SISTEM DETEKSI PENGGUNAAN *TUNNELING SOFTWARE*

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun Oleh:

SILVIA ARI SANTHY

NIM. 105060800111029

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2014

LEMBAR PERSETUJUAN

SISTEM DETEKSI PENGGUNAAN *TUNNELING SOFTWARE*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

SILVIA ARI SANTHY

NIM. 105060800111029

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I	Dosen Pembimbing II
R. Arief Setyawan, ST., MT.	**Aswin Suharsono, ST., MT.**
NIP. 19750819 199903 1 001	**NIK. 840919 06 1 1 0251**

LEMBAR PENGESAHAN
SISTEM DETEKSI PENGGUNAAN *TUNNELING SOFTWARE*

SKRIPSI

KONSENTRASI KOMPUTASI BERBASIS JARINGAN

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

SILVIA ARI SANTHY
NIM. 105060800111029

Skripsi ini telah diuji dan dinyatakan lulus pada

tanggal 21 Juli 2014

Penguji 1	Penguji 2
Achmad Basuki, ST., MMG., Ph.D	**Kasyful Amron, ST., M.Sc**
NIP. 19741118 200312 1 002	**NIP. 19750803 200312 1 003**

Penguji 3

Eko Sakti Pramukantoro, S.Kom., M.Kom

NIK. 86080506110252

Mengetahui

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, MT.

NIP. 19670801 199203 1 001

PERNYATAAN

ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam 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, 21 Juli 2014

Mahasiswa,

Silvia Ari Santhy

NIM. 105060800111029

KATA PENGANTAR

Puji Syukur kehadirat Tuhan Yang Maha Esa yang telah mencurahkan kasih dan rahmat-Nya sehingga skripsi yang berjudul "**Sistem Deteksi Penggunaan *Tunneling Software***" ini dapat diselesaikan.

Dalam menyelesaikan skripsi ini, penulis telah banyak mendapat bantuan dari berbagai pihak. Pada kesempatan ini, penulis mengucapkan banyak terima kasih kepada :

1. Kedua orang tua penulis, Ayahanda Subiyanto dan Ibunda Suharti yang selalu memberikan do'a dan harapan untuk terselesaikannya skripsi ini serta memberikan dukungan moral, material dan cinta kasihnya yang begitu besar.
2. Kakak penulis, Hendro Cahyono dan Eka Wahyuni yang senantiasa memberikan dorongan dan motivasi.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Edy Santoso, S.Si, M.Kom selaku Ketua, Wakil Ketua I, Wakil Ketua II dan Wakil Ketua III Program Teknologi Informasi dan Ilmu Komputer.
4. Bapak Drs. Marji, MT. Dan Bapak Issa Arwani S.Kom, MSc. Selaku Ketua Program dan Sekretaris Program Studi Informatika / Ilmu Komputer, segenap Bapak/Ibu Dosen dan seluruh Staff PTIIK Universitas Brawijaya.
5. Bapak R. Arief Setyawan, ST., MT. Selaku Dosen Pembimbing I yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyelesaian skripsi ini.
6. Bapak Aswin Suharsono, ST., MT. Selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyelesaian skripsi ini.
7. Senior penulis, Mas Amri, Mas Bagus dan senior lain yang telah membantu dan memberikan petunjuk.
8. Teman penulis, Mitta, Cah Cong, Yulis, Zul, Uli, Sari, Gopy yang selalu bertukar pikiran dan semangat selama pengerjaan skripsi ini.

9. Sahabat penulis, Hesti dan Sysy yang selalu saling menyemangati selama pengerjaan skripsi masing-masing.
10. Semua teman Teknik Informatika 2010.
11. Pihak lain yang tidak bisa penulis sebutkan satu per satu baik yang terlibat secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Hanya do'a yang bisa penulis berikan dan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Aamiin.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan masih memiliki banyak kekurangan. Untuk itu, kritik dan saran yang membangun sangat penulis harapkan. Semoga laporan skripsi ini membawa manfaat bagi penulis maupun pihak lain yang menggunakannya.

Malang, 24 Juni 2014

Penulis

ABSTRAK

SILVIA ARI SANTHY. 2014. : Sistem Deteksi Penggunaan *Tunneling Software*. Skripsi Program Studi Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
Dosen Pembimbing : R. Arief Setyawan, ST., MT. dan Aswin Suharsono, ST., MT.

Di era globalisasi saat ini, penggunaan internet sudah menjadi kebutuhan bagi sebagian orang. Berbagai layanan yang ada di internet seperti sosial media, portal berita dll merupakan layanan yang biasa diakses. Namun, biasanya ditempat-tempat seperti sekolah, kantor atau kampus terjadi pembatasan akses terhadap beberapa layanan seperti sosial media. Peraturan ini dibuat agar pengguna internet di wilayah tersebut bisa menggunakan fasilitas internet secara bijak di jam kerja. Tapi, ada beberapa orang yang berusaha melewati batasan tersebut menggunakan *tunneling software.*

Untuk membantu administrator jaringan dalam menegakkan kebijakan yang ada, maka perlu dibuat sebuah sistem deteksi penggunaan *tunneling software.* Sistem ini dibuat dengan menggunakan metode *naïve bayes classifier* untuk melakukan klasifikasi antara pengguna *tunneling software* dan pengguna normal. Dengan diketahuinya karakteristik tersebut diharapkan administrator jaringan dapat mengetahui pengguna yang menggunakan *tunneling software* dan kemudian bisa melakukan pemutusan koneksi terhadap pengguna yang bersangkutan.

Berdasarkan pengujian yang telah dilakukan, keseluruhan fungsional sistem deteksi berjalan dengan baik sesuai dengan perancangan yang telah dibuat. Hasil dari pengujian ini menyatakan tingkat akurasinya mencapai 100%. Tingkat akurasi ini bisa berubah karena tergantung pada data latih dan *tunneling software* yang digunakan.

Kata Kunci : Sistem deteksi, *Tunneling Software*, Klasifikasi

ABSTRACT

SILVIA ARI SANTHY. 2014. : Detection System of Using Tunneling Software. Skripsi Informatics Technology and Computer Science Study. Information Technology and Computer Science Program Brawijaya University. Supervisor : R. Arief Setyawan, ST., MT. and Aswin Suharsono, ST., MT.

In the current era of globalization, the use of the Internet has become a necessity for some people. A range of services available on the internet such as social media, news portals etc. are commonly accessed services. However, usually in places like schools, offices or campuses occurs restrictions on access to some services such as social media. This regulation was made that Internet users in the region could use the internet facility in working hours wisely. But, there are some people who tried to go beyond the use tunneling software.

To assist network administrators in enforcing the existing policy, it needs to make a detection system using tunneling software. This system is made by using naïve Bayes classifier to perform classification between tunneling software users and normal users. By knowing the expected characteristics of the network administrator can determine which users use tunneling software and can then terminate the connection to the user.

Based on the testing that has been done, the overall functional detection system running properly in accordance with the design that has been created. The results of these tests claim accuracy rate reaches 100%. This level of accuracy can be changed because it depends on the training data and tunneling software used.

KEYWORD : *Detection System, Tunneling Software, Classification*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR PERSAMAAN

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Aktifitas di dunia maya sangat erat hubungannya dengan akses suatu layanan ataupun server. Layanan yang biasa diminta antara lain browsing ke suatu situs, chatting, check email dan aktifitas-aktifitas lain yang dilakukan di internet. Pada saat ini, ada banyak layanan yang tidak bisa diakses secara bebas terutama ketika berada di lingkungan sekolah, kampus atau perkantoran yang menerapkan kebijakan-kebijakan khusus untuk instansinya.

Keterbatasan akses ini juga yang terjadi di lingkungan Universitas Brawijaya. Pada saat jam kerja, akses ke situs game online dan jejaring sosial seperti *http://facebook.com* akan diblokir. Hal ini dilakukan agar civitas akademika Universitas Brawijaya tidak lalai dalam menjalankan tugas yang seharusnya. Sayangnya, maksud baik dari Universitas Brawijaya ini terkadang malah membuat mahasiswa mencari-cari cara untuk bisa menembus sistem keamanan yang sudah ada. Menurut Civisec Project, ada banyak teknologi yang bisa digunakan untuk melakukan *bypass* antara lain *Web-Based Circumvention System* dan *Tunneling Software* yang dibagi menjadi 3 teknologi yaitu *Web Tunneling Software*, *Application Tunneling Software* dan *Anonymous Communications System* [CIV-07].

Ada beberapa *tunneling software* yang bisa dengan mudah didapat di internet, antara lain *Hotspot Shield, Ultrasurf, Vidalia, Freegate, Gpass Software, JAP – JonDo* dan masih banyak lagi yang lain. Dengan menggunakan *software proxy* ini, pengguna tetap bisa mengakses layanan yang diinginkan walaupun layanan tersebut merupakan layanan yang diblokir oleh kampus.

Beberapa dari *tunneling software* ini sebenarnya bisa diblok dengan beberapa cara, antara lain melalui *domain name, ip publik* dan *port-port* yang sering digunakan oleh *tunneling software*. Namun, karena *ip publik* dan *port* yang digunakan bersifat *random* maka hal ini akan menyulitkan administrator jaringan untuk membuat daftar *ip address* dan *port-port* yang harus diblokir. Karena itu

diperlukan sebuah sistem untuk bisa mendeteksi penggunaan *tunneling software* ini dengan melihat karakteristik paket dari trafik jaringan yang ada.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, maka dirumuskan masalah sebagai berikut:

1. Bagaimana cara mendeteksi penggunaan *tunneling software*?
2. Bagaimana efektifitas yang dihasilkan oleh sistem untuk mendeteksi penggunaan *tunneling software*?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah dikemukakan, penelitian ini mempunyai batasan-batasan masalah sebagai berikut:

1. Pendeteksian hanya dilakukan pada jaringan Universitas Brawijaya.
2. Metode yang digunakan adalah *naïve bayes classifier*.
3. Pengujian yang dilakukan berupa simulasi.

1.4 Tujuan

Tujuan yang ingin dicapai dalam pembuatan skripsi ini adalah:

1. Membuat klasifikasi karakteristik paket dari pengguna *tunneling software* dan pengguna biasa.
2. Mengetahui efektifitas sistem dalam mendeteksi penggunaan *tunneling software.*

1.5 Manfaat

Penulisan tugas akhir ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun manfaat yang diharapkan adalah sebagai berikut:

1. Bagi Penulis

a. Menambah pengetahuan tentang sistem keamanan di Universitas Brawijaya.

b. Sebagai media untuk pengimplementasian ilmu pengetahuan teknologi khususnya di bidang Keamanan Jaringan.

2. Bagi pembaca

a. Menambah wawasan akan pengimplementasian dari Keamanan Jaringan.

b. Mendapatkan wawasan terkait proses pengklasifikasian paket dan pendeteksian *tunneling software*.

3. Bagi Universitas Brawijaya

a. Mendapatkan tambahan referensi untuk membuat sistem keamanan yang lebih baik.

b. Sebagai evaluasi terhadap kinerja sistem keamanan yang sudah ada sebelumnya.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, sebagai berikut :

BAB I PENDAHULUAN

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika pembahasan.

BAB II KAJIAN PUSTAKA

Menguraikan tentang dasar teori dan referensi yang mendasari Sistem Deteksi Penggunaan *Tunneling Software*.

BAB III METODOLOGI PENELITIAN DAN PERANCANGAN

Menguraikan tentang metode dan langkah kerja yang terdiri dari studi literatur, metode pengambilan data, analisis kebutuhan, pengujian dan analisis serta pengambilan kesimpulan.

BAB IV IMPLEMENTASI

Pada bab ini berisi tentang proses pengklasifikasian pengguna berdasarkan trafik jaringan dan pendeteksian pengguna yang menggunakan *tunneling software* di jaringan Universitas Brawijaya.

BAB V PENGUJIAN DAN ANALISIS

Memuat hasil pengujian dan analisis terhadap sistem deteksi yang telah direalisasikan.

BAB VI PENUTUP

Memuat kesimpulan yang diperoleh dari pendeteksian dan pengujian jaringan yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II
KAJIAN PUSTAKA

2.1 Metode Bypass Sensor Internet

Sensor internet yang dilakukan oleh beberapa negara saat ini menjadi masalah global. Menurut penelitian oleh OpenNet Initiative (http://opennet.net) lebih dari 25 negara saat ini yang terlibat dalam praktik sensor internet. Negara yang paling banyak ditembus kebijakan penyaringannya diketahui bahwa secara rutin memblokir akses ke organisasi hak asasi manusia, berita, blog dan layanan web yang menantang status quo atau dianggap mengancam atau tidak diinginkan. Hal ini membuat kebebasan dari pengguna internet menjadi terbatasi. Oleh karena itu diperlukan teknologi untuk bisa melewati atau membobol sistem penyaringan yang diterapkan [CIV-07].

Alat, metode dan strategi yang digunakan untuk melewati internet *content filtering* disebut sebagai *circumvention technologies*. Ada banyak *circumvention technologies* yang dapat digunakan sesuai keadaan yang berbeda dengan berbagai macam pengguna. Tidak ada satu teknologi yang bisa digunakan untuk semua pengguna dengan keadaan dan kemampuan yang berbeda-beda [CIV-07].

Ada berbagai teknologi yang tersedia untuk pengguna yang ingin menghindari penyaringan internet. Bagaimanapun, menggunakan teknologi tersebut dengan sukses dan stabil tergantung pada berbagai faktor, termasuk tingkat kemampuan teknis dari pengguna, *potential security risk*, dan kontak yang tersedia di luar sensor yuridiksi [CIV-07].

Circumvention technologies ada 2 tipe yaitu :

a. *Web-Based Circumvention Systems*

Web-based circumvention systems adalah halaman web khusus yang memungkinkan pengguna untuk mengirimkan URL dan *web-based circumventor* mengambil halaman web yang diminta. Tidak ada koneksi antara pengguna dan situs web yang diminta seperti permintaan dari *circumventor transparently proxies* yang memungkinkan pengguna untuk menelusuri website yang diblokir tanpa terlihat. Karena alamat web dari *public circumventor* dikenal secara luas,

sebagian besar aplikasi penyaringan internet telah memiliki layanan ini di daftar blokir mereka. *Web-based circumvention system* dapat menjadi pilihan yang baik bagi pengguna yang berhubungan dengan kontak yang tidak terpercaya, dengan asumsi halaman belum terblokir [CIV-07].

b. *Tunneling Software*

Tunneling mengenkapsulasi satu bentuk trafik ke dalam bentuk trafik yang lain. Biasanya data menjadi tidak aman, karena itu trafik yang tidak dienkripsi dilewatkan dalam koneksi yang terenkripsi. Layanan yang normal pada komputer pengguna disediakan, tapi dijalankan melalui *tunnel* ke komputer yang tidak terfilter. Komputer tersebut yang akan meneruskan permintaan pengguna dan responnya secara transparan. Pengguna dengan kontak yang tidak terfilter dapat mengatur layanan *tunneling* pribadi, sementara pengguna tanpa kontak dapat membeli layanan *tunneling* komersial [CIV-07].

2.2 *Tunneling Software*

Tunneling software merupakan teknologi yang paling umum digunakan saat ini untuk melakukan pembobolan internet. Teknologi ini dibagi menjadi 3 jenis, yaitu :

a. *Web Tunneling Software*

Pada *web tunneling software*, *tunneling* terbatas hanya untuk lalu lintas web sehingga web browser akan berfungsi tapi aplikasi lain tidak [CIV-07]. Beberapa contoh dari teknologi ini, antara lain :

- UltraReach

UltraReach telah menciptakan perangkat lunak anti-sensor yang dikenal sebagai UltraSurf. UltraReach menyediakan klien download bagi pengguna windows (instalasi tidak dibutuhkan). Ini adalah perangkat lunak bebas dan tersedia dalam bahasa Inggris dan Cina. Saat dimulai, aplikasi membuka Internet Explorer yang secara otomatis dikonfigurasi agar pengguna bisa menjelajah internet melalui UltraSurf. Browser lain harus dikonfigurasi dengan manual. Secara default, koneksinya terenkripsi dan berbagai teknik yang digunakan untuk menemukan alamat IP diblokir [CIV-07].

- Freegate

Freegate adalah teknologi anti-sensor yang dikembangkan oleh DynaWeb, cara kerjanya hampir sama dengan UltraSurf. Yang berbeda adalah, secara default freegate tidak mengenkripsi URL. Jika pengguna ingin mengenkripsi permintaan URL, mereka harus mendownload paket perangkat lunak yang lain dan khusus menkonfigurasi freegate [CIV-07].

- Anonymizer

Anonymizer menyediakan klien download bagi pengguna windows. Setelah menyelesaikan proses instalasi yang mudah, pengguna bisa menggunakan "Anonymous Surfing ™" pilihan setelah trafik mereka melewati *tunnel* melalui Anonymizer. Namun, untuk memastikan keamanannya pengguna harus mengaktifkan "Surfing Security ™ SSL Encryption" sehingga semua trafik dienkripsi menggunakan HTTPS/SSL. Secara default, pilihan ini dinonaktifkan [CIV-07].

b. *Application Tunneling Software*

Application tunneling software memungkinkan pengguna untuk menggunakan *tunnel* pada beberapa aplikasi seperti email dan pesan instan [CIV-07]. Ada beberapa perangkat lunak yang menggunakan teknologi ini, yaitu :

- GPass

GPass menyediakan klien download bagi pengguna windows. Aplikasi ini adalah perangkat lunak bebas dan tersedia dalam bahasa Inggris dan Cina. Setelah GPass dimulai, ikon aplikasi yang menjadi proxy pada GPass dapat dimasukkan dalam antar muka dari GPass. Saat aplikasi ini dimulai melalui GPass, maka secara otomatis akan dikonfigurasi untuk berjalan melalui layanan ini. Secara default, Internet Explorer, Windows Media Player dan default klien email sudah dikonfigurasi. Koneksinya juga dienkripsi dan berbagai teknik yang digunakan untuk menemukan dan menghubungkan ke alamat IP diblokir. Aplikasi ini menyediakan kecepatan yang wajar dan memiliki kemampuan untuk menyimpan *bookmark* dan file lain secara terenkripsi [CIV-07].

- *HTTP Tunnel*

Seperti *psiphon* dan *Peacefire/Circumventor*, HTTP Tunnel juga menyediakan server dimana pengguna yang berasal dari negara yang tidak memiliki sensor bisa mendownload untuk setup layanan pribadi bagi pengguna

yang berada di negara yang memiliki sensor. HTTP Tunnel dapat digunakan secara gratis, walaupun juga tersedia aplikasi yang berbayar. Untuk menggunakan HTTP Tunnel, pengguna harus mengkonfigurasi manual beberapa aplikasi seperti web browser, klien email dan pesan instan[CIV-07].

- Relakks

Relakks menyediakan layanan berbayar yang disebut Relakks Safe Surf. Aplikasi ini adalah sistem VPN yang menggunakan tunnel yang terenkripsi untuk transportasi trafik dari pengguna melalui server Relakks. Aplikasi ini menggunakan klien asli VPN pada platform Windows dan Mac, sehingga pengguna tidak perlu menginstal perangkat lunak apapun. Banyak aplikasi yang berbeda bisa melewati tunnel menggunakan VPN seperti email, menjelajah web dan pesan instan [CIV-07].

c. *Anonymous Communication Systems*

Teknologi *anonymous* menyembunyikan alamat IP pengguna dari server hosting web situs yang dikunjungi. Beberapa teknologi *anonymous* menyembunyikan alamat IP pengguna dari layanan *anonymizing* itu sendiri dan mengenkripsi trafik antara pengguna dan layanan. Saat pengguna teknologi *anonymous* membuat permintaan untuk isi web melalui layanan proxy, teknologi *anonymous* dapat berguna untuk membobol sensor internet. Namun, ada beberapa teknologi *anonymous* yang mengharuskan pengguna untuk mendownload perangkat lunak yang dengan mudah bisa diblokir oleh pihak yang berwenang [CIV-07]. Teknologi ini digunakan untuk beberapa perangkat lunak, antara lain:

- JAP ANON

JAP ANON menyediakan klien download bagi pengguna windows/mac/ linux. Aplikasi ini tersedia dalam bahasa Inggris dan beberapa bahasa Eropa. Pengguna harus memilih “mix” untuk mencari rute trafik yang akan dilalui kemudian ikuti petunjuk yang tersedia untuk mengkonfigurasi penjelajah web menggunakan JAP ANON. “Mix” adalah perantara yang dilalui untuk permintaan rute dan saat banyak permintaan bergerak melalui mix, maka baik operator mix atau *host* yang diminta melalui mix harus tahu identitas asli dari pengguna [CIV-07].

