Desain dan Implementasi Web Proxy dan VPN Akses (Studi Kasus di Undip)

Adian Fatchur Rochim, Andrian Satria Martiyanto

***Abstract*—The development of computer and network technologies lead many organizations to expand its network. The intranet network is used initially to connect clients with geographically dispersed enterprise networks, but when it is need a more extensive and huge network intranets are not possible anymore, it is necessary development of a secure private network that uses Internet network. Methodology of research for this final task include the study of literature, design systems, and testing of the system. In the literature study used research methods from the literature study reference books that are related. The design of this final task using OpenVPN and Glype Proxy to access the local network from the Internet. Last is the testing of this network system, where in this stage the network system will be tested in order to create a reliable network and secure. The result obtained is a network system that allows users from outside the network can access some or all services available on the local network as still in a network, from anywhere via the Internet. For Internet users to become members of this private network they must go through the process of authentication, the data available on the private network is encrypted so it can not be opened except by the members of this private networks.**

***Index Terms*—Internet, intranet, VPN, Web Proxy**

I. PENDAHULUAN

PERKEMBANGAN teknologi komputer yang semakin pesat mengakibatkan badan usaha maupun lembaga akademik mengiplementasikan teknologi ini untuk banyak keperluan-keperluannya. Sebanding dengan bertambahnya fungsi teknologi komputer, bertambah pula keperluan akan luas jaringan komputer yang diperlukan oleh badan tersebut agar setiap anggota dari badan tersebut dapat menggunakan layanan-layanan teknologi yang disediakan.

Permasalahan muncul saat akan menghubungkan pengguna ataupun jaringan lain yang berjauhan atau terpisah secara geografis. Memang bisa dibangun *leased line* atau jaringan WAN (*Wide Area Network*) pribadi, tetapi pembangunan *leased line* ini sangat tidak efektif dari sisi biaya.

Tidak hanya pada badan usaha, lembaga akademis seperti UNDIP juga menghadapi permasalahan yang hampir sama. Layanan SIA (Sistem Informasi Akademik) yang sudah lama dimiliki UNDIP hanya dapat dibuka dari jaringan lokal saja, hal ini dikarenakan meletakkan *server* SIA pada jaringan publik sangatlah beresiko.

Salah satu solusi yang dapat memecahkan masalah ini adalah dengan membangun *Virtual Private Network* (VPN)[1], dengan adanya VPN dimungkinkan seorang pengguna atau jaringan yang berjauhan dapat berhubungan seperti dalam satu jaringan lokal. Sedangkan untuk aplikasi *web* dapat digunakan aplikasi *web proxy* untuk mengakses layanan lokal yang berbasis *web*.

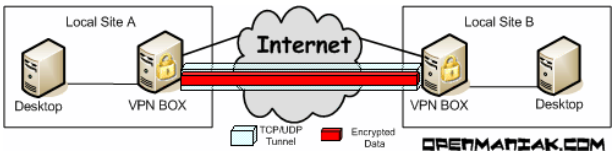
Tujuan dari penelitian ini adalah mempelajari penggunaan, cara kerja dan fungsi dari OpenVPN[2, 3, 4] dan Glype Proxy[5] untuk mengakses aplikasi-aplikasi lokal dari jaringan publik, dan mengimplementasikannya pada jaringan Universitas Diponegoro.

Agar pembahasan atau analisis tidak melebar dan terarah, maka permasalahan dibatasi pada: 1) menggunakan Linux sebagai sistem operasi, *2) server* VPN menggunakan perangkat lunak open source OpenVPN, 3) *Web* proxy yang memanfaatkan aplikasi open source Glype Proxy dan dibangun di atas *Secure Socket Layer*(SSL), 4) Otentikasi pada OpenVPN dan Glype Proxy, 5) Tidak membahas pemrograman Glype Proxy, dan 6) Impementasi untuk jaringan di Universitas Diponegoro.

