bertukar informasi agar dapat melakukan *monitoring* jaringan *wirelesss*. Implementasi sistem menggunakan bahasa pemrograman *python* dengan *library pysnmp*. Berdasarkan pengujian koneksi dan perhitungan *traffic*, sistem mampu terkoneksi ke jaringan *wireless* dengan baik dan dapat membaca jumlah *traffic*. Nilai rata – rata yang didapat *raspberry pi* pada saat *access point* tidak terkoneksi ke internet: $\overline{Tx}$ 2.174 MB dan $\overline{Rx}$ 1.234 MB, pada saat *access point* sudah terkoneksi ke internet: $\overline{Tx}$ 10.894 MB dan $\overline{Rx}$ 2.996 MB, pada saat 1 user melakukan browsing: $\overline{Tx}$ 23.694 MB dan $\overline{Rx}$ 4.366 MB, pada saat 3 user melakukan browsing: $\overline{Tx}$ 31.910 MB dan $\overline{Rx}$ 11.720 MB, pada saat 1 user melakukan download: $\overline{Tx}$ 96.954 dan $\overline{Rx}$ 10.173 MB dan pada saat 3 user melakukan download: $\overline{Tx}$ 143.500 dan $\overline{Rx}$ 19.750 MB. Hasil uji perhitungan jumlah *traffic* menggunakan *raspberry pi* mempunyai nilai yang relatif lebih kecil daripada *mikrotik router* dan mempunyai nilai yang relatif lebih besar daripada *the dude*. Perbandingan akurasi hasil *monitoring raspberry pi* terhadap *mikrotik router* dengan nilai: $\overline{Tx}$ akurasi 0.987 dan $\overline{Rx}$ akurasi 0.991, sedangkan perbandingan terhadap *the dude* dengan nilai: $\overline{Tx}$ akurasi 1.0007 dan $\overline{Rx}$ akurasi 1.0022.

Kata Kunci: *monitoring jaringan, protokol SNMP dan sistem embedded*

ABSTRACT

Kautsarani Permata Alam. 2014. ***Traffic Monitoring System Design on Wireless Network Using Embedded System*****.**
Lecturer Preceptor**: Ir. Heru Nurwarsito, M.Kom.** ***and*** **Gembong Edhi Setyawan, S.T,. M.T.**

By the increasing technology development at the age of information and globalization, the wireless technology users were also increasing. The wireless technology user enhancement in a certain area causing a density of the data traffic, so that required a media in order to do network monitoring. Then built a wireless network monitoring system which use for compiling network traffic capacity information. In this research, the traffic monitoring system designed using raspberry pi. The system design utilized SNMP protocol (Simple Network Management Protocol) as linking and exchange media in order to do wireless network monitoring. The system implementation utilized phyton programming language with pysnmp library. Based on connection and traffic calculation testing, system was capable to capture and connect also read wireless network traffic capacity. The average value that raspberry pi obtained when the access point not connected to internet: $\overline{Tx}$ 2.174 MB and $\overline{Rx}$ 1.234 MB, when the access point connected to internet: $\overline{Tx}$ 10.894 MB and $\overline{Rx}$ 2.996 MB, when 1 user browsing: $\overline{Tx}$ 23.694 MB and $\overline{Rx}$ 4.366 MB, when 3 users are browsing: $\overline{Tx}$ 31.910 MB and $\overline{Rx}$ 11.720 MB, when 1 user downloading: $\overline{Tx}$ 96.954 MB and $\overline{Rx}$ 10.173 MB and when 3 users are downloading: $\overline{Tx}$ 143.500 MB and $\overline{Rx}$ 19.750 MB. The test results of calculating traffic capacity using raspberry pi have smaller value relatively than the mikrotik router and greater value relatively than the dude. The results monitoring accuracy comparison of raspberry pi than the mikrotik router with value: $\overline{Tx}$ accuracy was 0.987 and $\overline{Rx}$ accuracy was 0.991, while the comparison than the dude with value: $\overline{Tx}$ accuracy was 1.0007 and $\overline{Rx}$ accuracy was 1.0022.

Keyword*: network monitoring, SNMP protocol and embedded system.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Di era informasi dan globalisasi, dunia komunikasi meningkat sangat drastis dengan ditandai banyak bermunculan sarana - sarana komunikasi baru dengan fitur beragam. Salah satu aspek perkembangan teknologi ditandai dengan semakin maraknya penggunaan internet menggunakan teknologi *wireless*. Penggunaan teknologi *wireless* pada suatu area dapat menyebabkan terjadinya kepadatan lalu lintas data, sehingga dibutuhkan sebuah media yang mampu melakukan *monitoring* pada jaringan. Maka dibangun sebuah sistem monitoring pada jaringan *wireless* yang berguna untuk mengumpulkan informasi dalam *traffic*, termasuk memantau kondisi *traffic* pada jaringan. Pada penelitian sebelumnya, yaitu oleh [MOS-09], dilakukan pemantauan lalu lintas pada jaringan internet menggunakan sebuah sistem ENTM (*Embedded Network Traffic Monitoring*) dengan *open source embedded linux* sebagai operasi sistem. Sistem ENTM (*Embedded Network Traffic Monitoring*) yang dibangun mampu meneliti paket jaringan, menganalisa data yang diteliti, dan memaparkan data kasar serta data hasil analisa tersebut. Sistem ini *handy device* bagi *administrator* untuk menganalisa data *traffic* yang masuk dan keluar. Dalam penelitian kali ini, akan dirancang sistem monitoring yang dapat melihat paket ak7 tual dari *traffic* pada jaringan *wireless* dan melaporkan berdasarkan *traffic* jaringan tersebut. Sistem monitoring pada jaringan internet penting adanya untuk memahami perilaku jaringan sehingga *administrator* dapat bereaksi tepat dan dapat membantu merancang serta memberikan masa depan jaringan yang lebih efisien [MOS-09].

Untuk membangun sebuah sistem monitoring, dibutuhkan sarana pendukung, yaitu sistem embedded. Perbedaan antara sistem embedded dengan sistem komputer terletak pada kegunaan dari masing – masing alat tersebut. Komputer banyak digunakan di berbagai bidang karena sifatnya yang “umum” (*general*) sehingga desain sistemnya sangat kompleks, sedangkan sistem embedded hanya dibuat untuk suatu kegunaan tertentu. Karena itu, spesifikasi yang dibutuhkan untuk sistem ini lebih “kecil” dan sederhana bila dibandingkan

dengan sistem komputer. Beberapa alasan penting yang menjadikan sistem embedded sering digunakan adalah karena perangkat ini mampu menjalankan aplikasi yang membutuhkan daya komputasi kecil, memiliki ukuran yang kecil, serta membutuhkan biaya yang rendah atau murah yang dapat digunakan dengan daya rendah untuk melakukan *monitoring* ataupun kontrol terhadap suatu mesin atau peralatan [SYU-09]. Sehingga dengan demikian, sistem embedded sangat cocok diimplementasikan untuk melakukan pekerjaan secara spesifik atau khusus, guna meminimalkan sumber daya.

Penelitian ini memanfaatkan adanya protokol SNMP (*Simple Network Management Protocol*) sebagai media penghubung antara pengirim dan penerima dalam berkomunikasi serta bertukar informasi agar dapat melakukan *monitoring* terhadap jaringan *wirelesss*. Protokol SNMP dirancang untuk memberikan kemampuan kepada pengguna/ *administrator* dalam memantau dan mengatur jaringan komputer secara sistematis dari jarak jauh atau yang sering disebut dengan *remotely* [NIK-07]. SNMP ini sangat membantu pekerjaan *administrator* dalam melakukan *monitoring* perangkat – perangkat jaringan dengan perintah – perintah yang mudah dipahami. SNMP akan mengambil informasi, biasanya disebut *query*, baik sumber daya yang digunakan maupun proses yang sedang berjalan [ARY-11]. Sehingga pesan SNMP yang diterima akan diolah untuk diketahui jumlah *traffic* yang melintas pada jaringan *wireless*. Hal tersebut penting dilakukan untuk mengelola dan memanfaatkan sumber daya jaringan secara efektif. Sistem ini dikembangkan dengan pemrograman bahasa *Python* dengan *library PySNMP* untuk pengaplikasian protokol SNMP.

Pada perancangan sistem monitoring *traffic* jaringan *wireless* ini digunakan sistem operasi *Raspbian Wheezy*, yaitu sistem operasi *linux* yang dikembangkan khusus untuk bekerja pada sebuah perangkat sistem embedded yakni *Raspberry Pi*. Dimana perangkat *Raspberry Pi* ini juga memanfaatkan perangkat berupa *wifi dongle* sebagai media transmisi agar dapat melakukan akses ke jaringan *wireless* melalui *access point*. Diharapkan penelitian ini dapat menghasilkan sebuah perangkat sistem monitoring *traffic* menggunakan sistem embedded yang mampu membaca data informasi pada jaringan *wireless*, yaitu jumlah *traffic* yang melintas dengan memanfaatkan protokol SNMP.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan diatas, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Merancang dan membangun sistem monitoring jaringan *wireless* menggunakan *Raspberry Pi.*
2. Bagaimana sistem monitoring yang dibangun mampu membaca data informasi di jaringan *wireless*, yaitu jumlah *traffic* yang melintas dengan memanfaatkan protokol SNMP.
3. Bagaimana hasil dari analisa kinerja *traffic* jaringan *wireless* dengan menggunakan *Raspberry Pi* dibandingkan *tools monitoring* yang lain.

1.3 Batasan Masalah

Agar diperoleh hasil pembahasan sesuai dengan yang diharapkan, maka perlu diberikan batasan masalah pada pengembangan sistem yang dibuat, yaitu:

- Perangkat *Embedded System* yang digunakan dalam perancangan adalah *Raspberry Pi*.
- Media transmisi yang digunakan yaitu *Wireless*.
- Platform yang digunakan yaitu *Linux*.
- Operasi sistem yang digunakan adalah *Raspbian Wheezy*.
- Analisis jaringan dilakukan pada protocol TCP/ IP dan UDP.
- Pengujian sistem menggunakan cara simulasi, yaitu membuat jaringan *WLAN* yang memanfaatkan *mikrotik router* dan *wireless access point.*
- Penelitian ini hanya mampu membaca data informasi di jaringan *wireless*, yaitu jumlah *traffic* yang melintas dengan memanfaatkan protokol SNMP.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah membangun sistem monitoring menggunakan sistem embedded, yaitu *Raspberry Pi* yang mampu membaca nilai data informasi berupa *traffic* di jaringan *wireless* dan melakukan analisa kinerja *traffic.* Sehingga nantinya memudahkan *administrator* jaringan dalam memantau kondisi *access point* yang ada, karena sistem dapat diakses dengan mudah dan cepat.

1.5 Manfaat Penelitian

Manfaat bagi pengguna yang nantinya akan didapat dari penelitian ini adalah menyediakan perangkat murah bersumber daya rendah yang dapat digunakan dengan mudah dan cepat untuk memantau kondisi teraktual *traffic* dari setiap *access point* yang ada melalui jaringan *wireless.*

1.6 Sistematika Pembahasan

Sistematika pembahasan dari penyusunan tugas akhir *Perancangan Sistem Monitoring Traffic pada Jaringan Wireless Menggunakan Sistem Embedded* direncanakan sebagai berikut:

BAB I PENDAHULUAN

Bab I terdiri dari latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika pembahasan penelitian. Bab I menjadi dasar dari keseluruhan pelaksanaan penelitian mengenai sistem monitoring *traffic* pada jaringan *wireless* menggunakan sistem embedded.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab II membahas kajian pustaka yang menjadi dasar acuan dilakukannya penelitian mengenai perancangan sistem monitoring *traffic* pada jaringan *wireless* menggunakan sistem embedded serta membahas teori – teori yang berkaitan dan menunjang dalam penyelesaian penelitian. Dasar teori yang diambil berasal dari jurnal, buku, dan sumber referensi lainnya yang berhubungan dengan topik yang diteliti.

BAB III METODE PENELITIAN DAN PERANCANGAN

Bab III menjelaskan metode atau langkah – langkah dalam perancangan sistem, implementasi sistem, pengujian sistem, dan analisis sistem yang dibangun. Serta membahas perancangan sistem yang terdiri dari perancangan perangkat lunak, perancangan perangkat keras, perancangan topologi jaringan dan perancangan cara kerja program yang digunakan dalam implementasi sistem.

BAB IV IMPLEMENTASI

Bab IV menjelaskan tentang cara implementasi sistem dan menjelaskan penelitian secara terperinci dengan menampilkan gambar – gambar dari hasil implementasi yang telah dilakukan.

BAB V PENGUJIAN DAN ANALISIS

Bab V berisi pengujian dan analisis terhadap sistem yang dibangun. Pengujian tersebut dilakukan secara bertahap sesuai dengan penjelasan pada Bab III.

BAB VI PENUTUP

Bab VI berisi kesimpulan dari pelaksanaan penelitian yang dibuat setelah dilakukan pengujian dan analisis. Untuk meningkatkan hasil dari kinerja sistem yang telah dibuat dalam penelitian ini, maka diberikan saran – saran untuk perbaikan dan penyempurnaan sistem yang telah dibangun

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

Pada Bab II membahas kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai perancangan sistem monitoring *traffic* pada jaringan *wireless* menggunakan sistem embedded. Pada penelitian ini, dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah: konsep dasar *embedded system, Raspberry Pi, embedded linux, Raspbian Wheezy, wireless access point, monitoring* jaringan, konsep dasar protokol SNMP dan penerapannya pada bahasa pemrograman *Python* serta referensi - referensi yang digunakan di dalam membuat sistem.

2.1 Kajian Pustaka

Penelitian yang membahas mengenai *monitoring* jaringan menggunakan sistem embedded, salah satunya adalah penelitian yang dilakukan oleh [MOS-09]. Yang dilakukan pada penelitian tersebut adalah pemantauan lalu lintas pada jaringan internet menggunakan sebuah sistem ENTM (*Embedded Network Traffic Monitoring*) yang menggunakan *open source embedded linux* sebagai operasi sistem. Peralatan utama yang dibutuhkan untuk membangun sistem ENTM adalah *TS-5400 SBC, LCD panel, keypad* dan *Compact Flash* (*CF*) *Card*. Sedangkan sistem perangkat lunak ENTM terdiri dari empat (4) modul, yaitu *System Control* (SC), *Network Packet Probe* (NPP), *Packet Analysis* (PA) and *View Module* (VM). Modul SC bertindak sebagai *interface*/ menu untuk menjalankan berbagai fungsi sistem dan integrasi perangkat eksternal (*keypad* dan LCD *panel*) ke SBC (*Single Board Computer*). Modul NPP melakukan *capture* paket dari segmen jaringan, mengekstrak informasi paket dan menyimpannya dalam *buffer* data sementara untuk analisa lebih lanjut. Modul PA melacak informasi *host* global dan individu untuk melihat ke dalam file. Modul VM digunakan untuk menampilkan, menganalisis data melalui browser web. Sistem ENTM (*Embedded Network Traffic Monitoring*) yang dibangun mampu meneliti paket jaringan, menganalisa data yang diteliti, dan memaparkan data kasar serta data hasil analisa tersebut. Sistem ini *handy device* bagi *administrator* untuk menganalisa data

traffic yang masuk dan keluar. Analisa kinerja sistem yang signifikan untuk menjamin kehandalan dan kepraktisan, dimana performa dari sistem ENTM (*Embedded Network Traffic Monitoring*) yang dibangun telah dibandingkan dengan *software* pada *Desktop PC* dan *Wireshark*. Hasilnya menunjukkan bahwa sistem ENTM (*Embedded Network Traffic Monitoring*) mempunyai daya saing tinggi walaupun spesifikasi pemrosesan dan memori kecil.

Penelitian lain yang berkaitan dengan *monitoring* jaringan adalah penelitian [ARY-11]. Penelitian ini berisi tentang membangun suatu sistem manajemen jaringan secara *online* berbasis PHP dan protokol *Simple Network Management Protokol* (SNMP) menggunakan *handphone* dengan sistem operasi *Android*. Sistem ini memanfaatkan parameter MIB untuk mendapatkan kondisi teraktual dari *router* yang dipantau. Karena *router* memegang peranan yang sangat penting dalam jaringan komputer, sehingga kondisinya harus dipantau untuk mengetahui sejak dini jika kerusakan terjadi pada *router* tersebut, sehingga tidak mengganggu stabilitas jaringan. Untuk memudahkan administrator jaringan dalam memantau kondisi *route*r, maka diperlukan suatu sistem peringatan dini yang dapat diakses lebih mudah dan cepat, yaitu melalui piranti yang selalu dibawa oleh *administrator*, yaitu *handphone*. Apabila terdapat suatu kondisi yang mengkhawatirkan, sistem akan menginformasikan kepada pengguna atau admin. Hasil uji coba menunjukkan bahwa sistem ini dapat menampilkan objek – objek (data informasi) yang di-*monitor* pada *handphone* dengan baik, ditunjukkan dengan berfungsinya sistem *monitoring* dengan baik dan informasi yang ditampilkan adalah akurat dan aktual.

Berdasarkan penelitian yang telah dibahas diatas, pemanfaatan teknologi sistem embedded tepat bila diterapkan pada perangkat SBC (*Single Board Computer*) dalam melakukan analisa kinerja *traffic* pada jaringan *wireless*. Pada kedua penelitian tersebut, perangkat – perangkat yang digunakan dalam membangun sebuah sistem monitoring menggunakan sistem embedded adalah perangkat *TS-5400 SBC* [MOS-09] dan *handphone android* [ARY-11]. Seperti yang diketahui bahwa teknologi *handphone android* merupakan *small system* pada *embedded system* yang disebut *system on-chip* (SOCs), dimana semua fitur/ sistem ditanamkan dalam teknologi *chip*. Hal ini membuktikan bahwa penerapan

teknologi sistem embedded menggunakan *embedded linux* dapat bekerja dengan baik pada perangkat SBC (*Single Board Computer*) yang hanya membutuhkan sumber daya rendah, terutama bila diterapkan menggunakan *Raspberry Pi*, yaitu salah satu sistem embedded yang mudah dirancang dan diimplementasikan. *Raspberry Pi* adalah *PC* kecil yang dapat digunakan untuk banyak hal seperti layaknya sebuah *PC desktop*, selain berukuran kecil sebesar kartu kredit, *raspberry pi* hanya membutuhkan sumber daya rendah, memiliki memori RAM yang cukup besar yaitu 512 MB untuk *type* B, serta murah.

2.2 Dasar Teori

2.2.1 Pengertian Sistem Embedded

Sistem embedded merupakan sistem komputer yang berorientasi pada aplikasi spesifik dalam skala yang bervariasi baik pada perangkat lunak maupun perangkat kerasnya. Sistem ini harus memenuhi kebutuhan akan kegunaan, kehandalan, biaya, kapasitas dan sumber daya dari suatu aplikasi. *Single Board Computer* (SBC) merupakan sebuah komputer lengkap yang dibangun pada sebuah *Printed Circuit Board* (PCB). Seperti halnya sebuah komputer pada umumnya, dalam sebuah SBC sudah terdapat *microprocessor* dengan *Random Access Memory* (RAM), IO, ADC/ DAC dan semua fitur lain yang diperlukan untuk berfungsi sebagai komputer dalam sebuah *board*. Sebuah SBC dapat diimplementasikan dalam berbagai *embedded system*, seperti pada sistem kendali waktu nyata, *monitoring*, ataupun *interface* IO [LIN-10].

Salah satu sistem embedded yang mudah dirancang dan diimplementasikan adalah *Raspberry Pi*, yaitu perangkat *mini computer* berukuran sebesar kartu kredit yang dapat dihubungkan ke *TV* dan *keyboard*. *Raspberry Pi* adalah *PC* kecil yang dapat digunakan untuk banyak hal selayaknya *PC desktop*, seperti *spreadshee*t, pengolah kata dan permainan. *Raspberry Pi* juga dapat digunakan untuk memainkan *video* dengan definisi tinggi. *Raspberry Pi* memiliki dua model yakni model A dan B, model A memiliki 256 MB *RAM*, 1 port USB dan tanpa *Ethernet* sedangkan pada tipe B memiliki 512 MB *RAM*, 2 port USB dan memiliki *Ethernet* [ANG-13]. *Raspberry Pi* model B ditunjukkan pada Gambar 2.1.

Gambar 2.1 *Raspberry Pi* model B
Sumber: [ANG-13]

Adapun spesifikasi *Raspberry Pi* model B berdasarkan [ANG-13] dijelaskan pada Tabel 2.1.

Tabel 2.1 Spesifikasi *Raspberry Pi* model B

Spesifikasi Raspberry Pi Type B	
Chip	Broadcom BCM2835 SoC full HD multimedia application processor
CPU	700 MHz Low Power ARM1176JZ-F Application Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	512 MB SDRAM
Ethernet	Onboard 10/100 ethernet RJ45 jack
USB 2.0	Dual USB connector
Video Output	HDMI (rev 1.3 &1.4) Coposite RCA (PAL and NTSC)
Audio Output	3.5 Jack, HDMI
Media Penyimpanan	SD. MMC, SDIO Card slot
Sistem Operasi	Linux
Dimensi	8,6 cm x 5,4 cm x 1,7 cm

Sumber: [ANG-13]

2.2.2 *Embedded Linux*

Linux merupakan sistem operasi yang bersifat modular, sehingga mudah untuk dilakukan penyesuaian (*configurable*) terhadap lingkungan pengoperasian dengan cara mengurangi program *utility*, *tools*, dan *services* lainnya yang tidak diperlukan dalam sistem embedded. *Linux* telah berhasil dijalankan pada prosesor dengan berbagai arsitektur dan berbagai tipe CPU. Keuntungan menggunakan *Linux* diantaranya, *Linux* merupakan sistem operasi *open source* yang telah

banyak mendukung *device*, *system file*, protokol jaringan, memperbaiki *bugs*, serta terus menambahkan layanan baru secara konstan dan teruji karena didukung oleh komunitas pemrograman dan pengguna yang sangat besar [LIN-10].

Salah satu sistem operasi yang banyak digunakan dalam sistem embedded adalah *embedded linux*, sebuah distro linux yang menggunakan kernel 2.x tertanam pada mikrokontroler. Keunggulan utama dari *embedded linux* adalah bahwa sistem operasi ini mempunyai fungsi yang lengkap. Dengan dukungan layanan yang baik terhadap jaringan komunikasi menyebabkan sistem operasi ini menjadi pilihan yang tepat dalam mengembangkan sistem embedded. *Embedded linux* memiliki sistem platform *cross-compilation*, Sistem embedded umumnya mempunyai *resources* yang terbatas sehingga kita tidak dapat menginstal *tool chain* dan *source code* diatasnya. Sebagai gantinya kita gunakan PC sebagai *computer host*, dan menggunakan *cross-compiler* untuk membuat program yang berjalan diatas *embedded system* (juga disebut sebagai *target system*) [LIN-10].

Embedded Linux (operasi sistem) yang umumnya digunakan pada perangkat *Raspberrry Pi* adalah *Raspbian Wheezy* yang berbasis *Debian. Raspbian Wheezy* dioptimalkan dan dikembangkan untuk kinerja terbaik pada *Raspberry Pi*, selesai pada Juni 2012. Raspbian dibuat lebih dari 35.000 paket, *Software pre-compiled* dibundel dalam format yang bagus untuk memudahkan instalasi pada *Raspberry Pi*. Namun, *Raspbian Wheezy* masih dalam pengembangan aktif dengan penekanan pada peningkatan stabilitas dan kinerja paket *Debian* sebanyak mungkin [ANG-13].

2.2.3 Monitoring Jaringan

Manajemen jaringan adalah kemampuan untuk memonitor, mengontrol dan merencanakan suatu jaringan komputer dan komponen sistem. Monitoring jaringan merupakan bagian dari manajemen jaringan. Hal yang paling mendasar dalam konsep manajemen jaringan adalah tentang adanya *manager* atau perangkat yang memanajemen dan *agent* atau perangkat yang dimanajemen. Dalam implementasinya, ada berbagai macam arsitektur manajemen jaringan yang didasarkan pada tipe dan ukuran masing - masing. Ada dua arsitektur yang dapat

digunakan yaitu manajemen terpusat (*centralized management*) dan manajemen menyebar (*distributed management*) [REZ-13].

Berdasarkan [WIL-99], tujuan *network monitoring* adalah mengumpulkan informasi mengenai status dan tingkah laku pada sebuah jaringan. Kegunaan *network monitoring* adalah:

- Mengetahui *down* atau *up* – nya suatu link dan menginformasikannya pada orang yang bertanggung jawab.
- Memantau baik buruknya suatu jalur dalam jaringan, jika ada paket yang hilang atau bertabrakan (*collision*), jaringan akan mengalami masalah.
- Mengetahui banyaknya paket data yang lewat (terkirim dan diterima).

Dengan kata lain *network monitoring* adalah suatu perangkat lunak yang memberikan kemampuan pada sebuah *workstation* untuk memantau lalu lintas jaringan.

2.2.4 *Wireless* LAN

Berdasarkan [RUD-11], WLAN (*Wireless* LAN) adalah suatu jaringan area lokal tanpa kabel dimana media transmisinya menggunakan frekuensi radio (RF) dan *infrared* (IR), untuk memberi sebuah koneksi jaringan ke seluruh pengguna dalam area disekitarnya. Area jangkauannya dapat berjarak dari ruangan kelas ke seluruh kampus atau dari kantor ke kantor yang lain dan berlainan gedung. Peranti yang umumnya digunakan untuk jaringan WLAN termasuk di dalamnya adalah *PC*, *notebook*, *PDA*, *handphone*, dan lain sebagainya. Teknologi *wireless* memiliki kelebihan dan kelemahan antara lain:

a. Kelebihan teknologi *wireless*:

a. Mobilitas
   - Bisa digunakan kapan saja
   - Kemampuan akses data pada jaringan *wireless* itu *real time*, selama masih di area *hot spot*

b. Kecepatan instalasi
   - Proses pemasangan cepat
   - Tidak perlu menggunakan kabel

c. Fleksibilitas tempat
   - Bisa menjangkau tempat yang tidak mudah dijangkau kabel

d. Jangkauan luas

e. Biaya pemeliharaannya murah (hanya mencakup stasiun bukan seperti pada jaringan kabel yang mencakup keseluruhan kabel)

f. Infrastrukturnya berdimensi kecil

g. Mudah dikembangkan

h. Mudah dan murah untuk direlokasi dan mendukung portabelitas

b. Kelemahan teknologi *wireless*:

a. Transmit data kecil, sedangkan jika menggunakan kabel akan lebih cepat

b. Alatnya cukup mahal

c. Mudah terjadi gangguan antara pengguna yang lain (interferensi gelombang)

d. Kapasitas jaringan terbatas

e. Keamanan data kurang terjamin

f. *Intermittence* (sinyal putus - putus)

g. Mengalami gejala yang disebut *multipath* yaitu propagasi radio dari pengirim ke penerima melalui banyak jalur yang LOS

h. Mempunyai *latency* yang cukup besar dibandingkan dengan media transmisi kabel

Secara umum, sistem manajemen WLAN (*Wireless* LAN) harus menyediakan layanan berikut [HAR-04]:

1. *Configuration Management*
2. *Performance Management*
3. *Accounting Management*
4. *Real-time Monitoring*
5. *Fault Management, and*
6. *Security Management*

Untuk membuat sebuah jaringan WLAN, dibutuhkan sebuah *router wireless. Router wireless* adalah sebuah *device* yang berfungsi untuk meneruskan paket – paket dari sebuah *network* ke *network* yang lainnya (baik LAN ke LAN atau LAN ke WAN) sehingga *host – host* yang ada pada sebuah *network* bisa berkomunikasi dengan *host – host* yang ada pada *network* yang lain. Mode *wireless router* dapat diatur sebagai *access point* dan juga berfungsi sebagai *gateway* (gerbang) penghubung dari satu jaringan ke jaringan lainnya [MAL-13].