- TOR

Tor merupakan aplikasi yang gratis, sistem komunikasi *anonymous* yang bekerja dengan me*routing* permintaan web melalui serangkaian *router* yang terenkripsi sehingga tidak ada router di jaringan yang dapat mengidentifikasi sumber atau tujuan dari permintaan tersebut. Tor juga memungkinkan pengguna untuk menggunakan *tunnel* dengan berbagai protokol lain melalui jaringannya, seperti trafik pesan instan dan email. Aplikasi ini juga memiliki fitur yang disebut "*hidden service*" yang memungkinkan pengguna untuk mempublikasikan web mereka sendiri secara anonim yang hanya bisa diakses melalui Tor. Setelah instalasi, layanan Tor dapat langsung digunakan dan pengguna dapat menggunakan browser Firefox yang disukai serta dilengkapi dengan "Torbutton" sehingga Tor dapat dengan mudah dinyalakan atau dimatikan. Jika pengguna menggunakan browser lain, maka diperlukan konfigurasi secara manual [CIV-07].

- I2P

I2P adalah jaringan *anonymization* yang ditujukan bagi pengguna untuk mempublikasikan dan mengakses konten secara anonim melalui I2P. Namun, aplikasi ini juga bisa digunakan untuk berselancar di internet secara anonim. I2P menyediakan klien download bagi pengguna windows/mac/linux. Browser pengguna harus dikonfigurasi secara manual untuk sampai ke tujuannya melalui jaringan I2P [CIV-07].

2.3 Deteksi *Tunneling Software*

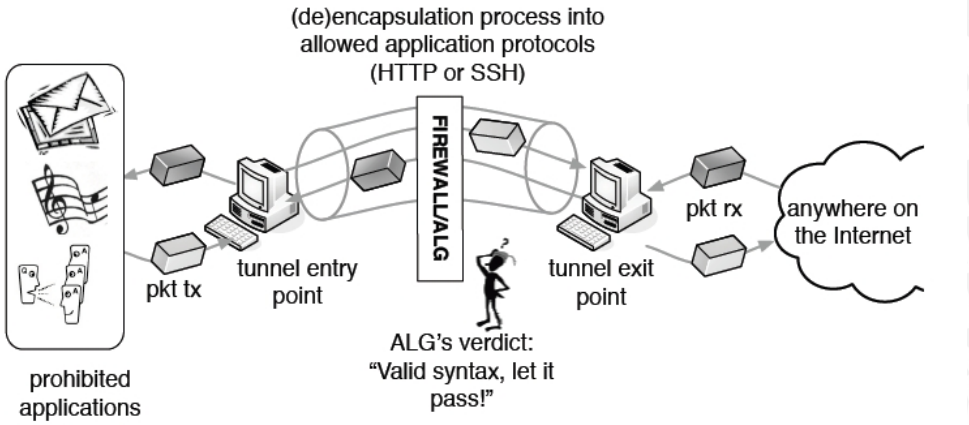
Firewall dan *Application Level Gateways* (ALG) telah digunakan untuk mengamankan batas-batas jaringan selama bertahun-tahun. Pertama, tujuannya adalah untuk mengendalikan situs lokal yang terhubung dengan pengguna dan protokol aplikasi yang akan digunakan. Kedua, aplikasi ini juga mencoba membatasi serangan yang dating dari internet [DUS-08].

Kebijakan keamanan dilakukan oleh *firewall* dan ALG dimulai dengan mendefinisikan protokol pada layer aplikasi yang diperbolehkan, dan alamat tujuan yang dapat dihubungi melalui aplikasi tersebut. Kedua jenis perangkat tersebut kemudian bekerjasama untuk menerapkan kebijakan yang telah dibuat: *firewall* memeriksa port TCP dan alamat tujuan, sementara ALG memverifikasi bahwa trafik yang melintasi jaringan telah sesuai dengan kebijakan dan tidak

berbahaya. Selain itu, ALG juga memverifikasi isi dari layer aplikasi pada saat koneksi sudah sesuai dengan kebijakan keamanan yang diinginkan [DUS-08].

Saat ini telah dirancang beberapa teknik untuk menyamarkan satu protokol dengan protokol yang lain, dengan tujuan untuk membobol kebijakan keamanan pada suatu jaringan. Teknik ini tergantung pada tunneling dari satu protokol di layer aplikasi yang satu dengan yang lain, yaitu pada dasarnya mereka mengenkapsulasi trafik yang dilarang ke dalam *payload* dari protokol aplikasi yang diperbolehkan. Teknik *tunneling* lainnya adalah dengan mengimplementasikan *Secure Shell* (SSH), yang dapat dikonfigurasi untuk melindungi trafik TCP antara klien SSH dan server SSH dengan cara kriptografi [DUS-08].

Tujuan dari mekanisme *tunneling* adalah untuk menyamarkan aplikasi protokol tertentu menjadi aplikasi protokol yang lain. Untuk menghindari ALG, proses menghasilkan aliran paket pada layer aplikasi, yang sama persis dengan protokol yang diperbolehkan. Terlepas dari protokol yang digunakan sebagai *tunnel*, mekanisme *tunneling* pada dasarnya bekerja dengan model klien-server: *host* klien yang berada di jaringan yang dilindungi berhubungan dengan server luar menggunakan aplikasi protokol yang diperbolehkan oleh kebijakan keamanan jaringan. Kemudian, setiap titik akhir menyediakan de-enkapsulasi dari protokol *tunnel* dan meneruskan data asli ke server yang sebenernya atau klien yang terlibat dalam komunikasi [DUS-08]. Secara umum, teknik *tunneling* dapat dilihat pada skema dibawah ini:



Gambar 2.1 Cara Kerja Tunnel : Skema *High-Level*

Sumber : [DUS-08]

Setidaknya ada tiga protocol yang digunakan untuk *tunnel* di trafik internet pada layer aplikasi, yaitu DNS, HTTP dan SSH.

a. DNS *Tunnel*

Tunnel di layer aplikasi dapat dibangun di atas DNS dengan hanya merubah merubah cara permintaan DNS biasa untuk domain yang diberikan dan diteruskan ke server asli. *Entry point* memecah paket IP yang akan melewati *tunnel* yang diminta untuk domain dari otoritas server yang merupakan *exit point* : server DNS lain (*resolvers*) diletakkan antara jaringan yang dilindungi dengan internet yang akan meneruskan permintaan tersebut ke *tunnel exit point*, pada saatnya paket yang asli akan dikumpulkan kembali dan dikirim ke tujuan mereka yang sebenarnya. DNS *respons* kemudian akan digunakan untuk mengirimkan paket pada arah yang berlawanan yaitu ke *tunnel entry point*. Karena protokol DNS jarang diblokir pada internet, teknik ini bisa sangat *powerful* [DUS-08].

b. HTTP *Tunnel*

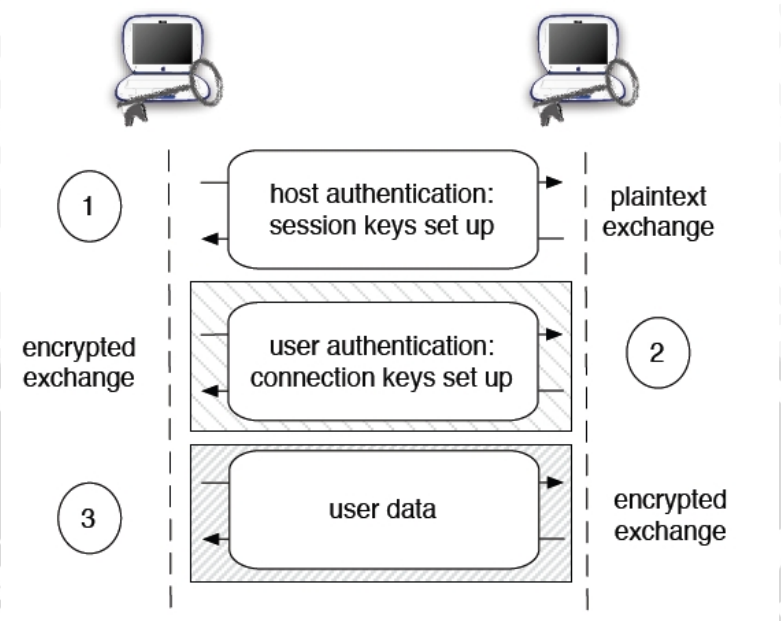
Administrator jaringan biasanya membiarkan trafik HTTP melewati jaringan mereka, walaupun biasanya disaring melalui aplikasi-aplikasi *proxy* dan *firewall*. Namun, baik pengguna yang sah dan perangkat lunak jaringan yang berbahaya (virus dan *malware*) dapat melanggar kebijakan ini dengan menggunakan protokol HTTP di aplikasi *tunnel* yang melalui jaringan, seperti digambarkan dalam gambar 2.1 bahwa aliran paket di dalam *tunnel* di*encoded* sehingga mereka dapat

tergabung secara teratur. Sebuah analisis tentang *payload* dari setiap *session*, jika dilakukan dengan cara pencocokan pola, tidak bisa menemukan perbedaan antara trafik yang menggunakan *tunnel* dengan trafik HTTP asli yang dihasilkan oleh penjelaj web dan server web yang sebenarnya [DUS-08].

Salah satu teknik HTTP *tunneling* yang paling umum digunakan adalah *httptunnel*. Alat ini memberikan sepasang *daemons* yang berjalan di *tunnel end-point*. Di sisi masuk, *client httptunnel* menunggu koneksi TCP pada *port* yang dikonfigurasi. Ketika sambungan dibuat, terjadi inisiasi HTTP *session* yang menuju *server httptunnel* yang berjalan di *tunnel exit point* [DUS-08].

c. SSH *Tunnel*

Protokol SSH berjalan di atas TCP dan dirancang untuk memberikan kerahasiaan data dan integritas antara dua *host* melalui jaringan yang tidak aman. Protokol ini biasanya digunakan untuk eksekusi perintah melalui *shell* yang aman dan penggandaan file yang aman antar *peer*. Protokol ini juga mendukung *tunneling* dari koneksi TCP, yang disebut *port forwarding*. SSH *tunneling* melindungi protokol dengan kriptografi, meningkatkan keamanan data dan sistem. Namun, sebagai akibat dari enkripsi data, setiap kebijakan keamanan jaringan yang mengandalkan teknik DPI dinetralkan secara total. Sementara dalam kasus HTTP *tunnel* yang kemungkinan menggunakan ALG yang lebih maju, berbasis *deep-packet inspection*, bisa memastikan sifat sebenarnya dari trafik yang berada di *tunnel*, pada kasus SSH penelitian terhadap *payload signature* menjadi tidak berguna, membuat *tunnel* yang dibangun di atas SSH menjadi sangat *powerful* [DUS-08].



Gambar 2.2 Tahap Autentikasi di SSH

Sumber : [DUS-08]

SSH *session* melibatkan dua tahap autentikasi sebelum *client* dan *server* dapat memulai pertukaran data : 1) autentikasi *host* dan 2) autentikasi pengguna. Secara umum, proses autentikasi bisa dilihat pada gambar 2.2.

Autentikasi *host* memberikan enkripsi yang kuat, autentikasi kriptografi dari *host* dan perlindungan integritas. Metode pertukaran kunci, algoritma *public key*, algoritma *symmetric encryption*, algoritma *message authentication* dan algoritma *hash* semuanya dinegosiasikan. Dengan harapan bahwa lingkungan hanya 2 *round-trip* saja yang dibutuhkan untuk pertukaran kunci secara penuh, autentikasi server, permintaan layanan dan penerimaan pemberitahuan terhadap permintaan layanan : dalam kasus terburuk, sebuah *round-trip* tambahan dapat diminta. Fase autentikasi ini ditransmisikan dengan tanpa enkripsi dan berakhir dengan pesan SSH MSG NEWKEYS. Semua pesan baru bisa dikirim setelah menggunakan *negotiated keys* dan algoritma, dan privasi serta integritas pun bisa dilindungi [DUS-08].

Autentikasi pengguna, dimaksudkan untuk dijalankan diatas layer *transport* dari protokol SSH, yaitu saluran yang terenkripsi didapat dari fase autentikasi *host*. *Public-key* adalah satu-satunya metode autentikasi yang wajib

digunakan, meskipun paswordnya juga diterima. Autentikasi password yang sukses di SSH membutuhkan *single round-trip* : *client* mengirimkan password ke *server*, yang kemudian membalas dengan sebuah ACK atau NACK. Pada kasus terakhir, *client* memiliki peluang lain untuk mengirim ulang password yang benar. Metode *public-key* membutuhkan satu atau dua *round-trip* : spesifikasi telah didefinisikan pada pertukaran awal dimana *client* bisa mengirimkan informasi *public key* ke *server* sebelum mengirimkan *signature* dengan *private key* miliknya [DUS-08].

2.4 Metode Klasifikasi Trafik Internet

Pada abad ke-21, jumlah pengguna internet meningkat secara signifikan. Para pengguna menggunakan beberapa aplikasi internet seperti WWW, FTP perangkat lunak berbasis *Peer-to-peer*, media web, pesan, email, VOIP dll. Hal ini menyebabkan peningkatan yang cepat pada trafik internet. Klasifikasi trafik internet menawarkan tiga fungsi utama yaitu untuk administrator jaringan, *Internet Service Provider* (ISP) dan pemerintah: Pertama, klasifikasi paket dapat digunakan dalam *intrusion detection system* (IDS) untuk mendeteksi pola dari *denial of service* (DoS) atau serangan berbahaya lainnya. Hal ini juga dapat digunakan oleh administrator untuk mengidentifikasi dan mengendalikan aplikasi jaringan bila diperlukan. Kedua, dapat digunakan oleh ISP untuk memantau arus jaringan, mendiagnosa jaringan untuk menemukan kesalahan, mengalokasikan bandwidth untuk aplikasi dan memastikan kinerja aplikasi dan layanan yang berjalan pada jaringan. Ketiga, dapat digunakan oleh pemerintah untuk melakukan "Lawful Inspection" (LI) dari *payload* paket untuk mendapatkan informasi pengguna. Sama seperti bagaimana perusahaan telepon menawarkan untuk memantau panggilan telepon kepada pemerintah, ISP menyediakan layanan LI untuk pemerintah [LIU-12].

Karakteristik trafik internet telah menjadi tantangan selama beberapa tahun terakhir. Hal ini membutuhkan pemahaman mendalam tentang struktur protokol jaringan yang canggih, karena ada berbagai jenis trafik untuk ISP serta volume yang besar dari *stream flows*. Dengan jumlah bandwidth dan layanan yang meningkat, pengguna dapat melakukan aktifitas yang jauh lebih rumit daripada

sebelumnya [LIU-12]. Metode klasifikasi trafik internet dibedakan menjadi tiga, yaitu klasifikasi berbasis *port*, klasifikasi berbasis *payload*, klasifikasi berbasis statistik.

2.4.1 Klasifikasi Berbasis *Port*

Cara paling mudah untuk mengklasifikasi trafik internet adalah dengan menggunakan nomor *port* UDP atau TCP. Alasannya adalah karena beberapa trafik menggunakan nomor *port*, dan nomor *port* tersebut dapat ditemukan pada *Internet Assigned Number Authority* (IANA). Contohnya, HTTP menggunakan *port* 80, POP3 menggunakan *port* 110 dan SMTP menggunakan *port* 25. Kita dapat mengatur aturan untuk mengklasifikasi aplikasi yang ditugaskan ke nomor *port*. Namun, banyak penelitian menyebutkan klasifikasi berbasis *port* itu tidak cukup. Moore dan Papagiannaki menyebutkan bahwa akurasi dari klasifikasi berbasis *port* adalah sekitar 70% selama penelitian yang mereka lakukan. Selain itu, Madhukar dan Williamson menyatakan dalam penelitian mereka bahwa kesalahan dalam klasifikasi berbasis *port* ini berkisar antara 30% dan 70%. Alasan utama untuk memilih nomor *port* yang statis adalah untuk membuat paket mampu melewati *server firewall*. Baru-baru ini banyak aplikasi yang mencoba untuk menghindari deteksi *firewall* dengan menyembunyikan nomor *port*. Beberapa aplikasi yang lain menggunakan nomor *port* dinamis, bukan statis. Dan server yang membagi alamat IP akan menggunakan nomor *port* yang bukan standar [LIU-12].

2.4.2 Klasifikasi Berbasis *Payload*

Pendekatan lain untuk mengklasifikasi paket adalah dengan menganalisa *payload* paket atau menggunakan teknologi *deep packet inspection* (DPI). Metode ini mengklasifikasi paket berdasarkan *signature* pada *payload* paket, dan telah disebut-disebut sebagai metode klasifikasi yang paling akurat, dengan hasil 100% benar dari semua paket yang diklasifikasikan jika *payload* tidak dienkripsi. *Signature* adalah string unik dalam *payload* yang membedakan target paket dengan trafik paket yang lain. Setiap protokol memiliki cara yang berbeda untuk berkomunikasi dengan protokol lain yang berbeda. Ada pola komunikasi dalam

payload dari paket. Kita dapat mengatur aturan untuk menganalisa *payload* paket untuk mencocokkan pola komunikasi untuk mengklasifikasikan aplikasi. Tapi ada beberapa masalah yang muncul yaitu pengguna dapat mengenkripsi *payload* untuk menghindari deteksi dan beberapa negara juga melarang melakukan inspeksi *payload* untuk melindungi informasi privasi dari pengguna. Selain itu, *classifier* akan mengalami beban operasional yang berat karena harus terus-menerus memperbaharui aplikasi *signature* untuk memastikan bahwa aplikasi tersebut berisi *signature* dari semua aplikasi terbaru [LIU-12].

2.4.3 Klasifikasi Berbasis Statistik

Karena keterbatasan klasifikasi berbasis *port* dan *payload*, baru-baru ini penelitian berfokus untuk menggunakan perilaku statistik dari layer *transport* dan *flow* untuk klasifikasi paket. Pendekatan ini menggunakan sekumpulan contoh *trace* trafik untuk melatih mesin klasifikasi untuk mengidentifikasi trafik di masa mendatang berdasarkan perilaku aplikasi *flow*, seperti panjang paket, waktu kedatangan antar paket, *flag* dan *checksum*. Target mereka adalah untuk mengklasifikasi trafik dengan pola yang sama dalam grup, atau mengklasifikasi trafik dalam aplikasi individu. Namun, akurasi dari klasifikasi trafik yang dienkripsi menggunakan pendekatan berbasis statistic relatif rendah, bervariasi dari 76% sampai 86% dengan tingkat *false positive* antara 0% sampai 8% berdasarkan pengaturan yang berbeda. Banyak peneliti menggunakan *Machine Learning* (ML) untuk melakukan klasifikasi berbasis statistic. Alasan memilih ML adalah karena dapat secara otomatis membuat *signature* untuk aplikasi dan mengidentifikasi aplikasi dalam trafik di masa mendatang. Alasan lain memilih ML adalah karena memiliki kemampuan untuk secara otomatis memilih fitur yang paling tepat untuk membuat *signature* [LIU-12]. Teknik ML terdiri dari banyak langkah, yaitu :

1. Menentukan fitur yang berhubungan dengan trafik. Kemungkinan fitur ini termasuk ukuran paket atau waktu kedatangan antar paket.
2. Menetapkan jenis aplikasi sebagai contoh.

3. Memilih contoh *trace* dari aplikasi untuk melatih mesin klasifikasi agar menghasilkan aturan, dan menggunakan algoritma ML untuk mengklasifikasikan trafik di masa depan.

2.5 Metode Klasifikasi Paket Data

Klasifikasi paket adalah proses menghubungkan paket, yang bercampur dengan paket lain secara acak, dengan aplikasi yang menghasilkan paket. Tugas yang paling menantang adalah menemukan hubungan antara *source packet* dengan paket yang dihasilkan oleh aplikasi yang menjadi target. Pendekatan klasifikasi membutuhkan fase pelatihan untuk menghubungkan antara pola dengan aplikasi. Fase pelatihan membutuhkan contoh dataset yang telah diklasifikasikan ke dalam trafik yang menjadi target. Oleh karena itu, pendekatan klasifikasi bekerja lebih baik ketika mengklasifikasi satu aplikasi atau sekelompok aplikasi. Namun, pendekatan ini memiliki batasan. *Classifier* harus dilatih dengan semua pola yang muncul di trafik yang dihasilkan oleh aplikasi. Jadi kinerja dari pendekatan klasifikasi ini sangat tergantung pada tahap pelatihan. Jika fase pelatihan ini mencakup semua kemungkinan, maka akurasinya akan tinggi. Dan jika tahap pelatihan tidak mencakup semua kemungkinan, maka akurasinya menjadi rendah [LIU-12].