II. LANDASAN TEORI

A. Sistem VPN

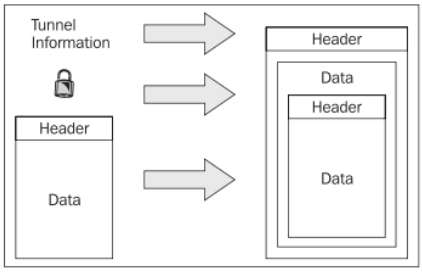
Virtual Private Network merupakan suatu cara untuk membuat sebuah jaringan bersifat *private* dan aman dengan memanfaatkan jaringan publik seperti *internet*[1, 6, 7]. Data dalam jaringan tersebut tidak dapat diketahui oleh pengguna lain di jaringan publik karena data tersebut dilewatkan pada *tunnel* yang dibentuk oleh VPN, seperti diperlihatkan dalam Gambar 1. *Tunnel* merupakan suatu mekanisme enkripsi-deskripsi data. Data yang dikirim melalui jalur publik adalah data terenkripsi yang hanya bisa dibuka oleh ujung dari *tunnel* yang memiliki kunci untuk medeskripsi data tersebut. Paket yang dikirim lewat VPN diperlihatkan dalam Gambar 2.



Gambar 1. *Tunneling* pada VPN melewati jarngan *internet*

Adian Fatchur Rochim: Program Studi Sistem Komputer, Universitas Diponegoro, Jl. Prof. Soedarto, Semarang, Indonesia, email: adian@undip.ac.id

Andrian Satria Martiyanto: Jurusan Teknik Elektro konsentrasi Komputer dan Informatika, Universitas Diponegoro, Jl. Prof. Soedarto, Semarang, Indonesia



Gambar 2. Paket yang dikirim pada VPN

Dengan adanya mekanisme *Tunneling* VPN[8]dapat dimanfaatkan pada beberapa kondisi, yaitu :

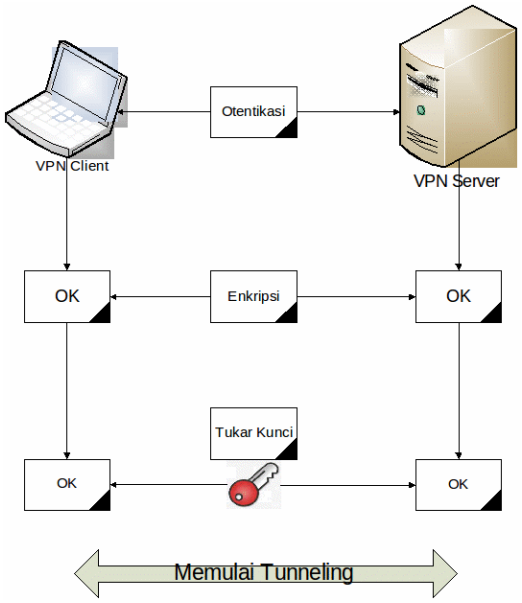
1) *Remote Access Client Connections*
   VPN berfungsi untuk mendukung *remote* dari komputer rumah ke komputer kantor atau sebaliknya dengan memanfaatkan jaringan *internet.*
2) *LAN-to-LAN internetworking*
   Dengan memanfaatkan VPN 2 jaringan atau lebih yang terpisah letak dapat digabungkan menjadi seperti dalam satu jaringan, dengan syarat semua jaringan tersebut tersambung ke jaringan *internet.*
3) Kontrol akses dalam suatu *intranet*
   VPN dapat dimanfaatkan untuk mengamankan pengguna suatu jaringan intranet, dengan tujuan agar data yang dikirimkan pengguna tersebut tidak diketahui pengguna lain pada jaringan yang sama. Fungsi ini sering dimanfaatkan pada jaringan yang terbuka untuk umum, contohnya jaringan nirkabel yang akhir-akhir ini menjamur di berbagai tempat.