Access point merupakan perangkat keras yang memungkinkan perangkat *wireless* seperti laptop, ponsel bisa terhubung ke jaringan kabel menggunakan *wireless* atau *wifi*, *bluetooth* atau perangkat standar lainnya. *Wireless Access Point* umumnya dihubungkan ke *router* melalui jaringan kabel, saat ini kebanyakan telah terintegrasi dengan *router* dan dapat digunakan untuk saling mengirim data antar perangkat *wireless* seperti laptop, *printer* yang memiliki *wifi* dan perangkat kabel pada jaringan. *Access point* berfungsi sebagai pengatur lalu lintas data, sehingga memungkinkan banyak *client* dapat saling terhubung melalui jaringan [ANG-13].

2.2.5 SNMP (*Simple Network Management Protokol*)

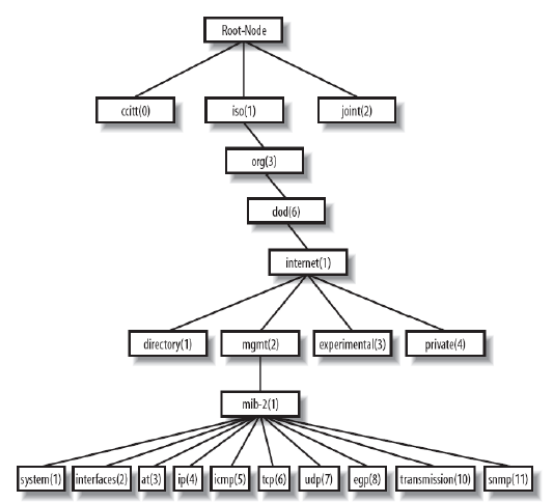
SNMP adalah sebuah protokol aplikasi pada jaringan TCP/ IP yang menangani manajemen jaringan. Protokol ini didesain sehingga pengguna dapat dengan mudah memantau kondisi jaringan komputer. Pemantauan kondisi jaringan dapat dilakukan dengan cara pengumpulan nilai – nilai informasi dari kondisi jaringan secara jarak jauh atau meggunakan satu pusat pengamatan. SNMP menjadi protokol yang terus dikembangkan karena banyak perangkat jaringan yang mendukung dan tersedia layanan SNMP, seperti *router, switch, server, workstation,* dan *printer.* Protokol TCP/ IP menggunakan *transport* UDP oleh karena itu dalam penggunaannya tidak akan membebani *traffic* jaringan [REZ-13].

Berdasarkan [ARY-11], SNMP (*Simple Network Management Protokol*) dibagi menjadi tiga elemen utama, yaitu:

1. ***Management Information Base*** **(MIB)**

MIB atau *Management Information Base* berfungsi sebagai struktur database variable elemen jaringan yang dikelola. MIB memiliki struktur bersifat hierarki yang diatur sedemikian rupa sehingga informasi nilai setiap variable dengan mudah diketahui maksudnya. Informasi dalam SNMP disimpan dalam bentuk variabel – variabel yang didefinisikan dalam MIB, dan masing – masing variabel tersebut memiliki tipe – tipe data tertentu, antara lain adalah *integer*, *octet string*, *display string*, *object identifier*, *null*, dan *sequence of*.

MIB dalam SNMP berbentuk menyerupai sebuah pohon dengan akar – akarnya. *Object Identifier* (OID) mengidentifikasikan atau memberi nama objek – objek dalam pohon MIB yang mendefinisikan tiga Cabang utama yaitu: *Consultative Committee for International Telegraph and Telephone* (CCITT), *International Organization for Standarization* (ISO), dan joint-ISO-CCITT. Sebagian besar aktifitas MIB adalah bagian dari Cabang ISO yang didefinisikan oleh OID 1.3.6.1 dan untuk komunikasi internet. Struktur pohon MIB ditunjukkan pada Gambar 2.2.



Gambar 2.2 Struktur Pohon MIB
Sumber: [ARY-11]

2. ***Agent***

Agent merupakan *software* yang berjalan pada setiap node atau elemen jaringan yang akan dipantau. Fungsi dari *agent* adalah mendapatkan informasi yang diperlukan dari MIB pada setiap node.

3. Manager

Manager adalah *software* yang berjalan di sebuah *host* pada jaringan. Fungsinya mengumpulkan informasi dari *agent – agent. Manager* hanya mengumpulkan informasi – informasi yang diminta oleh administrator jaringan.

Menurut standart IEFT, SNMP didesain untuk pemakaian internet yang didesain di atas protocol UDP (*User Datagram Protokol*) seperti yang digambarkan pada Tabel 2.2.

Tabel 2.2 Protokol SNMP

Sistem Manajemen Jaringan
Agent
SNMP
UDP
IP
Lapisan Bawah

Sumber: [ARY-11]

Cara kerja SNMP adalah dengan jalan *Manager* dan *Agent* saling berkirim pesan berupa permintaan *manager* dan jawaban dari *agent* tentang informasi jaringan yang dibawa oleh paket – paket data yang disebut PDU (*Protocol Data Unit*) [ARY-11]. SNMP menggunakan UDP sebagai protokol transportasi untuk mengirimkan data antara *manager* dan *agent*, UDP lebih dipilih daripada TCP karena sifatnya *connectionless* sehingga tidak ada koneksi *end-to-end* yang dibuat antara *agent* dengan *manager*. SNMP menggunakan port UDP 161 ketika mengirimkan dan menerima permintaan *query* [IBN-10].

2.2.6 *Communication Modes*

Protokol internet dan sistem pengalamatan jaringan mengenal 4 model komunikasi data, yaitu [KEV-03]:

- ***Unicast***

  Menyediakan komunikasi secara *point-to-point*, secara langsung antara 2 *host* dalam sebuah jaringan. *Single host* menerima semua lalu lintas (*traffic*) data.

- ***Anycast***

  Menyediakan metode penyampaian paket data kepada anggota terdekat dari sebuah *group*. Digunakan dalam komunikasi *one-to-one-of-many*. Alamat ini juga digunakan hanya sebagai alamat tujuan (*destination address*) dan diberikan hanya kepada *router*, bukan kepada *host- host* biasa.

- ***Broadcast***

  Proses pengiriman sinyal ke berbagai lokasi secara bersamaan baik melalui satelit, radio, televisi, komunikasi data pada jaringan, dan lain sebagainya.

- ***Multicast***

  Menyediakan metode untuk mengirimkan sebuah paket data ke banyak *host* yang berada dalam *group* yang sama. Digunakan dalam kumunikasi *one-to-many*.

2.3 Perhitungan Akurasi

Perhitungan akurasi dimaksudkan untuk mengetahui presentase dari keberhasilan perancangan sistem, perhitungan ini menggunakan rumus untung-rugi. Perhitungan dilakukan dengan membandingkan hasil pengujian jumlah *traffic* pada sistem emnbedded dengan *mikrotik router* dan *tools the dude*. Persamaannya dapat dilihat pada uraian dibawah:

$$X = a - b \quad (2\text{-}1)$$

$$Y = \frac{X}{b} \quad (2\text{-}2)$$

$$Z = 1 - \frac{\text{jumlah } Y}{\text{banyak } Y} \quad (2\text{-}3)$$

Keterangan:

a = Rata – rata hasil *monitoring raspberry pi*

b = Rata – rata hasil *monitoring mikrotik router/ the dude*

X = Selisih rata-rata hasil *Tx/ Rx*

Y = Nilai perbandingan selisih rata-rata hasil *monitoring raspberry pi* terhadap *mikrotik router/ the dude*

Z = Akurasi rata-rata hasil *monitoring raspberry pi* terhadap *mikrotik router/ the dude*

2.4 Referensi *Source Code*

Referensi *source code* digunakan peneliti sebagai bahan rujukan melakukan *editing source code* yang sudah tersedia untuk implementasi hitung jumlah *traffic*.

2.4.1 *Python* fdb.py

Python fdb.py merupakan program untuk memeriksa ketersediaan OID pada perangkat yang menjalankan layanan protocol SNMP. Di dalam *source code* ini terdapat nilai objek untuk memeriksa ketersediaan OID pada MIB SNMP. Referensi fdp.py dapat dilihat pada Gambar 2.3.

```
import sys
from pysnmp.entity.rfc3413.oneliner import cmdgen

def fetchFdb(ip, community):
    mib = '1.3.6.1.2.1.17.7.1.2.2.1.2'
    value = tuple([int(i) for i in mib.split('.')])
    generator = cmdgen.CommandGenerator()
    comm_data = cmdgen.CommunityData('server', community, 1) # 1
means version SNMP v2c
    transport = cmdgen.UdpTransportTarget((ip, 161))
    real_fun = getattr(generator, 'nextCmd')
    res     =     (errorIndication,     errorStatus,     errorIndex,
varBindTable)\
        = real_fun(comm_data, transport, value)

    if not errorIndication is None  or errorStatus is True:
        print "Error: %s %s %s %s" % res
    else:
        for varBindTableRow in varBindTable:
            varBindTableRow:
```

```
[(ObjectName(1.3.6.1.2.1.17.7.1.2.2.1.2.5.0.27.144.212.92.45),
                Integer(27))]

            data = varBindTableRow[0][0]._value[len(value):]

            vlan = data[0]
            mac = '%s' % ':'.join([hex(int(i))[2:] for i in data[-
6:]])
            mac = '%02x:%02x:%02x:%02x:%02x:%02x' % tuple(map(int,
data[-6:]))
            port = varBindTableRow[0][1]
            yield {'vlan': vlan, 'mac': mac, 'port': port}

if __name__ == '__main__':
    try:
        ip = sys.argv[1]
        community = sys.argv[2]
    except IndexError:
        print "PLease run command like:"
        print "python %s <ip> <community>" % __file__
        sys.exit(0)

    for fdb in fetchFdb(ip, community):
        print 'vlan: %(vlan)s mac: %(mac)s port: %(port)s' % (fdb)
```

Gambar 2.3 *Source code* fdb.py
Sumber: [snmpdesk-0.0.91]

2.4.2 *Python* traffic.py

Python traffic.py merupakan program untuk membaca jumlah *traffic* pada jaringan melalui perangkat yang menjalankan layanan protocol SNMP. Di dalam *source code* ini terdapat nilai objek untuk menghitung jumlah *traffic* yang melintas melalui protokol SNMP. Referensi traffic.py dapat dilihat pada Gambar 2.4.

```
import sys
import collections
from pysnmp.entity.rfc3413.oneliner import cmdgen

'''def datafrommib(mib, community, ip):
    value = tuple([int(i) for i in mib.split('.')])
    generator = cmdgen.CommandGenerator()
    comm_data = cmdgen.CommunityData('server', community, 1) # 1
means version SNMP v2c
    transport = cmdgen.UdpTransportTarget((ip, 161))
```

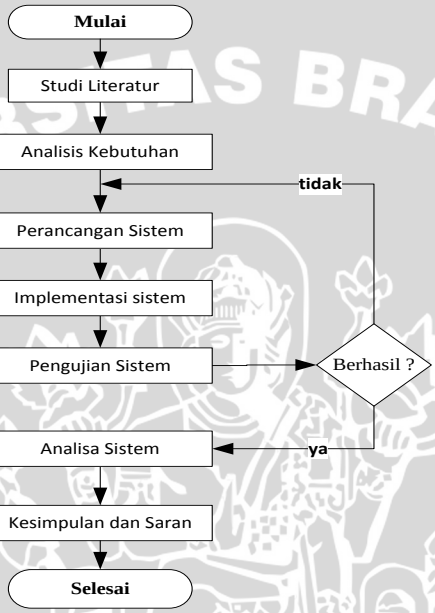
```
    real_fun = getattr(generator, 'nextCmd')
    res     =     (errorIndication,     errorStatus,     errorIndex,
varBindTable)\
            = real_fun(comm_data, transport, value)

    if not errorIndication is None  or errorStatus is True:
           print "Error: %s %s %s %s" % res
           yield None
    else:
        for varBindTableRow in varBindTable:
           varBindTableRow:
              in:           [(ObjectName(1.3.6.1.2.1.2.2.1.10.8),
Counter32(180283794))]
           data = varBindTableRow[0]
           port = data[0]._value[len(value):]
           octets = data[1]

           yield {'port': port[0], 'octets': octets}
'''

def datafrommib(mib, community, conn):
    value = tuple([int(i) for i in mib.split('.')])
    res     =     (errorIndication,     errorStatus,     errorIndex,
varBindTable)\
            = real_fun(comm_data, transport, value)
    res     =     (errorIndication,     errorStatus,     errorIndex,
varBindTable)\
            = conn[3](conn[1], conn[2], value)

    if not errorIndication is None  or errorStatus is True:
           print "Error: %s %s %s %s" % res
           yield None
    else:
        for varBindTableRow in varBindTable:
            varBindTableRow:
              in:           [(ObjectName(1.3.6.1.2.1.2.2.1.10.8),
Counter32(180283794))]
           data = varBindTableRow[0]
           port = data[0]._value[len(value):]
           octets = data[1]

           yield {'port': port[0], 'octets': octets}

def load(ip, community):
     for use snmptool try:
     In: snmpwalk -c mymypub -v2c <ip> 1.3.6.1.2.1.2.2.1.10.2
     Out: snmpwalk -c mymypub -v2c <ip> 1.3.6.1.2.1.2.2.1.16.2
     e.t.c...
    generator = cmdgen.CommandGenerator()
    comm_data  =  cmdgen.CommunityData('server',  community,  1)  #  1
```

```
means version SNMP v2c
    transport = cmdgen.UdpTransportTarget((ip, 161))
    real_fun = getattr(generator, 'nextCmd')
    conn = (generator, comm_data, transport, real_fun)
    mibs = [('1.3.6.1.2.1.2.2.1.16', 'out'),
            ('1.3.6.1.2.1.2.2.1.10', 'in'),
            ('1.3.6.1.2.1.2.2.1.11', 'ucast'),
            ('1.3.6.1.2.1.2.2.1.12', 'nucast'),
            ('1.3.6.1.2.1.2.2.1.13', 'discards'),
            ('1.3.6.1.2.1.2.2.1.14', 'errors')]

    ports = collections.defaultdict(dict)

    for mib in mibs:
        data = datafrommib(mib[0], community, conn)
        for row in data:
            if row:
                ports[row['port']][mib[1]] = int(row['octets'])
            else:
                return None

    return ports

if __name__ == '__main__':
    try:
        ip = sys.argv[1]
        community = sys.argv[2]
    except IndexError:
        print "Please run command like:"
        print "python %s <ip> <community>" % __file__
        sys.exit(0)
     == debug ==
    import profile
    profile.run("load('%s', '%s')" % (ip, community))
    ports = load(ip, community)
    if ports:
        for key, value in ports.items():
            print key, ('in: %(in)s out: %(out)s ucast: %(ucast)s'
+\
                        '     nucast:     %(nucast)s     discards:
%(discards)s' +\
                        ' errors: %(errors)s') % value
```

Gambar 2.4 *Source code* traffic.py
Sumber: [snmpdesk-0.0.91]

BAB III
METODE PENELITIAN DAN PERANCANGAN

Pada Bab III ini dijelaskan langkah – langkah yang akan dilakukan dalam perancangan, implementasi, analisis dan pengujian dari sistem monitoring atas sistem yang dibuat dan kemungkinan arah pengembangan selanjutnya. Diagram alir dari pelaksanaan penelitian secara keseluruhan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Pelaksanaan Penelitian

3.1 Studi Literatur

Studi literatur menjelaskan seluruh dasar teori yang mendukung perancangan sistem monitoring *traffic* pada jaringan *wireless* menggunakan sistem embedded, adapun yang dijadikan dalam bahan studi literatur adalah dasar – dasar teori untuk dapat merancang, membangun beserta cara pengujian sistem, meliputi:

1. *Embedded System*
   a. *Raspberry Pi*
   b. *Raspbian Wheezy*
2. Bahasa Pemrograman
   a. *Python*
3. Topologi Jaringan

a. *Wireless access point*

4. Monitoring Jaringan
   a. Analisa *traffic* jaringan
   b. SNMP

3.2 Perancangan

Perancangan sistem dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Tahap perancangan dibutuhkan agar peneliti dapat dengan mudah melakukan proses selanjutnya terhadap sistem. Pada tahap perancangan ini, seluruh perangkat lunak dan perangkat keras tersebut dirancang sehingga menjadi sebuah sistem yang dikehendaki. Seluruh perangkat lunak digunakan sebagai bagian proses perancangan, pengujian serta analisis sistem jaringan *wireless*. Sedangkan perangkat keras berupa *Raspberry Pi* berfungsi sebagai media kinerja sistem yang diterapkan.

Perancangan sistem memberikan gambaran mengenai sistem yang akan dibuat dalam penelitian yang meliputi sistem perangkat keras dan perangkat lunak, disertai dengan perancangan topologi jaringan sistem dan cara kerja program yang digunakan. Perancangan sistem bertujuan sebagai acuan dalam implementasi sistem.

3.2.1 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk menganalisa dan mendapatkan semua kebutuhan yang diperlukan oleh peneliti dalam perancangan sistem. Analisis kebutuhan menggambarkan keadaan yang mendorong diperlukannya pengukuran jumlah *traffic* pada jaringan *wireless*, yang mencakup analisis perangkat lunak, analisis perangkat keras, dan analisis kebutuhan sistem sebagai bahan analisis kebutuhan yang harus dipenuhi dalam perancangan sistem yang diterapkan.

3.2.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam perancangan sistem monitoring *traffic* pada jaringan *wireless* menggunakan sistem embedded dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan Perangkat Keras

Embedded System	• *Raspberry Pi Type* B • *Processor* ARM • 512 MB SDRAM, > 4 GB SD *Card*
Perlengkapan Jaringan	• Konektor RJ 45 , Kabel UTP Cat 5 • *Wifi dongle* Edimax tipe nano • *Mikrotik router board* • *Wireless Access Point*
Alat Pengujian	• *Processor* Intel *core* i3 CPU • 2GB RAM, >256 MB *Hard Drive*

3.2.3 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam perancangan sistem monitoring *traffic* pada jaringan *wireless* menggunakan sistem embedded dapat dilihat pada Tabel 3.2.

Tabel 3.2 Spesifikasi Kebutuhan Perangkat Lunak

Raspberry Pi	• Operasi Sistem : *Raspbian Wheezy* (*Linux*) • Bahasa Pemrograman : *Python*
Router Board	• *Software* pendukung: *Winbox*
HP G42	• Operasi Sistem : *Windows*7 32-*bit* • Bahasa Pemrograman : *Python* • *Software* pendukung : *Connectify*, *Putty*, *Win SCP* • *Tools* pengujian : *The Dude*, *Advance IP Scanner*

Penjelasan Tabel 3.2 adalah sebagai berikut:

a. Sistem Operasi yang digunakan pada *Raspberry Pi* adalah *Raspbian Wheezy*, sistem operasi turunan dari *Linux*.
b. Bahasa pemrograman yang digunakan untuk membangun sistem adalah bahasa pemrograman *Python*.
c. *Winbox* digunakan untuk melakukan konfigurasi pada *Mikrotik Router Board*.
d. Sistem Operasi yang digunakan pada *PC* adalah *Windows*7 32-*bit*.
e. *Connectify* adalah *software* yang digunakan pada *PC* dengan operasi sistem *Windows*. Fungsi penggunaan *Connectify* agar *PC* dapat memberikan alamat *IP* pada *Raspberry Pi*.

f. *WinSCP* adalah *software* yang digunakan untuk memudahkan memindah data atau *file* melalui jaringan dari *PC* ke *Raspberry Pi* dan sebaliknya.
g. *Putty* digunakan untuk melakukan konfigurasi pada *Raspberry Pi*.
h. *Tools The Dude* digunakan sebagai perbandingan sistem terhadap hasil uji *traffic*.

Analisis kebutuhan perangkat lunak mencakup analisis terhadap, kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional adalah kebutuhan utama yang harus ada agar sistem monitoring ini dapat terwujud, berikut adalah kebutuhan fungsional yang dimiliki sistem, yaitu:

- Sistem mampu melakukan *monitoring* pada jaringan *wireless*.
- Sistem mampu membaca data informasi di jaringan *wireless*, yaitu jumlah *traffic* yang melintas dengan memanfaatkan protokol SNMP.
- Sistem dapat memberikan tanda *timeout* pada *access point* yang layanan SNMP –nya tidak dapat diakses.

Sedangkan kebutuhan non-fungsional adalah suatu kebutuhan yang diperlukan sistem monitoring agar dapat berjalan dengan optimal, antara lain:

- *Availability* : sistem dapat diakses secara *realtime* (24 jam) selama masih di area *hot spot.*
- *Usability* : sistem yang dibuat mudah untuk digunakan serta mudah untuk dikembangkan lebih lanjut.
- *User-Friendly* : membangun sistem yang *friendly*, sehingga memudahkan pengguna untuk mempelajari dan menggunakan.

3.2.4 Perancangan Perangkat Keras dan Perangkat Lunak

Pada perancangan sistem monitoring ini, dibutuhkan beberapa faktor pendukung berupa beberapa perangkat, yakni perangkat keras dan perangkat lunak. Seluruh perangkat lunak dan perangkat keras yang digunakan sudah ditentukan sebelumnya. Pada tahap perancangan sistem, ditentukan MIB yang terdapat pada *agent* SNMP yang akan dianalisa. Kemudian digunakan sistem

operasi *Raspbian Wheezy* dan instalasi *software* pendukung seperti P*ython* untuk bahasa pemrograman. Perancangan sistem monitoring meliputi implementasi program pengambilan nilai SNMP, serta menampilkan hasil nilai atau jumlah *traffic* yang melintas.

- **Perancangan Perangkat Keras**

Perangkat keras diperlukan untuk membangun seluruh komponen yang terdapat pada perancangan sistem monitoring. Adapun perangkat yang dibutuhkan adalah *Raspberry Pi type* B yang akan bertindak sebagai media kinerja sistem yang akan diterapkan. Pemilihan *Raspberry Pi type* B dikarenakan memiliki *memory* RAM yang cukup besar yaitu 512 MB, selain itu adanya kebutuhan untuk terhubung ke jaringan internet, sehingga *Ethernet* serta port *USB* yang terdapat pada *Raspberry Pi type* B dapat digunakan untuk *wifi dongle* maupun kabel LAN.

Perancangan dilakukan pada jaringan WLAN yang terdiri dari *mikrotik router* dan *wireless access point* yang mengaktifkan layanan SNMP. Adanya layanan SNMP pada *access point* ini, akan memudahkan sistem untuk dapat membaca data informasi yang dibutuhkan, yaitu jumlah *traffic* yang melintas pada jaringan *wireless.*

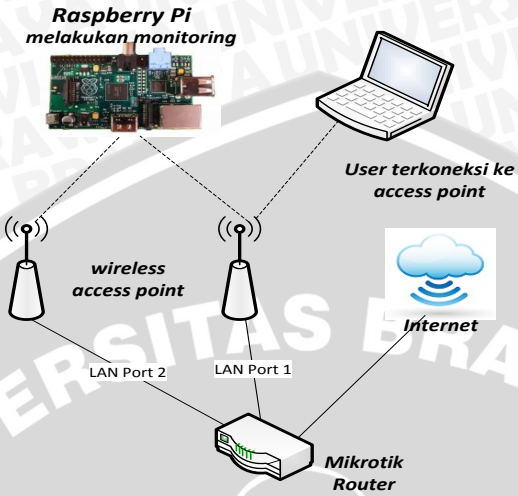
- **Perancangan Perangkat Lunak**

Sistem operasi yang digunakan dalam perancangan sistem monitoring *traffic* adalah *Raspbian Wheezy* yaitu sistem operasi *Linux* yang dikembangkan khusus untuk *Raspberry Pi*. Alasan digunakannya operasi sistem *Raspbian Wheezy* karena mendukung bahasa pemrograman *Python,* dimana seluruh pengimplementasian program menggunakan bahasa pemrograman *Python*. Selain itu, dibutuhkan program *Notepad++* untuk melakukan *editing source code.*

3.2.5 Perancangan Topologi Jaringan

Perancangan topologi jaringan merupakan gambaran bagaimana perangkat sistem embedded bekerja di dalam sistem jaringan *wireless*. Penelitian dilaksanakan pada jaringan W*ireless Local Area Network* (WLAN) yang terdiri dari *wireless access point, mikrotik router*, serta *raspberry pi* yang bertindak sebagai media kinerja sistem SNMP yang terhubung dengan jaringan WLAN.

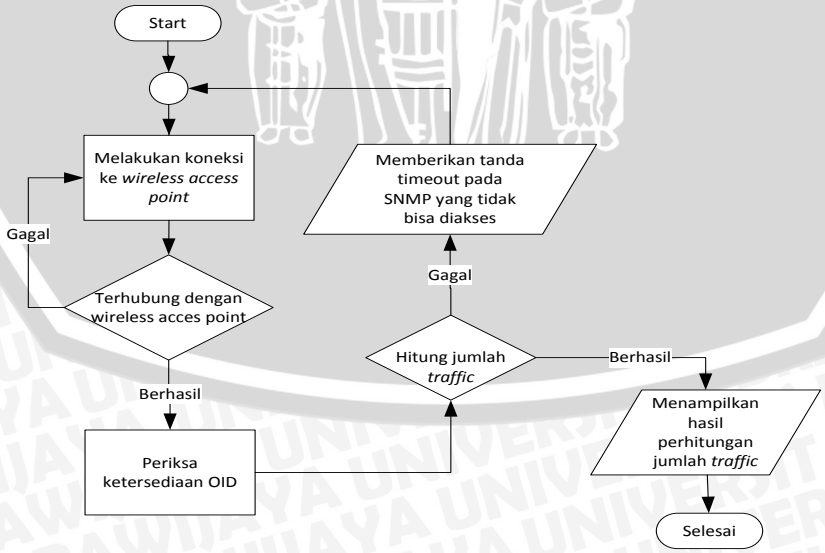
Gambar 3.2 memperlihatkan topologi sistem monitoring *traffic* pada jaringan *wireless* menggunakan *Raspberry Pi*:



Gambar 3.2 Topologi sistem monitoring *traffic*

3.2.6 Perancangan Cara Kerja Program

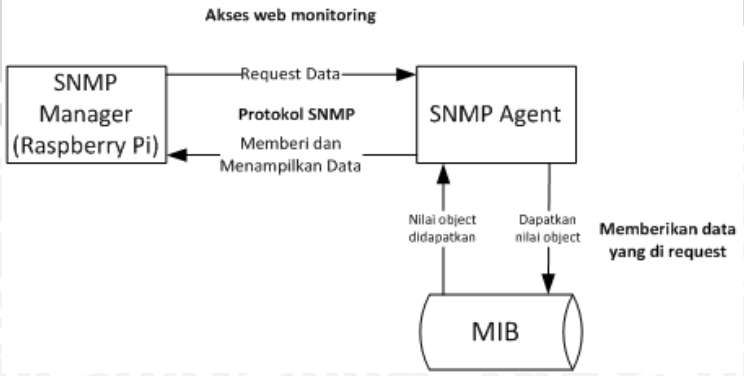
Untuk dapat melakukan proses perhitungan jumlah *traffic* pada jaringan *wireless*, dibutuhkan sebuah program yang ditujukan untuk membaca nilai dari objek *traffic* tersebut. Dimana sistem kerja perhitungan tersebut berdasarkan proses pengiriman paket data keseluruhan yang ada dalam jaringan. Adapun diagram alir dari program yang digunakan seperti pada Gambar 3.3.