Mesin klasifikasi mengambil file *trace* yang tidak diketahui sebagai masukan, dan kemudian mengidentifikasi keberadaan jenis yang emnjadi target dalam file *trace*. Keluarannya harus "ya" jika trafik termasuk dalam jenis target, dan "tidak" jika trafik tidak termasuk pada jenis target. Kunci membedakan teknik klasifikasi yang baik dan buruk adalah akurasi klasifikasi. Untuk itu perlu mempertimbangkan metric berikut : *false positive*, *false negative*, *true positive*, *true negative*, *recall* dan *presisi*. Thuy Nguyen T.T. dan Grenville Armitage telah memberikan definisi mereka [LIU-12].

- *False Negative* : Persentase jenis target yang salah diklasifikasikan sebagai yang lain.
- *False Positive* : Persentase jenis trafik yang diklasifikasikan sebagai jenis yang ditargetkan.

- *True Positive* : Persentase trafik yang diklasifikasikan dengan benar sebagai jenis yang ditargetkan.
- *True Negative* : Persentase trafik lainnya dengan benar tidak diklasifikasikan sebagai jenis yang ditargetkan.
- *Recall* : Persentase trafik dengan benar diklasifikasikan sebagai jenis yang ditargetkan.
- *Presisi* : Persentase contoh trafik yang benar-benar memiliki jenis yang ditargetkan, diantara semua yang diklasifikasikan sebagai jenis yang ditargetkan.

Metode untuk klasifikasi paket data sendiri dibedakan menjadi 3, yaitu pendekatan berbasis *signature*, pendekatan *naïve bayes estimator* dan pendekatan berbasis *fingerprints*.

2.5.1 Pendekatan Berbasis *Signature*

Roughan telah melakukan penelitian tentang penggunaan *quadratic discriminat analysis, nearest neighbours* dan *linear discriminate analysis* untuk mengklasifikasi aplikasi yang berbeda. Penulis menggunakan fitur yang bermacam-macam untuk mengatur aturan klasifikasi. Fitur tersebut dikategorikan menjadi lima kategori. Dalam fitur *packet level*, penulis menggunakan fitur seperti ukuran paket dan *root mean square size*. Untuk fitur *flow level*, mereka menggunakan fitur seperti *mean flow duration* dan *mean number of packet*. Pada fitur *connection level*, fitur yang digunakan termasuk *advertisement window sizes* serta fitur-fitur yang digunakan pada *flow-level*. Fitur *intra-flow*/*connection* menggunakan *inter arrival time, latencies loss rate* dan sebagainya. Untuk kategori *multi-flow*, fitur yang digunakan lebih rumit daripada kategori yang lain. Kategori ini lebih berguna untuk aplikasi P2P yang menggunakan beberapa koneksi menuju *end-system* untuk mendownload file. Diantara semua fitur yang disebutkan, mereka menggunakan *duration* dan *average packet length* sebagai fitur yang paling berharga [LIU-12].

2.5.2 Pendekatan Berbasis *Naïve Bayes Estimator*

Moore dan Zuev menggunakan estimator *naïve bayes* untuk mengklasifikasi trafik dalam aplikasi yang berbeda. Berbeda dengan pendekatan yang lain, mereka menggunakan dataset yang telah diklasifikasikan untuk membuat pengujian yang dilakukan menjadi lebih akurat.

Mereka menggunakan *trust* dan *accuracy by bytes* untuk mengevaluasi hasil pengujian mereka. *Trust* merepresentasikan seberapa baik anda bisa mempercayai klasifikasi. *Accuracy-by-bytes* adalah persentase dari *flow bytes* yang diklasifikasikan dengan benar.

2.5.3 Pendekatan Berbasis *Fingerprints*

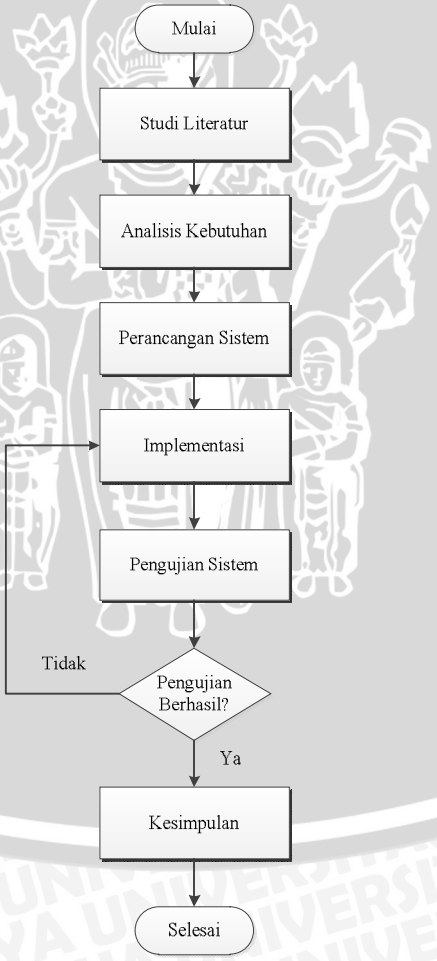
Crotti mengusulkan metode klasifikasi menggunakan panjang paket, urutan paket dan waktu kedatangan. Fitur-fitur ini telah digunakan oleh peneliti lain, tetapi perbedaannya adalah struktur dari fitur yang disebut *fingerprints*. *Fingerprints* adalah cara yang lebih efisien dan terstruktur untuk mengatur fitur. Mereka menggunakan algoritma *normalized threshold* dalam penelitiannya.

BAB III
METODOLOGI PENELITIAN DAN PERANCANGAN

Pada bab ini akan dijelaskan tentang metodologi penelitian, perancangan dan analisis kebutuhan dari sistem yang akan dibuat. Penelitian ini dilakukan pada jaringan di Universitas Brawijaya.

3.1 Metode Penelitian

Secara umum, metodologi penelitian yang akan dilakukan pada penelitian ini dapat dilihat pada diagram alir berikut ini:



Gambar 3.1 Diagram Alir Keseluruhan Proses Penelitian

3.1.1 Studi Literatur

Studi literatur digunakan untuk mempelajari dan memahami konsep yang dibutuhkan dalam mengerjakan penelitian dan penulisan laporan penelitian. Studi literatur yang diperlukan adalah dasar-dasar teori untuk dapat mengetahui karakteristik pengguna *tunneling software* dan cara pengujian sistem. Dasar-dasar teori tersebut meliputi :

a. Karakteristik paket dari pengguna *tunneling software*
b. Wireshark
c. Metode yang digunakan untuk mendeteksi

3.1.2 Analisis Kebutuhan

Analisis kebutuhan berguna untuk melakukan pendataan tentang kebutuhan sistem. Pada tahap ini akan dilakukan identifikasi perangkat keras dan perangkat lunak yang digunakan pada sistem yang akan dirancang. Proses identifikasi ini bertujuan untuk mempermudah dalam mendesain sistem dan analisis yang dilakukan terhadap sistem bisa lebih akurat.

3.1.2.1 Analisa Perangkat Keras

Perangkat keras yang akan digunakan dalam penelitian ini dapat dilihat pada tabel berikut:

Tabel 3.1 Kebutuhan Perangkat Keras

Perangkat Keras	Pengguna	Detektor
CPU	Processor Intel Core i5	Processor Intel Core i5
Memory	4 GB	4 GB
Sistem Operasi	Microsoft Windows 7	Microsoft Windows 8

3.1.2.2 Analisa Perangkat Lunak

Kebutuhan perangkat lunak yang akan digunakan dalam penelitian ini dibagi menjadi 2 jenis, yaitu kebutuhan perangkat lunak untuk pengguna dan kebutuhan perangkat lunak untuk detektor.

1. Kebutuhan Perangkat Lunak untuk Pengguna

Perangkat lunak yang dibutuhkan oleh pengguna adalah sebagai berikut :

a. Browser

Browser adalah suatu program atau perangkat lunak yang digunakan untuk menjelajahi internet. Dalam penelitian ini, browser digunakan oleh pengguna untuk mengakses internet melalui jaringan di Universitas Brawijaya.

b. Ultrasurf

Ultrasurf merupakan salah satu jenis *tunneling software* yang banyak digunakaan saat ini. Program ini bekerja dengan sistem *tunneling* atau mencari celah dari jaringan lokal untuk dapat menghindari blocking jaringan tersebut. Ultrasurf dipilih sebagai salah satu *tunneling software* yang digunakan dalam penelitian ini karena aplikasi ini menggunakan sistem *tunneling* untuk menghindari pemblokiran.

c. Hotspot Shield

Hotspot Shield adalah sebuah aplikasi *tunneling software* yang biasa digunakan untuk *anonymous browsing*. Aplikasi ini membangun jaringan pribadi yang menghubungkan perangkat komputer pengguna langsung ke *internet gateway* dengan cara dengan cara membobol *firewall*. Dengan cara kerja aplikasi ini yang menggunakan VPN, pengguna tidak akan merasakan dampak dari pemblokiran yang ada. Hotspot Shield digunakan dalam penelitian ini karena aplikasi ini menggunakan sistem VPN untuk membobol pemblokiran yang ada.

d. Freegate

Freegate adalah salah satu perangkat lunak yang dapat digunakan untuk membobol alamat IP yang berada dibawah sebuah *proxy* tertentu. Freegate dikembangkan dan dikelola oleh perusahaan Dynamic Internet Technology (DIT). Freegate bekerja dengan menyadap *backbone* anti sensor dari DynaWeb, yaitu produk lain dari DIT yang bekerja sebagai P2P bagi DIT seperti sistem jaringan *proxy*.

e. ChrisPC Anonymous Proxy

Aplikasi ini biasa digunakan untuk *anonymous browsing*. Cara kerja perangkat lunak ini adalah dengan menggunakan salah satu jaringan *proxy* yang paling kuat di dunia. Jenis jaringan tersebut terdiri dari ratusan *proxy* server dari seluruh dunia yang dapat dibentuk untuk membuat *user* menjadi anonim. Dalam

penelitian ini, aplikasi ChrisPC Free Anonymous Proxy digunakan untuk menguji sistem.

f. ExpatShield

Tunneling software ini tidak memberikan batasan bandwidth atau bisa disebut *unlimited*. Karena bersifat gratis, ketika digunakan aplikasi ini akan menampilkan iklan-iklan yang cukup mengganggu. Dalam penelitian ini, ExpatShield digunakan sebagai salah satu perangkat lunak penguji.

g. CyberGhost VPN

Aplikasi ini menjadi salah satu layanan VPN populer saat ini. CyberGhost VPN adalah premi layanan VPN yang digunakan untuk menjelajah web secara anonym melalui koneksi SSL yang sangat aman dengan menggunakan enkripsi SSL 1024-bit. Perangkat lunak ini menjadi salah satu perangkat lunak penguji dalam penelitian ini.

2. Kebutuhan Perangkat Lunak untuk Detektor

Kebutuhan perangkat lunak yang digunakan untuk detektor adalah :

a. Wireshark

Wireshark merupakan perangkat lunak untuk melakukan analisa lalu lintas jaringan komputer. Dalam penelitian ini, wireshark digunakan untuk melihat IP Source, IP Destination dan Protokol yang digunakan oleh pengguna.

b. Xampp

Xampp merupakan perangkat lunak untuk windows yang terdiri dari beberapa layanan diantaranya adalah Apache, MySQL dan PHP. Untuk membuat sebuah web di komputer maka dibutuhkan sebuah server web. Salah satu server web yang bisa digunakan adalah Xampp. Xampp menyediakan berbagai macam layanan salah satunya adalah Apache untuk server web [UPT-13].

c. phpMyAdmin

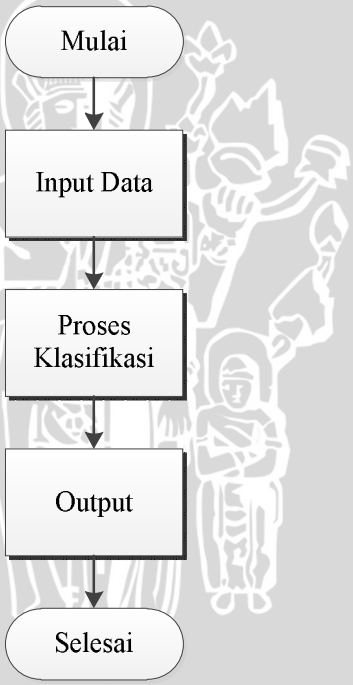
phpMyAdmin adalah sebuah aplikasi/perangkat lunak *opensource* yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi database MySQL melalui jaringan lokal maupun internet [HAK-13]. phpMyAdmin digunakan sebagai database untuk menampung data- data trafik jaringan dari pengguna.

d. Macromedia Dreamweaver 8

Macromedia Dreamweaver 8 atau biasa disebut Dreamweaver 8 adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman web [SGS-13]. Aplikasi ini digunakan untuk membuat program berbasis PHP untuk melakukan perhitungan *naïve bayes classifier* dan membuat antar muka.

3.2 Perancangan Sistem

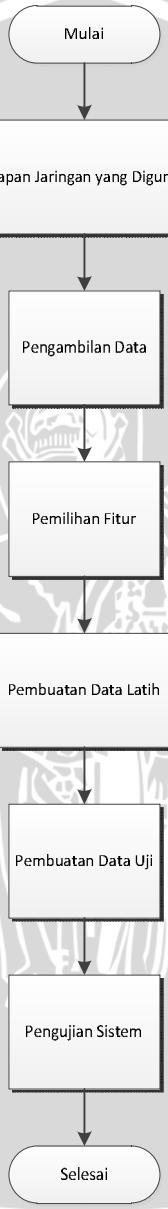
Perancangan sistem akan memberikan gambaran tentang sistem yang akan dibuat dalam penelitian ini. Perancangan sistem bertujuan sebagai acuan dalam implementasi sistem dan untuk melakukan analisis kebutuhan yang akan dipergunakan dalam penelitian. Secara umum, alur dari sistem dapat dilihat pada bagan dibawah ini:



Gambar 3.2 Diagram Perancangan Sistem Secara Umum

Pada diagram diatas dapat dilihat bahwa terdapat 3 langkah utama yang dilakukan untuk mendeteksi penggunaan *tunneling software* yaitu input data, proses klasifikasi dan output. Input data disini adalah proses pemasukan data uji yang akan diklasifikasi. Proses klasifikasi adalah proses untuk melakukan klasifikasi data berdasarkan metode *naïve bayes classifier*. Sedangkan output

adalah hasil dari klasifikasi yang telah dilakukan. Hasil output tersebut ada 2 jenis yaitu pengguna normal dan pengguna *tunneling software*. Untuk lebih jelas tentang perancangan sistem secara lebih mendetail, dapat dilihat pada diagram dibawah ini:

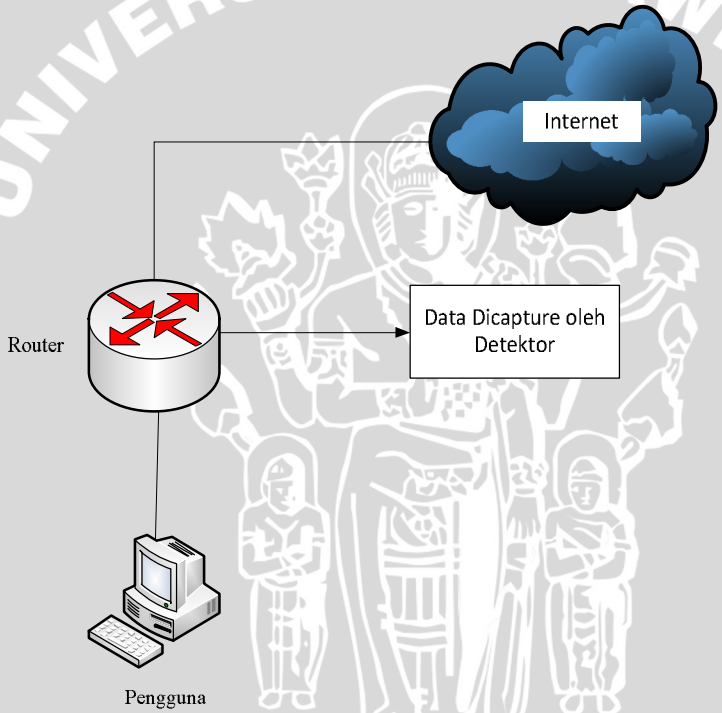


Gambar 3.3 Diagram Perancangan Sistem Secara Detail

3.2.1 Topologi Jaringan

Penelitian ini dilakukan pada jaringan LAN yang terdiri dari *router* dan

minimal 2 *host* yaitu satu host sebagai pengguna dan satu host sebagai detektor. Komputer yang bertindak sebagai pengguna ini bisa *browsing* dengan menggunakan *tunneling software* atau menggunakan request HTTP biasa. Selanjutnya paket data yang dikirimkan oleh pengguna akan dideteksi pada komputer yang bertindak sebagai detektor. Komputer ini akan mendeteksi pengguna tersebut menggunakan *tunneling software* atau tidak. Jika pengguna tersebut terdeteksi sebagai pengguna *tunneling software*, maka koneksinya akan langsung diputus. Secara umum, topologi sistem dapat digambarkan seperti berikut ini:

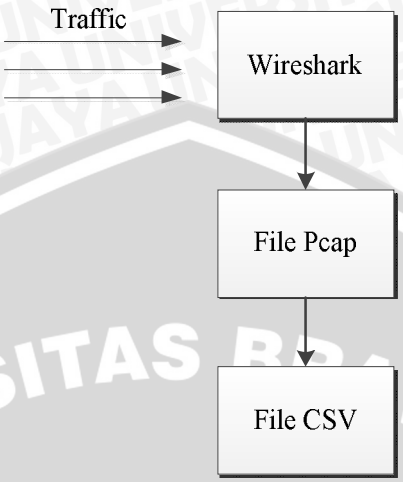


Gambar 3.4 Topologi Sistem

3.2.2 Pengambilan Data

Pengambilan data dilakukan dengan cara menangkap paket yang masuk dari setiap *host* menggunakan Wireshark. File yang ditangkap oleh wireshark akan ditulis dalam file pcap. File pcap merupakan file *binary* dari wireshark yang tidak dapat dibaca oleh aplikasi lain. Agar dapat dibaca maka file pcap tersebut harus dikonversi menjadi file CSV (*Comma Separated Values File*). Diagram untuk

pengambilan data dapat dilihat seperti dibawah ini:

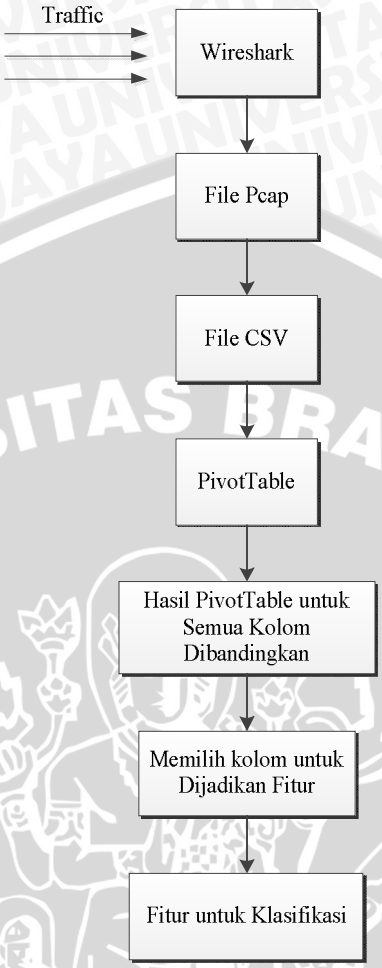


Gambar 3.5 Diagram Pengambilan Data

3.2.3 Pemilihan Fitur

Untuk melakukan klasifikasi, diperlukan fitur-fitur yang bisa dijadikan sebagai acuan. Fitur dipilih jika memiliki perbedaan yang jelas antara pengguna *tunneling software* dengan pengguna normal. Dengan pemilihan fitur yang benar, maka klasifikasi yang dilakukan bisa semakin valid.