B. Protocol Tunneling VPN

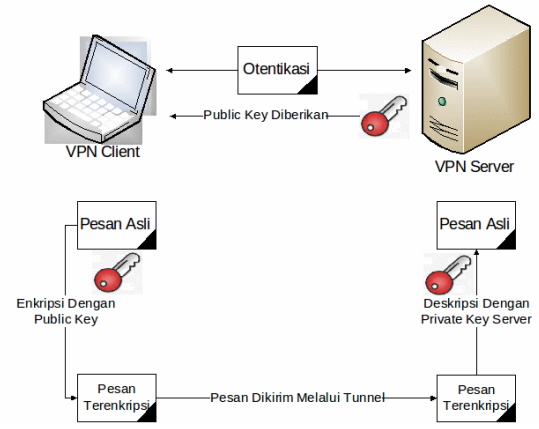
Pengembang sistem VPN sangat beragam, dimana pada umumnya tidak kompatibel antara satu dan lain. Perbedaan yang dimiliki oleh jenis-jenis VPN tersebut antara lain adalah pada *protocol-protocol tunneling* yang digunakan, jenis dari protocol-protocol tersebut adalah :

1) *Point-to-Point Tunneling Protocol* (PPTP)
   Protokol tunneling jenis ini banyak digunakan pada produk-produk microsoft, protokol ini berjalan pada lapisan ke dua model layer OSI.
2) *Layer Two Tunneling Protocol* (L2TP)
   Awalnya bernama *Layer 2 Forwarding*, yang kemudian dikembangkan dengan menambahkan kelebihan-kelebihan PPTP. Protokol jenis ini dikembangkan oleh perusahaan CISCO[9, 4]. Seperti namanya protocol ini berjalan pada lapisan ke dua OSI.
3) *Internet Protocol Security* (IPsec)
   VPN dengan Protokol ini paling banyak jenisnya, protokol ini digunakan pada produk-produk microsoft, cisco dan berbagai vendor jaringan lain. Merupakan pengembangan dari dua protokol *tunneling* VPN sebelumnya. Protocol ini bekerja pada lapisan ke 3 model OSI.
4) *Secure Socket Layer* (SSL)
   SSL sering juga disebut *Transport Layer Security*(TLS) karena bekerja pada lapisan ke 4 model OSI yaitu lapisan *transport*[10]. Protokol ini belum banyak digunakan pada vendor VPN, tetapi dengan fitur-fitur yang dimilikinya protokol jenis ini memiliki potensi berkembang yang sangat besar. Protokol ini banyak digunakan pada prodok-produk *Open Source*.

Tunneling dengan IPSec dan SSL memiliki perbedaan baik dalam proses enkripsi maupun otentikasi. Perbedaan tersebut dapat dilihat pada Gambar 3 dan Gambar 4.



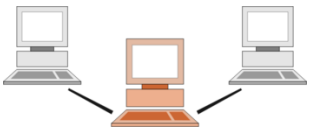
Gambar 3. Model *Tunneling* IPSec VPN



Gambar 4. Model *Tunneling* dengan *protocol* SSL

C. OpenVPN

OpenVPN[3], adalah salah satu jenis aplikasi penyedia layanan VPN yang gratis. OpenVPN menggunakan SSL untuk menangani *tunneling.* OpenVPN memiliki dukungan yang luas terhadap berbagai macam produk-produk *opensource,* terutama untuk aplikasi-aplikasi



Gambar 5. Proxy server

yang menangani proses enkripsi SSL/TLS dan Otentikasi. Secara *default,* OpenVPN menggunakan *library* OpenSSL[11] untuk membangun *tunnel.*

OpenSSL adalah suatu aplikasi OpenSource yang menangani protokol SSL/TLS, Aplikasi ini dapat mendukung beberapa mekanisme enkripsi dan juga otentikasi dengan memanfaatkan tanda pengenal berupa sertifikat. Aplikasi OpenSSL banyak digunakan untuk membangun protokol HTTPS[12] dan juga tunneling dalam VPN.