Gambar 3.3 Diagram Alir Cara Kerja Program

Penjelasan dari diagram alir cara kerja program pada Gambar 3.3 diuraikan sebagai berikut:

a. Melakukan koneksi ke *access point* yang sudah ditentukan sebelumnya melalui jaringan *wireless*.
b. Pada tahap kedua, yaitu seleksi kondisi. Bila koneksi ke *access point* berhasil, maka proses akan dilanjutkan ke tahap selanjutnya. Namun bila tidak berhasil, maka akan dilakukan koneksi ulang terhadap *access point* yang ada.
c. Setelah sistem dapat terhubung dengan perangkat jaringan yang ada, proses selanjutnya adalah memeriksa ketersediaan OID pada setiap *access point* menggunakan *source code* yang dibangun pada bahasa pemrograman *Python.*
d. Perhitungan jumlah *traffic* yang dilakukan adalah membaca nilai dari objek – objek pada MIB menggunakan *source code* yang dibangun dengan bahasa pemrograman *Python*. Nilai objek yang dibaca adalah data informasi jumlah *traffic* yang melintas dengan memanfaatkan protokol SNMP.
e. Di dalam proses perhitungan jumlah *traffic* juga terdapat seleksi kondisi, apabila perhitungan berhasil maka akan dilanjutkan ke proses selanjutnya dan apabila tidak berhasil maka sistem akan memberikan tanda *timeout* pada *access point* yang layanan SNMP –nya tidak dapat diakses.
f. Proses akhir dari sistem adalah menampilkan hasil dari perhitungan jumlah *traffic*.



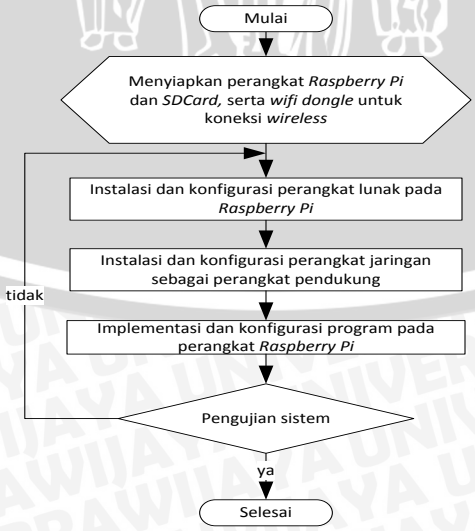
Gambar 3.4 Desain Umum Perancangan Sistem

Pada Gambar 3.4 memperlihatkan desain umum perancangan sistem, dimana *Raspberry Pi* yang melakukan *request* kepada *agent* - *agent* SNMP untuk mendapatkan nilai objek yang terdapat pada MIB. Setelah nilai objek didapatkan, SNMP *agent* akan melakukan *reply* kepada *Raspberry Pi* sehingga dapat ditampilkan data berupa nilai objek yang diminta. Nilai objek atau yang sering disebut dengan OID (*Object Identifier*) SNMP merupakan informasi terstruktur dari database SNMP (MIB). Hasil dari pengambilan nilai objek ini akan diolah untuk diketahui jumlah *traffic* yang melintas melalui protokol SNMP. Hal tersebut penting dilakukan untuk nantinya dapat mengelola dan memanfaatkan sumber daya jaringan secara efektif. Gambar 3.5 adalah perangkat jaringan yang dibutuhkan untuk perancangan sistem

Gambar 3.5 *Raspberry Pi* dan perangkat jaringan

3.3 Implementasi

Implementasi dilakukan berdasarkan perancangan yang telah ditentukan sebelumnya. Proses implementasi dilakukan bertahap agar peneliti dapat melakukan analisa dengan mudah, serta dapat melakukan perbaikan jika terjadi kesalahan pada sistem. Tahapan implementasi pada penelitian ini dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Alir Implementasi Sistem

Tahap awal yang dilakukan pada implementasi sistem adalah menyiapkan perangkat *Raspberry Pi* dan *SDCard* beserta *wifi dongle* sebagai perangkat pendukung dalam perancangan sistem. Kemudian dilakukan instalasi dan konfigurasi perangkat lunak pada *Raspberry Pi* yang terdiri dari instalasi operasi sistem *Raspbian Wheezy*, instalasi bahasa pemrograman *Python*, serta instalasi *library pySNMP* untuk pengaplikasian protokol SNMP. Untuk dapat melakukan akses terhadap *Raspberry Pi* melalui *PC* dengan sistem operasi *Windows*, dibutuhkan beberapa *software* pendukung antara lain adalah melakukan instalasi *connectify*, *win scp*, dan *putty*.

Tahap selanjutnya menyiapkan perangkat jaringan yang nantinya dibutuhkan dalam pengujian sistem, antara lain menyiapkan *mikrotik router board* dan *wireless access point* yang juga harus dikonfigurasi agar sesuai dengan sistem yang dibangun. *Software* penunjang yang dibutuhkan untuk melakukan konfigurasi pada *mikrotik router* adalah *winbox*. Tahap akhir pada perancangan adalah implementasi *python source code* pada *Raspberry Pi*, dan dilakukan pengujian serta analisis sistem. Implementasi sistem akan dijelaskan secara terperinci pada Bab IV.

3.4 Pengujian dan Analisis

Pada tahap ini dilakukan uji coba koneksi pada *access point* melalui jaringan *wireless* dan melakukan analisis sistem. Pengujian koneksi dimaksudkan agar sistem embedded dapat bekerja dengan optimal dan mampu melakukan perhitungan jumlah *traffic* pada *access point* yang ada. Penelitian ini hanya mampu membaca nilai objek berupa data informasi, yaitu jumlah *traffic* yang melintas pada jaringan *wireless* dengan memanfaatkan protokol SNMP. Sistem yang dibangun dapat diakses lebih mudah dan cepat yaitu melalui piranti yang bersumber daya rendah dan murah, dimana dengan tujuan untuk memudahkan *administrator* jaringan dalam memantau kondisi *access point*. Sistem ini memanfaatkan MIB (Management Information Base) untuk mendapatkan kondisi teraktual dari *access point* yang dipantau. Apabila terjadi suatu kondisi yang mengkhawatirkan, *administrator* akan dapat dengan segera mengatasinya.

Tahap pengujian selanjutnya adalah melakukan perbandingan hasil perhitungan jumlah *traffic yang* didapatkan dari sistem yang dibangun dalam sistem embedded dengan hasil yang didapat dengan *tools the dude*, sehingga dapat menguraikan permasalahan yang timbul dalam melakukan analisis maupun dalam melakukan pengujian. Tahap akhir adalah dilakuan analisis terhadap hasil pengujian sistem yang telah dibangun. Jika terjadi kekurangan dan kesalahan pada sistem embedded untuk perhitungan jumlah *traffic*, maka diperlukan analisis yang menyeluruh sehingga mampu memberikan solusi terhadap kesalahan pada sistem. Tahap pengujian dan analisis secara terperinci akan dijelaskan pada Bab V.

3.5 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah perancangan, implementasi, pengujian dan analisis selesai. Kesimpulan disusun berdasarkan pada hasil pengujian dan analisis terhadap sistem yang telah dibangun. Isi pada kesimpulan diharapkan dapat menjadi acuan untuk pengembangan dan penyempurnaan sistem. Pada akhir penulisan terdapat saran yang bertujuan untuk memperbaiki kesalahan dan melakukan penyempurnaan terhadap sistem yang telah dibuat.

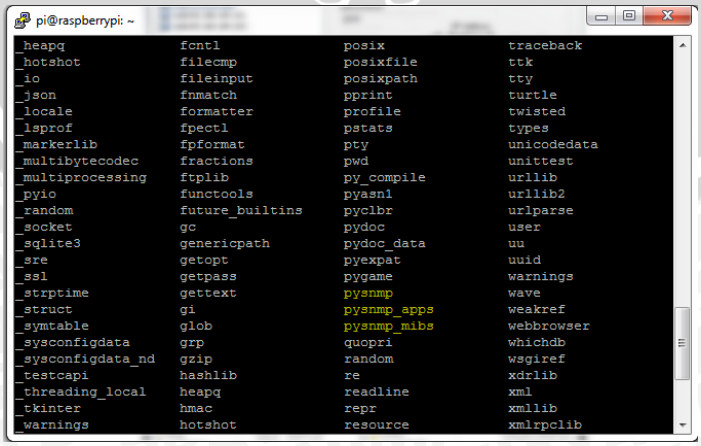
BAB IV
IMPLEMENTASI

Pada bab implementasi dibahas mengenai langkah – langkah dalam implementasi dan konfigurasi terhadap sistem monitoring *traffic* menggunakan sistem embedded. Langkah – langkah dalam implementasi mengacu pada perancangan yang telah dijelaskan pada bab sebelumnya. Implementasi meliputi instalasi dan konfigurasi dalam *Raspberry Pi.*

4.1 Instalasi dan Konfigurasi *Raspberry Pi*

Raspberry Pi digunakan sebagai perangkat keras dalam melakukan perhitungan jumlah *traffic* pada jaringan *wireless.* Sistem operasi yang digunakan adalah *Raspbian Wheezy*. Penggunaan sistem operasi *Raspbian Wheezy* dimaksudkan agar dapat dilakukan konfigurasi terhadap *Raspberry Pi* yang ada sehingga menjadi sebuah sistem embedded yang dapat melakukan perhitungan *traffic.* Selain itu dilakukan instalasi bahasa pemrograman *Python* beserta instalasi *library pysnmp* untuk pengaplikasian protokol SNMP.

Sebelumnya lakukan instalasi bahasa pemrograman *python*, dimana bahasa pemrograman tersebut digunakan dalam seluruh implementasi program. Kemudian, lakukan instalasi *library pysnmp*. *Pysnmp* merupakan SNMP v1/ v2/ v3 *engine* dan aplikasi yang murni ditulis dalam *python*, mendukung *Manager*/ *Agent*/ *Proxyroles*, *scriptable* MIBs, operasi *asynchronous* dan beberapa transportasi spesifikasi ASN.1.



Gambar 4.1 *Library pysnmp* sudah terinstal dalam *raspberry pi*

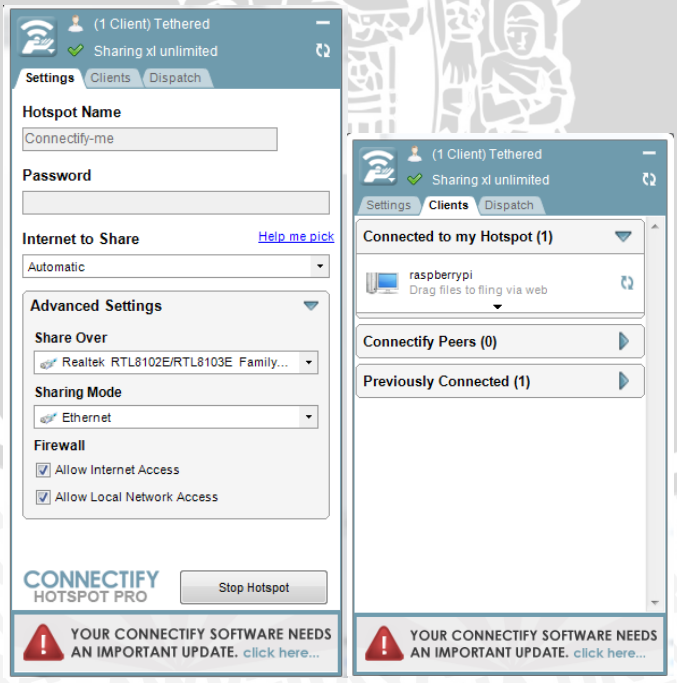
Gambar 4.1 diatas menunjukkan bahwa, *library pysnmp* yang dibutuhkan oleh bahasa pemrograman *python* untuk melakukan perhitungan jumlah *traffic* pada jaringan *wireless* melalui protokol SNMP telah berhasil terinstall dalam perangkat *Raspberry Pi*. Untuk dapat melakukan *monitoring* terhadap *traffic* pada jaringan *wireless*, *Raspberry Pi* menggunakan *IP address* yang dinamis. Hal ini dimaksudkan agar *raspberry pi* dapat digunakan untuk melakukan *monitoring* pada area jaringan yang berbeda.

4.1.1 Instalasi dan Konfigurasi Software Pendukung

Untuk dapat melakukan akses terhadap *Raspberry Pi* melalui *PC* dengan sistem operasi *Windows*, dibutuhkan beberapa *software* pendukung antara lain adalah *connectify*, *win scp*, dan *putty*.

- ***Connectify***

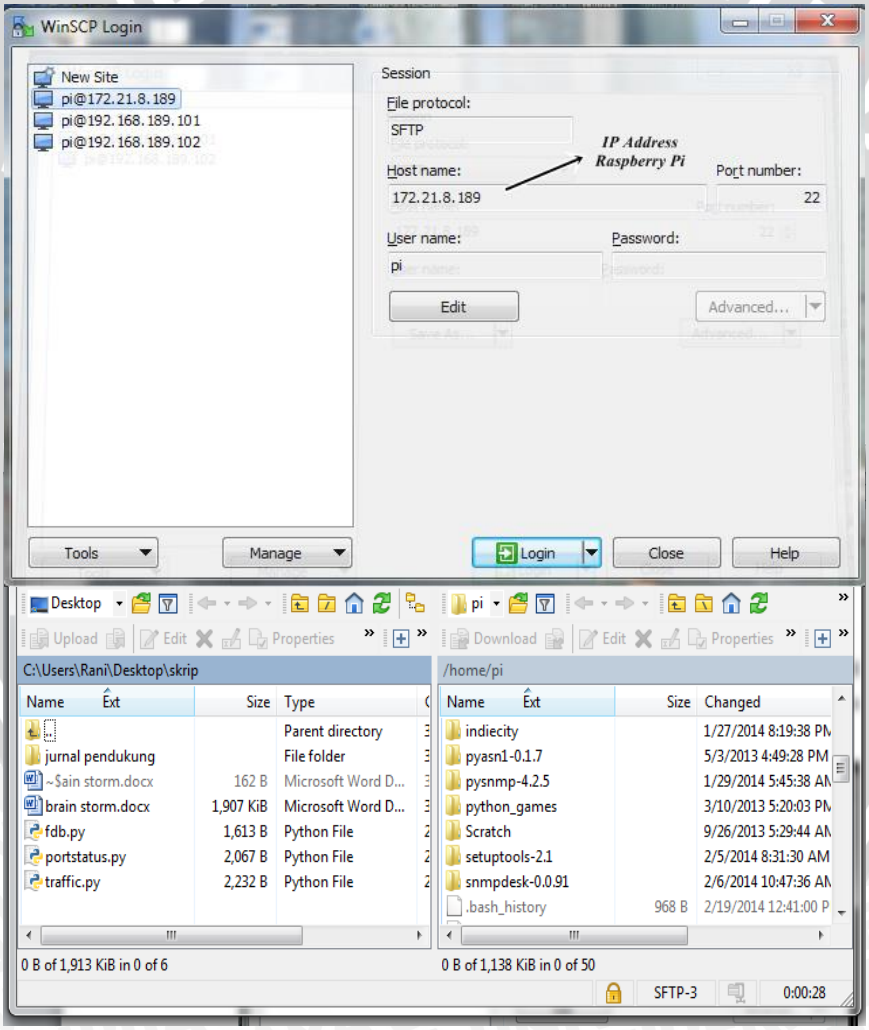
Kegunaan umum dari *connectify* adalah untuk melakukan *sharing* koneksi internet di tempat yang tidak memiliki koneksi internet dengan modal berupa modem hp ataupun *usb stick* atau biasa disebut dengan istilah *hotspot*. *Connectify* merupakan *software* pendukung untuk menghubungkan *Raspberry Pi* dengan server melalui *Ethernet*. Tampilan *connectify* dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Connectify* melakukan *sharing* koneksi internet terhadap *raspberry pi*

- ***WinSCP***

WinSCP adalah aplikasi yang berfungsi untuk transfer file atau copy file antara *Windows* dengan *Linux* (*Raspberry Pi*). Kegunaan dari *WinSCP* adalah sebagai alat atau media untuk *transfer*, atau lebih familiar kita kenal dengan sebutan *upload* dan *download* file melalui protokol ftp dan *secure shell* (SSH). Penggunaan *WinSCP* mempermudah dalam memanajemen *file* yang berada dalam *Raspberry Pi*, yaitu memindahkan *file* dari *PC* ke *Raspberry Pi* dan sebaliknya, serta dapat melakukan editorial seperti mengedit isi file, merubah nama file, menghapus file dan lain sebagainya. Tampilan *WinSCP* dapat dilihat pada Gambar 4.3.

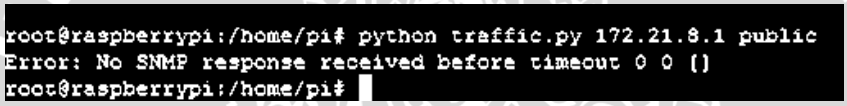


Gambar 4.3 Manajemen file dengan *Win SCP*

- ***Putty***

Putty merupakan *software* yang berjalan pada sistem operasi *Windows*, dimana berfungsi untuk menghubungkan antara *Windows* dengan *Raspberry Pi*. *Software* ini dibutuhkan agar dapat melakukan konfigurasi terhadap perangkat *Raspberry Pi*/ melakukan *remote connection* melalui terminal *shell* (port SSH). SSH adalah protokol jaringan yang memungkinkan pertukaran data melalui saluran aman antara dua perangkat jaringan. Tampilan *putty* saat melakukan perhitungan *traffic* dapat dilihat pada Gambar 4.4 dan Gambar 4.5.

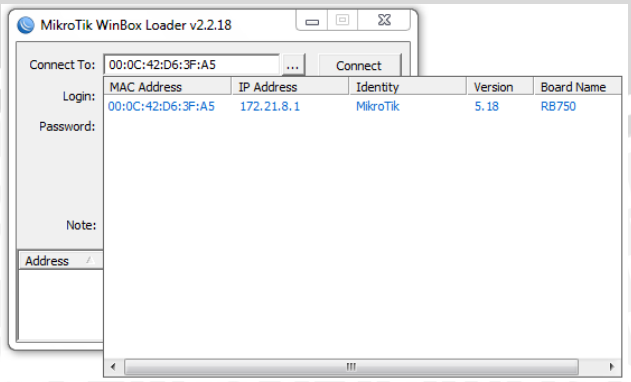
Gambar 4.4 *Putty* saat menampilkan hasil hitung *traffic*



Gambar 4.5 *Putty* menampilkan tanda *timeout* saat layanan SNMP tidak tersedia/ tidak dapat diakses

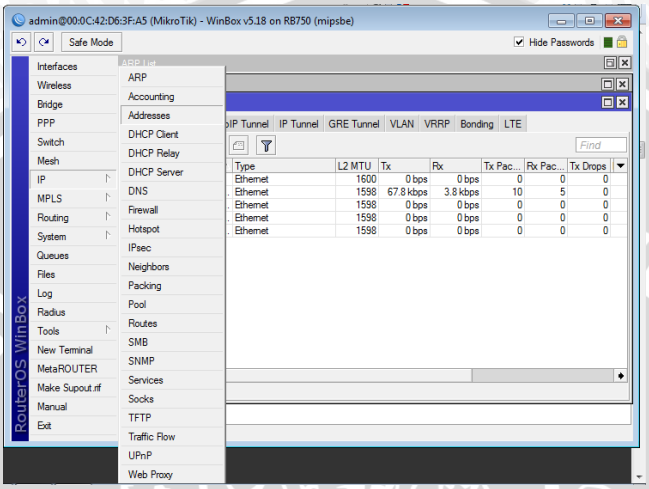
4.2 Konfigurasi *Mikrotik Router*

Untuk dapat melakukan *remote* ke server *mikrotik router*, digunakan sebuah *utility* yang beroperasi dalam mode GUI yaitu *Winbox*. Melakukan konfigurasi *mikrotik router* melalui *winbox* ini lebih banyak digunakan karena selain penggunaannya yang mudah, juga tidak harus menghafalkan perintah – perintah *console*. Fungsi utama dari penggunaan *winbox* adalah untuk melakukan *setting* atau mengatur *mikrotik router* dengan GUI atau tampilan *desktop*. Gambar 4.6 dibawah adalah tampilan *winbox*.



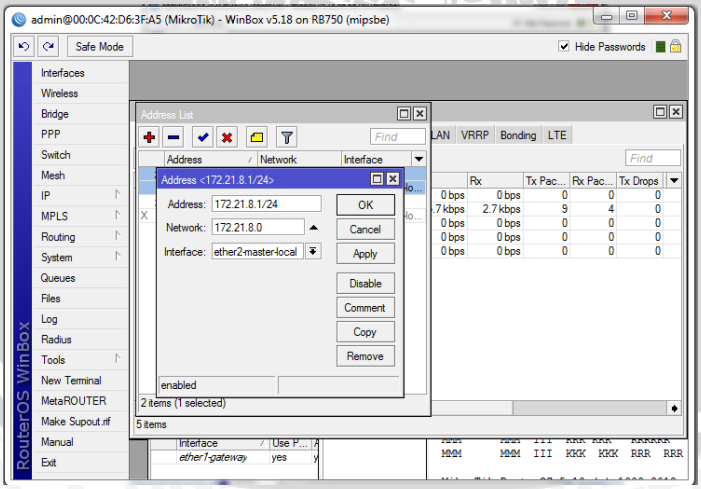
Gambar 4.6 Melakukan koneksi ke *mikrotik router* melalui *winbox*

Untuk dapat melakukan konfigurasi terhadap *mikrotik router*, hal pertama yang dilakukan adalah melakukan koneksi melalui *winbox* menggunakan MAC *address* (00:0C:42:D6:3F:A5) seperti terlihat pada Gambar 4.6. Konfigurasi *IP address* pada *mikrotik router* dilakukan dengan cara klik menu *IP* kemudian pilih *Addressing* seperti terlihat pada Gambar 4.7 dibawah.



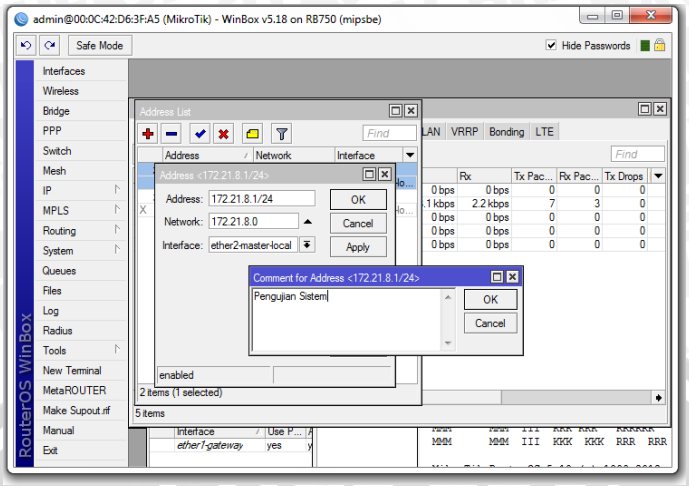
Gambar 4.7 Konfigurasi *IP address* pada *mikrotik router*

Kemudian, *setting IP address* untuk *mikrotik router* dapat segera dilakukan yaitu dengan memberikan *IP address* 172.21.8.1/ 24 pada kolom yang telah tersedia, seperti terlihat pada Gambar 4.8.

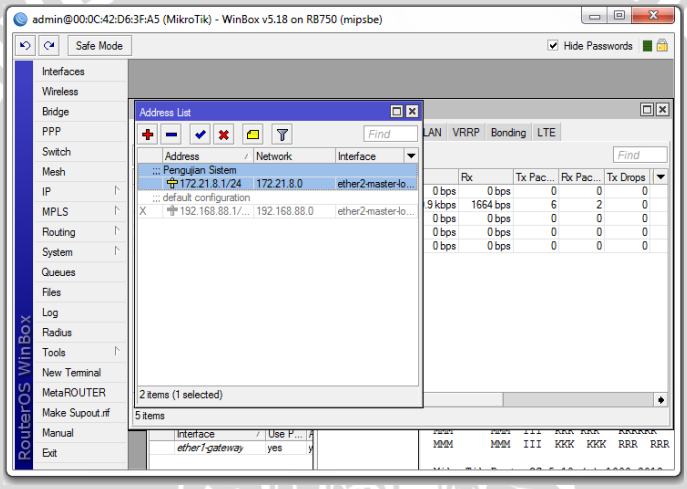


Gambar 4.8 Memberikan *IP address* pada *mikrotik router*

Setelah selesai memberikan *IP address*, lalu klik *apply* hingga muncul perintah "*Comment for Address*", terlihat pada Gambar 4.9. Dan saat *setting IP address* telah berhasil, hasilnya dapat dilihat pada Gambar 4.10.