Cara untuk menentukan fitur adalah dengan melakukan filterisasi. Data yang telah didapat sebelumnya dan sudah dikonversi menjadi file csv akan difilter menggunakan fasilitas *PivotTable* dari Microsoft Excel. Semua data dari kolom yang dimasukkan ke dalam *PivotTable*, kemudian dibandingkan dan dicari kolom mana yang memiliki perbedaan paling jelas. Setelah didapat kolom mana saja yang memiliki perbedaan antara pengguna normal dan pengguna *tunneling software*, maka kolom tersebut yang akan dijadikan sebagai fitur. Pada saat klasifikasi, fitur inilah yang akan dijadikan patokan dalam menentukan karakteristik dari pengguna normal dan pengguna *tunneling software*. Untuk lebih jelas, skema pemilihan fitur dapat dilihat pada bagan dibawah ini:



Gambar 3.6 Diagram Pemilihan Fitur

Secara umum, fitur yang diperlukan seperti tampak pada tabel dibawah ini:

Tabel 3.2 Fitur untuk Klasifikasi

No.	Nama Fitur	Tipe
1.	IP Source	Varchar (39)
2.	IP Destination	Varchar (39)
3.	Protokol	Varchar (8)

3.2.4 Pembuatan Data Latih

Pada penelitian ini, data latih didapat dengan mengakses 14 website yang berbeda dengan menggunakan 2 cara yaitu secara normal dan menggunakan

tunneling software. Ada 3 buah *tunneling software* yang digunakan yaitu Hotspot Shield, Ultrasurf dan Freegate. Berikut ini daftar website yang digunakan untuk data latih:

Tabel 3.3 Daftar Nama Website untuk Data Latih

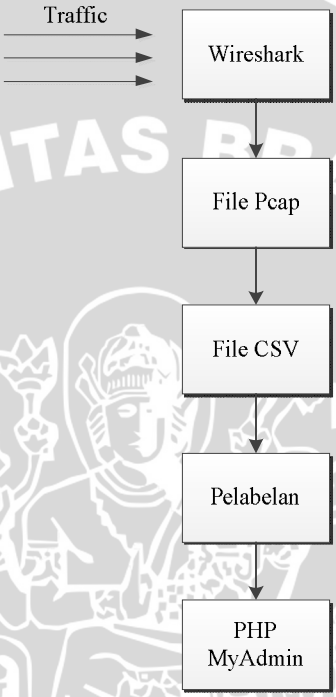
No.	Nama Website
1.	www.facebook.com
2.	http://id.yahoo.com
3.	www.detik.com
4.	www.kompas.com
5.	www.okezone.com
6.	http://ub.ac.id
7.	http://ptiik.ub.ac.id
8.	http://jpc.ub.ac.id
9.	www.waptrick.com
10.	http://cilpie.esy.es
11.	www.gudanglagu.com
12.	http://prasetya.ub.ac.id
13.	http://selma.ub.ac.id
14.	www.mytrans.detik.com

Tabel 3.4 Jenis Data Latih

No.	Jenis Data Latih	Jumlah Pengguna Normal	Jumlah Pengguna *Tunneling software*	Total
1.	Ultrasurf	14	14	28
2.	Hotspot Shield	14	14	28
3.	Freegate	14	14	28

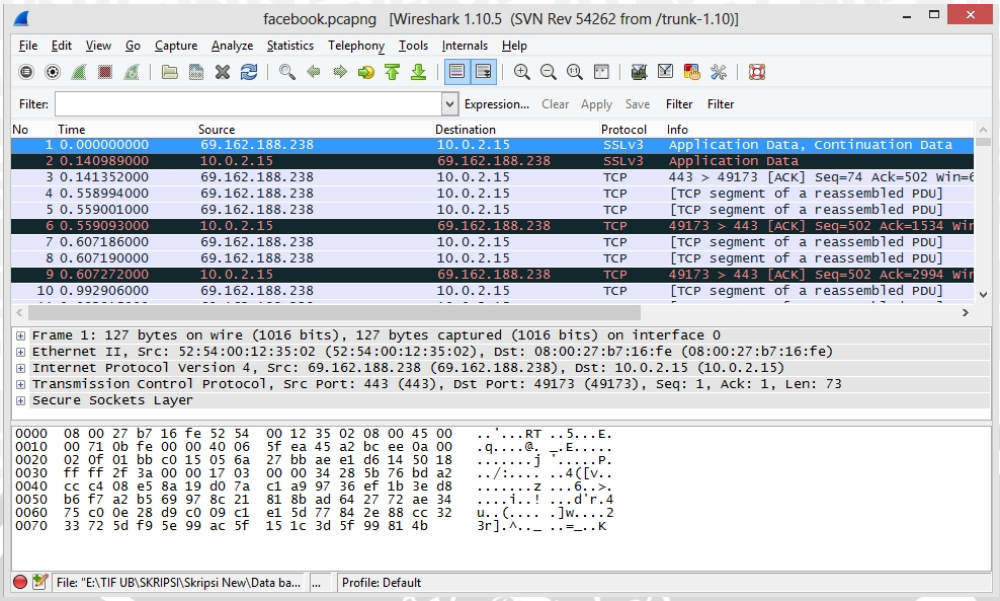
Cara untuk membuat data latih hampir sama dengan pengambilan data sebelumnya, yaitu data dari pengguna normal dan pengguna *tunneling software* ditangkap menggunakan wireshark. Kemudian data tersebut dikonversi ke dalam

file csv. Untuk data latih, dilakukan pelabelan data sebelum dimasukkan dalam database. Untuk data dari pengguna normal akan diberi label ‘0’ sedangkan untuk data dari pengguna *tunneling software* diberi label ‘1’. Setelah mendapat label, data tersebut dimasukkan dalam database. Secara umum, proses pembuatan data latih dapat dilihat pada bagan dibawah ini:



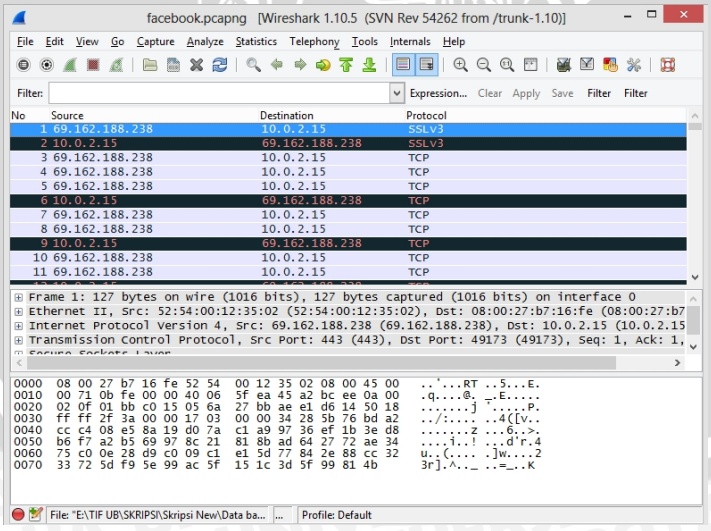
Gambar 3.7 Diagram Pembuatan Data Latih

Untuk memudahkan pembuatan data latih, antar muka dari wireshark pun disesuaikan dengan fitur yang ada. Hanya kolom dari fitur seperti IP Source, IP Destination dan Protocol yang akan ditampilkan. Hal ini dilakukan untuk mempermudah proses sterilisasi data, yaitu menghilangkan duplikasi yang berdasarkan label yang telah diberikan.



Gambar 3.8 Tampilan Wireshark Secara Umum

Tampilan wireshark tersebut akan diubah sesuai dengan fitur yang dibutuhkan. Antar muka wireshark setelah dirubah akan menjadi seperti berikut ini :



Gambar 3.9 Tampilan Wireshark Setelah Dirubah

Data dari wireshark tersebut kemudian dimasukkan dalam database

phpMyAdmin yang berisikan atribut sebagai berikut:

Tabel 3.5 Atribut pada Database

No.	Nama Atribut	Tipe	Keterangan
1.	No	Integer (4)	Nomor urut data
2.	IP Source	Varchar (39)	IP sumber dari paket data tersebut
3.	IP Destination	Varchar (39)	IP tujuan dari paket data tersebut
4.	Protokol	Varchar (8)	Protokol yang dipakai oleh paket data tersebut

3.2.5 Pembuatan Data Uji

Pembuatan data uji dilakukan dengan mengakses 10 website yang berbeda dengan menggunakan 2 cara yaitu secara normal dan menggunakan *tunneling software*. Ada 3 buah *tunneling software* yang digunakan ChrisPC Anonymous Proxy, Cyberghost VPN dan ExpatShield. Berikut ini daftar website yang digunakan untuk data uji:

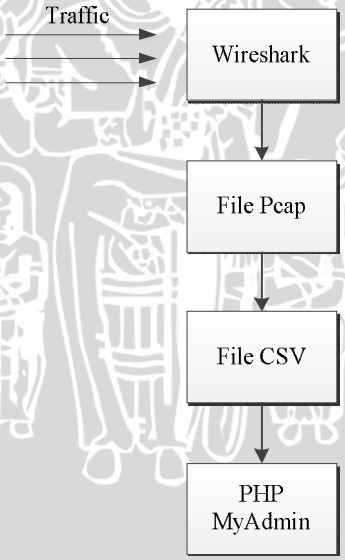
Tabel 3.6 Daftar Nama Website untuk Data Uji

No.	Nama Website
1.	www.facebook.com
2.	www.kompas.com
3.	http://ptiik.ub.ac.id
4.	http://id.yahoo.com
5.	http://twitter.com
6.	http://ub.ac.id
7.	www.gameskeren.com
8.	www.hokagame.com
9.	www.bri.co.id
10.	www.olx.co.id

Tabel 3.7 Jenis Data Uji

No.	Jenis Data Latih	Jumlah Pengguna Normal	Jumlah Pengguna *Tunneling software*	Total
1.	ChrisPC Anonymous Proxy	10	10	20
2.	CyberGhost VPN	10	10	20
3.	ExpatShield	10	10	20

Cara untuk pembuatan data uji hampir sama dengan pembuatan data latih, yaitu data dari pengguna normal dan pengguna *tunneling software* ditangkap menggunakan wireshark. Kemudian data tersebut dikonversi ke dalam file csv. Pada pembuatan data uji, file yang sudah dikonversi menjadi file csv langsung dimasukkan dalam database. Secara umum, proses pembuatan data latih dapat dilihat pada bagan dibawah ini :



Gambar 3.10 Diagram Pembuatan Data Uji

3.2.6 Pengujian Sistem

Dalam penelitian ini, penulis menggunakan pengujian simulasi. Penulis menyediakan data uji yang telah diketahui merupakan paket dari pengguna *tunneling software* atau paket dari pengguna normal.

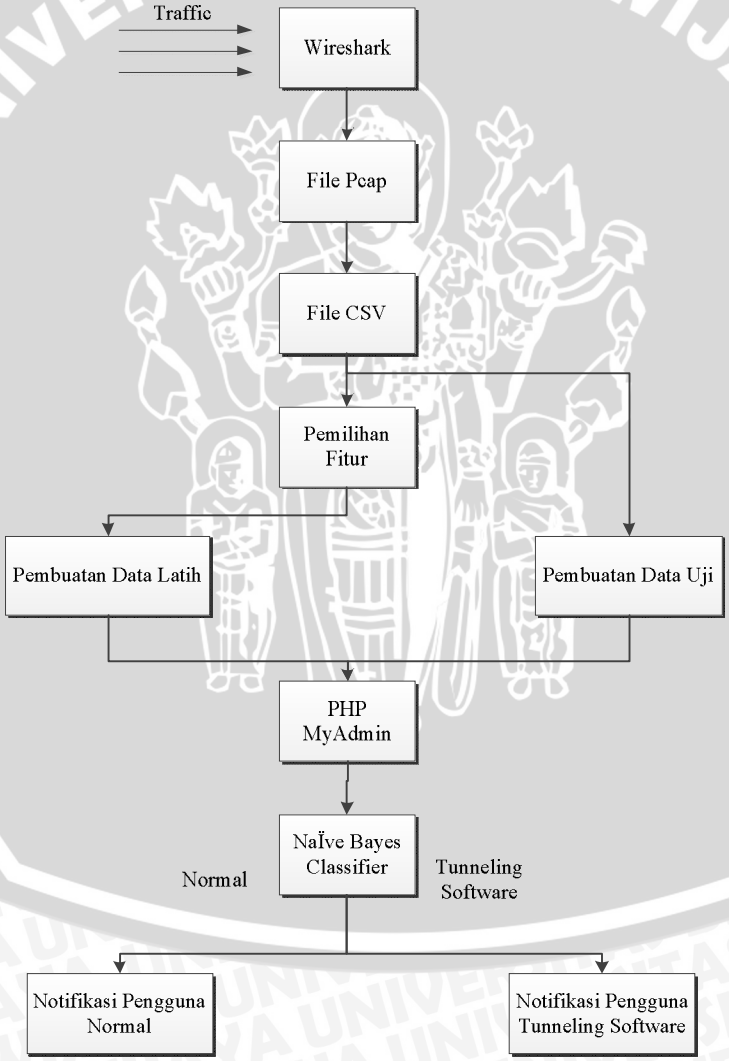
Penulis menggunakan 3 skenario pengujian untuk mengukur tingkat akurasi dari program yang telah dibuat. Dalam tiap skenario tersebut, data latihnya memiliki jumlah paket pengguna *tunneling software* dan pengguna normal berbeda untuk setiap data uji sehingga bisa dilihat akurasi dari skenario tersebut.

Tabel 3.8 Skenario Pengujian

Skenario Ke-	Jumlah Data Latih	*Tunneling software* untuk Data Latih	Jumlah Data Uji	*Tunneling software* untuk Data Uji
1	84 paket data yang terdiri dari : - 42 paket data pengguna normal - 42 paket data pengguna *tunneling software*	Ultrasurf HotspotShield Freegate	20 paket data yang terdiri dari : - 10 paket data pengguna normal - 10 paket data pengguna *tunneling software*	- ChrisPC Anonymous Proxy
2	104 paket data yang terdiri dari : - 52 paket data pengguna normal - 52 paket data pengguna *tunneling software*	Ultrasurf HotspotShield Freegate ChrisPC Anonymous Proxy	20 paket data yang terdiri dari : - 10 paket data pengguna normal - 10 paket data pengguna *tunneling software*	- CyberGhost VPN
3	104 paket data yang terdiri dari :	Ultrasurf HotspotShield Freegate	20 paket data yang terdiri dari :	- ExpatShield

	- 52 paket data pengguna normal - 52 paket data pengguna *tunneling software*	ChrisPC Anonymous Proxy CyberGhost VPN	- 10 paket data pengguna normal - 10 paket data pengguna *tunneling software*	

Alur klasifikasi paket data secara umum bisa dilihat pada diagram dibawah ini:



Gambar 3.11 Diagram Klasifikasi Paket Data

BAB IV
IMPLEMENTASI

Pada bab ini akan dibahas mengenai langkah-langkah dalam mendeteksi pengguna yang menggunakan *tunneling software*. Langkah penerapan mengacu pada tahapan perancangan yang terdiri dari sebuah komputer pengguna dan sebuah komputer detektor. Komputer pengguna disini berperan untuk mengakses layanan internet baik secara normal atau menggunakan *tunneling software*, sedangkan komputer detektor bertugas untuk mendeteksi layanan yang diakses oleh komputer pengguna tersebut menggunakan *tunneling software* atau tidak. Implementasi meliputi lingkungan perangkat keras dan perangkat lunak.

4.1 Implementasi Lingkungan

Dalam penelitian ini dibutuhkan perangkat yang menunjang tahap implementasi. Implementasi lingkungan dibagi menjadi 2 yaitu implementasi lingkungan perangkat keras dan implementasi lingkungan perangkat lunak.

4.1.1 Implementasi Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini antara lain :

1. Komputer pengguna
   - Processor Intel Core i5 CPU
   - 4 GB RAM
   - 465 GB Harddisk Drive
   - Berada di Vbox
2. Komputer detektor
   - Processor Intel Core i5 CPU
   - 4 GB RAM
   - 465 GB Harddisk Drive
   - Berada di windows induk

4.1.2 Implementasi Lingkungan Perangkat Lunak

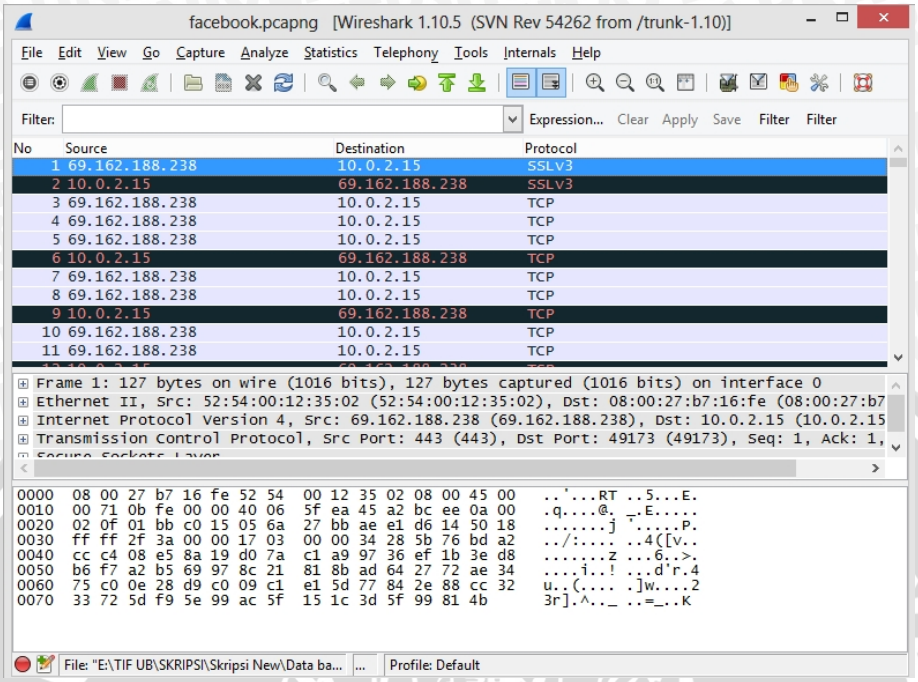
1. Komputer pengguna
   - Sistem Operasi Windows 7
   - Internet Explorer
   - Ultrasurf
   - Hotspot Shield
   - Freegate
   - ChrisPC Anonymous Proxy
   - ExpatShield
   - CyberGhost VPN
2. Komputer Detektor
   - Sistem Operasi Windows 8
   - Wireshark
   - Xampp
   - phpMyAdmin
   - Macromedia Dreamweaver 8

4.2 Proses Pembentukan Sistem

Berdasarkan hasil perancangan pada bab III tentang analisa proses pembentukan dataset dalam sistem deteksi pengguna *tunneling software*, maka pada bab ini akan dijelaskan implementasi dari perancangan yang telah dibuat tersebut.