OpenVPN memiliki dukungan yang terbatas terhadap mekanisme otentikasi, terutama dalam hal penyimpanan data pengguna, untuk menutupi kekurangan ini bisa digunakan FreeRADIUS[13] . Aplikasi FreeRADIUS merupakan aplikasi yang menangani protokol RADIUS (*Remote Authentication Dial In User Service*) yaitu suatu protokol yang menangani fungsi-fungsi AAA (*Authentication, Authorization, Accounting*)[14]. FreeRADIUS dapat bekerja dengan basis data MySQL sebagai penyimpan data untuk keperluan otentikasi, otoritas maupun akuntingnya.

D. Web Proxy

Dalam jaringan komputer, *server* proxy adalah server yang bertindak sebagai perantara untuk melayani permintaan dari klien yang mencari sumber daya dari *server* lain, seperti diperlihatkan dalam Gambar 5. *Server* proxy akan menghubungi *server* yang memiliki sumber daya dan meminta data yang diminta oleh klien, dengan cara ini *server* yang memiliki data hanya mengetahui bahwa yang meminta datanya adalah *server* proxy.

Sebuah proxy *server* memiliki dua tujuan:

- Untuk menjaga mesin di baliknya anonymous atau tak dikenali (terutama untuk alasan keamanan) baik yang meminta maupun penyedia layanan
- Untuk mempercepat akses ke sumber daya (*caching*). Biasanya digunakan untuk cache halaman *web* dari *web server.*

Proxy bisa berupa aplikasi *server*, sebagai contoh squid, atau bisa juga hanya berupa aplikasi berbasis *web* yang menyediakan layanan-layanan proxy yang tentu saja dikembangakan dengan bahasa pemrograman berbasis *web* seperti PHP. *Web* proxy biasanya digunakan untuk menyembunyikan identitas pengguna internet dan berguna untuk melewati batasan sensor. Salah satu aplikasi proxy yang berbasis *web* adalah Glype Proxy.

E. Glype Proxy

Glype Proxy, merupakan aplikasi proxy berbasis *web* yang bersifat gratis dan *open source*. Aplikasi ini ditulis dengan bahasa PHP, untuk menjalankan aplikasi ini dibutuhkan *web server* yang mendukung bahasa PHP dan cURL. cURL yaitu piranti baris perintah untuk mentransfer file dengan sintaks URL, mendukung FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, dan LDAPS FILE. cURL mendukung sertifikat SSL, HTTP POST, HTTP PUT, FTP uploading, upload berbasis HTTP form, proxy, cookies, otentikasi user/password, kerberos, proxy tunneling dan banyak yang lainnya. PHP memiliki library untuk menjalankan perintah-perintah cURL ini.

Glype Proxy memiliki banyak fasilitas antara lain, dapat langsung dipakai tanpa proses instalasi, memiliki halaman admin yang dapat mengkonfigurasi fasilitas *web* proxy, memiliki fasilitas caching sehingga dapat mempercepat pencarian layanan, mendukung javascript, memiliki fasilitas untuk mengenkripsi URL *server* yang dituju sehingga pengguna tidak dapat menambahkan kode-kode pada URL tersebut.

Selain untuk menyembunyikan pengguna proxy, Glype Proxy juga bisa digunakan untuk menyembunyikan dan mengamankan suatu *server*, tentu saja jika letak *server* tersebut harus dibelakang *server web* proxy. Yaitu dengan memanfaatkan fasilitas enkripsi URL dan *web* page source, serta kontrol akses yang juga tersedia pada Glype Proxy

Saat ini Aplikasi Glype Proxy terbaru adalah versi 1.1, untuk mendapatkan program ini bisa melalui alamat *web* www.glype.com. Program ini juga telah banyak digunakan di berbagai situs di internet sebagai pendukung layanan mereka.