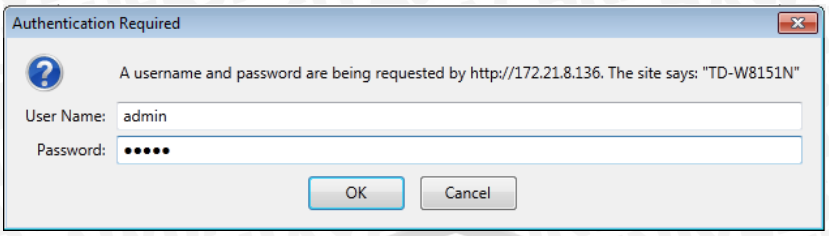
Gambar 4.9 *Apply IP address* pada *router mikrotik*



Gambar 4.10 *IP address* yang berhasil *disetting* pada *router mikrotik*

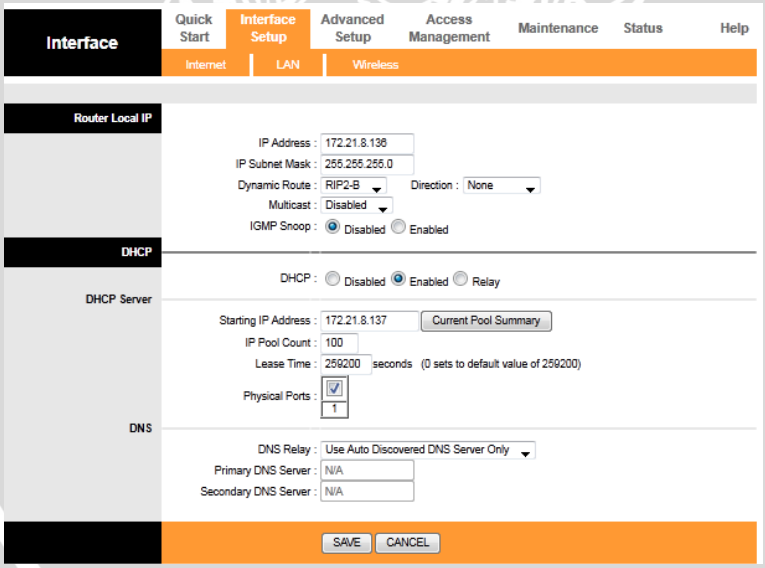
4.3 Konfigurasi *Access Point*

Access Point dalam jaringan komputer adalah sebuah perangkat komunikasi nirkabel yang memungkinkan antar perangkat untuk terhubung ke jaringan nirkabel dengan menggunakan *wifi*, *bluetooth*, atau standar terkait. *Access point* berfungsi sebagai Hub. Switch yang bertindak untuk menghubungkan jaringan lokal dengan jaringan *wireless*/ nirkabel. Dalam *access point* inilah koneksi data/ internet dipancarkan atau dikirim melalui gelombang radio. Ukuran kekuatan sinyal juga mempengaruhi area *coverage* yang akan dijangkau, semakin besar kekuatan sinyal semakin luas jangkauannya (ukuran dalam satuan dBm atau mW).



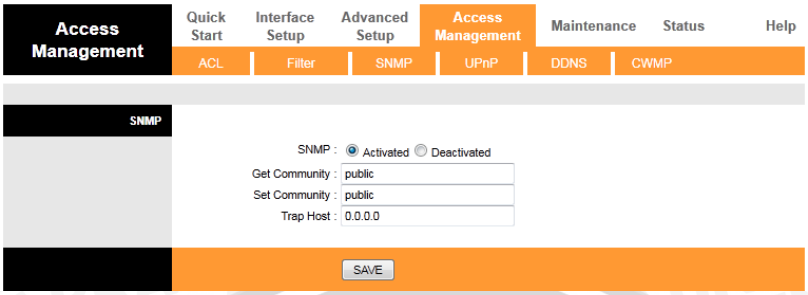
Gambar 4.11 Melakukan koneksi *Access Point*

Gambar 4.11 diatas adalah langkah awal untuk dapat melakukan akses atau koneksi terhadap *access point*, yaitu dengan cara mengakses melalui *web browser* menggunakan *IP address* dari *access point* yang akan dikonfigurasi. Kemudian isikan *username* dan *password* agar dapat *login* ke *access point* tersebut. Setelah itu dapat segera melakukan *setting IP address* pada *access point* melalui menu *Interface Setup*. *IP address* yang diberikan adalah 172.21.8.136 dengan *IP subnet mask* 255.255.255.0, terlihat pada Gambar 4.12. Sedangkan untuk penambahan perangkat *access point*, diberikan *IP address* sesuai dengan jaringan yang telah dibuat.



Gambar 4.12 Melakukan *setting IP address* pada *access point*

Untuk mengaktifkan layanan SNMP, lakukan *setting* melalui menu *Access Management.* Layanan SNMP dapat langsung diaktifkan serta dapat dilakukan *setting* nama *community* jaringan, terlihat pada Gambar 4.13.



Gambar 4.13 Mengaktifkan dan *setting* nama *community* SNMP pada *access point*

4.4 Implementasi *Python Source Code*

Pemrograman bahasa *Python* digunakan dalam seluruh pengimplementasian berkas program perhitungan jumlah *traffic* pada jaringan *wireless*. *Python code source* untuk memeriksa ketersediaan OID dalam MIB SNMP pada *access point* dapat dilihat pada Gambar 4.14.

```
mib = '1.3.6.1.2.1.17.7.1.2.2.1.2' #parameter mib
    value = tuple([int(i) for i in mib.split('.')])

    generator = cmdgen.CommandGenerator()
    comm_data = cmdgen.CommunityData('server', community, 1)
    transport = cmdgen.UdpTransportTarget((ip, 161))

    real_fun = getattr(generator,'nextCmd')
    res     =     (errorIndication,     errorStatus,     errorIndex,
varBindTable)\
    = real_fun(comm_data, transport, value)
    if not errorIndication is None  or errorStatus is True:
           print "Error: %s %s %s" % res
```

.**Gambar 4.14** *Source Code Fbd*

Pada Gambar 4.15, dapat dilihat *python code source* untuk menampilkan nilai objek yaitu jumlah *traffic* yang melintas di jaringan *wireless* dengan memanfaatkan protokol SNMP.

```
generator = cmdgen.CommandGenerator()
    comm_data = cmdgen.CommunityData('server', community, 1)
    transport = cmdgen.UdpTransportTarget((ip, 161))

    real_fun = getattr(generator, 'nextCmd')
    conn = (generator, comm_data, transport, real_fun)
    mibs = [('1.3.6.1.2.1.2.2.1.16', 'out'),
            ('1.3.6.1.2.1.2.2.1.10', 'in'),
            ('1.3.6.1.2.1.2.2.1.11', 'ucast'),
```

```
            ('1.3.6.1.2.1.2.2.1.12', 'nucast'),
            ('1.3.6.1.2.1.31.1.1.1.3', 'bcast'),
            ('1.3.6.1.2.1.2.2.1.14', 'errors')]
    ports = collections.defaultdict(dict)

    for mib in mibs:
        data = datafrommib(mib[0], community, conn)
        for row in data:
            if row:
                ports[row['port']][mib[1]] = int(row['octets'])
            else:
                return None
    return ports
```

Gambar 4.15 *Source Code Traffic*

Keterangan:

a. 1.3.6.1.2.1.2.2.1.16 (*IfOutOctets*)

   Adalah OID untuk menampilkan jumlah *octets* yang ditransmisikan keluar pada *interface*, termasuk *framing* karakter.

b. 1.3.6.1.2.1.2.2.1.10 (*IfInOctets*)

   Adalah OID untuk menampilkan jumlah *octets* yang diterima pada *interface*, termasuk *framing* karakter.

c. 1.3.6.1.2.1.2.2.1.11 (*IfInUcastPkts*)

   Adalah OID untuk menampilkan jumlah paket subnetwork-unicast.

d. 1.3.6.1.2.1.2.2.1.12 (*IfInNucastPkts*)

   Adalah OID untuk menampilkan jumlah paket non-unicast (subnetwork-broadcast/ multicast).

e. 1.3.6.1.2.1.31.1.1.1.3 (*IfInBroadcastPkts*)

   Adalah OID untuk menampilkan jumlah paket yang ditujukan kepada *broadcast*.

f. 1.3.6.1.2.1.2.2.1.14 (*IfInErrors*)

   Adalah OID untuk menampilakan jumlah dari semua kesalahan input pada *interface.*

BAB V
PENGUJIAN DAN ANALISIS

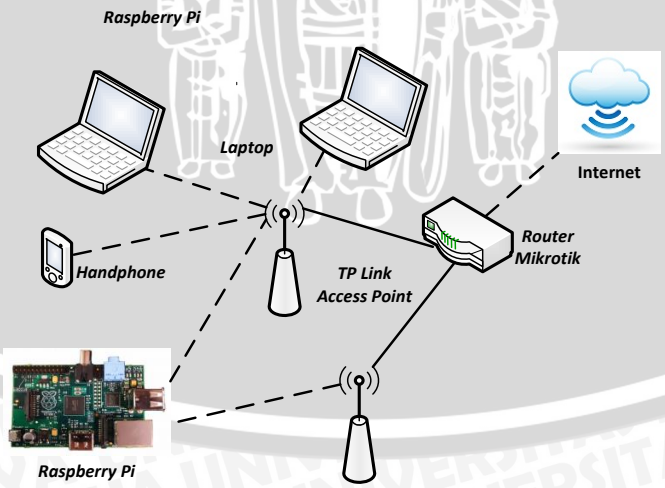
Pada bab ini dilakukan proses pengujian dan analisis terhadap sistem yang telah dibuat, yaitu analisis perhitungan jumlah *traffic* pada jaringan *wireless* menggunakan *Raspberry Pi.* Tahap pengujian sistem dilakukan menggunakan cara simulasi, yaitu membuat jaringan WLAN yang memanfaatkan *mikrotik router board* dan *wireless access point.*

5.1 Pengujian

Proses pengujian dilakukan melalui dua tahap, yaitu pengujian konektifitas dan pengujian perhitungan *traffic* pada jaringan *wireless*.

5.1.1 Topologi Pengujian

Dalam tahap pengujian sistem, dibuat sebuah skenario pengujian dengan melakukan akses ke internet oleh pengguna untuk kemudian diketahui jumlah *traffic* yang melintas pada *access point* melalui jaringan *wireless*. Gambar 5.1 dibawah menjelaskan topologi simulasi pengujian terhadap sistem yang telah dibuat.



Gambar 5.1 Topologi pengujian terhadap sistem

Penjelasan:

- Garis putus – putus merupakan koneksi via *wireless* dari perangkat ke *access point.*
- Garis lurus merupakan koneksi via kabel.

Langkah Pengujian:

Pengujian yang dilakukan untuk melihat *traffic* melalui perangkat laptop, *access point,* atau *router*/ *gateway* yang ada adalah sebagai berikut:

1. Laptop sebagai *user* yang mengakses *web, email,* dan jaringan internet sebaiknya mengaktifkan *service* SNMP terlebih dahulu. Sehingga, nilai dari SNMP dapat diambil oleh perangkat *Raspberry Pi.*
2. *Router*/ *gateway* sebagai penerima paket dan jalur keluar menuju internet, sehingga dapat mengetahui status dari *traffic* yang keluar dari jaringan lokal ke internet.
3. *Wireless Access Point* sebagai jalur distribusi dari *user* (laptop dan *handphone*) ke internet, sehingga dapat dilakukan perhitungan jumlah *traffic* yang melintas melalui perangkat *Raspberry Pi*. Hal ini dapat dilakukan bila mengetahui *username* dan *password* dari *access point* yang ada, serta SNMP yang ada pada *access point* telah diaktifkan dan dilakukan setting *community* yang diinginkan.

Proses pengujian dilakukan dengan melakukan koneksi terhadap *access point* melalui jaringan *wireless* dan melakukan perhitungan jumlah *traffic* yang melintas kemudian dilakukan analisis.

5.1.2 Pengujian Koneksi Perangkat Jaringan

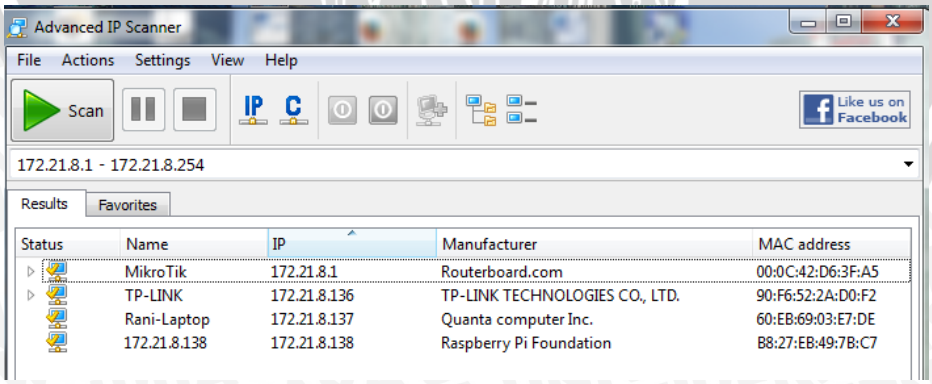
Pengujian koneksi dilakukan untuk memastikan bahwa perangkat sistem embedded yang dibangun sudah terhubung dalam jaringan internal, sehingga nantinya dapat melakukan *monitoring* terhadap *access point* yang dikehendaki. Pengujian koneksi dilakukan dengan cara simulasi yaitu *scanning IP address* perangkat jaringan berupa *access point* dan laptop serta *handphone* sebagai *user*

yang dihubungkan menggunakan *mikrotik router*. Pada Tabel 5.1, dijelaskan mekanisme pengujian koneksi perangkat jaringan.

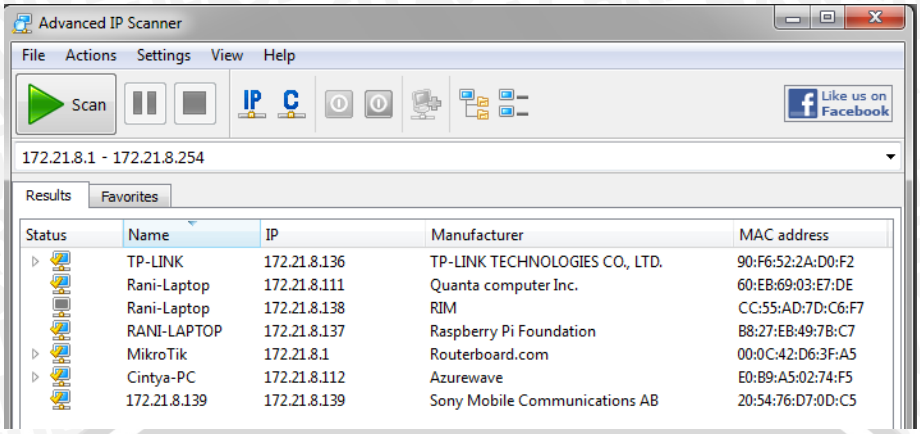
Tabel 5.1 Mekanisme pengujian koneksi perangkat jaringan

Nama Pengujian	Pengujian koneksi perangkat jaringan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa perangkat *raspberry pi* dapat terhubung dengan perangkat jaringan yang akan diuji.
Mekanisme Pengujian	1. Penguji menjalankan aplikasi *Advance IP scanner*. 2. Masukkan *IP address* perangkat jaringan yang akan diuji (172.21.8.1 – 172.21.8.254). 3. Menghitung jumlah perangkat jaringan yang terdeteksi oleh *IP scanner*.
Hasil yang diharapkan	*Raspberry pi* dapat melakukan koneksi terhadap perangkat jaringan berupa *access point*, serta 2 laptop dan *handphone* sebagai *user*.

Pengujian koneksi terhadap perangkat jaringan berupa *access point*, laptop dan *handphone* dilakukan dengan *scanning IP address*. *IP address* 172.21.8.1 pada *mikrotik router* sebagai *gateway*, dan *IP Address* 172.21.8.136 pada *access point*. Hasil dari *scanning* dapat dilihat pada Gambar 5.2 dan Gambar 5.3.



Gambar 5.2 *Scanning IP address* perangkat jaringan yang digunakan dengan 1 *user*



Gambar 5.3 *Scanning IP address* perangkat jaringan yang digunakan dengan 3 *user*

Dari pengujian koneksi yang telah dilakukan, didapatkan bahwa perangkat jaringan yang digunakan dalam simulasi pengujian dapat terhubung ke *raspberry pi* dan *user* dengan baik, seperti terlihat pada Gambar 5.2 (1 *user* menggunakan laptop) dan Gambar 5.3 (2 *user* menggunakan laptop, 1 *user* menggunakan *handphone*). Hasil pengujian koneksi merupakan hasil yang didapatkan dari mekanisme pengujian yang dijelaskan sebelumnya. Hasil pengujian koneksi dapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil pengujian koneksi perangkat jaringan

Hasil yang diharapkan	*Raspberry pi* dapat melakukan koneksi terhadap perangkat jaringan yang akan diuji berupa *access point* dan beberapa *device* berupa laptop serta *handphone*
Hasil yang didapatkan	*Raspberry pi* dapat terhubung dengan perangkat jaringan yang akan diuji

5.1.3 Pengujian Perhitungan *Traffic*

Pengujian perhitungan *traffic* dilakukan untuk mengetahui bahwa sistem yang dibangun telah mampu memberikan hasil perhitungan jumlah *traffic*. Pengujian perhitungan jumlah *traffic* terdiri dari empat mekanisme, yaitu pengujian 1: saat *access point* tidak terkoneksi ke internet, pengujian 2: saat *access point* sudah terkoneksi ke internet, pengujian 3a: saat 1 *user* melakukan *browsing*, pengujian 3b: saat 3 *user* melakukan *browsing*, pengujian 4: saat 1 *user*

melakukan *download* dan pengujian 4b: saat 3 *user* melakukan *download*. Mekanisme pengujian dapat dilihat pada Tabel 5.3 dibawah ini:

Tabel 5.3 Mekanisme pengujian hitung jumlah *traffic*

Nama Pengujian	Pengujian hitung jumlah *traffic*
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan perhitungan jumlah *traffic* terhadap *access point*.
Mekanisme Pengujian 1 (Saat *access point* tidak terkoneksi ke internet)	1. Penguji menyiapkan perangkat jaringan berupa *mikrotik router board*, *wireless access* point, *pc*, dan *raspberry pi*. 2. Penguji melakukan *scanning IP address* untuk memastikan *raspberry pi* dapat terhubung dengan perangkat jaringan yang akan diuji. 3. Penguji menjalankan sistem pada *raspberry pi*. 4. Penguji memeriksa ketersediaan OID pada MIB SNMP dengan menjalankan program *fdb.py*. 5. Penguji melakukan perhitungan jumlah *traffic* menggunakan *raspberry pi* dengan menjalankan program *traffic.py*, menjalankan penghitung jumlah *traffic* pada *the dude* dan *mikrotik router*. 6. Membandingkan hasil pengujian jumlah *traffic* yang didapat dari *raspberry pi*, *the dude*, dan *mikrotik router*.
Mekanisme Pengujian 2 (Saat *access point* sudah terkoneksi ke internet)	1. Penguji menyiapkan perangkat jaringan berupa *mikrotik router board*, *pc*, *raspberry pi* dan 2 *wireless access point*. 2. Penguji melakukan *scanning IP address* untuk memastikan *raspberry pi* dapat terhubung dengan perangkat jaringan yang akan diuji. 3. Penguji melakukan koneksi ke internet melalui *access point* yang tersedia.

	4. Penguji menjalankan sistem pada *raspberry pi*. 5. Penguji memeriksa ketersediaan OID pada MIB SNMP dengan menjalankan program *fdb.py*. 6. Penguji melakukan perhitungan jumlah *traffic* menggunakan *raspberry pi* dengan menjalankan program *traffic.py*, menjalankan penghitung jumlah *traffic* pada *the dude* dan *mikrotik router*. 7. Membandingkan hasil pengujian jumlah *traffic* yang didapat dari *raspberry pi*, *the dude*, dan *mikrotik router*.
Mekanisme Pengujian 3 a (Saat 1 *user* melakukan *browsing*)	8. Penguji melakukan *browsing* melalui 1 *device* berupa laptop yang terhubung ke *access point*, kemudian melakukan perhitungan jumlah *traffic* menggunakan *raspberry pi* dengan menjalankan program *traffic.py*, menjalankan penghitung jumlah *traffic* pada *the dude* dan *mikrotik router*. 9. Membandingkan hasil pengujian jumlah *traffic* yang didapat dari *raspberry pi*, *the dude*, dan *mikrotik router*.
Mekanisme Pengujian 3 b (Saat 3 *user* melakukan *browsing*)	10. Penguji melakukan *browsing* melalui beberapa *device* berupa 2 laptop dan 1 *handphone* yang terhubung ke *access point*, kemudian melakukan perhitungan jumlah *traffic* menggunakan *raspberry pi* dengan menjalankan program *traffic.py*, menjalankan penghitung jumlah *traffic* pada *the dude* dan *mikrotik router*. 11. Membandingkan hasil pengujian jumlah *traffic* yang didapat dari *raspberry pi*, *the dude*, dan *mikrotik router* saat banyak *user*.

Mekanisme Pengujian 4 a (Saat 1 *user* melakukan *download*)	12. Penguji melakukan *download* melalui 1 *device* berupa laptop yang terhubung ke *access point*, kemudian melakukan perhitungan jumlah *traffic* menggunakan *raspberry pi* dengan menjalankan program *traffic.py*, menjalankan penghitung jumlah *traffic* pada *the dude* dan *mikrotik router*. 13. Membandingkan hasil pengujian jumlah *traffic* yang didapat dari *raspberry pi*, *the dude*, dan *mikrotik router*.
Mekanisme Pengujian 4 b (Saat 3 *user* melakukan *download*)	14. Penguji melakukan *download* melalui beberapa *device* berupa 2 laptop, dan 1 *handphone* yang terhubung ke *access point*, kemudian melakukan perhitungan jumlah *traffic* menggunakan *raspberry pi* dengan menjalankan program *traffic.py*, menjalankan penghitung jumlah *traffic* pada *the dude* dan *mikrotik router*.
Hasil yang diharapkan	Hasil pengujian *traffic* menggunakan *raspberry pi* memiliki selisih atau perbandingan yang tidak jauh berbeda dengan hasil pengujian dengan *the dude* dan *mikrotik router*. Sehingga pengujian dengan *raspberry pi* bisa dipastikan kevalidannya.

Dari mekanisme pengujian 1 yang telah dilakukan, didapatkan hasil seperti terlihat pada Tabel 5.4.

Tabel 5.4 Hasil pengujian 1 (saat *access point* tidak terkoneksi ke internet)

Waktu Pengujian	Perangkat					
	Mikrotik Router		*Raspberry Pi*		*The Dude*	
	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)
Menit ke-1	0.443	0.809	0.325	0.776	0.302	0.761
Menit ke-2	1.023	0.949	0.936	0.912	0.946	0.913
Menit ke-3	1.259	1.012	1.175	0.977	1.175	0.977
Menit ke-4	1.863	1.185	1.798	1.161	1.812	1.152
Menit ke-5	2.098	1.248	2.018	1.215	2.018	1.215
Menit ke-6	2.563	1.366	2.484	1.349	2.484	1.334
Menit ke-7	2.822	1.427	2.732	1.410	2.732	1.399
Menit ke-8	3.150	1.475	3.095	1.468	3.011	1.445
Menit ke-9	3.368	1.511	3.287	1.487	3.296	1.487
Menit ke-10	3.928	1.570	3.888	1.585	3.844	1.563
Rata – rata	2.251	1.252	2.174	1.234	2.162	1.225

Pada pengujian 1 (saat *access point* tidak terkoneksi internet) dapat diketahui bahwa hasil *monitoring Raspberry Pi* dan *The Dude* mempunyai nilai yang selalu lebih kecil dibanding *Mikrotik Router*. Sedangkan hasil *monitoring Raspberry Pi* dengan *Mikrotik Router* mempunyai nilai selisih yang lebih besar daripada hasil *monitoring Raspberry Pi* dengan *The Dude*, dengan perhitungan sebagai berikut:

Raspberry Pi dengan *Mikrotik Router*

$\overline{Tx_1}$ = 2.251 MB - 2.174 MB = 0.077 MB (0.034)

$\overline{Rx_1}$ = 1.252 MB - 1.234 MB = 0.018 MB (0.014)

Raspberry Pi dengan *The Dude*

$\overline{Tx_1}$ = 2.162 MB - 2.174 MB = - 0.012 MB (- 0.005)

$\overline{Rx_1}$ = 1.225 MB - 1.234MB = - 0.009 MB (- 0.007)

Dari mekanisme pengujian 2, didapatkan hasil seperti terlihat pada Tabel 5.5.

Tabel 5.5 Hasil pengujian 2 (saat *access point* sudah terkoneksi ke internet)

Waktu Pengujian	Perangkat					
	Mikrotik Router		*Raspberry Pi*		*The Dude*	
	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)
Menit ke-1	8.100	2.702	7.799	2.660	8.000	2.677
Menit ke-2	8.900	2.797	8.493	2.735	8.700	2.774
Menit ke-3	10.200	2.910	10.195	2.904	9.800	2.867
Menit ke-4	10.600	2.954	10.502	2.933	10.500	2.927
Menit ke-5	11.100	3.017	10.810	2.967	11.000	2.984
Menit ke-6	11.400	3.060	11.350	3.033	11.300	3.032
Menit ke-7	11.900	3.124	11.777	3.111	11.700	3.099
Menit ke-8	12.400	3.182	12.340	3.157	12.300	3.158
Menit ke-9	12.700	3.219	12.602	3.194	12.600	3.196
Menit ke-10	13.200	3.289	13.123	3.264	13.100	3.264
Rata – rata	11.050	3.025	10.894	2.996	10.900	2.998

Pada pengujian 2 (saat *access point* sudah terkoneksi internet) dapat diketahui bahwa hasil *monitoring Raspberry Pi* dan *The Dude* mempunyai nilai yang selalu lebih kecil dibanding *Mikrotik Router*. Sedangkan hasil *monitoring Raspberry Pi* dengan *Mikrotik Router* mempunyai nilai selisih yang lebih besar daripada hasil *monitoring Raspberry Pi* dengan *The Dude*, dengan perhitungan sebagai berikut:

Raspberry Pi dengan *Mikrotik Router*

$\overline{Tx_2}$ = 11.050 MB - 10.894 MB = 0.156 MB (0.014)

$\overline{Rx_2}$ = 3.025 MB - 2.996 MB = 0.029 MB (0.009)

Raspberry Pi dengan *The Dude*

$\overline{Tx_2}$ = 10.900 – 10.894 MB = 0.006 MB (0.0005)

$\overline{Rx_2}$ = 2.998 – 2.996 MB = 0.002 MB (0.0007)

Mekanisme pengujian 3 terdiri dari dua bagian, dari mekanisme pengujian 3a didapatkan hasil seperti terlihat pada Tabel 5.6.

Tabel 5.6 Hasil pengujian 3a (saat 1 *user* melakukan *browsing*)

Waktu Pengujian	Perangkat					
	Mikrotik Router		*Raspberry Pi*		*The Dude*	
	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)
Menit ke-1	18.100	3.710	17.286	3.628	17.900	3.686
Menit ke-2	19.200	3.831	18.500	3.738	19.000	3.807
Menit ke-3	23.200	4.321	22.926	4.277	23.000	4.296
Menit ke-4	23.900	4.404	23.408	4.341	23.700	4.371
Menit ke-5	24.600	4.491	25.295	4.459	24.500	4.466
Menit ke-6	24.900	4.536	24.774	4.504	24.800	4.514
Menit ke-7	25.600	4.618	25.275	4.569	25.300	4.581
Menit ke-8	26.200	4.687	25.974	4.652	26.000	4.662
Menit ke-9	26.800	4.762	26.475	4.715	26.700	4.739
Menit ke-10	27.600	4.900	27.027	4.777	27.500	4.800
Rata – rata	23.920	4.426	23.694	4.366	23.840	4.392

Pada pengujian 3a (saat 1 *user* melakukan *browsing*) dapat diketahui bahwa hasil *monitoring Raspberry Pi* dan *The Dude* mempunyai nilai yang relatif lebih kecil dibanding *Mikrotik Router*. Hasil *monitoring Raspberry Pi* dengan *Mikrotik Router* mempunyai nilai selisih yang lebih besar daripada hasil *monitoring Raspberry Pi* dengan *The Dude*, dengan perhitungan sebagai berikut:

Raspberry Pi dengan *Mikrotik Router*

$\overline{Tx_{3a}}$ = 23.920 MB - 23.694 MB = 0.226 MB (0.009)

$\overline{Rx_{3a}}$ = 4.426 MB - 4.366 MB = 0.060 MB (0.014)

Raspberry Pi dengan *The Dude*

$\overline{Tx_{3a}}$ = 23.840 – 23.694 MB = 0.146 MB (0.006)

$\overline{Rx_{3a}}$ = 4.392 – 4.366 MB = 0.026 MB (0.006)

Dari mekanisme pengujian 3b, didapatkan hasil seperti terlihat pada Tabel 5.7.