4.2.1 Pengambilan Data

Dataset ini didapat dari aplikasi wireshark yang menangkap trafik jaringan yang dilalui oleh pengguna. Berikut ini adalah contoh hasil trafik yang ditangkap oleh wireshark:



Gambar 4.1 Hasil *Capture* Wireshark

Hasil file yang ditangkap oleh wireshark adalah berupa format binary yang tidak bisa dibaca langsung oleh pengguna sehingga harus dikonversi terlebih dahulu ke file ASCII. Konversi file wireshark menjadi ASCII dilakukan secara langsung melalui aplikasi wireshark sendiri. Contoh hasil data trafik yang sudah dikonversi menjadi file .csv adalah sebagai berikut ini:

Hasil Konversi dari .pcap menjadi .csv
No,"Source","Destination","Protocol
1,"69.162.188.238","10.0.2.15","SSLv3
2,"10.0.2.15","69.162.188.238","SSLv3
3,"69.162.188.238","10.0.2.15","TCP"
4,"69.162.188.238","10.0.2.15","TCP"
5,"69.162.188.238","10.0.2.15","TCP"
6,"10.0.2.15","69.162.188.238","TCP"
7,"69.162.188.238","10.0.2.15","TCP"

Gambar 4.2 Hasil Konversi Data

4.2.2 Pemilihan Fitur

Cara untuk memilih fitur adalah dengan melakukan filterisasi. Data yang telah didapat sebelumnya dan sudah di*convert* menjadi file csv akan difilter menggunakan fasilitas *PivotTable* dari Microsoft Excel. Semua data dari kolom yang dimasukkan ke dalam *PivotTable*, kemudian dibandingkan dan dicari kolom mana yang memiliki perbedaan paling jelas. Setelah didapat kolom mana saja yang memiliki perbedaan antara pengguna normal dan pengguna *tunneling software*, maka kolom tersebut yang akan dijadikan sebagai fitur. Data yang dibandingkan adalah data dari pengguna normal dan data dari pengguna 3 jenis *tunneling software* yang digunakan sebagai data latih. Berikut ini adalah contoh tampilan dari filterisasi yang telah dilakukan:

A	B	C	D	E	F	G	H
Normal		Ultrasurf		Hotspot		Freegate	
IP Source	Jumlah	IP Source	Jumlah	IP Source	Jumlah	IP Source	Jumlah
10.0.2.15	2373	08:00:27:54:e9:ac	1	10.0.2.15	387	08:00:27:54:e9:ac	2
10.0.2.2	3	10.0.2.15	404	50.117.56.58	3	10.0.2.15	1219
111.221.29.20	5	10.0.2.2	6	66.171.229.67	88	10.0.2.2	26
111.221.74.254	35	175.45.184.165	2	69.22.170.146	370	119.112.86.221	204
114.4.39.201	365	52:54:00:12:35:02	1	74.115.0.51	2	175.45.184.165	11
114.4.39.229	440	61.231.69.234	4	**Grand Total**	**850**	52:54:00:12:35:02	2
114.4.39.230	418	65.49.14.98	5			**Grand Total**	**1464**
173.194.117.14	19	65.55.184.151	2				
173.194.117.7	3	66.34.91.76	941				
173.252.110.27	21	fe80::2145:40f5:fad1:b35e	2				
175.45.184.164	38	**Grand Total**	**1368**				
175.45.189.132	1359						
31.13.79.65	27						
fe80::2145:40f5:fad1:b35e	2						
Grand Total	**5108**						

Gambar 4.3 Filterisasi Data

Dari filterisasi yang telah dilakukan, terlihat bahwa kolom yang memiliki perbedaan yang jelas antara pengguna normal dengan pengguna *tunneling software* adalah IP *Source*, IP *Destination* dan Protokol. Oleh karena itu, ketiga fitur tersebut yang dijadikan sebagai fitur dalam sistem deteksi penggunaan *tunneling software* ini.

4.2.3 Pembuatan Data Latih

Cara membuat data latih hampir sama dengan pengambilan data sebelumnya, yaitu data dari pengguna normal dan pengguna *tunneling software* ditangkap menggunakan wireshark. Kemudian data tersebut dikonversi ke dalam file csv. Untuk data latih, dilakukan pelabelan data sebelum dimasukkan dalam database. Untuk data dari pengguna normal akan diberi label '0' sedangkan untuk data dari pengguna *tunneling software* diberi label '1'. Setelah mendapat label, data tersebut dimasukkan dalam database.

Sebelum dimasukkan dalam database, file .csv ini dihilangkan duplikasinya agar data yang masuk dalam database bisa benar-benar valid. Inilah contoh data yang telah dimasukkan dalam database phpMyAdmin:

No	Source	Destination	Protocol	Label
126	10.0.2.15	114.38.88.56	UDP	1
127	10.0.2.15	200.147.242.103	NBNS	1
128	23.74.230.135	10.0.2.15	TCP	0
129	10.0.2.15	23.74.230.135	TCP	0

Gambar 4.4 Data Trafik dalam Database

4.2.4 Pembuatan Data Uji

Cara untuk pembuatan data uji hampir sama dengan pembuatan data latih, yaitu data dari pengguna normal dan pengguna *tunneling software* di*capture* menggunakan wireshark. Kemudian data tersebut dikonversi ke dalam file csv. Pada pembuatan data uji, file yang sudah dikonversi menjadi file csv langsung dimasukkan dalam database. Data inilah yang kemudian akan diproses dalam sistem deteksi yang telah dibuat. Berikut ini adalah contoh dari data uji yang telah dibuat:

No	Source	Destination	Protocol
1	10.0.2.15	192.221.112.253	HTTP
2	173.194.117.114	10.0.2.15	TCP
3	192.221.112.253	10.0.2.15	TCP
4	192.221.112.253	10.0.2.15	TCP
5	192.221.112.253	10.0.2.15	TCP
6	192.221.112.253	10.0.2.15	TCP
7	10.0.2.15	192.221.112.253	TCP
8	192.221.112.253	10.0.2.15	TCP
9	192.221.112.253	10.0.2.15	TCP
10	10.0.2.15	192.221.112.253	TCP

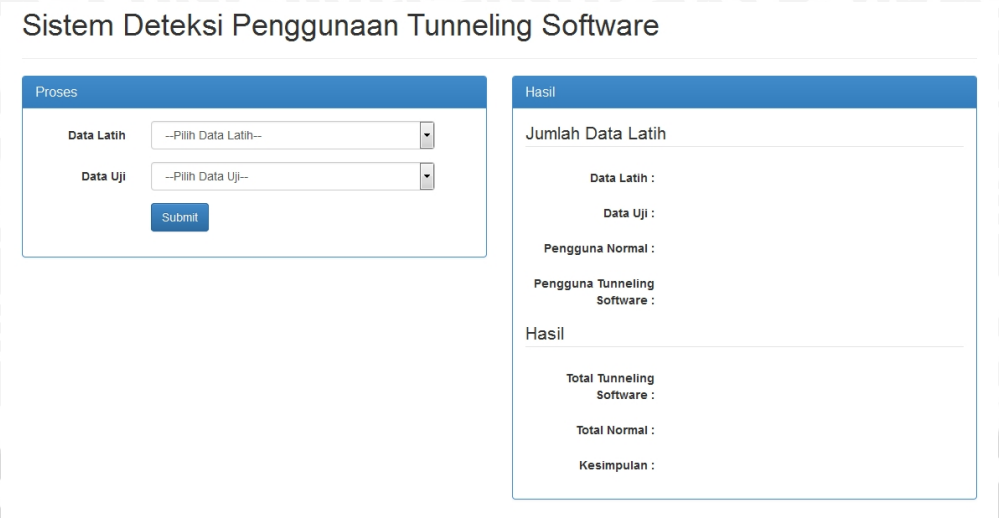
Gambar 4.5 Contoh Data Uji

4.2.5 Pengujian Sistem

Setelah semua data baik data latih maupun data uji dimasukkan dalam database, maka data itu kemudian akan diolah melalui sebuah program untuk diklasifikasikan. Metode untuk melakukan klasifikasi adalah dengan menggunakan metode *naïve bayes classifier*. Klasifikasi ini didasarkan pada pelabelan yang telah dilakukan, yaitu data label '1' untuk pengguna *tunneling software* dan data label '0' untuk pengguna normal. Data-data tersebut kemudian dihitung menggunakan rumus dasar dari *naïve bayes classifier*. Rumus yang digunakan adalah sebagai berikut :

$$P(C|F_1, \dots, F_n) = \frac{P(C)P(F_1,\dots,F_n|C)}{P(F_1,\dots,F_n)} \quad (4\text{-}1)$$

Keseluruhan proses ini ditunjukkan di kode program pada lampiran 1. Contoh tampilan program dapat dilihat pada gambar dibawah ini:



Gambar 4.6 Tampilan Awal Program

Gambar 4.7 Tampilan Untuk Pengguna Normal

Gambar 4.8 Tampilan Untuk Pengguna *Tunneling Software*

Dalam sekali *running*, program melakukan klasifikasi sebanyak data uji. Jika dalam sebuah data uji terdapat 100 baris data maka dalam sekali eksekusi program tersebut melakukan 100 kali klasifikasi dan menghasilkan keluaran yaitu data uji yang dimasukkan tersebut termasuk pengguna normal maupun pengguna dari *tunneling software*.

BAB V
PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis ini, penulis akan menjabarkan tentang parameter dan hasil pengujian yang telah dilakukan pada sistem. Selain itu, dalam bab ini juga terdapat analisa hasil klasifikasi pengguna yang melakukan *bypass*.

5.1 Pengujian *Black Box*

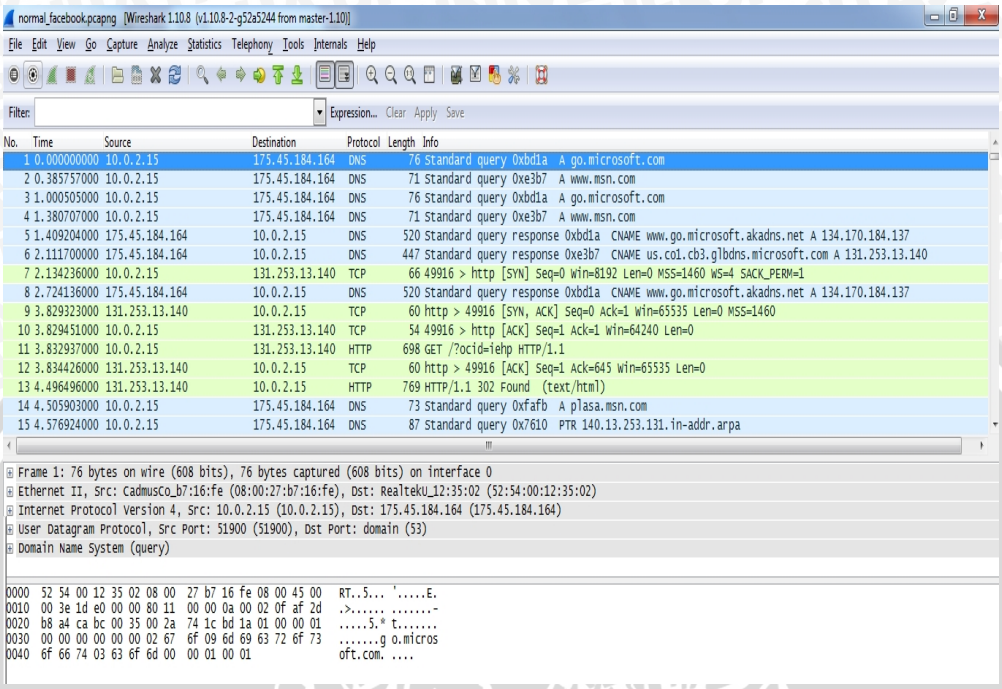
Proses pengujian adalah proses mencoba aplikasi yang sudah dibangun untuk menemukan adanya eror pada aplikasi [SUS-11]. Pada penelitian ini, penulis menggunakan metode pengujian *Black Box Testing*. Metode pengujian ini berfokus pada persyaratan untuk memverifikasi dan memvalidasi persyaratan fungsionalitas suatu aplikasi. Pengujian ini akan menentukan sejauh mana sistem dapat berjalan sesuai dengan fungsinya.

5.2 Skenario Pengujian

Pengujian dilakukan sejak awal proses klasifikasi data. Hal ini dilakukan untuk memastikan bahwa setiap proses yang dilakukan telah sesuai dengan apa yang diharapkan.

a. Pengambilan Data

Pada tahap pertama, dilakukan pengambilan data. Data trafik dari pengguna yang sedang mengakses suatu website ditangkap menggunakan wireshark.

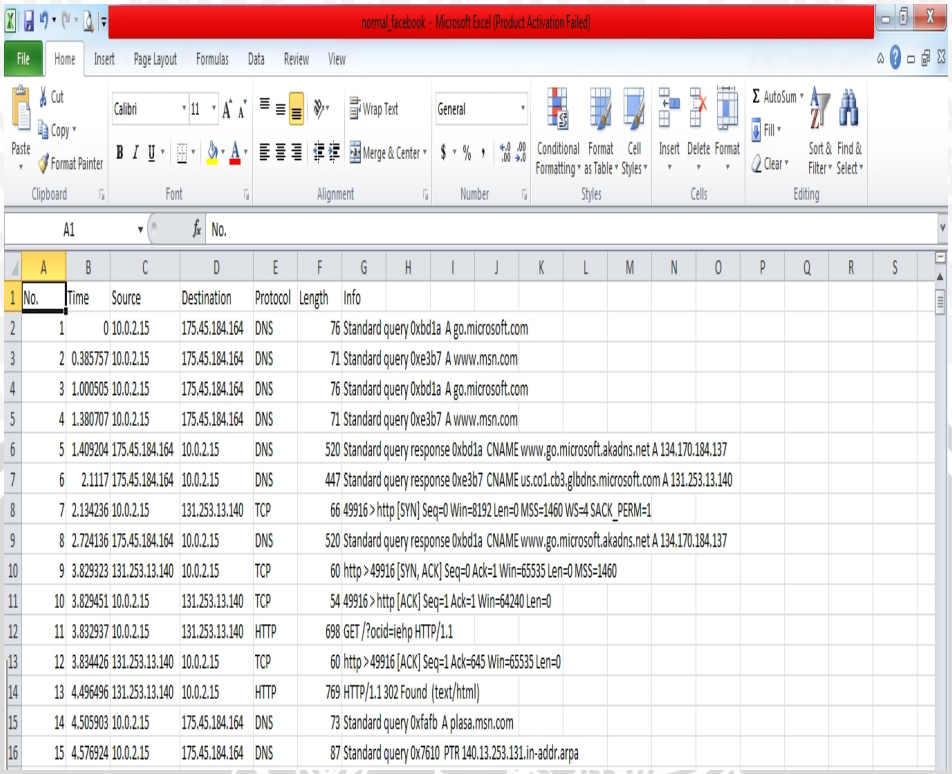


Gambar 5.1 Hasil *Capture* Data dari Wireshark

Semua data trafik dari pengguna bisa ditangkap dengan baik oleh wireshark. Hal ini berarti bahwa tahap pertama berstatus *valid*.

b. Proses Konversi Data

Hasil file yang ditangkap oleh wireshark merupakan file pcap yang tidak bisa dibaca oleh aplikasi lain. Oleh karena itu, data trafik tersebut perlu dirubah ke dalam bentuk csv agar bisa dibaca oleh aplikasi lain.



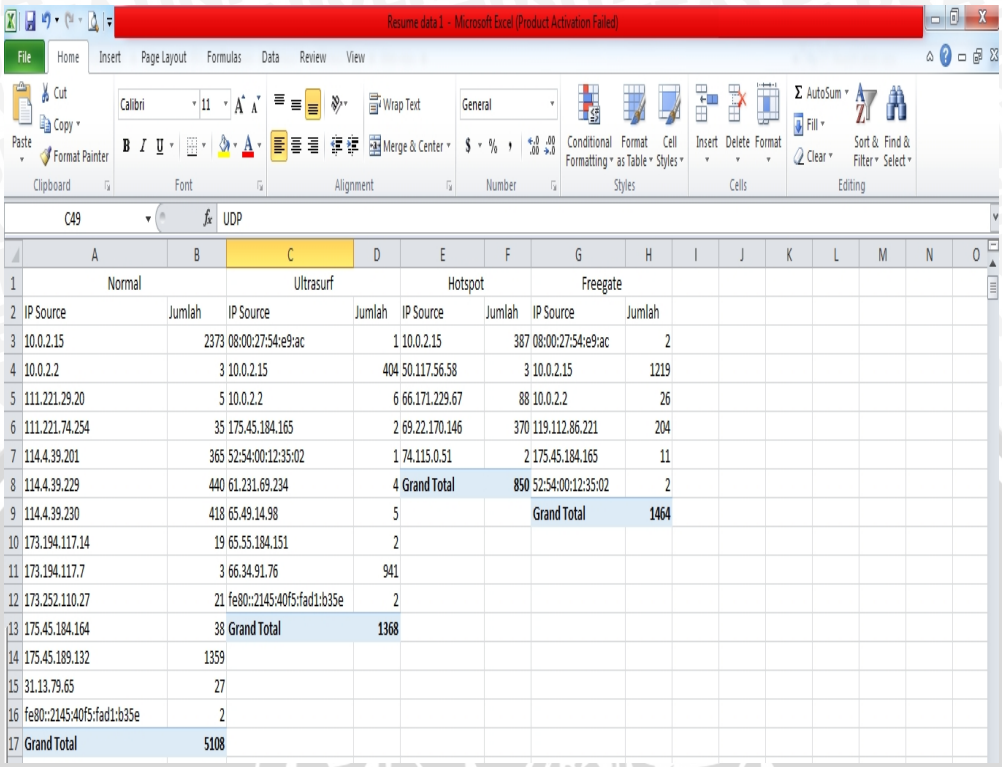
No.	Time	Source	Destination	Protocol	Length	Info
1	0	10.0.2.15	175.45.184.164	DNS	76	Standard query 0xbd1a A go.microsoft.com
2	0.385757	10.0.2.15	175.45.184.164	DNS	71	Standard query 0xe3b7 A www.msn.com
3	1.000505	10.0.2.15	175.45.184.164	DNS	76	Standard query 0xbd1a A go.microsoft.com
4	1.380707	10.0.2.15	175.45.184.164	DNS	71	Standard query 0xe3b7 A www.msn.com
5	1.409204	175.45.184.164	10.0.2.15	DNS	520	Standard query response 0xbd1a CNAME www.go.microsoft.akadns.net A 134.170.184.137
6	2.1117	175.45.184.164	10.0.2.15	DNS	447	Standard query response 0xe3b7 CNAME us.co1.cb3.glbdns.microsoft.com A 131.253.13.140
7	2.134236	10.0.2.15	131.253.13.140	TCP	66	49916 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
8	2.724136	175.45.184.164	10.0.2.15	DNS	520	Standard query response 0xbd1a CNAME www.go.microsoft.akadns.net A 134.170.184.137
9	3.829323	131.253.13.140	10.0.2.15	TCP	60	http > 49916 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
10	3.829451	10.0.2.15	131.253.13.140	TCP	54	49916 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
11	3.832937	10.0.2.15	131.253.13.140	HTTP	698	GET /?ocid=iehp HTTP/1.1
12	3.834426	131.253.13.140	10.0.2.15	TCP	60	http > 49916 [ACK] Seq=1 Ack=645 Win=65535 Len=0
13	4.496496	131.253.13.140	10.0.2.15	HTTP	769	HTTP/1.1 302 Found (text/html)
14	4.505903	10.0.2.15	175.45.184.164	DNS	73	Standard query 0xfafb A plasa.msn.com
15	4.576924	10.0.2.15	175.45.184.164	DNS	87	Standard query 0x7610 PTR 140.13.253.131.in-addr.arpa

Gambar 5.2 Hasil Konversi Data

Proses konversi data dari file pcap menjadi file csv berhasil dilakukan dengan baik. Oleh karena itu, tahap kedua berstatus *valid*.

c. Pemilihan Fitur

Setelah semua data dari pengguna normal dan pengguna *tunneling software* dirubah menjadi bentuk csv, maka akan dilakukan filterisasi data untuk menentukan fitur yang akan digunakan untuk klasifikasi. Proses filterisasi ini menggunakan fasilitas *PivotTable* yang disediakan oleh Microsoft Excel. Filterisasi ini dilakukan pada semua kolom di data pengguna normal dan pengguna *tunneling software*.