F. HTTPS

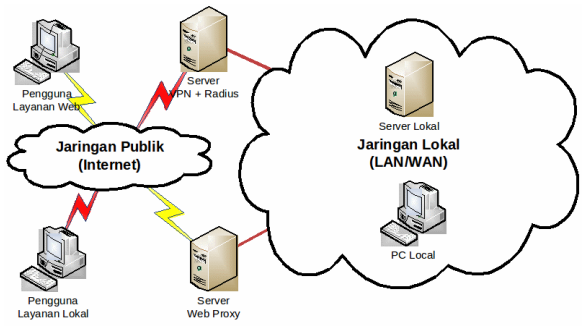
Pada *Web Server* bisa ditambahkan plugin SSL untuk menghasilkan protokol HTTPS. HTTPS adalah versi aman dari HTTP, protokol komunikasi dari *World Wide Web*. Ditemukan oleh Netscape Communications Corporation untuk menyediakan otentikasi dan komunikasi tersandi dan penggunaan dalam e-commerce.

Selain menggunakan komunikasi plain text, HTTPS menyandikan data sesi menggunakan protokol SSL atau protokol TLS. Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks*. Pada umumnya port HTTPS adalah 443.

Tingkat keamanan tergantung pada ketepatan dalam mengimplementasikan pada *browser web* dan perangkat lunak *server* dan didukung oleh algorithma penyandian yang aktual. Oleh karena itu, pada halaman *web* digunakan HTTPS, dan URL yang digunakan dimulai dengan ”https://” bukan dengan ”http://”.Fasilitas HTTPS bisa ditambahkan pada *server web* yang digunakan oleh *web* proxy sehingga menambahkan fungsi enkripsi pada aplikasi *web* proxy tersebut.

III. PERANCANGAN SISTEM

Dalam perancangan ini dibutuhkan dua *server* yaitu *server* untuk OpenVPN dan *Web* Proxy[15], keduanya memiliki fungsi yang hampir sama yaitu agar pengguna



Gambar 6. Skema jaringan secara penuh

dari luar jaringan lokal dapat mengakses layanan-layanan yang ada di *server* lokal. Skema jaringan secara penuh diperlihatkan dalam Gambar 6. Penggunaan dua *server* tersebut ditujukan untuk membatasi akses untuk tiap-tiap *user*, dimana dalam implementasi di UNDIP ini harus ada pemisahan antara *user* yang hanya memerlukan layanan *web* saja dan *user* yang memerlukan semua layanan jaringan yang ada yang mengharuskan *user* tersebut terdata sebagai pengguna lokal.

Kedua *server* tersebut bisa juga digabungkan untuk menghemat *server*, karena fungsi VPN dan *Web* Proxy tidak saling mengganggu. Aplikasi-aplikasi pendukung untuk kedua fungsi tersebut hampir sama sehingga bisa digunakan satu *server* saja.

A. *Enkripsi*

Enkripsi pada *web* proxy dan OpenVPN menggunakan library yang dimiliki OpenSSL. Digunakan kunci publik dan kunci privat sebagai media enkripsi dan deskripsinya. Pada *web* proxy kunci publik diberikan melalui fitur sertifikat *server* yang akan dicek keabsahannya oleh browser klien sedangkan pada OpenVPN digunakan sertifikat klien yang juga berfungsi untuk otentikasi klien oleh *server*.

B. *Otentikasi*

Otentikasi pada OpenVPN memanfaatkan sertifikat klien, sertifikat ini digunakan sebagai pengenal klien pada *server*. Sertifikat hanya dihasilkan oleh *server* yang bersangkutan. Selain sertifikat klien juga harus memiliki username dan password yang sesuai, pada *server* pencocokan username password ini diproses pada aplikasi FreeRADIUS, data user disimpan pada basis data MySQL.

Otentikasi pada Glype Proxy hanya menggunakan metode session yang dimiliki bahasa pemrograman PHP. Diperlukan data username dan password untuk masuk ke halaman utama Glype Proxy, data username dan password didapat dari basis data SIA UNDIP.