Tabel 5.7 Hasil pengujian 3b (saat 3 *user* melakukan *browsing*)

Waktu Pengujian	Perangkat					
	Mikrotik Router		*Raspberry Pi*		*The Dude*	
	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)
Menit ke-1	29.800	11.500	29.800	11.500	29.700	11.400
Menit ke-2	30.500	11.500	30.400	11.500	30.300	11.400
Menit ke-3	30.900	11.600	30.800	11.600	30.700	11.500
Menit ke-4	31.300	11.600	31.200	11.600	31.100	11.500
Menit ke-5	31.700	11.700	31.600	11.700	31.500	11.600
Menit ke-6	32.200	11.800	32.000	11.700	32.000	11.700
Menit ke-7	32.500	11.800	32.500	11.800	32.300	11.700
Menit ke-8	33.000	11.900	32.900	11.900	32.700	11.800
Menit ke-9	33.900	12.000	33.800	12.000	33.700	11.900
Menit ke-10	34.100	12.000	34.100	12.000	34.000	12.000
Rata – rata	31.990	11.740	31.910	11.720	31.800	11.650

Pada pengujian 3b (saat 3 *user* melakukan *browsing*) dapat diketahui bahwa hasil *monitoring Raspberry Pi* dan *The Dude* mempunyai nilai yang relatif lebih kecil dibanding *Mikrotik Router*. Hasil *monitoring Raspberry Pi* dengan *Mikrotik Router* mempunyai nilai selisih yang lebih kecil daripada hasil *monitoring Raspberry Pi* dengan *The Dude*, dengan perhitungan sebagai berikut:

Raspberry Pi dengan *Mikrotik Router*

$\overline{Tx_{3b}}$ = 31.990 MB – 31.910 MB = 0.080 MB (0.003)

$\overline{Rx_{3b}}$ = 11.740 MB – 11.720 MB = 0.020 MB (0.002)

Raspberry Pi dengan *The Dude*

$\overline{Tx_{3b}}$ = 31.800 – 31.910 = - 0.110 MB (- 0.003)

$\overline{Rx_{3b}}$ = 11.650 – 11.720 MB = - 0.070 MB (- 0.006)

Mekanisme pengujian 4 terdiri dari dua bagian, dari mekanisme pengujian 4a didapatkan hasil seperti terlihat pada Tabel 5.8.

Tabel 5.8 Hasil pengujian 4a (saat 1 *user* melakukan *download*)

Waktu Pengujian	Perangkat					
	Mikrotik Router		*Raspberry Pi*		*The Dude*	
	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)
Menit ke-1	49.700	6.800	48.870	6.671	49.500	6.700
Menit ke-2	52.000	7.000	51.452	6.891	51.800	6.900
Menit ke-3	53.900	7.200	53.283	7.151	53.800	7.200
Menit ke-4	56.400	7.500	55.499	7.367	56.200	7.400
Menit ke-5	94.800	10.500	93.946	10.408	84.300	9.900
Menit ke-6	102.500	11.000	98.706	10.760	102.100	10.900
Menit ke-7	116.900	11.700	116.056	11.588	116.700	11.600
Menit ke-8	137.700	12.900	133.995	12.670	136.700	12.800
Menit ke-9	155.000	13.900	152.832	13.715	152.700	13.600
Menit ke-10	165.300	14.600	164.897	14.511	165.000	14.500
Rata – rata	98.420	10.310	96.954	10.173	96.880	10.150

Pada pengujian 4a (saat 1 *user* melakukan *download*) dapat diketahui bahwa hasil *monitoring Raspberry Pi* dan *The Dude* mempunyai nilai yang selalu lebih kecil dibanding *Mikrotik Router*. Hasil *monitoring Raspberry Pi* dengan *Mikrotik Router* mempunyai nilai selisih yang lebih besar daripada hasil *monitoring Raspberry Pi* dengan *The Dude*, dengan perhitungan sebagai berikut:

Raspberry Pi dengan *Mikrotik Router*

$\overline{Tx_{4a}}$ = 98.420 MB - 96.954 MB = 1.466 MB (0.014)

$\overline{Rx_{4a}}$ = 10.310 MB - 10.173 MB = 0.137 MB (0.013)

Raspberry Pi dengan *The Dude*

$\overline{Tx_{4a}}$ = 96.880 MB - 96.954 MB = - 0.074 MB (- 0.0008)

$\overline{Rx_{4a}}$ = 10.150 MB - 10.173 MB = - 0.023 MB (- 0.002)

Dari mekanisme pengujian 4b, didapatkan hasil seperti terlihat pada Tabel 5.9.

Tabel 5.9 Hasil pengujian 4b (saat 3 *user* melakukan *download*)

Waktu Pengujian	Perangkat					
	Mikrotik Router		*Raspberry Pi*		*The Dude*	
	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)	Tx (MB)	Rx (MB)
Menit ke-1	137.700	19.300	136.700	19.300	135.600	19.200
Menit ke-2	141.000	19.400	140.900	19.400	140.800	19.300
Menit ke-3	141.300	19.500	141.200	19.500	141.100	19.400
Menit ke-4	141.500	19.500	141.400	19.500	141.300	19.400
Menit ke-5	142.500	19.600	141.800	19.600	141.600	19.500
Menit ke-6	142.800	19.800	142.700	19.800	142.600	19.700
Menit ke-7	143.000	19.800	143.000	19.800	142.800	19.700
Menit ke-8	144.300	20.000	143.600	20.000	143.200	19.800
Menit ke-9	149.300	20.200	148.300	20.200	148.200	20.100
Menit ke-10	156.300	20.500	155.400	20.400	155.000	20.400
Rata – rata	143.970	19.760	143.500	19.750	143.220	19.650

Pada pengujian 4b (saat 3 *user* melakukan *download*) dapat diketahui bahwa hasil *monitoring Raspberry Pi* dan *The Dude* mempunyai nilai yang selalu lebih kecil dibanding *Mikrotik Router*. Hasil *monitoring Raspberry Pi* dengan *Mikrotik Router* mempunyai nilai selisih yang lebih besar daripada hasil *monitoring Raspberry Pi* dengan *The Dude*, dengan perhitungan sebagai berikut:

Raspberry Pi dengan *Mikrotik Router*

$\overline{Tx_{4b}}$ = 143.970 MB – 143.500 MB = 0.470 MB (0.003)

$\overline{Rx_{4b}}$ = 19.760 MB – 19.750 MB = 0.010 MB (0.0005)

Raspberry Pi dengan *The Dude*

$\overline{Tx_{4b}}$ = 143.220 MB – 143,500 MB = - 0.280 MB (- 0.002)

$\overline{Rx_{4b}}$ = 19.650 MB – 19.750 MB = - 0.100 MB (- 0.005)

Dari keempat mekanisme pengujian yang telah dilakukan menggunakan *raspberry pi, mikrotik router* dan *the dude*, menunjukkan hasil *monitoring* yang memiliki selisih tidak jauh berbeda. Perbandingan akurasi hasil *monitoring raspberry pi, mikrotik router* dan *the dude*, sebagai berikut:

Akurasi *Raspberry Pi* dengan *Mikrotik Router*

$\overline{Tx}$ Akurasi = $1 - \frac{\overline{Tx1} + \overline{Tx2} + \overline{Tx3a} + \overline{Tx3b} + \overline{Tx4a} + \overline{Tx4b}}{6}$

$= 1 - \frac{0.034 + 0.014 + 0.009 + 0.003 + 0.014 + 0.003}{6}$

$= 1 - 0.013 = 0.9870$

$\overline{Rx}$ Akurasi = $1 - \frac{\overline{Rx1} + \overline{Rx2} + \overline{Rx3a} + \overline{Rx3b} + \overline{Rx4a} + \overline{Rx4b}}{4}$

$= 1 - \frac{0.014 + 0.009 + 0.014 + 0.002 + 0.013 + 0.0005}{6}$

$= 1 - 0.009 = 0.9910$

Akurasi *Raspberry Pi* dengan *The Dude*

$\overline{Tx}$ Akurasi = $1 - \frac{\overline{Tx1} + \overline{Tx2} + \overline{Tx3a} + \overline{Tx3b} + \overline{Tx4a} + \overline{Tx4b}}{6}$

$= 1 - \frac{(-0.005) + 0.0005 + 0.006 + (-0.003) + (-0.0008) + (-0.002)}{6}$

$= 1 - (-0.0007) = 1.0007$

$\overline{Rx}$ Akurasi = $1 - \frac{\overline{Rx1} + \overline{Rx2} + \overline{Rx3a} + \overline{Rx3b} + \overline{Rx4a} + \overline{Rx4b}}{4}$

$= 1 - \frac{(-0.007) + 0.0007 + 0.006 + (-0.006) + (-0.002) + (-0.005)}{6}$

$= 1 - (-0.0022) = 1.0022$

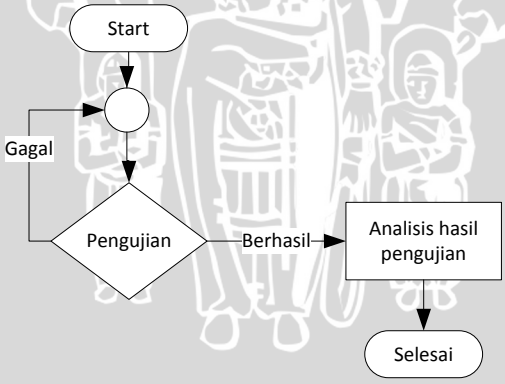
Hasil pengujian perhitungan *traffic* merupakan hasil yang didapatkan dari mekanisme pengujian yang dijelaskan sebelumnya. Hasil pengujian hitung jumlah *traffic* dapat dilihat pada Tabel 5.10.

Tabel 5.10 Hasil pengujian hitung jumlah *traffic*

Hasil yang diharapkan	*Raspberry pi* mampu melakukan hitung jumlah *traffic* dengan selisih nilai yang tidak jauh berbeda dengan hasil pengujian dengan *the dude* dan *mikrotik router*. Sehingga pengujian dengan *raspberry pi* bisa dipastikan kevalidannya
Hasil yang didapatkan	*Raspberry pi* mampu melakukan hitung jumlah *traffic* dengan hasil yang tidak jauh berbeda dengan *the dude* dan *mikrotik router*.

5.2 Analisis

Analisis bertujuan untuk menganalisa data hasil pengujian sehingga didapatkan kesimpulan. Tahapan analisis dilakukan sesuai dengan hasil pengujian terhadap sistem. Mekanisme analisis yang dilakukan antara lain adalah analisis koneksi perangkat jaringan, dan analisis perhitungan jumlah *traffic* yang melintas pada jaringan *wireless*. Tahapan analisis hasil pengujian dapat dilihat pada Gambar 5.4.



Gambar 5.4 Tahapan analisis hasil pengujian

Berdasarkan pengujian koneksi, perangkat sistem embedded dapat terkoneksi ke perangkat jaringan yang akan diuji dengan baik. Hal ini dibuktikan dari hasil *scanning IP address*, terlihat bahwa perangkat *raspberry pi* mendapat *IP address* (172.21.8.138 saat menggunakan 1 *device* dan 172.21.8.137 saat menggunakan 3 *device*) dari *access point* yang memiliki *IP address* 172.21.8.136. Dengan terkoneksinya *raspberry pi* ke *access point* tersebut, perhitungan jumlah

traffic yang melintas di jaringan *wireless* dapat dilakukan dengan memanfaatkan protokol SNMP

Dalam pengujian perhitungan *traffic*, perangkat sistem embedded dapat menghitung jumlah *traffic* melalui empat mekanisme yaitu [1)] pengujian saat *access point* tidak terhubung dengan jaringan internet, [2)] saat *access point* yang ada telah terhubung ke jaringan internet, [3a)] saat 1 *user* melakukan *browsing*, [3b)] saat 3 *user* melakukan *browsing*, [4a)] saat 1 *user* melakukan *download* dan [4b)] saat 3 *user* melakukan *download*. Dari hasil pengujian terhadap *access point* menunjukkan bahwa jumlah *traffic* yang didapatkan melalui *raspberry pi* mempunyai selisih sedikit dibanding jumlah *traffic* yang didapat melalui *mikrotik router* dan *the dude*. Nilai rata – rata yang didapat *raspberry pi* pada pengujian 1: $\overline{\text{Tx}}$ 2.174 MB dan $\overline{\text{Rx}}$ 1.234 MB, pada pengujian 2: $\overline{\text{Tx}}$ 10.894 MB dan $\overline{\text{Rx}}$ 2.996 MB, pada pengujian 3a: $\overline{\text{Tx}}$ 23.694 MB dan $\overline{\text{Rx}}$ 4.366 MB, pada pengujian 3b: $\overline{\text{Tx}}$ 31.910 MB dan $\overline{\text{Rx}}$ 11.720 MB, pada pengujian 4a: $\overline{\text{Tx}}$ 96.954 MB dan $\overline{\text{Rx}}$ 10.173 MB dan pada pengujian 4b: $\overline{\text{Tx}}$ 143.500 MB dan $\overline{\text{Rx}}$ 19.750 MB.

Secara keseluruhan hasil pengujian, perangkat *raspberry pi* dapat berjalan dengan baik sesuai dengan tujuan yang ingin dicapai. Dari hasil pengujian tersebut didapatkan perbandingan akurasi hasil *monitoring raspberry pi* terhadap *mikrotik router* dengan nilai: $\overline{Tx}$ akurasi 0.987 dan $\overline{Rx}$ akurasi 0.991, sedangkan perbandingan akurasi hasil *monitoring raspberry pi* terhadap *the dude* didapatkan hasil: $\overline{Tx}$ akurasi 1.0007 dan $\overline{Rx}$ akurasi 1.0022. Sehingga dapat disimpulkan bahwa hasil uji perhitungan jumlah *traffic* menggunakan *raspberry pi* mempunyai nilai yang relatif lebih kecil daripada menggunakan *mikrotik router* dan mempunyai nilai yang relatif lebih besar daripada menggunakan *the dude*.

BAB VI
PENUTUP

Pada bab ini berisi kesimpulan dari hasil implementasi, pengujian dan analisis sistem serta diberikan saran untuk pengembangan sistem.

6.1 Kesimpulan

Dari hasil penelitian, didapatkan hasil sebagai berikut:

1. Mampu merancang dan membangun sistem monitoring jaringan *wireless* dengan *raspberry pi* yang bekerja dengan baik. Penambahkan *wifi dongle* edimax tipe nano pada perangkat *raspberry pi* agar mampu menangkap jaringan *wireless*. Dalam implementasi seluruh berkas program digunakan *python*, dengan *library pysnmp* untuk pengaplikasian protokol SNMP.
2. Sistem mampu membaca jumlah *traffic* pada jaringan *wireless*, hal ini dibuktikan dari hasil pengujian sistem terhadap *access point.* Hasil *monitoring raspberry pi* tidak jauh berbeda dengan hasil *monitoring the dude* dan *mikrotik router*. Nilai rata – rata yang didapat *raspberry pi* pada pengujian 1: $\overline{\text{Tx}}$ 2.174 MB dan $\overline{\text{Rx}}$ 1.234 MB, pada pengujian 2: $\overline{\text{Tx}}$ 10.894 MB dan $\overline{\text{Rx}}$ 2.996 MB, pada pengujian 3a: $\overline{\text{Tx}}$ 23.694 MB dan $\overline{\text{Rx}}$ 4.366 MB, pada pengujian 3b: $\overline{\text{Tx}}$ 31.910 MB dan $\overline{\text{Rx}}$ 11.720 MB, pada pengujian 4a: $\overline{\text{Tx}}$ 96.954 MB dan $\overline{\text{Rx}}$ 10.173 MB dan pada pengujian 4b: $\overline{\text{Tx}}$ 143.500 MB dan $\overline{\text{Rx}}$ 19.750 MB.
3. Hasil kinerja dari perangkat *raspberry pi* menunjukkan bahwa *raspberry pi* mampu menangkap jaringan *wireless* dan terkoneksi dengan baik, hal ini terlihat dari hasil pengujian yang dilakukan oleh *raspberry pi*. Hasil uji perhitungan jumlah *traffic* menggunakan *raspberry pi* mempunyai nilai yang relatif lebih kecil daripada menggunakan *mikrotik router* dan mempunyai nilai yang relatif lebih besar daripada menggunakan *the dude*. Perbandingan akurasi hasil *monitoring raspberry pi* terhadap *mikrotik router* dengan nilai: $\overline{Tx}$ akurasi 0.987 dan $\overline{Rx}$ akurasi 0.991, sedangkan perbandingan akurasi hasil *monitoring raspberry pi* terhadap *the dude* didapatkan hasil: $\overline{Tx}$ akurasi 1.0007 dan $\overline{Rx}$ akurasi 1.0022.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem, antara lain:

1. Untuk penelitian lebih lanjut disarankan sistem monitoring *traffic* ini dapat dikembangkan dengan meneliti isi dari paket jaringan dan memaparkannya.
2. Untuk penelitian lebih lanjut disarankan membangun sistem basis data untuk menyimpan nilai – nilai objek SNMP sehingga performa dari perangkat jaringan yang akan dimonitoring dapat dimaintenance dalam jangka waktu tertentu.

DAFTAR PUSTAKA

[WIL-99] Stallings, William. 1999. *Data and Computer Communications*. Prentice Hall PTR. Page 45

[KEV-03] Miller, Kevin. 2003. *Deploying IP Anycast*. Cornegie Mellon Network Group.

[HAR-04] Li, Harry Ph.D., Guangjing Chen. 2004. *Wireless LAN Network Management System*. San Jose. Computer Engineering Department, College of Engineering, San Jose State University.

[NIK-07] Kakanakov, Nikolay., Elena Kostadinova. 2007. *Using SNMP for Remote Measurement and Automation*. Bulgaria. Department of Computer Systems and Technologies, Technical University of Sofia.

[MOS-09] Rahman, M.D Mostafijur. 2009.*Network Traffic Monitoring System Based On Embedded Linux And Single Board Computer*. Malaysia. School Of Computer And Communication Engineering, Universiti Malaysia Perlis.

[SYU-09] Ab-Rahman, Mohamad Syuhaimi., Mohamad Najib., Kasmiran Jumari. 2009. *Optical Switch Controller Using Embedded Internet Based System*. Bangi, Selangor, Malaysia.Faculty of Engineering and Built Environtment, Universiti Kebangsaan Malaysia.

[MIS-10] Schwartz, Mischa. 2010. *History Of Communication*.IEEE Communications Magazine.

[LIN-10] F, Lintang Dwi., Brilliant A., Dianadewi R., Sandra Y. 2010.*Pengembangan Sistem Kendali Waktu Nyata dengan Embedded System berbasis Embedded Linux*. Bandung, Jawa Barat. Pusat Penelitian Informatika, Lembaga Ilmu Pengetahuan Indonesia.

[IBN-10] K, Ibnu Febry., Wahyu S S.Kom.,M.Kom., Bagus J.S S.Kom.,M.Kom. 2010. *Implementasi Pemantauan Simpul- Simpul Jaringan Dengan Protokol SNMP Menggunakan Python – Fuse*. Surabaya. Institut Teknologi Sepuluh Nopember (ITS).

[RUD-11] Hartono, Rudi, S.Si., Agus Purnomo, S.Si. 2011. *Wireless Network 802.11*. Surabaya. TI, FMIPA, Universitas Negeri Surabaya.

[ARY-11] Shiddiqi, Ary M., Andhika Panji N. 2011. *Sistem Monitoring Jaringan Dengan Protokol SNMP Menggunakan Piranti Bergerak.*Surabaya. Institut Teknologi Sepuluh Nopember (ITS).

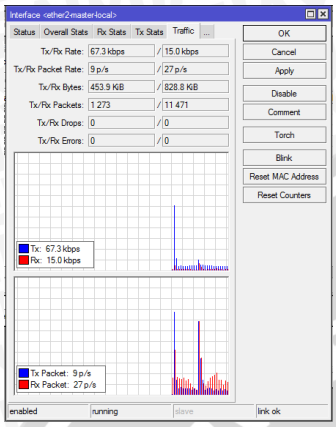
[REZ-13] Pradikta, Reza., Achmad Affandi, Eko Setijadi. 2013. *Rancang Bangun Aplikasi Monitoring Jaringan dengan Menggunakan Simple Network Management Protocol.* Surabaya. Institut Teknologi Sepuluh Nopember (ITS).

[MAL-13] Hakim, Malik Abdillah,. Yeffry Handoko,.2013. *Pemanfaatan Mini PC Raspberry Pi Sebagai Pengontrol Jarak Jauh Berbasis Web Pada Rumah.* Bandung, Universitas Komputer Indonesia.

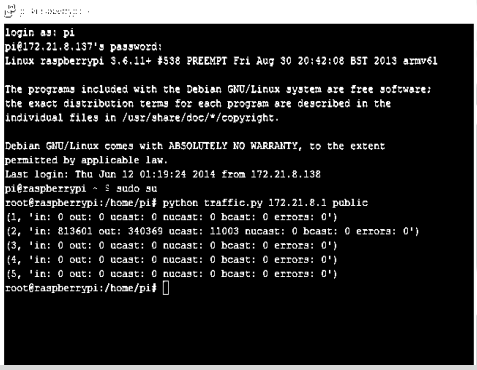
[ANG-13] Adhitama, Angie P. 2013. *Implementasi dan Analisis QOS Wifi Menggunakan Embedded System*. Malang. PTIIK, Universitas Brawijaya.

Lampiran 1. PENGUJIAN 1 (saat *access point* tidak terkoneksi ke internet)