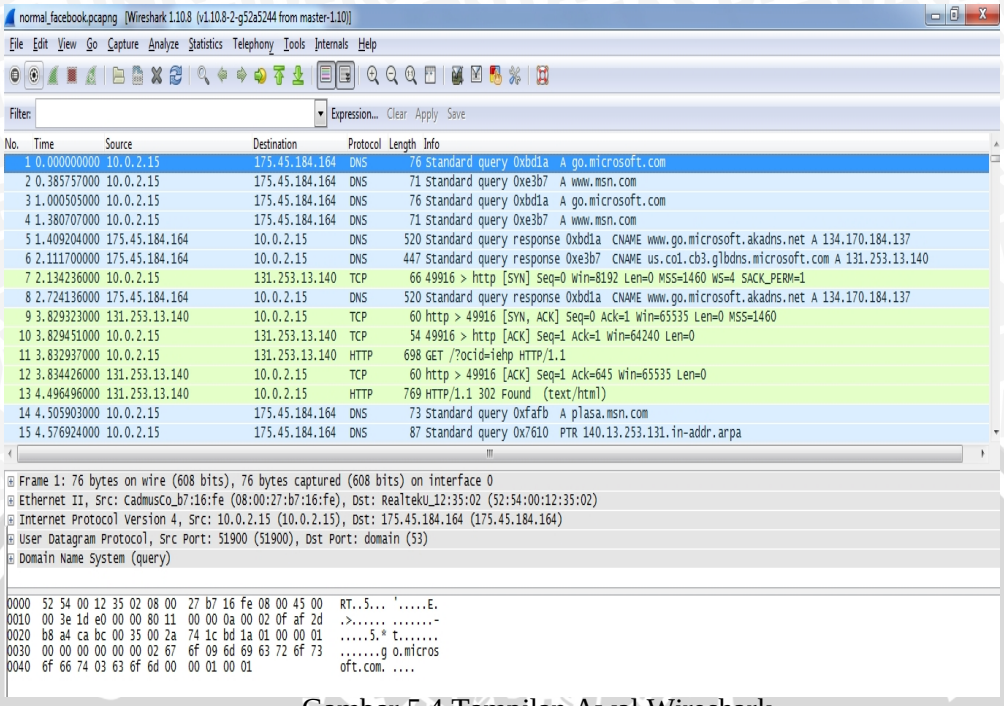
	A	B	C	D	E	F	G	H
1	Normal		Ultrasurf		Hotspot		Freegate	
2	IP Source	Jumlah	IP Source	Jumlah	IP Source	Jumlah	IP Source	Jumlah
3	10.0.2.15	2373	08:00:27:54:e9:ac	1	10.0.2.15	387	08:00:27:54:e9:ac	2
4	10.0.2.2	3	10.0.2.15	404	50.117.56.58	3	10.0.2.15	1219
5	111.221.29.20	5	10.0.2.2	6	66.171.229.67	88	10.0.2.2	26
6	111.221.74.254	35	175.45.184.165	2	69.22.170.146	370	119.112.86.221	204
7	114.4.39.201	365	52:54:00:12:35:02	1	74.115.0.51	2	175.45.184.165	11
8	114.4.39.229	440	61.231.69.234	4	Grand Total	850	52:54:00:12:35:02	2
9	114.4.39.230	418	65.49.14.98	5			Grand Total	1464
10	173.194.117.14	19	65.55.184.151	2				
11	173.194.117.7	3	66.34.91.76	941				
12	173.252.110.27	21	fe80::2145:40f5:fad1:b35e	2				
13	175.45.184.164	38	Grand Total	1368				
14	175.45.189.132	1359						
15	31.13.79.65	27						
16	fe80::2145:40f5:fad1:b35e	2						
17	Grand Total	5108						

Gambar 5.3 Proses Filterisasi Data

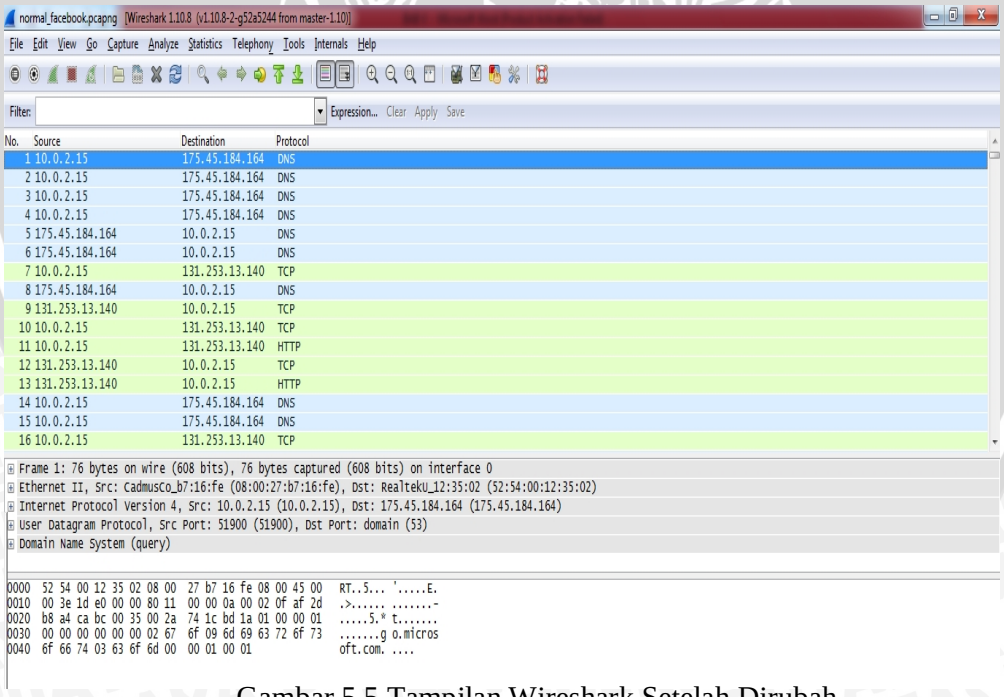
Dari proses filterisasi yang dilakukan maka terlihat bahwa kolom yang memiliki perbedaan paling jelas antara pengguna normal dengan pengguna *tunneling software* adalah kolom IP *Source*, IP *Destination* dan *Protocol*. Karena itu ketiga kolom tersebut dipilih untuk menjadi fitur. Proses pemilihan fitur ini telah berhasil dilakukan. Oleh karena itu, tahap ketiga berstatus *valid*.

d. Pembuatan Data Latih

Setelah fitur untuk klasifikasi didapatkan, maka tahap selanjutnya adalah pembuatan data latih. Untuk lebih mempermudah proses pengambilan data, maka tampilan dari wireshark pun dirubah sesuai dengan fitur yang dibutuhkan. Sehingga data yang ditangkap hanya data dari fitur yang dibutuhkan, yaitu IP *Source*, IP *Destination* dan *Protocol*.

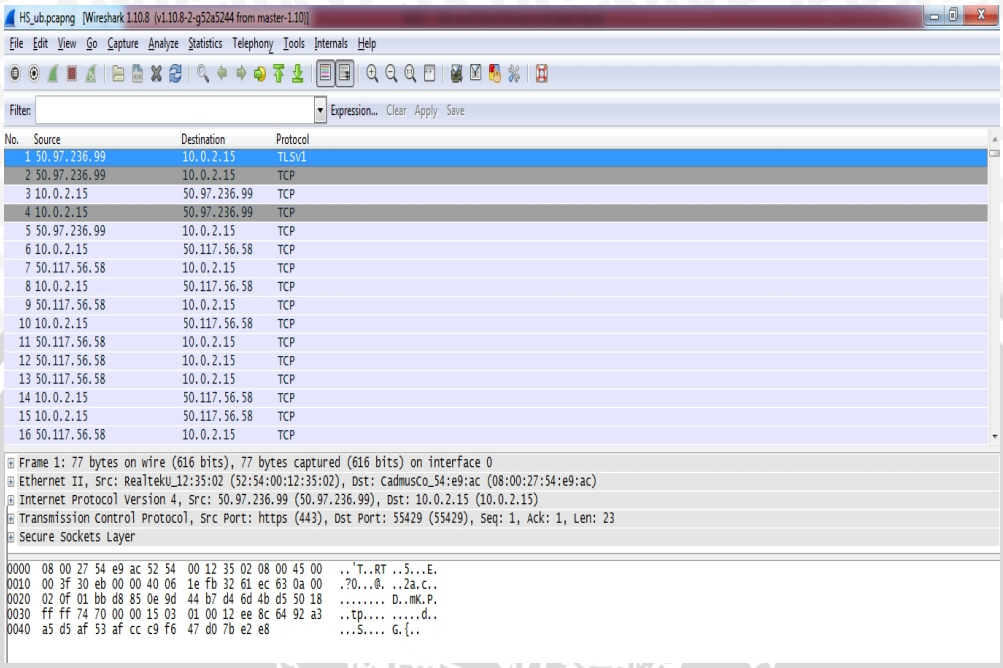


Gambar 5.4 Tampilan Awal Wireshark

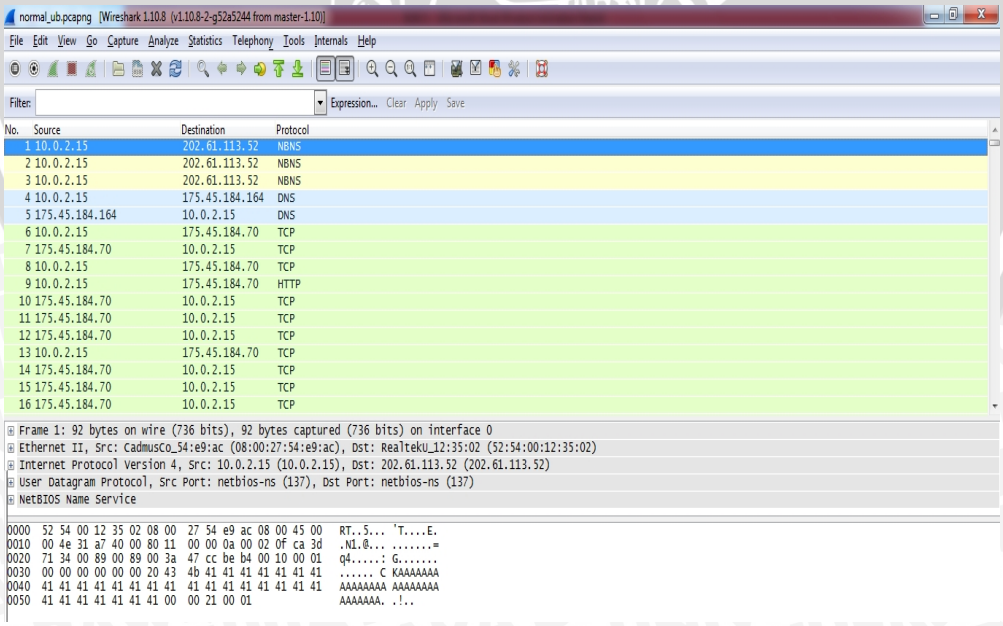


Gambar 5.5 Tampilan Wireshark Setelah Dirubah

Proses pengambilan data latih hampir sama dengan pengambilan data sebelumnya. Data dari pengguna normal dan pengguna *tunneling software* ditangkap menggunakan aplikasi wireshark.

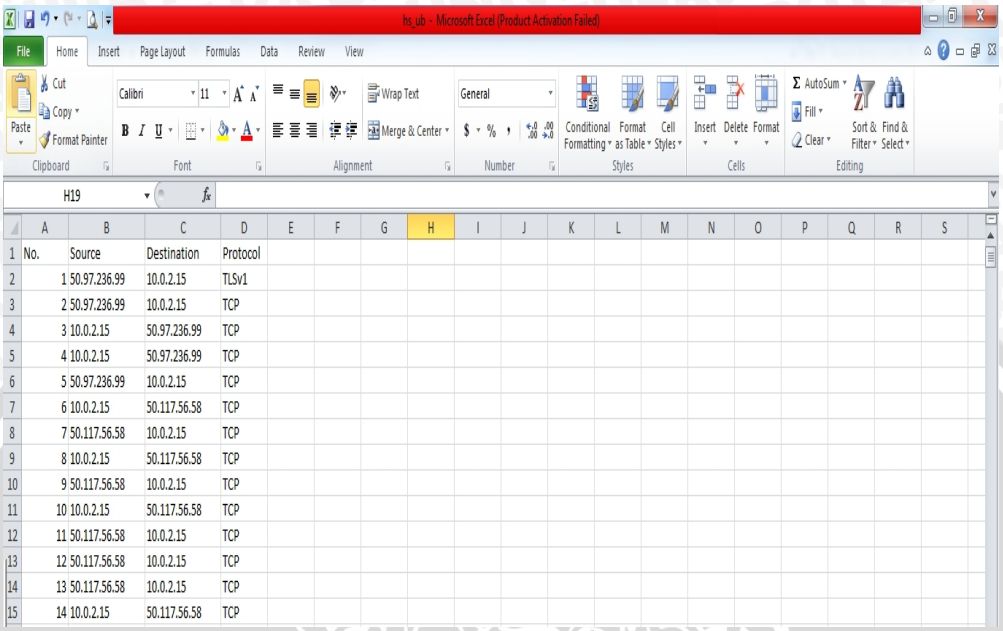


Gambar 5.6 Data Trafik dari Pengguna *Tunneling Software*



Gambar 5.7 Data Trafik dari Pengguna Normal

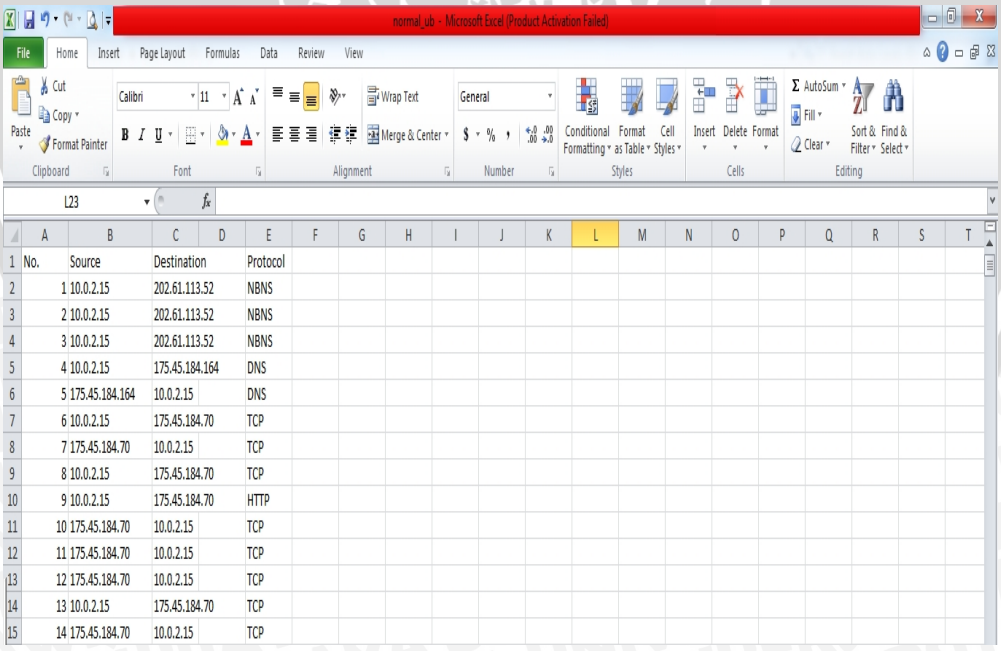
Setelah data trafik ditangkap, kemudian dikonversi dalam bentuk file csv agar bisa dibaca oleh aplikasi yang lain.



hs_ub - Microsoft Excel (Product Activation Failed)

No.	Source	Destination	Protocol
1	50.97.236.99	10.0.2.15	TLSv1
2	50.97.236.99	10.0.2.15	TCP
3	10.0.2.15	50.97.236.99	TCP
4	10.0.2.15	50.97.236.99	TCP
5	50.97.236.99	10.0.2.15	TCP
6	10.0.2.15	50.117.56.58	TCP
7	50.117.56.58	10.0.2.15	TCP
8	10.0.2.15	50.117.56.58	TCP
9	50.117.56.58	10.0.2.15	TCP
10	10.0.2.15	50.117.56.58	TCP
11	50.117.56.58	10.0.2.15	TCP
12	50.117.56.58	10.0.2.15	TCP
13	50.117.56.58	10.0.2.15	TCP
14	10.0.2.15	50.117.56.58	TCP

Gambar 5.8 Hasil Konversi Data Trafik Pengguna *Tunneling Software*

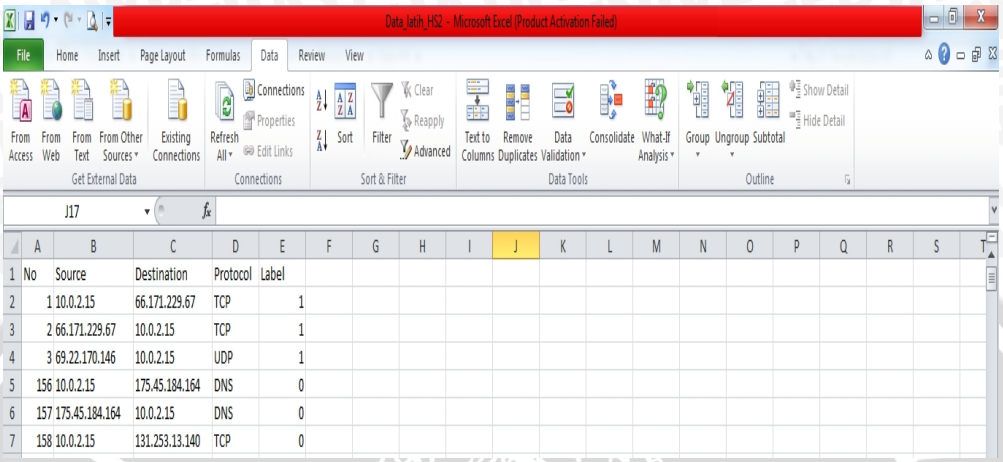


normal_ub - Microsoft Excel (Product Activation Failed)

No.	Source	Destination	Protocol
1	10.0.2.15	202.61.113.52	NBNS
2	10.0.2.15	202.61.113.52	NBNS
3	10.0.2.15	202.61.113.52	NBNS
4	10.0.2.15	175.45.184.164	DNS
5	175.45.184.164	10.0.2.15	DNS
6	10.0.2.15	175.45.184.70	TCP
7	175.45.184.70	10.0.2.15	TCP
8	10.0.2.15	175.45.184.70	TCP
9	10.0.2.15	175.45.184.70	HTTP
10	175.45.184.70	10.0.2.15	TCP
11	175.45.184.70	10.0.2.15	TCP
12	175.45.184.70	10.0.2.15	TCP
13	10.0.2.15	175.45.184.70	TCP
14	175.45.184.70	10.0.2.15	TCP

Gambar 5.9 Hasil Konversi Data Trafik Pengguna Normal

Setelah semua data trafik dirubah menjadi file csv, data tersebut diberikan label. Label “0” untuk pengguna normal dan label “1” untuk pengguna *tunneling software*.



	A	B	C	D	E
1	No	Source	Destination	Protocol	Label
2	1	10.0.2.15	66.171.229.67	TCP	1
3	2	66.171.229.67	10.0.2.15	TCP	1
4	3	69.22.170.146	10.0.2.15	UDP	1
5	156	10.0.2.15	175.45.184.164	DNS	0
6	157	175.45.184.164	10.0.2.15	DNS	0
7	158	10.0.2.15	131.253.13.140	TCP	0

Gambar 5.10 Hasil Pelabelan Data Latih

Semua data yang sudah diberi label, kemudian dimasukkan ke dalam database. Data latih ini digunakan untuk mengklasifikasikan pengguna berdasarkan karakteristik paket yang ada.

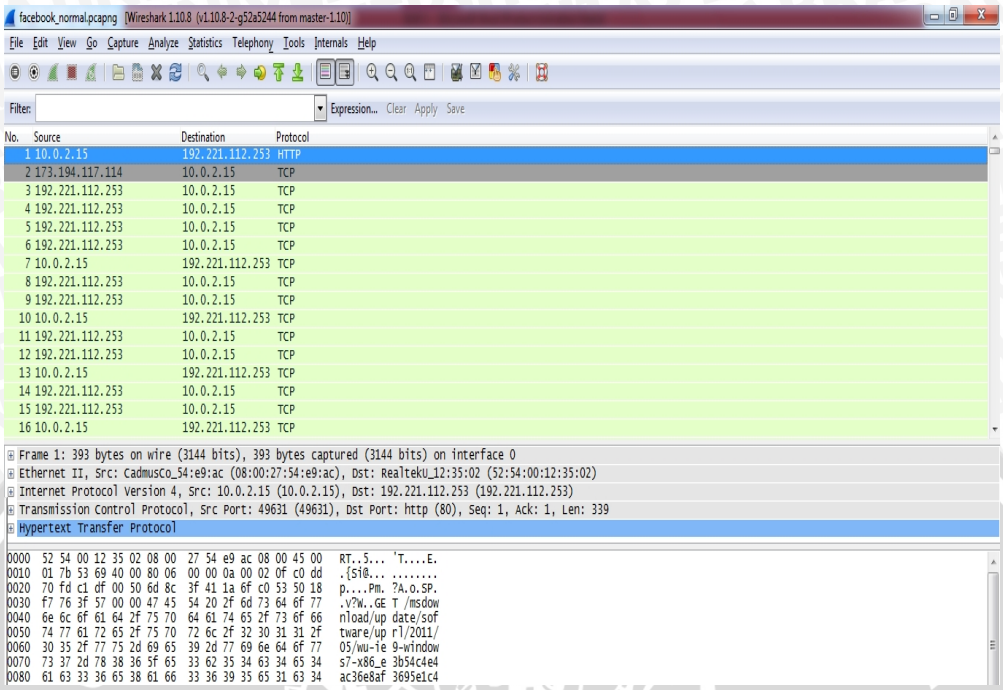
No	Source	Destination	Protocol	Label
126	10.0.2.15	114.38.88.56	UDP	1
127	10.0.2.15	200.147.242.103	NBNS	1
128	23.74.230.135	10.0.2.15	TCP	0
129	10.0.2.15	23.74.230.135	TCP	0

Gambar 5.11 Data Latih dalam Database

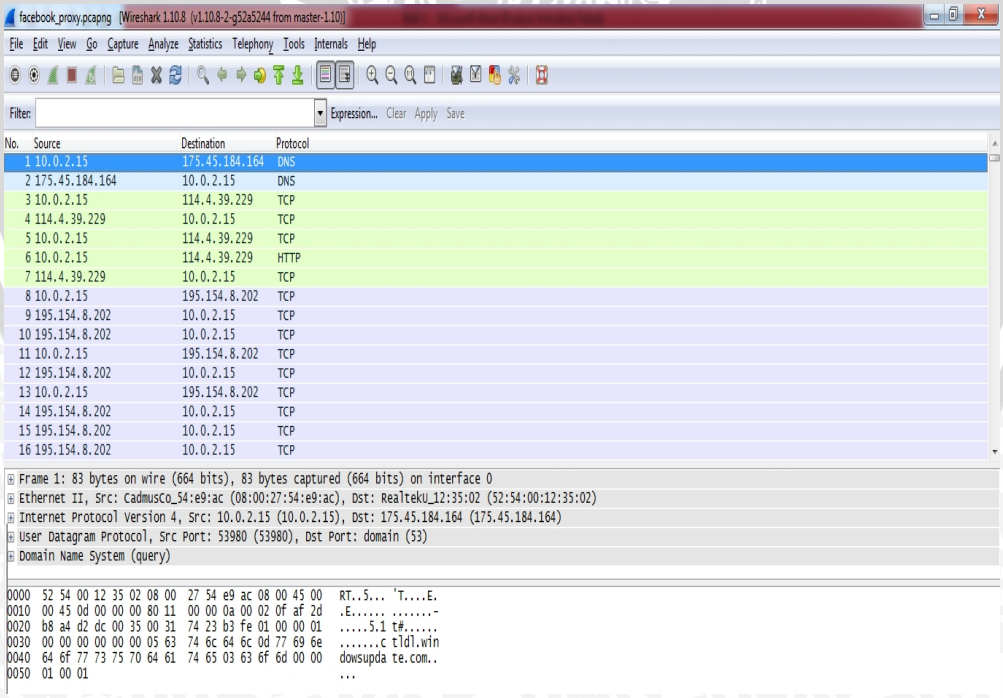
Semua proses pembuatan data latih ini berhasil dilakukan, sehingga memiliki status *valid*.

e. Pembuatan Data Uji

Proses pembuatan data uji ini hampir sama dengan pembuatan data latih. Data dari pengguna normal dan pengguna *tunneling software* ditangkap menggunakan wireshark.

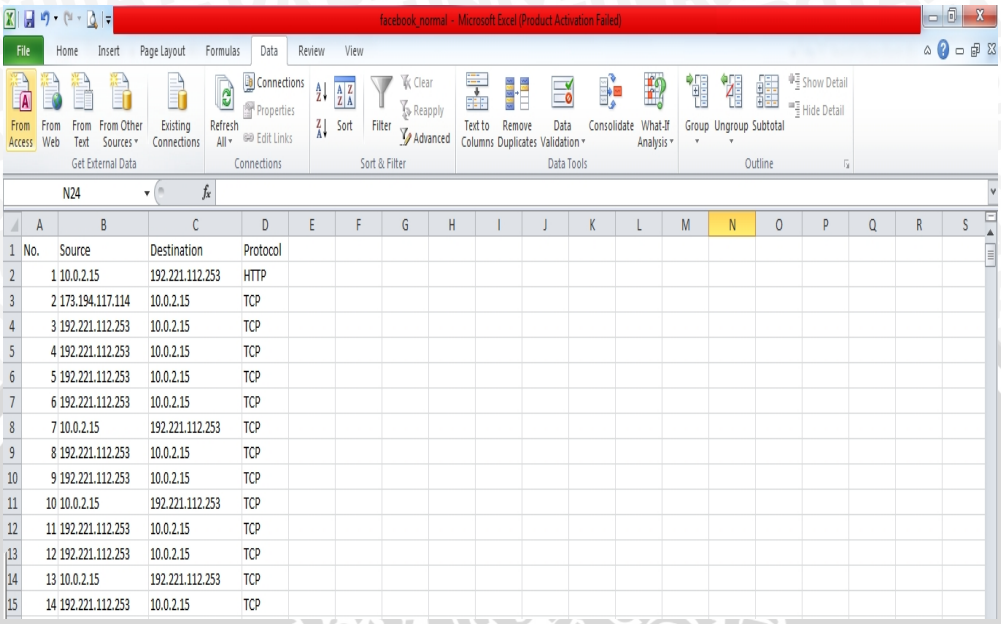


Gambar 5.12 Hasil *Capture* Data Uji Pengguna Normal



Gambar 5.13 Hasil *Capture* Data Uji Pengguna *Tunneling Software*

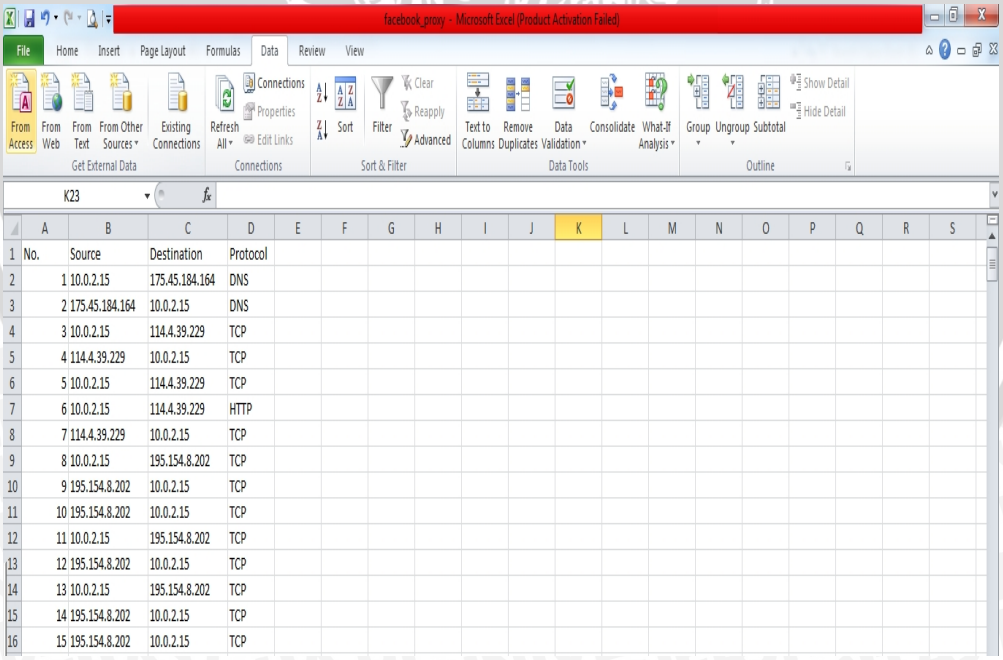
Data uji tersebut kemudian dikonversi ke dalam file csv agar bisa dibaca oleh aplikasi lain.



facebook_normal - Microsoft Excel (Product Activation Failed)

No.	Source	Destination	Protocol
1	10.0.2.15	192.221.112.253	HTTP
2	173.194.117.114	10.0.2.15	TCP
3	192.221.112.253	10.0.2.15	TCP
4	192.221.112.253	10.0.2.15	TCP
5	192.221.112.253	10.0.2.15	TCP
6	192.221.112.253	10.0.2.15	TCP
7	10.0.2.15	192.221.112.253	TCP
8	192.221.112.253	10.0.2.15	TCP
9	192.221.112.253	10.0.2.15	TCP
10	10.0.2.15	192.221.112.253	TCP
11	192.221.112.253	10.0.2.15	TCP
12	192.221.112.253	10.0.2.15	TCP
13	10.0.2.15	192.221.112.253	TCP
14	192.221.112.253	10.0.2.15	TCP

Gambar 5.14 Hasil Konversi Data Uji Pengguna Normal



facebook_proxy - Microsoft Excel (Product Activation Failed)

No.	Source	Destination	Protocol
1	10.0.2.15	175.45.184.164	DNS
2	175.45.184.164	10.0.2.15	DNS
3	10.0.2.15	114.4.39.229	TCP
4	114.4.39.229	10.0.2.15	TCP
5	10.0.2.15	114.4.39.229	TCP
6	10.0.2.15	114.4.39.229	HTTP
7	114.4.39.229	10.0.2.15	TCP
8	10.0.2.15	195.154.8.202	TCP
9	195.154.8.202	10.0.2.15	TCP
10	195.154.8.202	10.0.2.15	TCP
11	10.0.2.15	195.154.8.202	TCP
12	195.154.8.202	10.0.2.15	TCP
13	10.0.2.15	195.154.8.202	TCP
14	195.154.8.202	10.0.2.15	TCP
15	195.154.8.202	10.0.2.15	TCP

Gambar 5.15 Hasil Konversi Data Uji Pengguna *Tunneling Software*

Setelah semua data uji dirubah menjadi bentuk file csv, maka data uji tersebut dimasukkan ke dalam database agar dapat dilakukan proses klasifikasi.

No	Source	Destination	Protocol
1	10.0.2.15	192.221.112.253	HTTP
2	173.194.117.114	10.0.2.15	TCP
3	192.221.112.253	10.0.2.15	TCP
4	192.221.112.253	10.0.2.15	TCP
5	192.221.112.253	10.0.2.15	TCP
6	192.221.112.253	10.0.2.15	TCP
7	10.0.2.15	192.221.112.253	TCP
8	192.221.112.253	10.0.2.15	TCP
9	192.221.112.253	10.0.2.15	TCP
10	10.0.2.15	192.221.112.253	TCP

Gambar 5.16 Data Uji dalam Database

Semua proses pembuatan data uji ini telah berhasil dilakukan, maka statusnya adalah *valid*.

f. Klasifikasi

Tahap selanjutnya adalah proses klasifikasi. Data yang sudah berada dalam database kemudian akan langsung diklasifikasikan menggunakan algoritma *naïve bayes classifier*. Hasil dari sistem berupa notifikasi paket data uji tersebut merupakan paket data dari pengguna *tunneling software* atau pengguna normal.

Evaluasi dilakukan dengan membandingkan hasil pendeteksian yang dilakukan oleh sistem dengan pengamatan langsung terhadap paket data yang ada. Pengujian yang dilakukan ini berguna untuk mengetahui presisi, recall dan akurasi dari sistem yang telah dibangun.

Tingkat akurasi ini sangat menentukan keberhasilan suatu system, untuk itu perlu mempertimbangkan metric berikut : *false positive*, *false negative*, *true positive*, *true negative*, *recall* dan *presisi*. Thuy Nguyen T.T. dan Grenville Armitage telah memberikan definisi mereka [LIU-12].

- *False Negative* : Persentase jenis target yang salah diklasifikasikan sebagai yang lain.
- *False Positive* : Persentase jenis trafik yang diklasifikasikan sebagai jenis yang ditargetkan.
- *True Positive* : Persentase trafik yang diklasifikasikan dengan benar sebagai jenis yang ditargetkan.
- *True Negative* : Persentase trafik lainnya dengan benar tidak diklasifikasikan sebagai jenis yang ditargetkan.

Secara umum, keempat alarm tersebut dapat ditunjukkan pada tabel dibawah ini:

Tabel 5.1 Jenis Alarm

Prediksi / Aktual	Pengguna *Tunneling software*	Pengguna Normal
Pengguna *Tunneling software*	TP	FN
Pengguna Normal	FP	TN

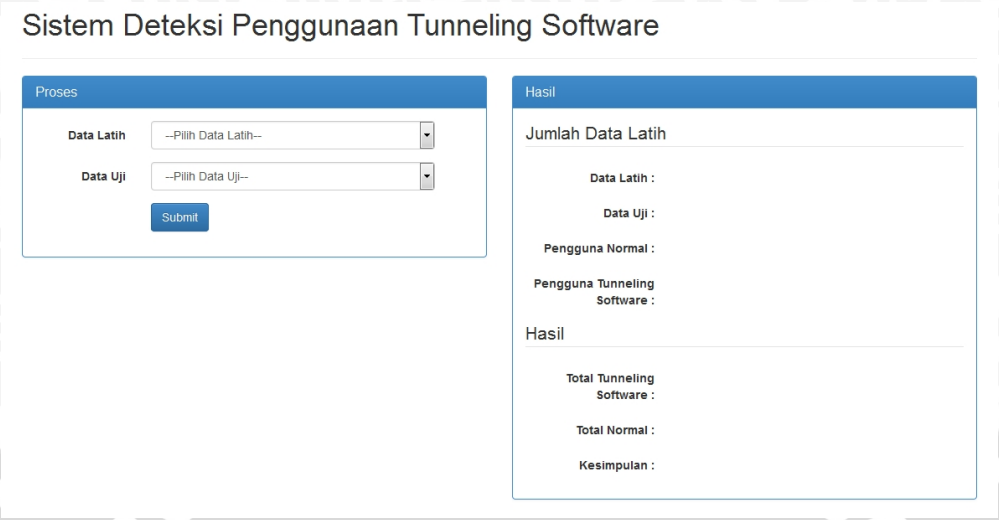
Setelah jumlah setiap *alert* telah diketahui, presisi dapat dihitung menggunakan persamaan 1, *recall* menggunakan persamaan 2 dan akurasi menggunakan rumus pada persamaan 3.

$$\text{Presisi} = \frac{TP}{TP+FP} 100\% \quad (5\text{-}1)$$

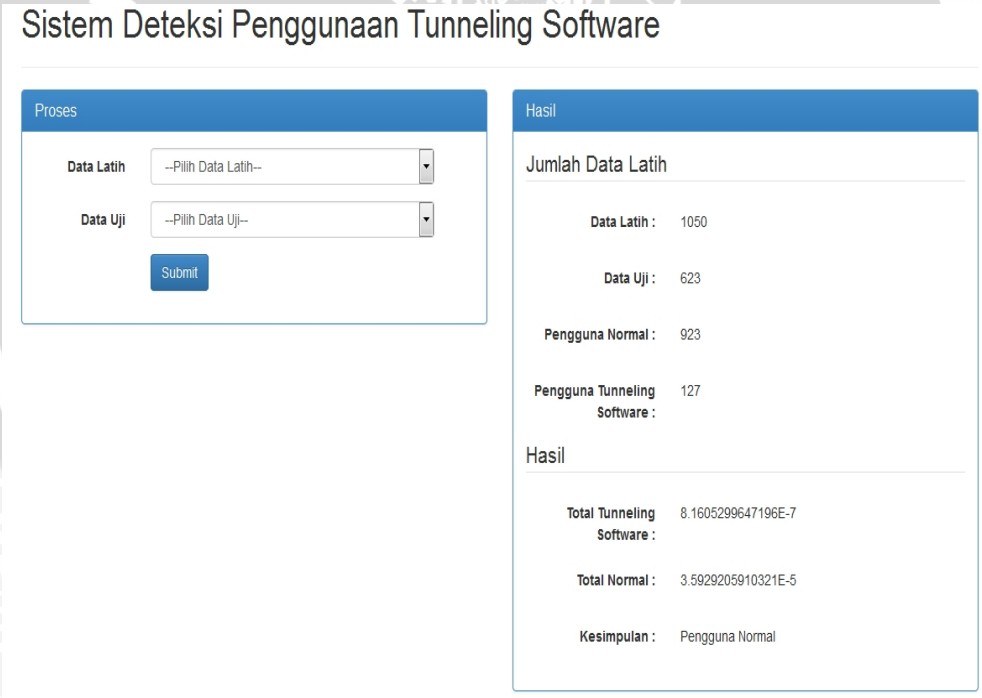
$$\text{Recall} = \frac{TP}{TP+FN} 100\% \quad (5\text{-}2)$$

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} 100\% \quad (5\text{-}3)$$

Berikut ini adalah tampilan dari sistem deteksi yang telah dibuat.



Gambar 5.17 Tampilan Awal Program



Gambar 5.18 Tampilan Untuk Pengguna Normal

Gambar 5.19 Tampilan Untuk Pengguna *Tunneling Software*

Keseluruhan proses klasifikasi bisa berhasil dilakukan, maka statusnya adalah *valid.*

5.3 Hasil Pengujian

Sistem hanya dapat memberikan keluaran berupa jumlah pengguna normal dan pengguna *tunneling software* yang terdeteksi dalam sistem. Untuk itu penulis perlu membandingkan apakah keluaran dari sistem sudah benar atau terdapat kesalahan dalam klasifikasi.

Pada pengujian yang telah dilakukan, tingkat akurasi paling tinggi didapat pada skenario 2 dan 3. Data lengkap dari pengujian yang telah dilakukan dapat dilihat pada tabel 5.2 dan 5.3 dibawah ini:

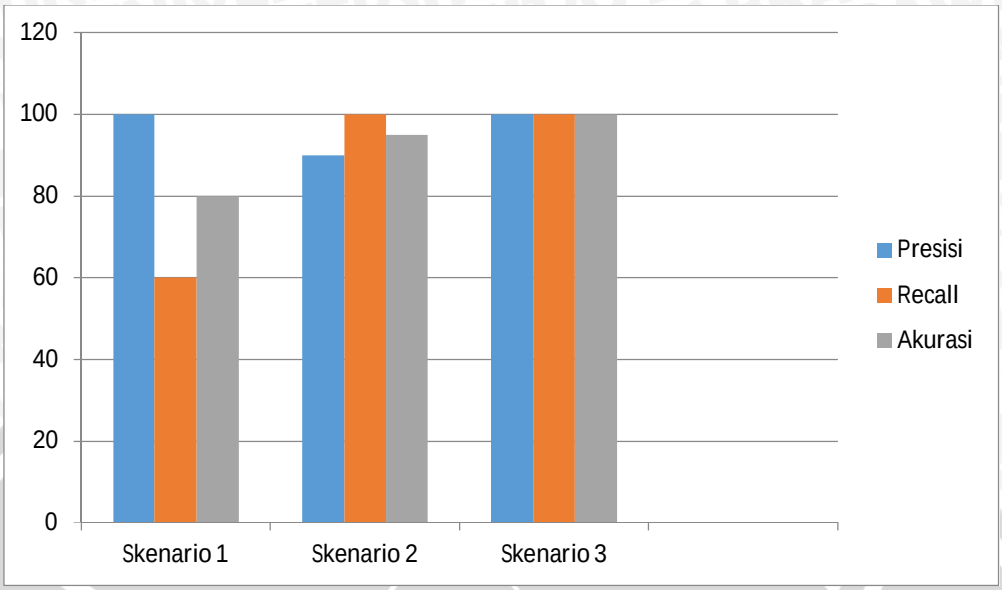
Tabel 5.2 Hasil Pengujian Alarm

Skenario Ke-	Prediksi / Aktual	Pengguna *Tunneling software*	Pengguna Normal
1	- Pengguna *Tunneling software* - Pengguna Normal	6 0	4 10
2	- Pengguna *Tunneling software* - Pengguna Normal	10 1	0 9
3	- Pengguna *Tunneling software* - Pengguna Normal	10 0	0 10

Tingkat persentase keberhasilan pengujian yang telah dilakukan dapat dilihat pada tabel dibawah ini:

Tabel 5.3 Data Pengujian Simulasi

Skenario Ke-	Jenis Alarm	Presisi	Recall	Akurasi
1	- TP : 6 / 10 = 60 % - FN : 4 / 10 = 40 % - FP : 0 / 10 = 0 % - TN : 10 / 10 = 100 %	100 %	60 %	80 %
2	- TP : 10 / 10 = 100 % - FN : 0 / 10 = 0 % - FP : 1 / 10 = 10 % - TN : 9 / 10 = 90 %	90 %	100 %	95 %
3	- TP : 10 / 10 = 100 % - FN : 0 / 10 = 0 % - FP : 0 / 10 = 0 % - TN : 10 / 10 = 100 %	100 %	100 %	100 %



Gambar 5.20 Grafik Tingkat Pengujian Sistem

BAB VI
PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat penulis ambil dari penelitian yang telah dilakukan dengan berbagai skenario pengujian antara lain :

1. Tahap-tahap untuk mendeteksi penggunaan *tunneling software* adalah pengambilan data, pemilihan fitur, pembuatan data latih, pembuatan data uji dan klasifikasi.
2. Tingkat akurasi dari sistem deteksi penggunaan *tunneling software* tergantung pada metode dan jumlah data latih yang digunakan.
3. Pada pengujian yang telah dilakukan di penelitian ini, sistem deteksi penggunaan *tunneling software* mampu menghasilkan akurasi mencapai 100 %.