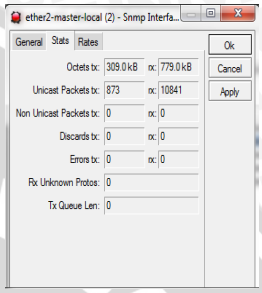
Menit ke-1



Mikrotik router

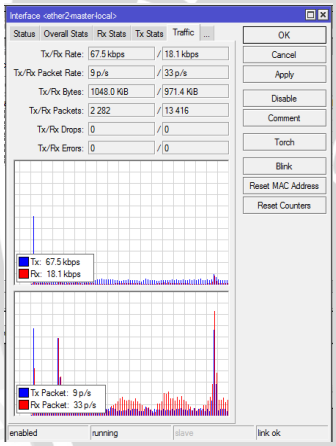


Raspberry pi

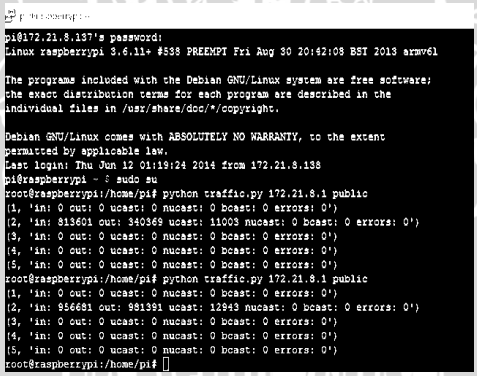


The dude

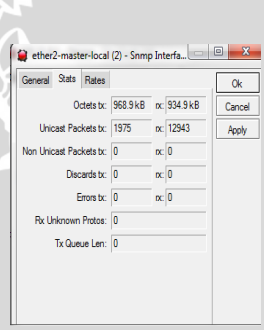
Menit ke-2



Mikrotik router

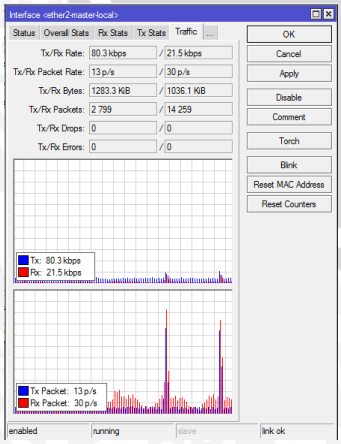


Raspberry pi

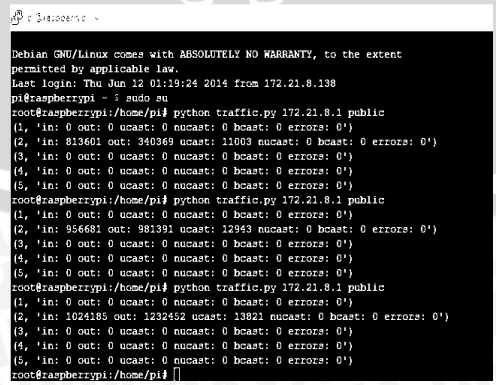


The dude

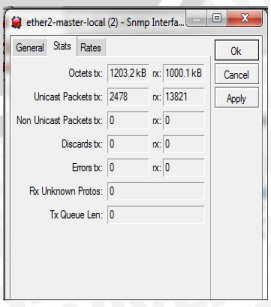
Menit ke-3



Mikrotik router

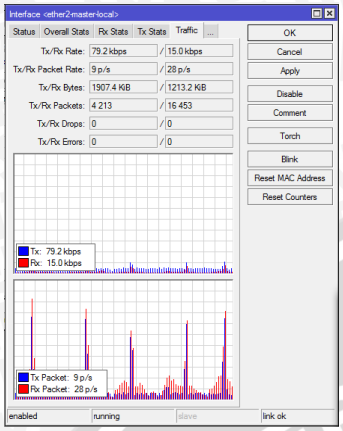


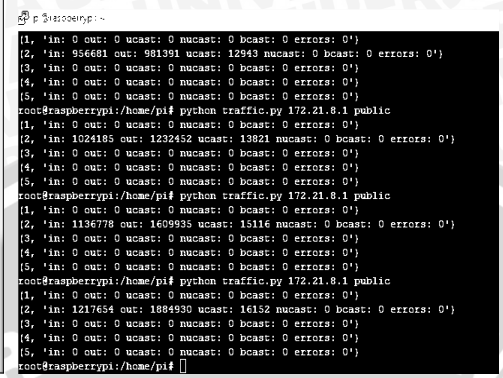
Raspberry pi

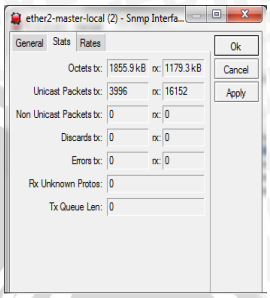


The dude

Menit ke-4

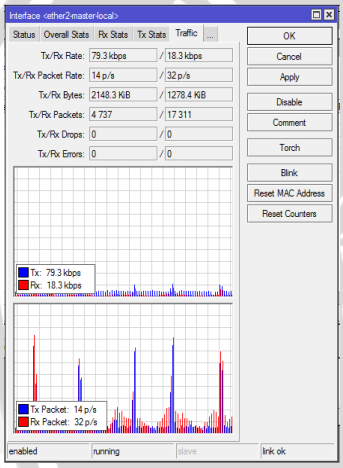


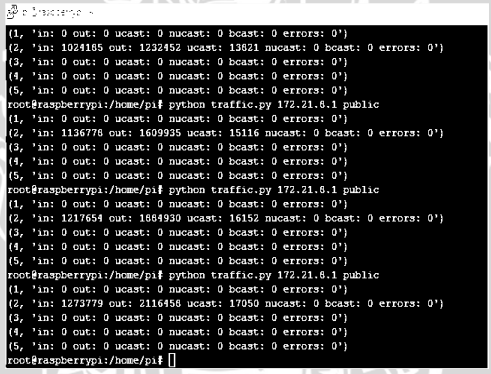


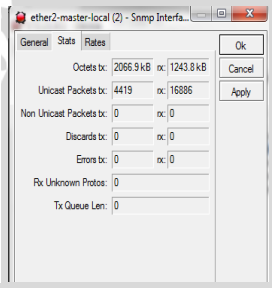


Mikrotik router *Raspberry pi* *The dude*

Menit ke-5

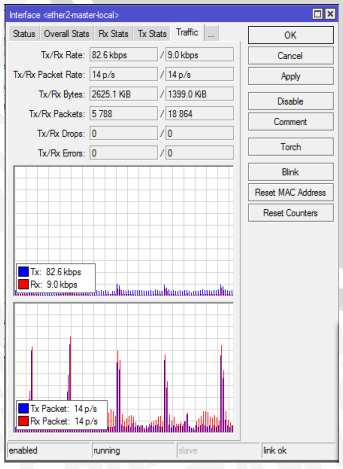


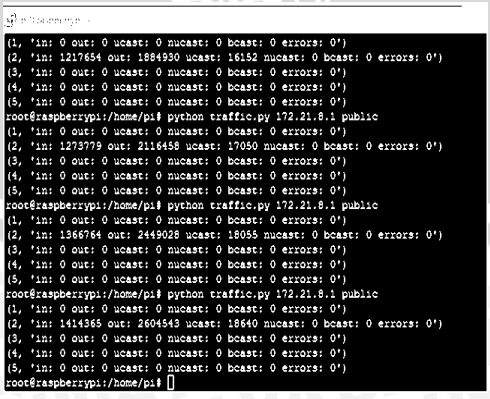


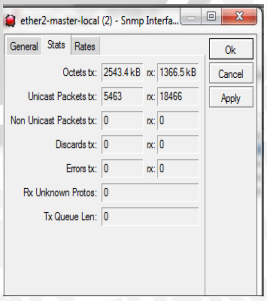


Mikrotik router *Raspberry pi* *The dude*

Menit ke-6

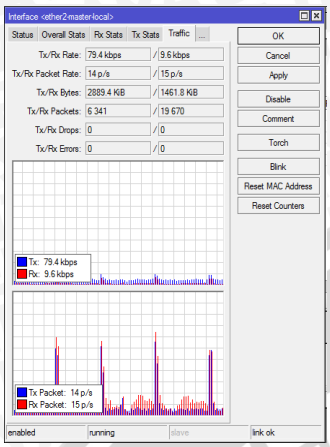




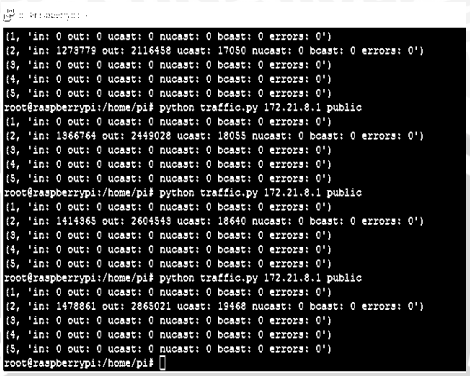


Mikrotik router *Raspberry pi* *The dude*

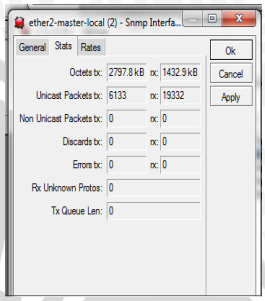
Menit ke-7



Mikrotik router

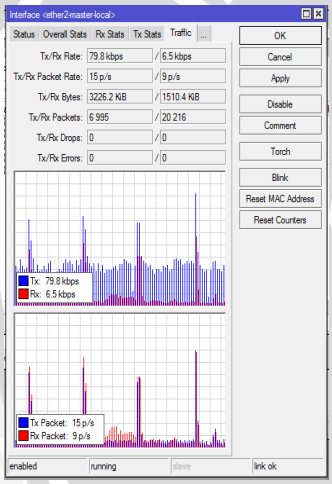


Raspberry pi

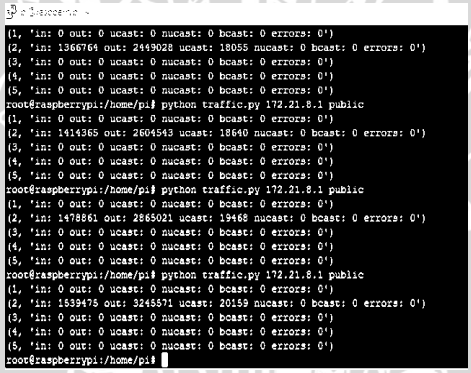


The dude

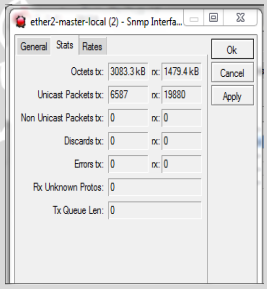
Menit ke-8



Mikrotik router

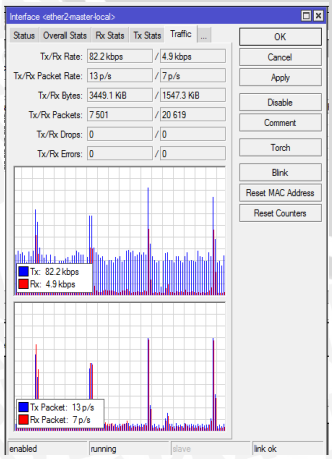


Raspberry pi

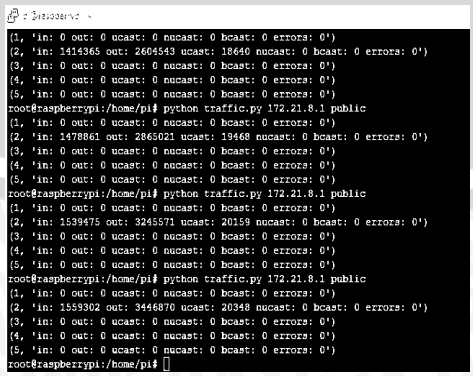


The dude

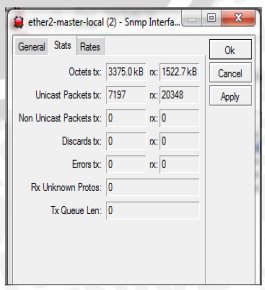
Menit ke - 9



Mikrotik router

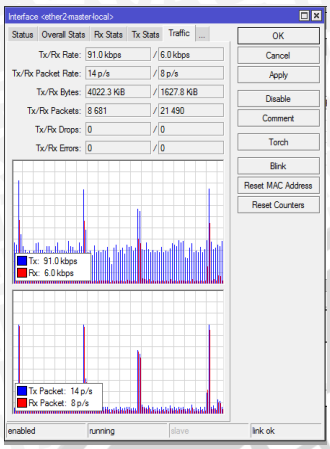


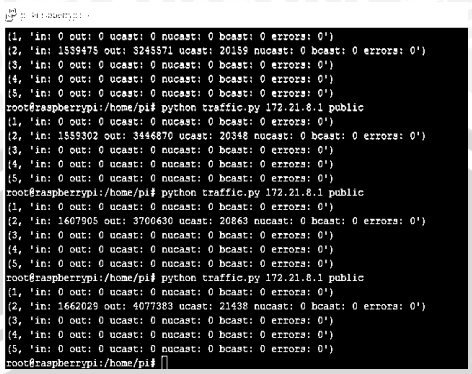
Raspberry pi

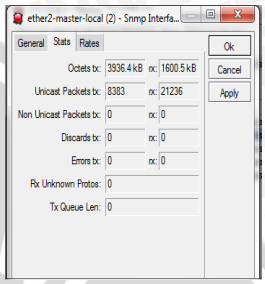


The dude

Menit ke 10



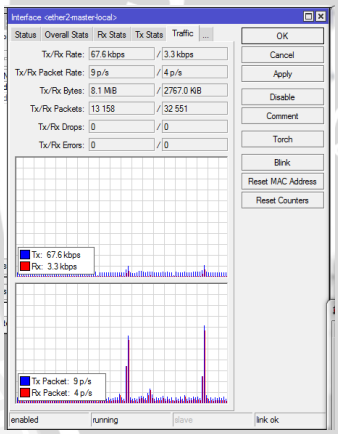


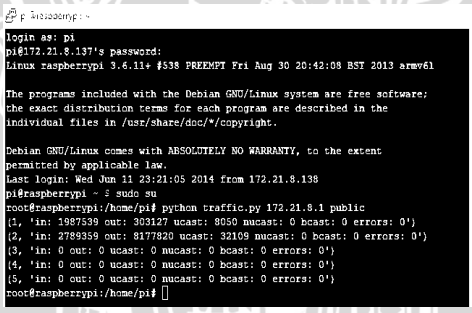


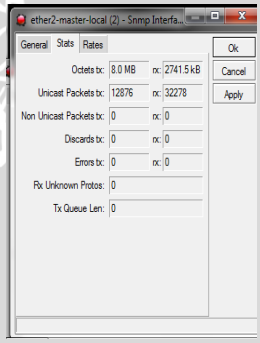
Mikrotik router *Raspberry pi* *The dude*

Lampiran 2. PENGUJIAN 2 (saat *access point* sudah terkoneksi ke internet)

Menit ke-1

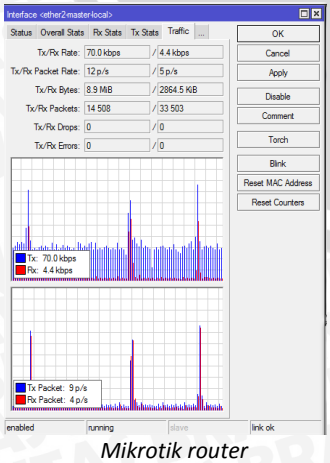


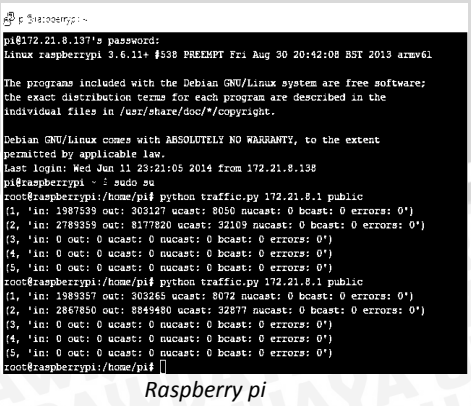


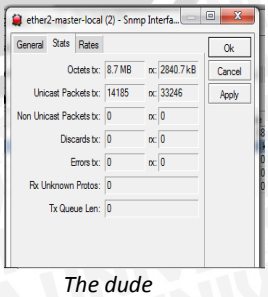


Mikrotik router *Raspberry pi* *The dude*

Menit ke-2

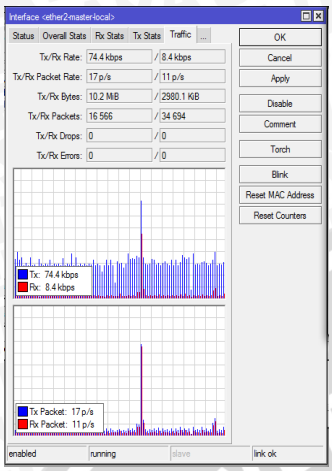


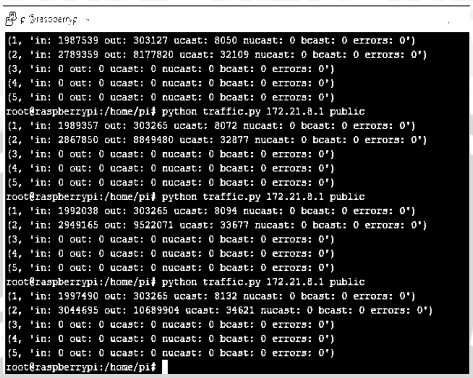


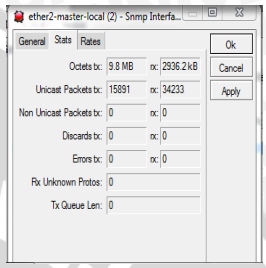


Mikrotik router *Raspberry pi* *The dude*

Menit ke-3

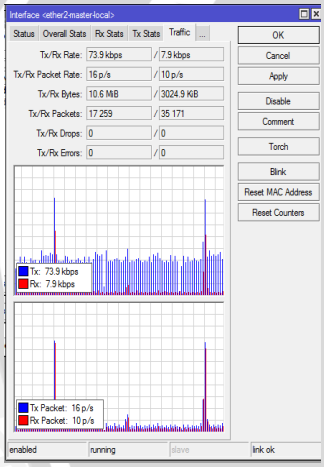


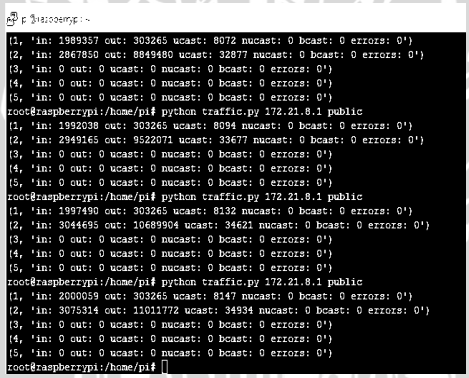


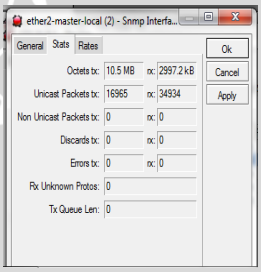


Mikrotik router *Raspberry pi* *The dude*

Menit ke-4

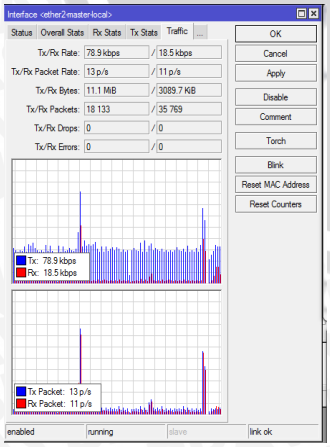


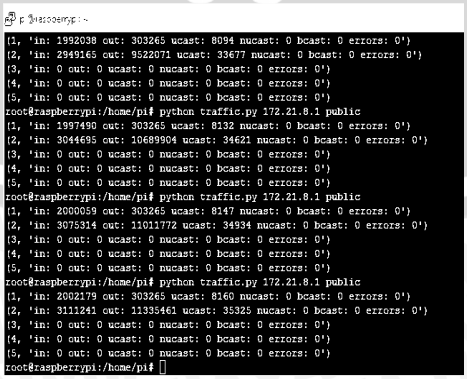


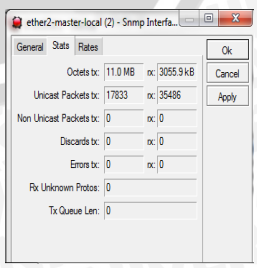


Mikrotik router *Raspberry pi* *The dude*

Menit ke-5

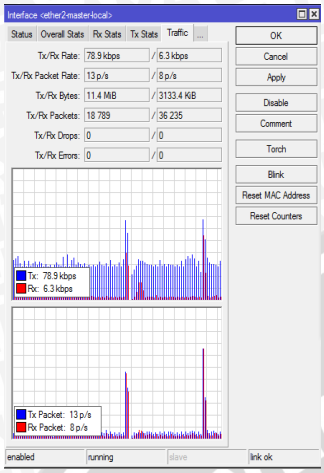




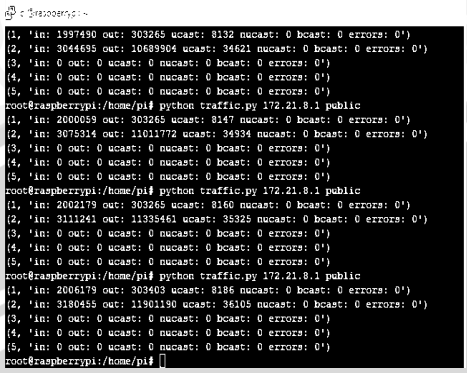


Mikrotik router *Raspberry pi* *The dude*

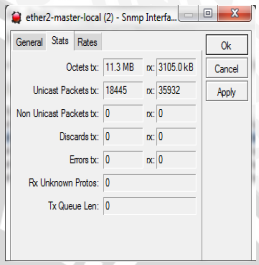
Menit ke-6



Mikrotik router

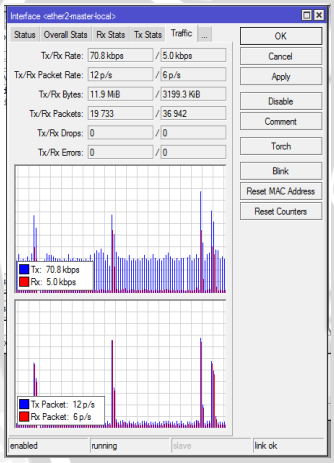


Raspberry pi

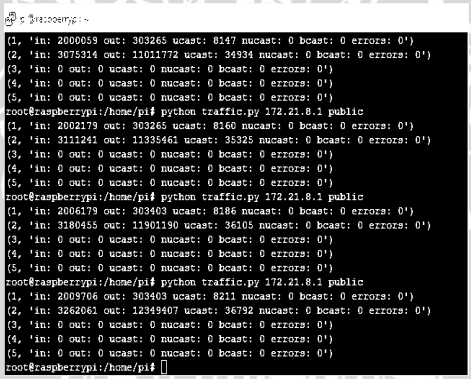


The dude

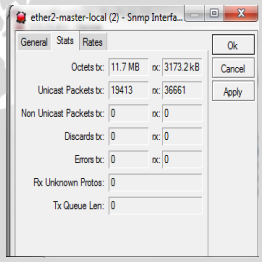
Menit ke-7



Mikrotik router

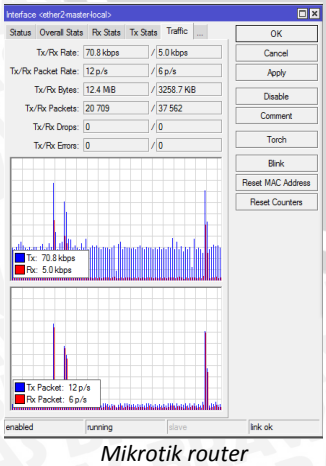


Raspberry pi

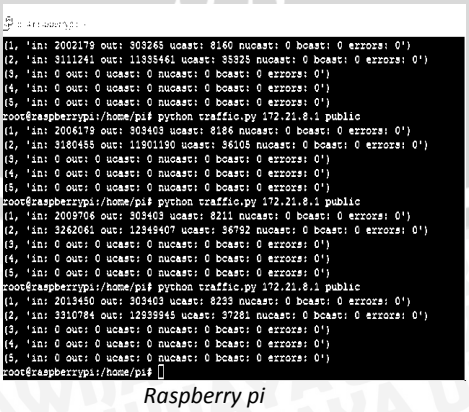


The dude

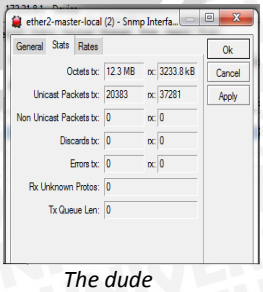
Menit ke-8



Mikrotik router

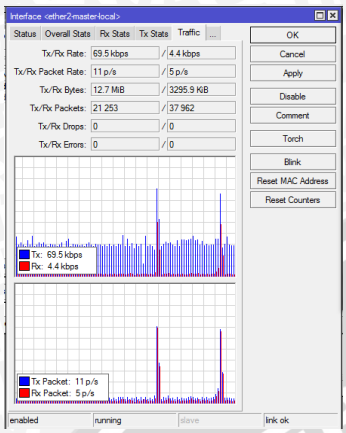


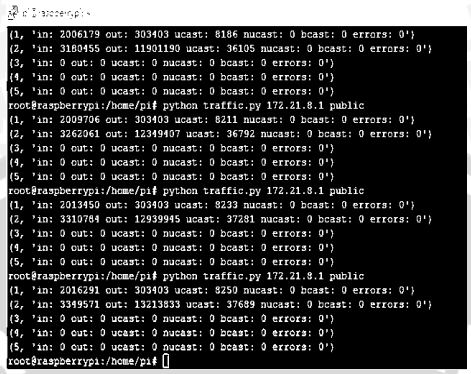
Raspberry pi

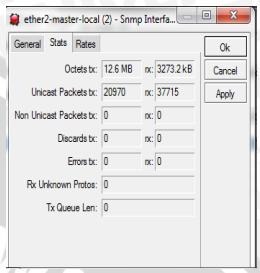


The dude

Menit ke-9

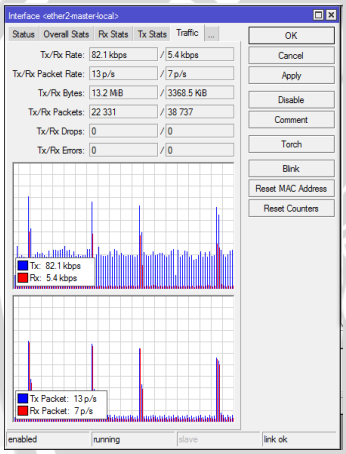


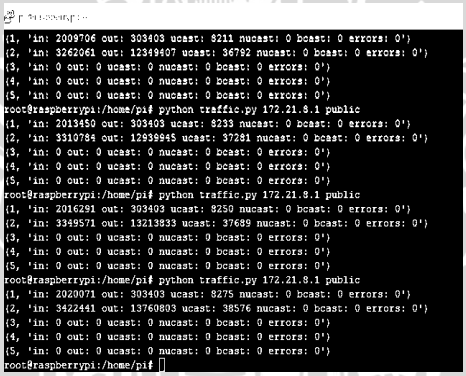


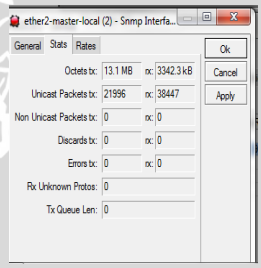


Mikrotik router *Raspberry pi* *The dude*

Menit ke-10



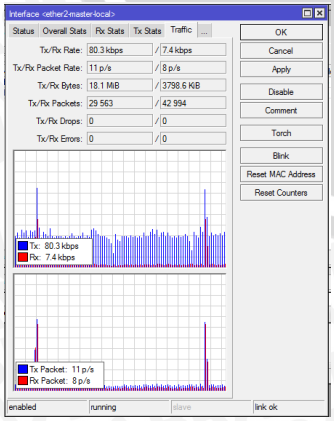


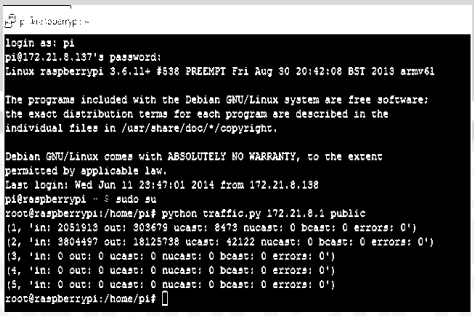


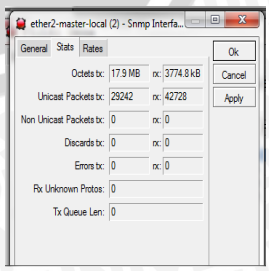
Mikrotik router *Raspberry pi* *The dude*

Lampiran 3. PENGUJIAN 3a (saat 1 *user* melakukan *browsing*)

Menit ke-1

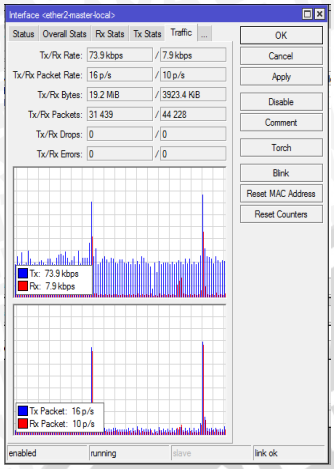




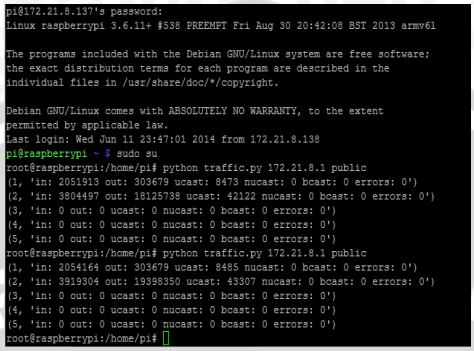


Mikrotik router *Raspberry pi* *The dude*

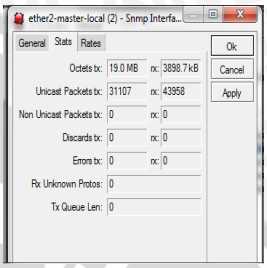
Menit ke-2



Mikrotik router

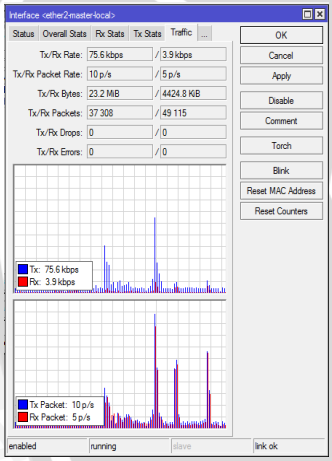


Raspberry pi

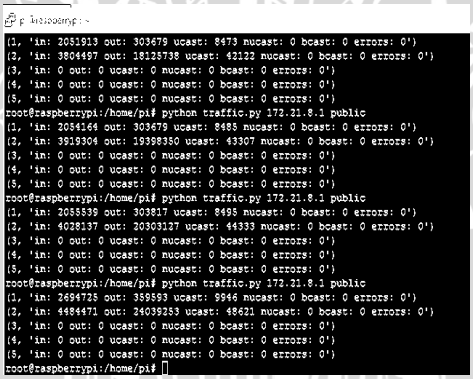


The dude

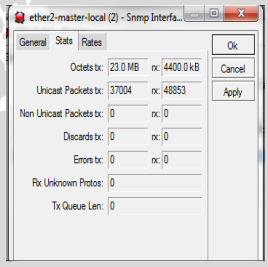
Menit ke-3



Mikrotik router

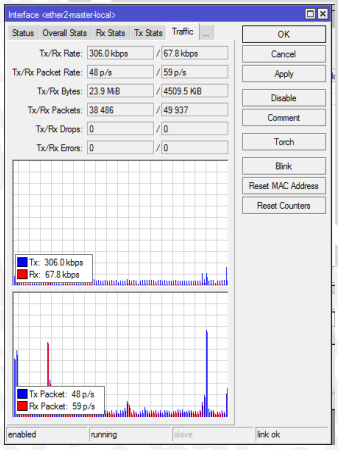


Raspberry pi

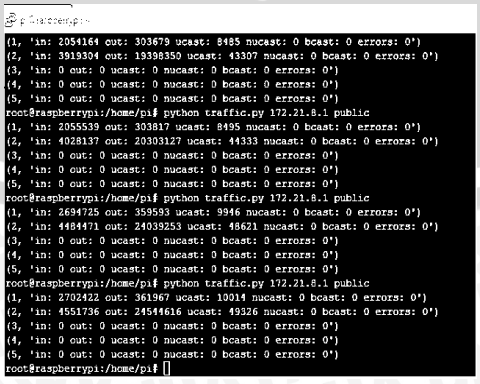


The dude

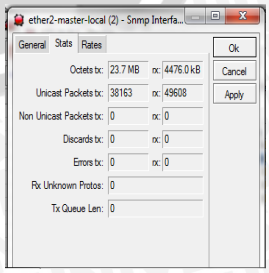
Menit ke-4



Mikrotik router

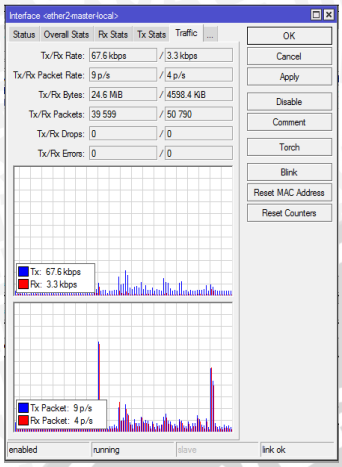


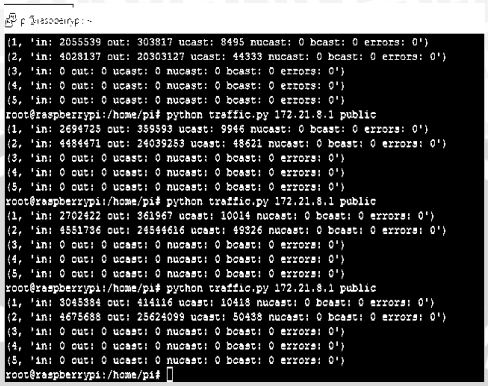
Raspberry pi

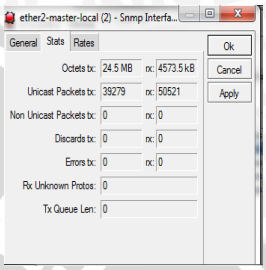


The dude

Menit ke-5

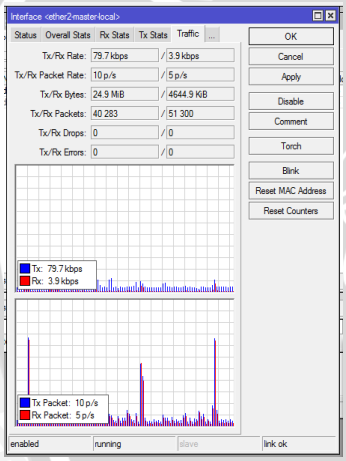


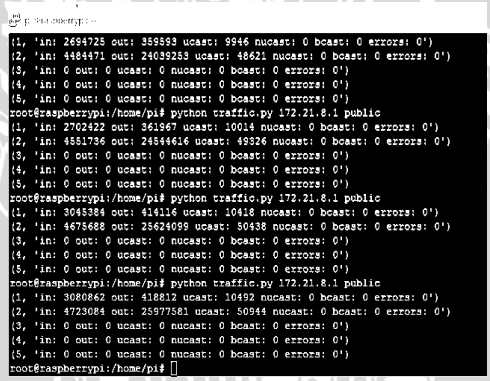


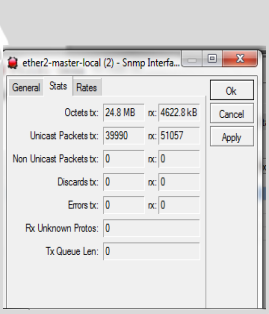


Mikrotik router *Raspberry pi* *The dude*

Menit ke-6

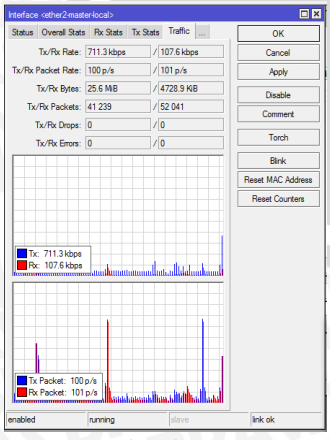


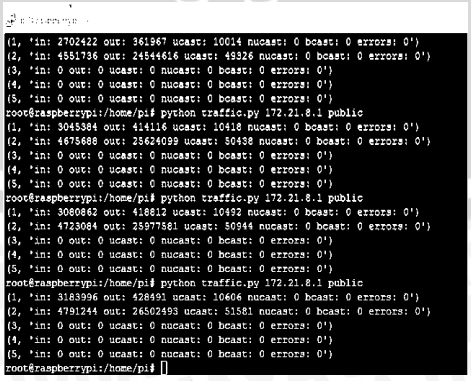


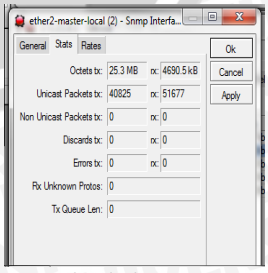


Mikrotik router *Raspberry pi* *The dude*

Menit ke-7

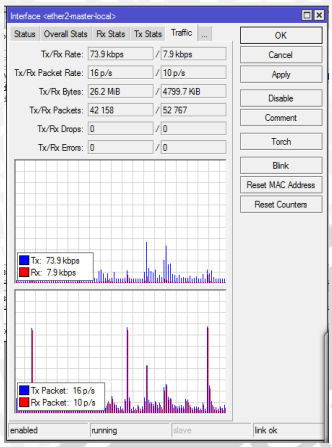




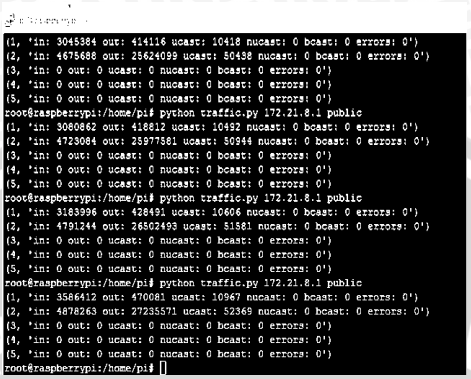


Mikrotik router *Raspberry pi* *The dude*

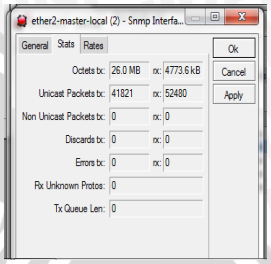
Menit ke-8



Mikrotik router

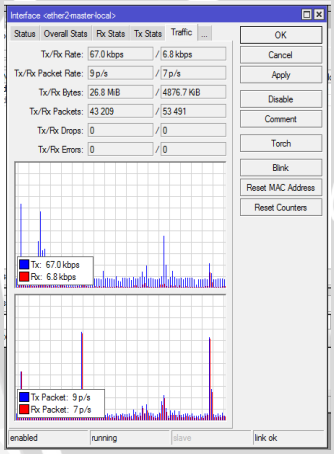


Raspberry pi

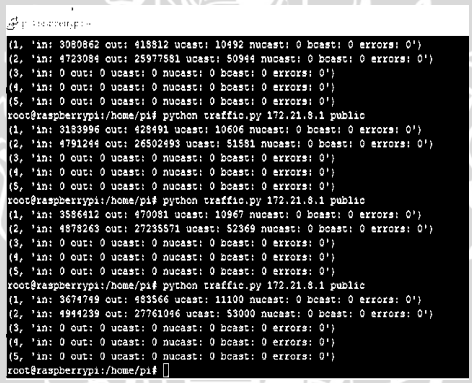


The dude

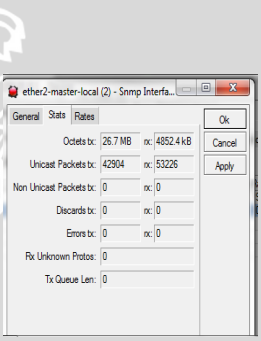
Menit ke-9



Mikrotik router

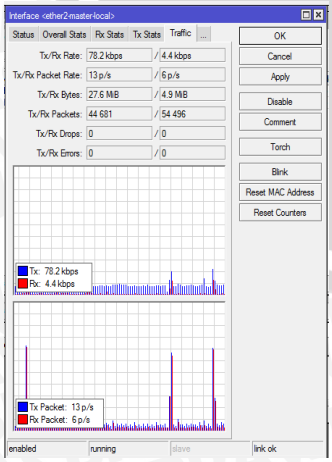


Raspberry pi

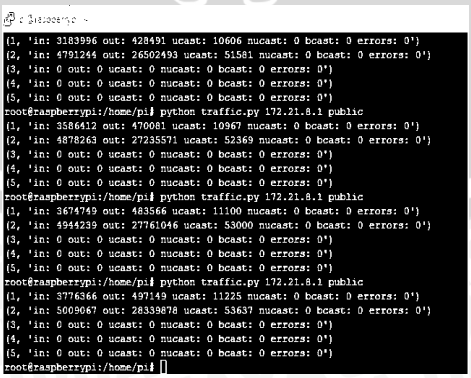


The dude

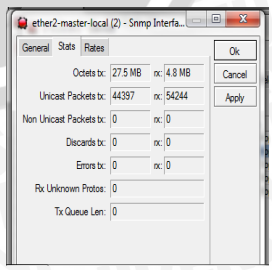
Menit ke-10



Mikrotik router



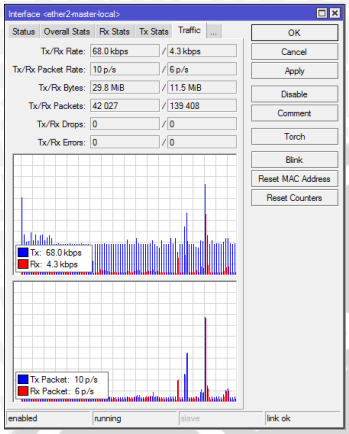
Raspberry pi

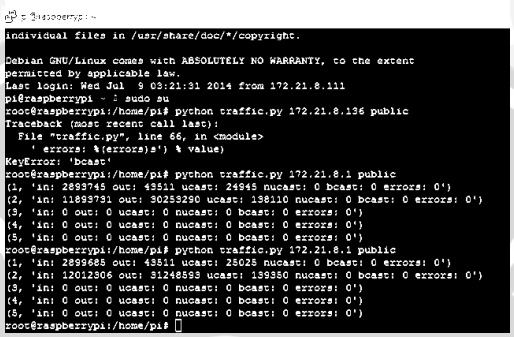


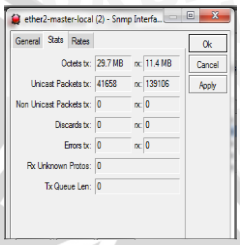
The dude

Lampiran 4. PENGUJIAN 3b (saat 3 *user* melakukan *browsing*)

Menit ke-1

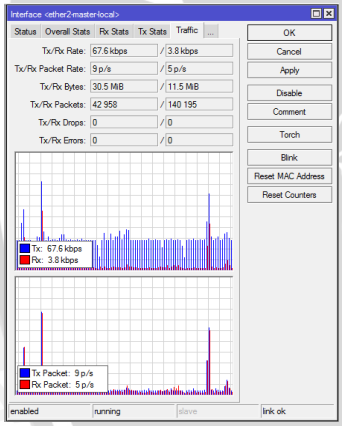


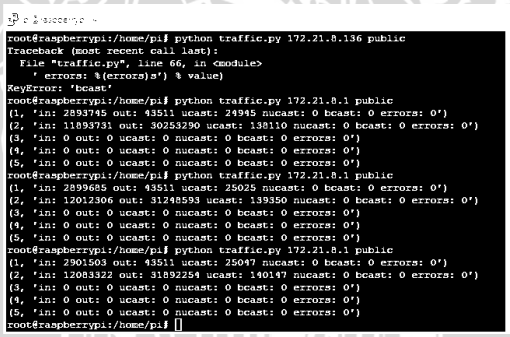


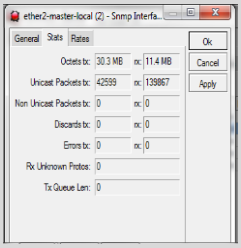


Mikrotik router *Raspberry pi* *The dude*

Menit ke-2

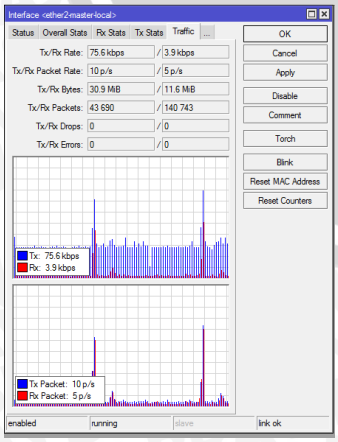


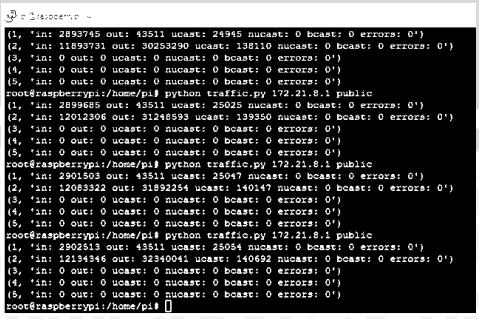


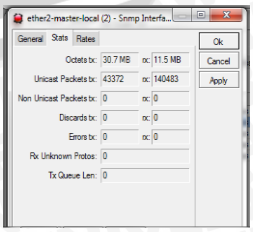


Mikrotik router *Raspberry pi* *The dude*

Menit ke-3

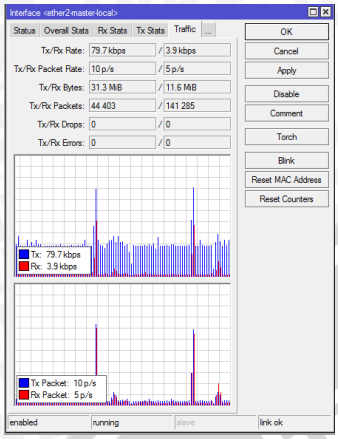


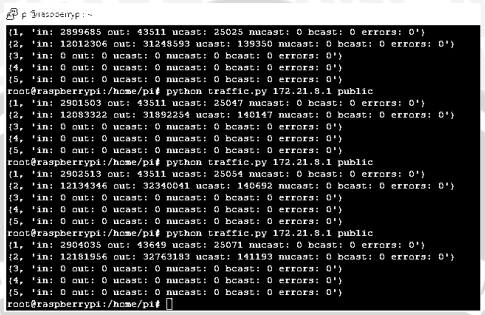


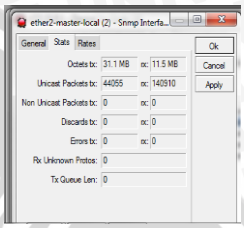


Mikrotik router *Raspberry pi* *The dude*

Menit ke-4

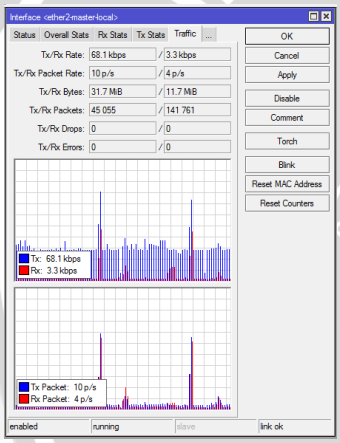


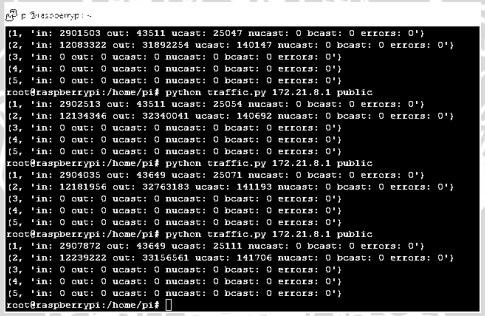


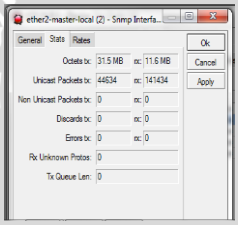


Mikrotik router *Raspberry pi* *The dude*

Menit ke-5

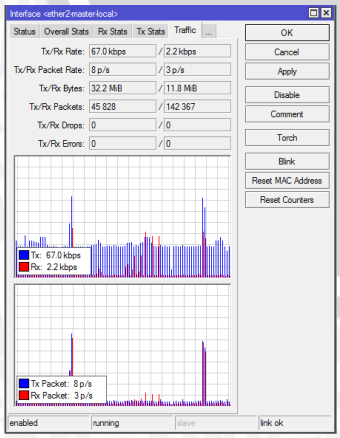


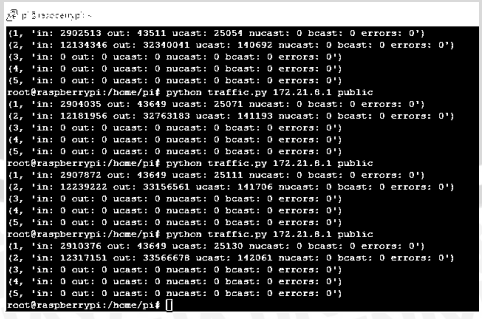


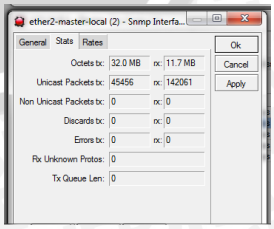


Mikrotik router *Raspberry pi* *The dude*

Menit ke-6

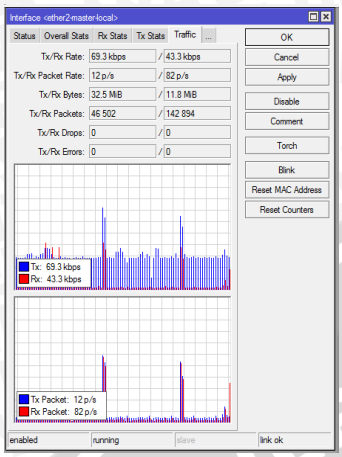




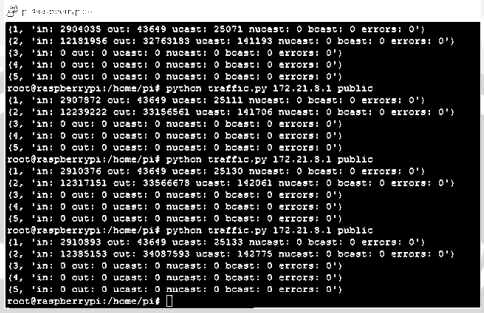


Mikrotik router *Raspberry pi* *The dude*

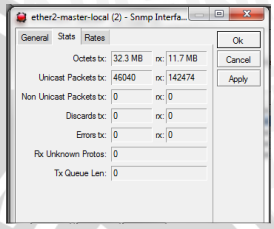
Menit ke-7



Mikrotik router

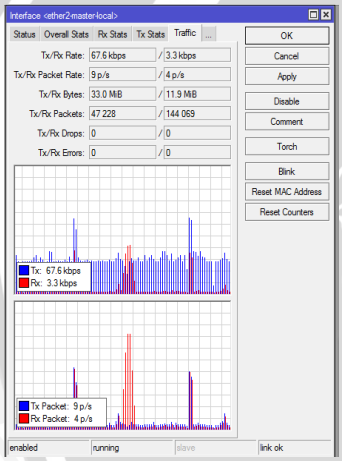


Raspberry pi

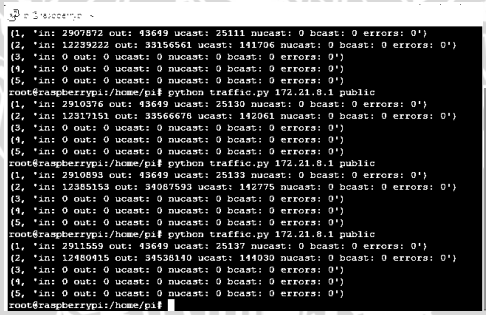


The dude

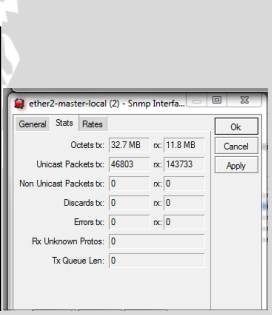
Menit ke-8



Mikrotik router

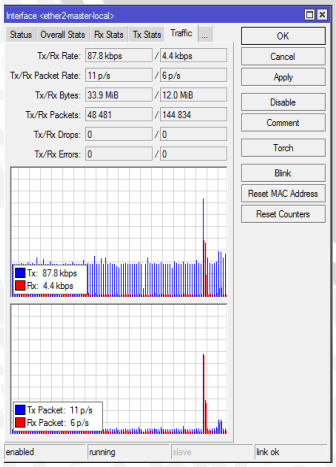


Raspberry pi

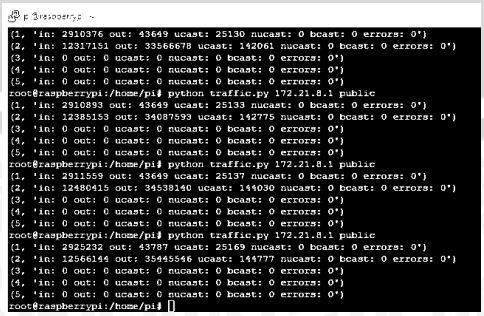


The dude

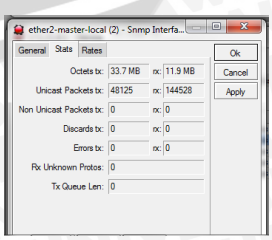
Menit ke-9



Mikrotik router

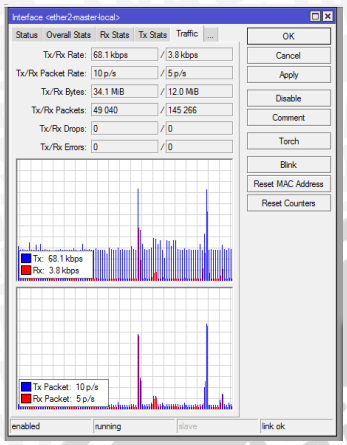


Raspberry pi

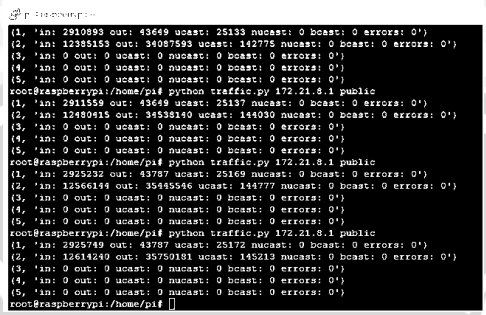


The dude

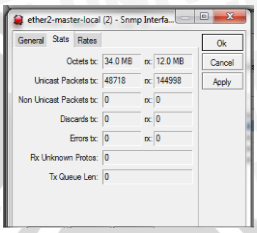
Menit ke-10



Mikrotik router



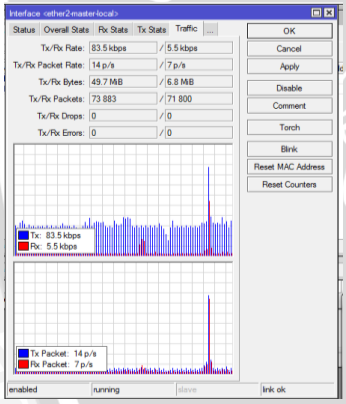
Raspberry pi



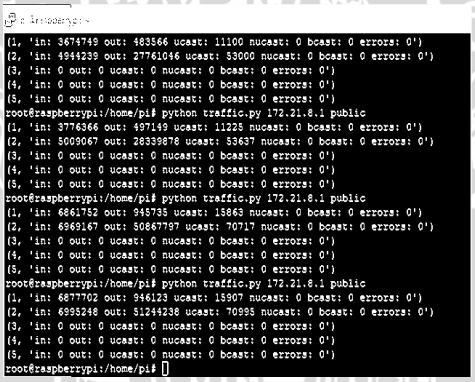
The dude

Lampiran 5. PENGUJIAN 4a (saat 1 *user* melakukan *browsing*)