6.2 Saran

Diperlukan lebih banyak jenis aplikasi *tunneling software* yang harus dipelajari sehingga menambah jumlah kombinasi data latih. *Tunneling software* yang digunakan untuk data latih antara lain Ultrasurf, Hotspot Shield dan Freegate. Sedangkan pada pengujian, penulis menggunakan ChrisPC Anonymous Proxy, ExpatShield dan CyberGhost VPN.

DAFTAR PUSTAKA

[CIV-07] Civisec. 2007. Everyone's Guide To By-Passing Internet Censorship For Citizens Worldwide. Toronto: The University of Toronto.

[DUS-08] Dusi, M. M. Crotti. F. Gringoli. L. Salgarelli. Tunnel Hunter: Detecting Application-Layer Tunnels with Statistical Fingerprinting. Italy: Universita degli Studi di Brescia.

[HAK-13] Hakim, Zainal. 2013. Apa itu phpmyadmin?. Diakses dari http://www.zainalhakim.web.id/apa-itu-phpmyadmin.html. Tanggal akses 14-6-2013

[LIU-12] Liu, Yu. 2012. A Survey of Machine Learning Based Packet Classification. Canada: University of British Columbia.

[SGS-13] Student Guide Series. 2013. Macromedia Dreamweaver 8. Diakses dari http://blog.akmi-baturaja.ac.id/naniktriana/wp-content/uploads/2013/02/SGS-Macromedia-Dreamweaver-8.pdf. Tanggal akses 14-6-2013

[SUS-11] Susanto, Ardian. Sistem Informasi Penggajian Karyawan Berbasis Web Pada Kejaksaan Negeri Tangerang. Jakarta : Universitas Mercu Buana.

[UPT-13] Unit Pelaksana Teknis Pusat Komputer. 2013. Modul Pelatihan Website. Diakses dari http://www.unila.ac.id/wp-content/uploads/2013/05/MODUL-PELATIHAN-WEBSITE-WORDPRESS-SECARA-OFFLINE-2013.pdf. Tanggal akses 14-6-2013

LAMPIRAN

Lampiran 1 : Kode Program

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="shortcut icon"
href="../../assets/ico/favicon.ico">

    <title>Theme Template for Bootstrap</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- Bootstrap theme -->
    <link href="css/bootstrap-theme.min.css"
rel="stylesheet">

    <!-- Just for debugging purposes. Don't actually copy
this line! -->
    <!--[if lt IE 9]><script src="../../assets/js/ie8-
responsive-file-warning.js"></script><![endif]-->

    <!-- HTML5 shim and Respond.js IE8 support of HTML5
elements and media queries -->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.j
s"></script>
      <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.mi
n.js"></script>
    <![endif]-->
  </head>

     <?php
     $link = mysqli_connect('localhost', 'root', '',
'csv_db');
     if($_POST){
     $query1 = "SELECT count(*) jumlah1 FROM
".$_POST['dataLatih'];
     $result1 = mysqli_query($link, $query1);

     $query2 = "SELECT count(*) jumlah2 FROM
".$_POST['dataUji'];
```

```
    $result2 = mysqli_query($link, $query2);

    $query3 = "SELECT count(*) jumlah3 FROM
".$_POST['dataLatih']." where `Label` = '0'";
    $result3 = mysqli_query($link, $query3);

    $query4 = "SELECT count(*) jumlah4 FROM
".$_POST['dataLatih']." where `Label` = '1'";
    $result4 = mysqli_query($link, $query4);
    }
?>
  <body role="document">

   <div class="container theme-showcase" role="main">

     <div class="page-header" style="margin: 20px 0 20px
!important;">
       <h1>Klasifikasi Pengguna Software Proxy</h1>
     </div>
      <div class="row">
          <div class="col-lg-6">
               <div class="panel panel-primary">
            <div class="panel-heading">
              <h3 class="panel-title">Proses</h3>
            </div>
            <div class="panel-body">
              <form class="form-horizontal"
action="home.php" method="post">
                    <fieldset>

                    <!-- Select Basic -->
                    <div class="form-group">
                      <label class="col-md-3 control-
label" for="selectbasic">Data Latih</label>
                      <div class="col-md-8">
                      <?php
                          //$link =
mysqli_connect('localhost', 'root', '', 'csv_db');
                       $qry1 = "SELECT table_name FROM
information_schema.tables WHERE table_type =  'base table'
AND table_schema =  'csv_db' and table_name
in('ultrasurf','hotspotshield','freegate')";
                          $rslt1 = mysqli_query($link,
$qry1);
                    ?>
                          <select id="" name="dataLatih"
class="form-control">
                          <option>--Pilih Data Latih--
</option>
                            <?php
```

```
                                while ($temp1 =
mysqli_fetch_array($rslt1)){
                                    $dta1 = $temp1;
                                    echo "<option
value='".$dta1['table_name']."'>".$dta1['table_name']."</opt
ion>";
                                    } ?>
                                </select>
                            </div>
                        </div>

                        <!-- Select Basic -->
                        <div class="form-group">
                          <label class="col-md-3 control-
label" for="">Data Uji</label>
                            <div class="col-md-8">
                            <?php

                             $qry = "SELECT table_name FROM
information_schema.tables WHERE table_type =  'base table'
AND table_schema =  'csv_db' limit 40";
                                $rslt = mysqli_query($link,
$qry);
                            ?>
                                <select id="" name="dataUji"
class="form-control">
                                <option>--Pilih Data Uji--
</option>
                                <?php


                                while ($temp =
mysqli_fetch_array($rslt)){
                                    $dta = $temp;
                                    echo "<option
value='".$dta['table_name']."'>".$dta['table_name']."</optio
n>";
                                    } ?>
                                </select>
                            </div>
                        </div>

                        <!-- Button -->
                        <div class="form-group">
                          <label class="col-md-3 control-
label" for=""></label>
                            <div class="col-md-4">
                                <input id="" name=""
type="submit" class="btn btn-primary" />
                            </div>
                        </div>

                        </fieldset>
```

```
                    </form>
                  </div>
                </div>
                 </div>
                 <div class="col-lg-6">
                      <div class="panel panel-primary">
                  <div class="panel-heading">
                    <h3 class="panel-title">Hasil</h3>
                  </div>
                  <div class="panel-body">
                    <form class="form-horizontal">
                            <fieldset>

                            <!-- Form Name -->
                            <legend>Jumlah Data Latih</legend>

                            <!-- Text input-->
                            <div class="form-group">
                              <label class="col-md-4 control-
label" for="textinput">Data Latih : </label>
                              <div class="col-md-8">
                              <p style="margin-top:7px">
                              <?php
                                  if($_POST){
                                    while ($temp =
mysqli_fetch_array($result1)){
                                        $dataL = $temp;
                                        echo $dataL['jumlah1'];
                                        }
                                  }
                                  ?>
                              </p>

                              </div>
                            </div>

                            <!-- Text input-->
                            <div class="form-group">
                              <label class="col-md-4 control-
label" for="textinput">Data Uji : </label>
                              <div class="col-md-8">
                              <p style="margin-top:7px">
                            <?php
                            if($_POST){
                            while ($temp =
mysqli_fetch_array($result2)) {
                            $dataU = $temp;
                                  echo $dataU['jumlah2'];
                                  }
                            }?>
                              </p>
```

```
                    </div>
                  </div>

                  <!-- Text input-->
                  <div class="form-group">
                    <label class="col-md-4 control-
label" for="textinput">Pengguna Normal : </label>
                    <div class="col-md-8">
                    <p style="margin-top:7px">
                        <?php
                        if($_POST){
                              while ($temp =
mysqli_fetch_array($result3)){
                                  $data0 = $temp;
                                  echo
$data0['jumlah3'];
                                  }
                             }?>
                    </p>

                    </div>
                  </div>

                  <!-- Text input-->
                  <div class="form-group">
                    <label class="col-md-4 control-
label" for="textinput">Pengguna Proxy : </label>
                    <div class="col-md-8">
                    <p style="margin-top:7px">
                   <?php
                    if($_POST){
                   while ( $temp =
mysqli_fetch_array($result4)){
                      $data1 = $temp;
                  echo $data1['jumlah4'];
                      }
                      } ?>
                    </p>

                    </div>
                  </div>

                  <!-- Form Name -->
                  <legend>Hasil</legend>

                  <!-- Text input-->
                  <div class="form-group">
                    <label class="col-md-4 control-
label" for="textinput">Total Proxy : </label>
                    <div class="col-md-8">
                    <p style="margin-top:7px">
                    <?php
```

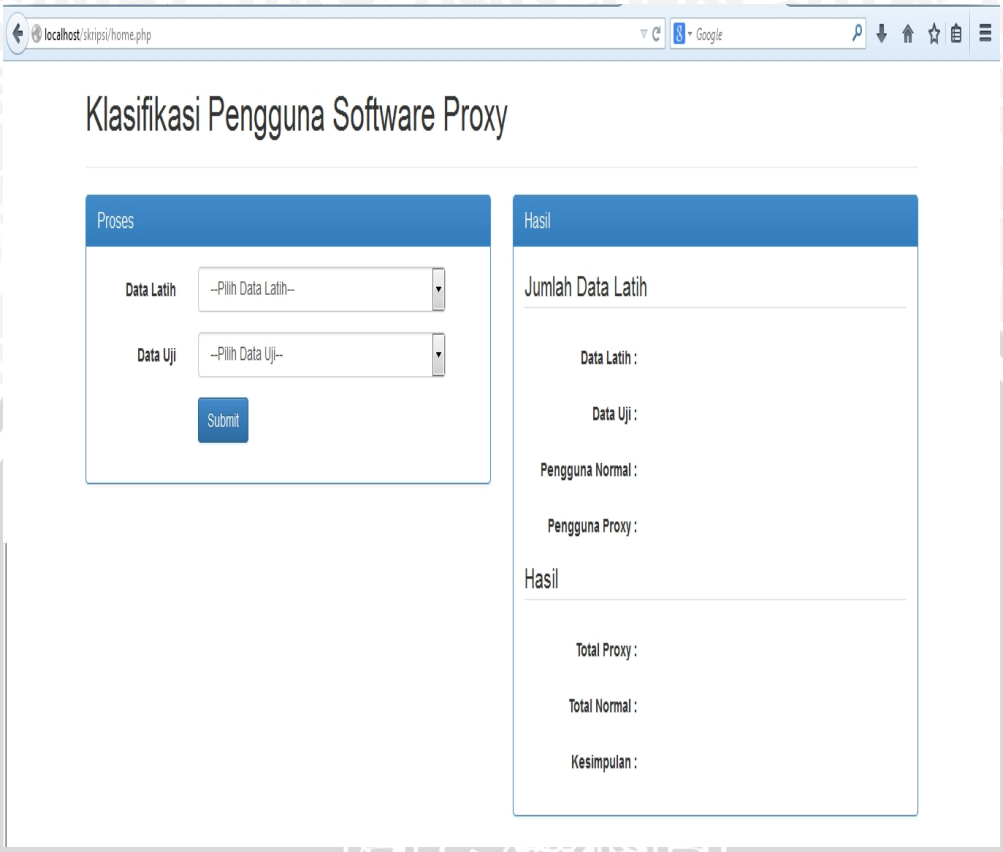
```
                        if($_POST){
                        $queryNo = "select No from
".$_POST['dataUji']."";
$resultNo = mysqli_query($link, $queryNo);

$total1=0;
$total0=0;


while($row = mysqli_fetch_array($resultNo)){

      $No = $row['No'];

      #Menghitung IP Source

      $query15 = "SELECT `Source` FROM ".$_POST['dataUji']."
where `No` = $No";
      $result15 = mysqli_query($link, $query15);
      $dataSrcuji = mysqli_fetch_array($result15);

      $query16 = "SELECT count(*) jumlah5 FROM
".$_POST['dataLatih']." where `Label` = '1' and `Source` =
'".$dataSrcuji['Source']."'";
      $result16 = mysqli_query($link, $query16);
      $dataSrc1 = mysqli_fetch_array($result16);

      $query17 = "SELECT count(*) jumlah6 FROM
".$_POST['dataLatih']." where `Label` = '0' and `Source` =
'".$dataSrcuji['Source']."'";
      $result17 = mysqli_query($link, $query17);
      $dataSrc0 = mysqli_fetch_array($result17);

      #Menghitung IP Destination

      $query18 = "SELECT `Destination` FROM
".$_POST['dataUji']." where `No` = $No";
      $result18 = mysqli_query($link, $query18);
      $dataDestuji = mysqli_fetch_array($result18);

      $query19 = "SELECT count(*) jumlah7 FROM
".$_POST['dataLatih']." where `Label` = '1' and
`Destination` = '".$dataDestuji['Destination']."'";
      $result19 = mysqli_query($link, $query19);
      $dataDest1 = mysqli_fetch_array($result19);

      $query20 = "SELECT count(*) jumlah8 FROM
".$_POST['dataLatih']." where `Label` = '0' and
`Destination` = '".$dataDestuji['Destination']."'";
      $result20 = mysqli_query($link, $query20);
      $dataDest0 = mysqli_fetch_array($result20);

      #Menghitung protokol
```

```
        $query5 = "SELECT `Protocol` FROM
".$_POST['dataUji']." where `No` = $No";
        $result5 = mysqli_query($link, $query5);
        $dataProuji = mysqli_fetch_array($result5);

        $query6 = "SELECT count(*) jumlah9 FROM
".$_POST['dataLatih']." where `Label` = '1' and `Protocol` =
'".$dataProuji['Protocol']."'";
        $result6 = mysqli_query($link, $query6);
        $dataPro1 = mysqli_fetch_array($result6);

        $query7 = "SELECT count(*) jumlah10 FROM
".$_POST['dataLatih']." where `Label` = '0' and `Protocol` =
'".$dataProuji['Protocol']."'";
        $result7 = mysqli_query($link, $query7);
        $dataPro0 = mysqli_fetch_array($result7);

        $nilai1 = $data1['jumlah4']/$dataL['jumlah1'];
        $nilai0 = $data0['jumlah3']/$dataL['jumlah1'];

        $nilaiSrc1 = $dataSrc1['jumlah5']/$data1['jumlah4'];
        $nilaiSrc0 = $dataSrc0['jumlah6']/$data0['jumlah3'];

        $nilaiDest1 = $dataDest1['jumlah7']/$data1['jumlah4'];
        $nilaiDest0 = $dataDest0['jumlah8']/$data0['jumlah3'];

        $nilaipro1 = $dataPro1['jumlah9']/$data1['jumlah4'];
        $nilaipro0 = $dataPro0['jumlah10']/$data0['jumlah3'];

        $nilaitotal1 = $nilai1 * $nilaiSrc1 * $nilaiDest1 *
$nilaipro1;
        $nilaitotal0 = $nilai0 * $nilaiSrc0 * $nilaiDest0 *
$nilaipro0;

        //$total1[]=$nilaitotal1;
        //$total0[]=$nilaitotal0;
        $total1=$total1+$nilaitotal1;
        $total0=$total0+$nilaitotal0;


}

$totala=$total1/$dataU['jumlah2'];
echo $totala;
                        }
                         ?>
                        </p>

                        </div>
                      </div>

                      <!-- Text input-->
```

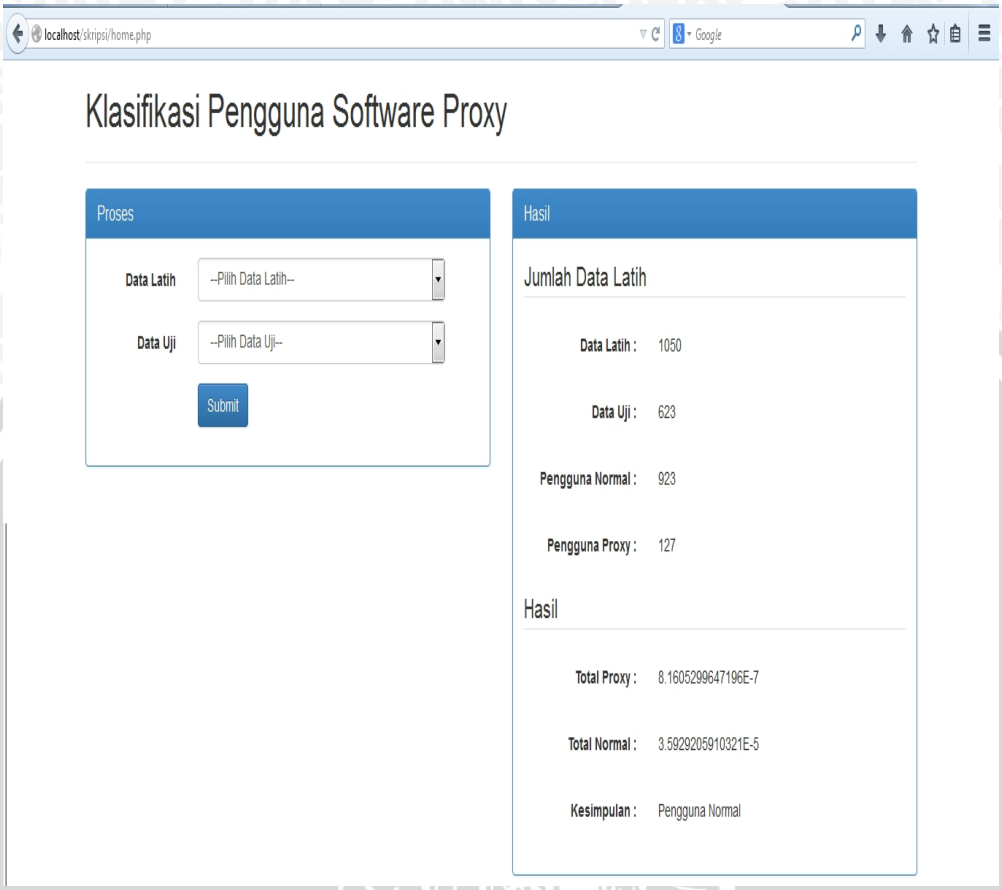
```
                    <div class="form-group">
                      <label class="col-md-4 control-
label" for="textinput">Total Normal : </label>
                      <div class="col-md-8">
                      <p style="margin-top:7px">
                     <?php
                     if($_POST){
                     $totalb=$total0/$dataU['jumlah2'];
                     echo $totalb;} ?>
                      </p>

                      </div>
                    </div>

                    <div class="form-group">
                      <label class="col-md-4 control-
label" for="textinput">Kesimpulan : </label>
                      <div class="col-md-8">
                      <p style="margin-top:7px">
                     <?php
                     if($_POST){
                     if ($totala >= $totalb)
                          {
                                echo "Pengguna Software
Proxy"; echo '<br />';
                                //echo "Pengguna Software
Proxy - Pengguna Normal = ".$nilai1-$nilai0; echo '<br />';
                                //echo "IP Source = ".
$nilaiSrc1-$nilaiSrc0; echo '<br />';
                                //echo "IP Destination = ".
$nilaiDest1-$nilaiDest0; echo '<br />';
                                //echo "Protokol = ".
$nilaipro1-$nilaipro0; echo '<br />';
                                //echo "Total = ".
$nilaitotal1-$nilaitotal0; echo '<br />';
                          }
                          else
                          {
                                echo "Pengguna Normal";
echo '<br />';
                                //echo "Pengguna Normal -
Pengguna Software Proxy = ". $nilai0-$nilai1; echo '<br />';
                                //echo "IP Source = ".
$nilaiSrc1-$nilaiSrc0; echo '<br />';
                                //echo "IP Destination = ".
$nilaiDest1-$nilaiDest0; echo '<br />';
                                //echo "Protokol = ".
$nilaipro1-$nilaipro0; echo '<br />';
                                //echo "Total = ".
$nilaitotal1-$nilaitotal0; echo '<br />';
                          }
                          }?>
```

```
                                </p>

                                </div>
                            </div>

                            </fieldset>
                            </form>
                        </div>
                    </div>
                    </div>
                </div>


    </div> <!-- /container -->


    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load
faster -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jqu
ery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>

  </body>
  <?php
  if($_POST)
  {
      mysqli_close($link);
      }
      ?>
</html>
```

Lampiran 2 : Tampilan Awal Program



Lampiran 3 : Tampilan Untuk Pengguna Normal



Lampiran 4 : Tampilan Untuk Pengguna *Tunneling Software*