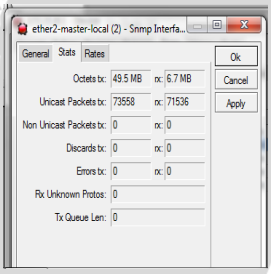
Menit ke-1



Mikrotik router

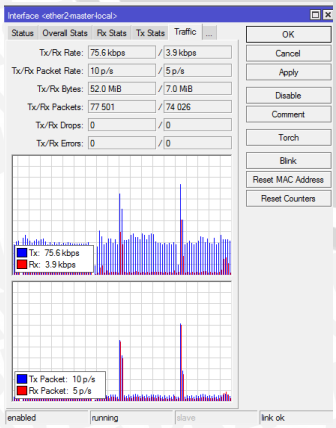


Raspberry pi

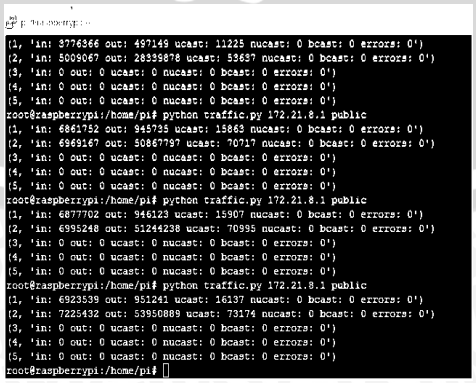


The dude

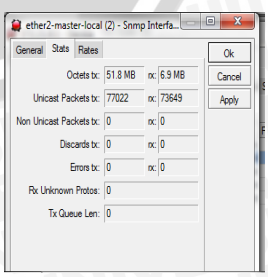
Menit ke-2



Mikrotik router

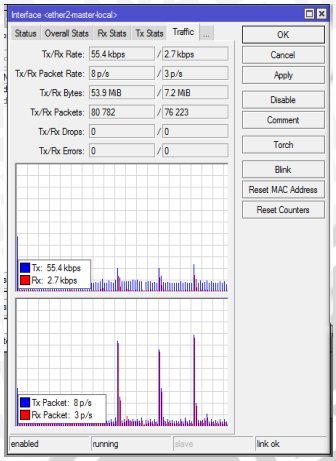


Raspberry pi

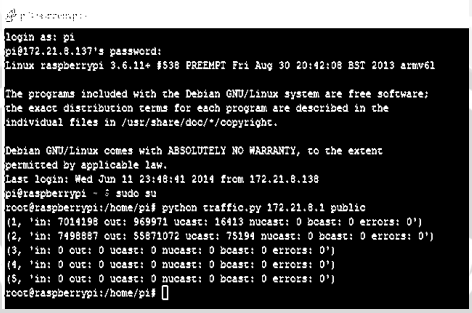


The dude

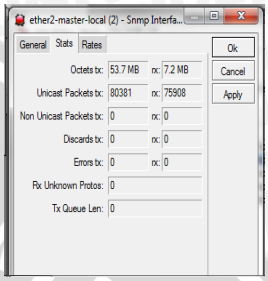
Menit ke-3



Mikrotik router

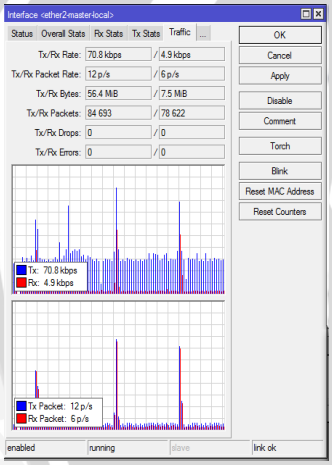


Raspberry pi

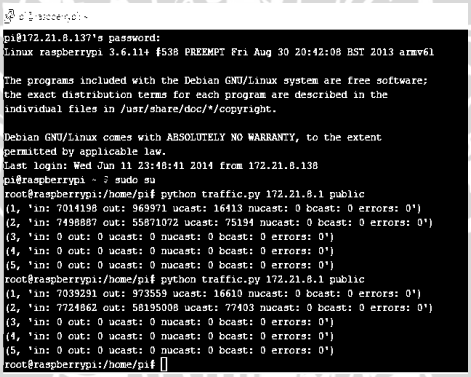


The dude

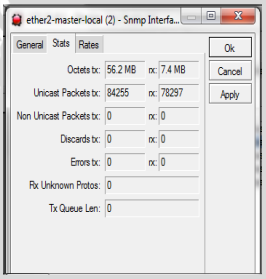
Menit ke-4



Mikrotik router

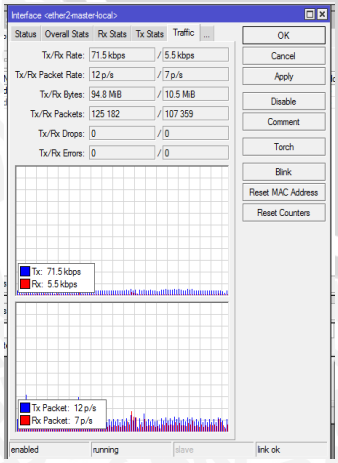


Raspberry pi

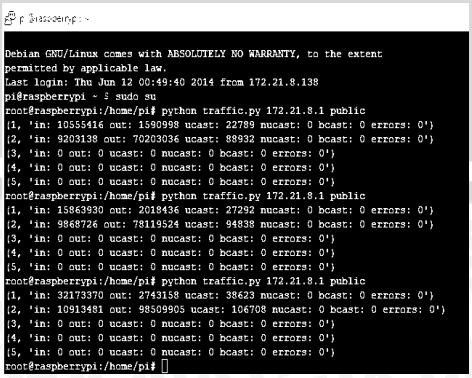


The dude

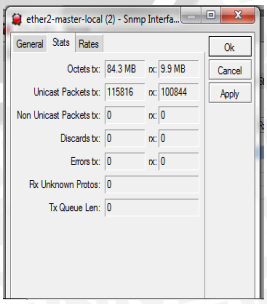
Menit ke-5



Mikrotik router

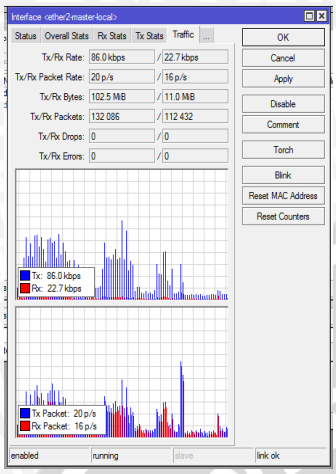


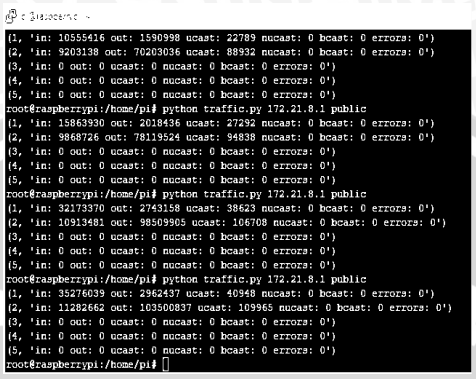
Raspberry pi

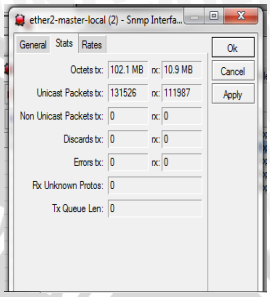


The dude

Menit ke-6

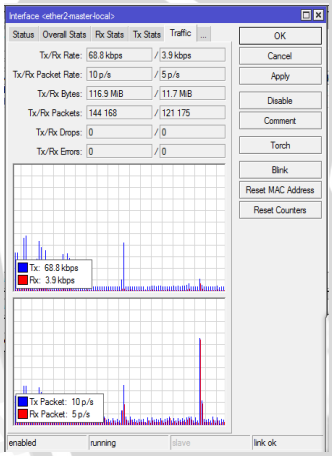


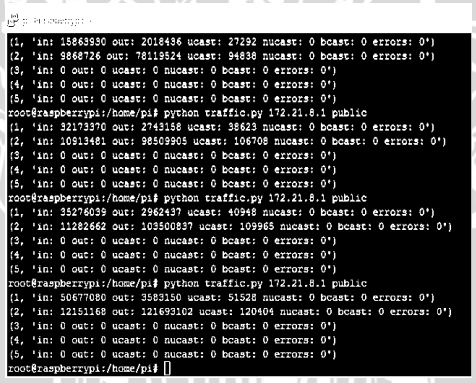


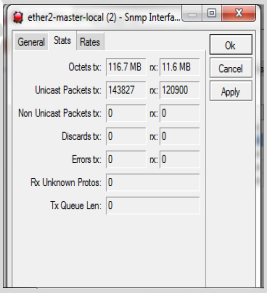


Mikrotik router *Raspberry pi* *The dude*

Menit ke-7

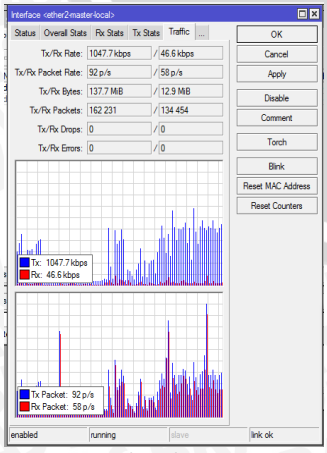


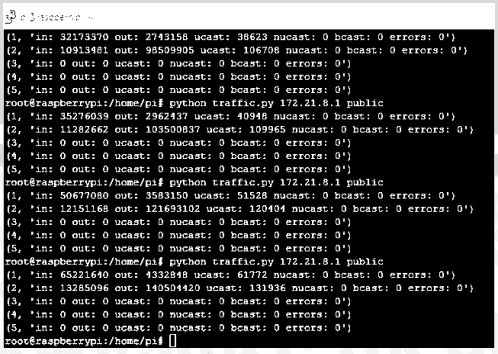


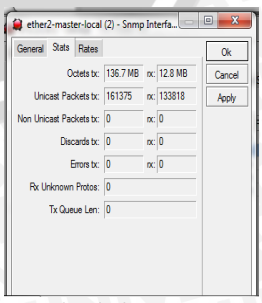


Mikrotik router *Raspberry pi* *The dude*

Menit ke-8

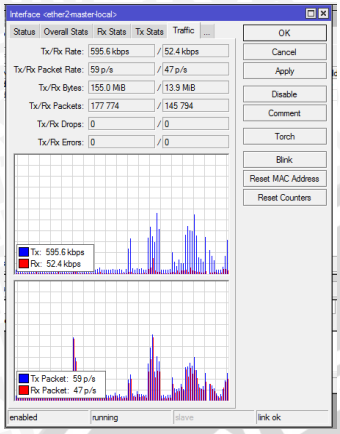




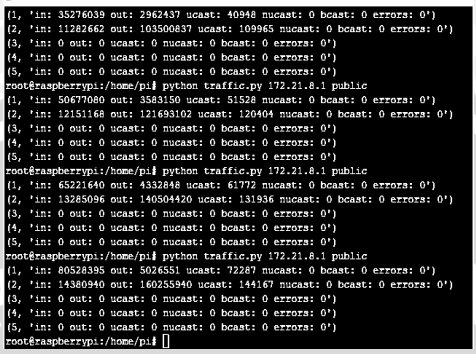


Mikrotik router *Raspberry pi* *The dude*

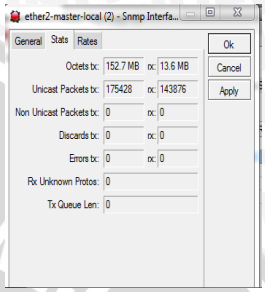
Menit ke-9



Mikrotik router

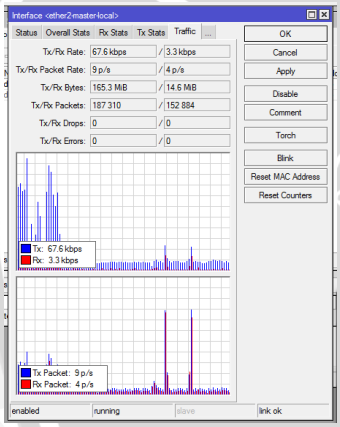


Raspberry pi

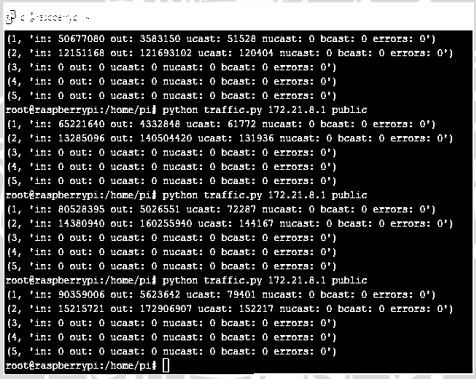


The dude

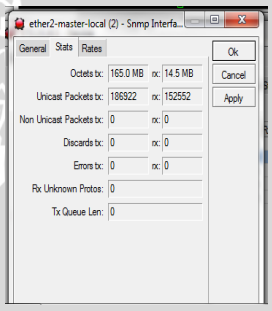
Menit ke-10



Mikrotik router



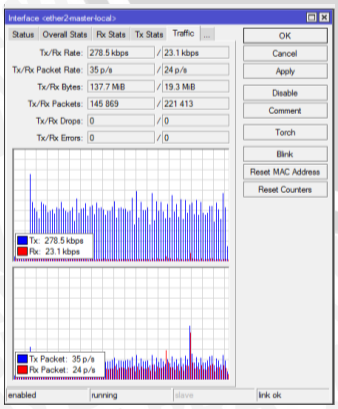
Raspberry pi



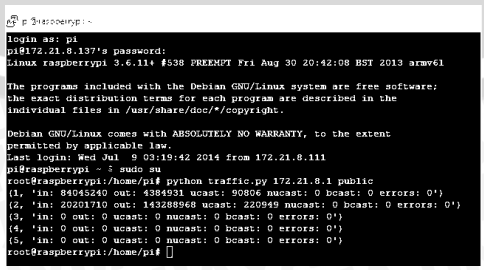
The dude

Lampiran 6. PENGUJIAN 4b (saat 3 *user* melakukan *download*)

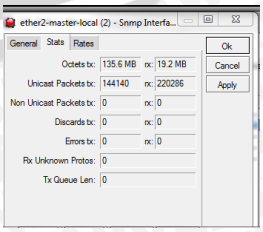
Menit ke-1



Mikrotik router

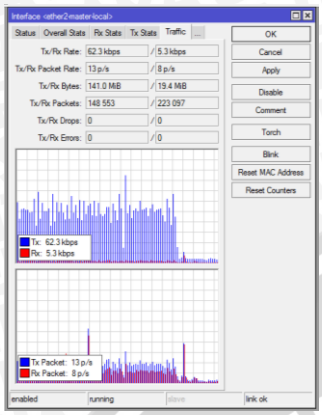


Raspberry pi

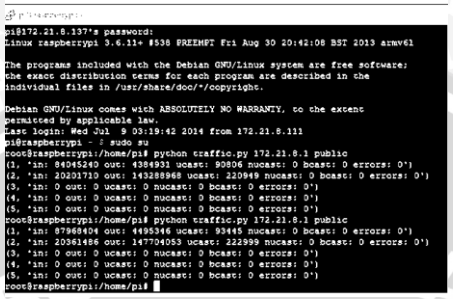


The dude

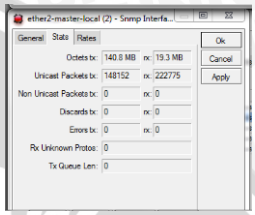
Menit ke-2



Mikrotik router

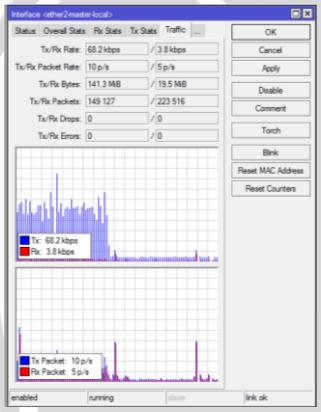


Raspberry pi

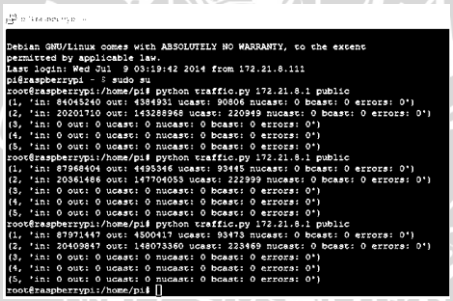


The dude

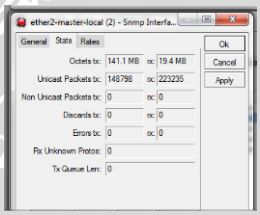
Menit ke-3



Mikrotik router

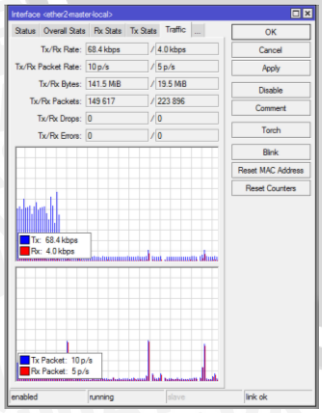


Raspberry pi

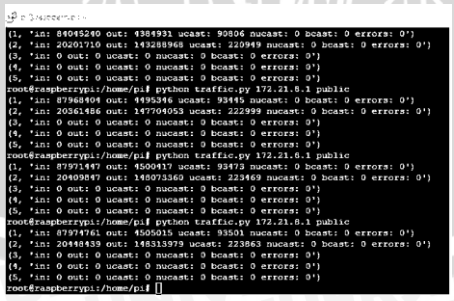


The dude

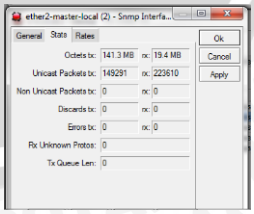
Menit ke-4



Mikrotik router

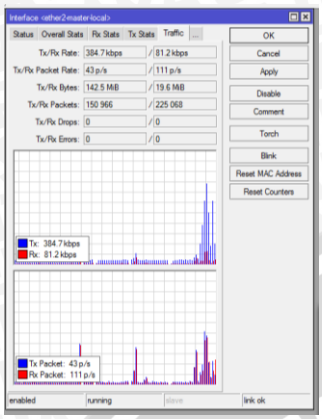


Raspberry pi

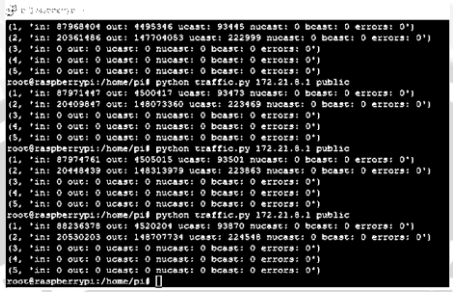


The dude

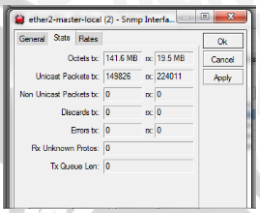
Menit ke-5



Mikrotik router

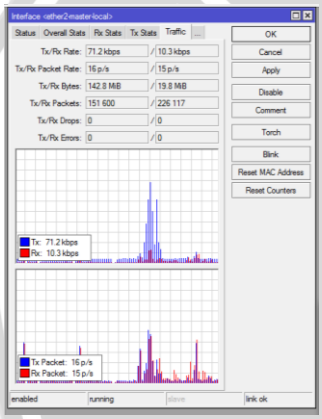


Raspberry pi

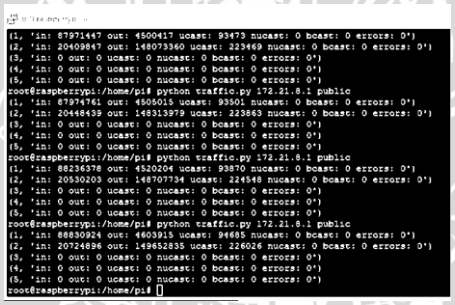


The dude

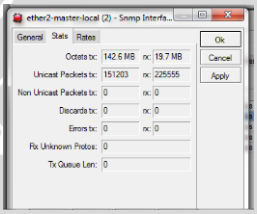
Menit ke-6



Mikrotik router

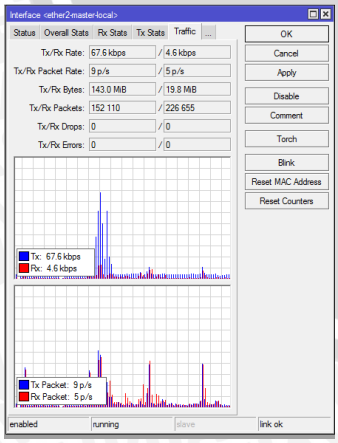


Raspberry pi

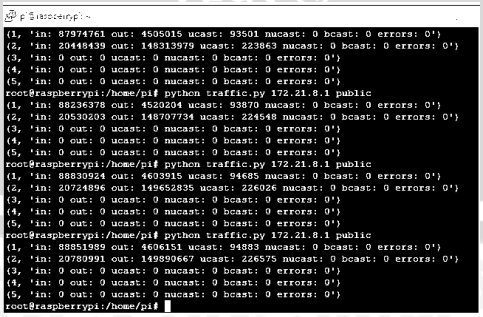


The dude

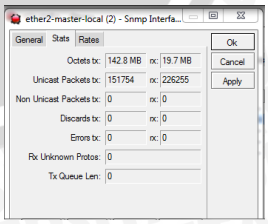
Menit ke-7



Mikrotik router

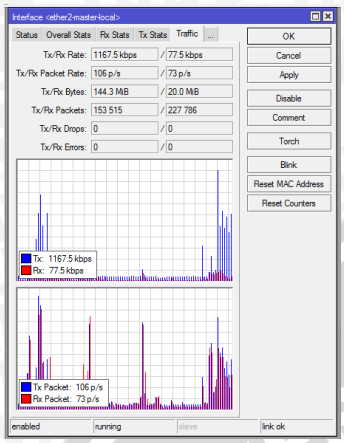


Raspberry pi

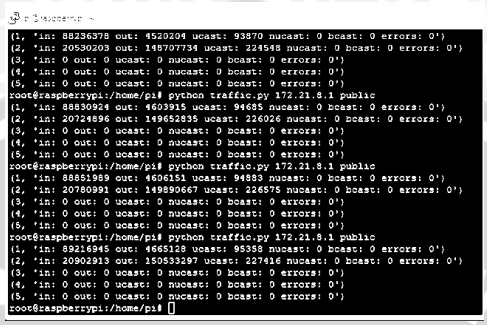


The dude

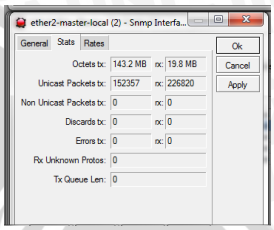
Menit ke-8



Mikrotik router

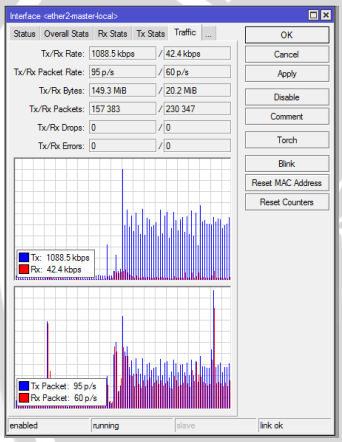


Raspberry pi

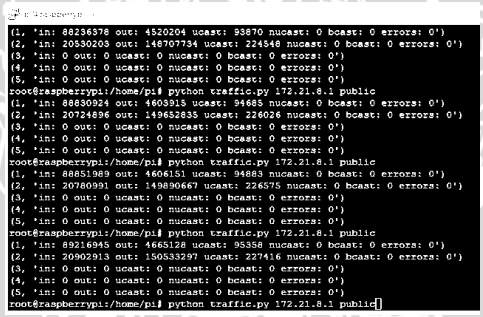


The dude

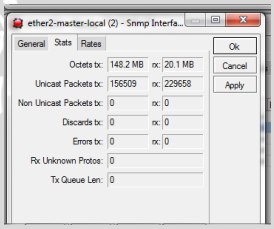
Menit ke-9



Mikrotik router

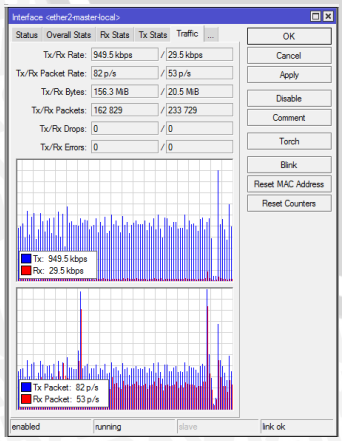


Raspberry pi

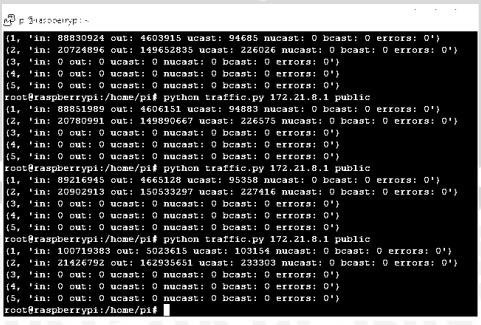


The dude

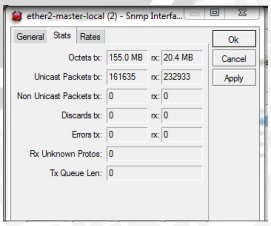
Menit ke-10



Mikrotik router



Raspberry pi



The dude

Lampiran 7. *Python source code* fdb.py

```
#!/usr/bin/env python

import os, sys
import socket
import random
from datetime import datetime as dt

from pysnmp.entity.rfc3413.oneliner import cmdgen
from pysnmp.proto.rfc1902 import Integer, IpAddress, OctetString

def fetchFdb(ip, community):#mendeklarasikan fungsi fdb yang berisi ip dan
community/ komunitas  pada snmp yang biasanya bersifat publik

    mib = '1.3.6.1.2.1.17.7.1.2.2.1.2' #parameter mib
    value = tuple([int(i) for i in mib.split('.')])

    generator = cmdgen.CommandGenerator()
    comm_data = cmdgen.CommunityData('server', community, 1) # 1 berarti versi
SNMP v2c
    transport = cmdgen.UdpTransportTarget((ip, 161))#161 menunjukkan port udp

    real_fun = getattr(generator,'nextCmd') #'getCmd')
    res = (errorIndication, errorStatus, errorIndex, varBindTable)\
    = real_fun(comm_data, transport, value)
    if not errorIndication is None  or errorStatus is True:
           print "Error: %s %s %s" % res
    else:
        for varBindTableRow in varBindTable:

            data = varBindTableRow[0][0]._value[len(value):]

            vlan = data[0]
            mac = '%02x:%02x:%02x:%02x:%02x:%02x' % tuple(map(int, data[-6:]))
            port = varBindTableRow[0][1]
            yield {'vlan': vlan, 'mac': mac, 'port': port}

if __name__ == '__main__':
    try:
        ip = sys.argv[1]
        community = sys.argv[2]
    except IndexError:
       print "PLease run command like:"
       print "python %s <ip> <community>" % __file__
       sys.exit(0)

    for fdb in fetchFdb(ip, community):
        print 'vlan: %(vlan)s mac: %(mac)s port: %(port)s' % (fdb)
```

Lampiran 8. *Python source code* traffic.py

```
#!usr/bin/env python

import sys
import collections
from pysnmp.entity.rfc3413.oneliner import cmdgen

def datafrommib(mib, community, conn):#mendeklarasi data dari mib, berisi mib,
komunitas dan koneksi
```

```
    value = tuple([int(i) for i in mib.split('.')])

    res = (errorIndication, errorStatus, errorIndex, varBindTable)\
          = conn[3](conn[1], conn[2], value)

    if not errorIndication is None  or errorStatus is True:
          print ("Error: %s %s %s %s") % res
          yield None
    else:
        for varBindTableRow in varBindTable:

            data = varBindTableRow[0]
            port = data[0]._value[len(value):]
            octets = data[1]

            yield {'port': port[0], 'octets': octets}

def load(ip, community):
    generator = cmdgen.CommandGenerator()
    comm_data = cmdgen.CommunityData('server', community, 1) # 1 berarti versi
SNMP
    transport = cmdgen.UdpTransportTarget((ip, 161))
    real_fun = getattr(generator, 'nextCmd')
    conn = (generator, comm_data, transport, real_fun)
    mibs = [('1.3.6.1.2.1.2.2.1.16', 'out'),
            ('1.3.6.1.2.1.2.2.1.10', 'in'),
            ('1.3.6.1.2.1.2.2.1.11', 'ucast'),
            ('1.3.6.1.2.1.2.2.1.12', 'nucast'),
            ('1.3.6.1.2.1.2.2.1.13', 'discards'),
            ('1.3.6.1.2.1.2.2.1.14', 'errors')]

    ports = collections.defaultdict(dict)

    for mib in mibs:
        data = datafrommib(mib[0], community, conn)
        for row in data:
            if row:
                ports[row['port']][mib[1]] = int(row['octets'])
            else:
                return None

    return ports

if __name__ == '__main__':
    try:
        ip = sys.argv[1]
        community = sys.argv[2]
    except IndexError:
        print ("Please run command like:")
        print ("python %s <ip> <community>") % __file__
        sys.exit(0)

    ports = load(ip, community)
    if ports:
        for key, value in ports.items():
            print (key, ('in: %(in)s out: %(out)s ucast: %(ucast)s' +\
                       ' nucast: %(nucast)s discards: %(discards)s' +\
                       ' errors: %(errors)s') % value)
```