IV. Implementasi dan Pengujian

Sebelum menanamkan sistem VPN dan *Web Proxy* perlu disiapkan terlebih dahulu sistem operasi *server*nya. Sistem operasi yang dipilih adalah turunan dari Debian, yaitu Ubuntu *Server* 8.10[16]. Alasan digunakan turunan dari debian yaitu Ubuntu adalah, karena memiliki repository *server* diberbagai negara termasuk Indonesia yang memungkinkan melakukan update sistem maupun paket-paket terbaru. Versi 8.10 dipilih karena versi tersebut memiliki paket-paket untuk dukungan RADIUS yang lebih baik, dibanding versi sesudahnya. Sama dengan instalasi kebanyakan sistem operasi, instalasi Ubuntu menggunakan paket instalasi yang berupa CD.

Setelah sistem operasi terinstall dengan baik, dilakukan pengaturan IP pada *server* tersebut serta daftar DNS agar *server* tersebut dapat terhubung dengan internet dan jaringan lokal. Setelah *server* dapat terhubung dengan baik ke internet dan jaringan lokal, pengaturan terhadap *server* tersebut dapat dilakukan melalui fasilitas *remote,* hal ini memungkinkan *server* dapat di akses darimana saja.

Pada Ubuntu instalasi program dapat dilakukan dengan menjalankan perintah "`apt-get install nama program`>". Dengan syarat file `/etc/apt/source.list` telah berisi daftar repository. Penyedia repository yang digunakan pada implementasi ini adalah repo.undip.ac.id.

Aplikasi-aplikasi yang dipasangkan pada *server* untuk memungkinkan jalannya sistem VPN dengan OpenVPN dan *Web* Proxy dengan Glype Proxy yaitu :

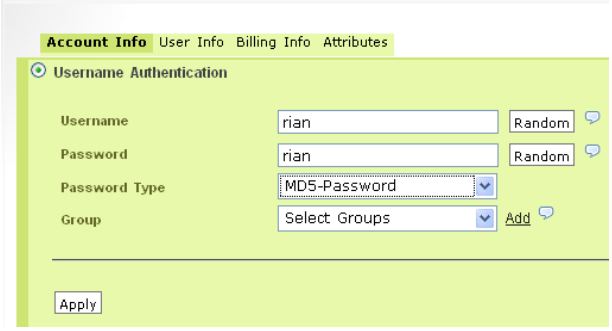
- MySQL untuk penyimpanan data, paket yang perlu di install adalah mysql-*server*-5.0
- Apache sebagai *Web Server* yang mendukung PHP, cukup dengan meng-install paket phpmyadmin
- RADIUS sebagai aplikasi untuk Otentikasi, dengan nama paket freeradius dan freeradius-mysql agar RADIUS dapat bekerjasama dengan MySQL
- OpenSSL sebagai pustaka enkripsi dan deskripsi, dengan nama paket openssl dan ssl-cert untuk membuat sertifikat.

Aplikasi OpenVPN dan Glype Proxy bisa ditambahkan setelah semua aplikasi di atas terpasang dengan baik. Untuk menjalankan sistem VPN dan *Web* Proxy, perlu dilakukan beberapa pengaturan pada *server* serta penambahan plugin-plugin.

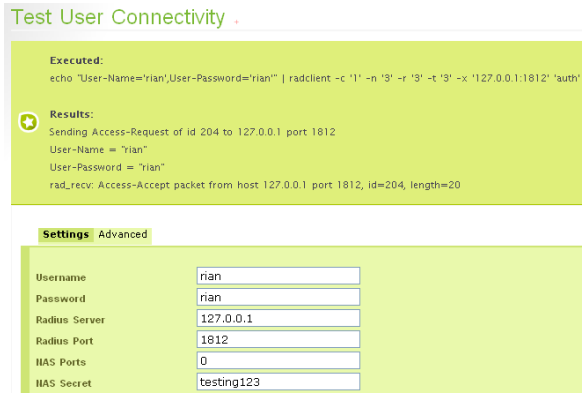
A. *Konfigurasi FreeRADIUS*

Pada FreeRADIUS, terdapat aplikasi DaloRADIUS untuk memudahkan pengaturannya. Aplikasi DaloRADIUS bisa didapat di www.daloradius.com. DaloRADIUS merupakan aplikasi berbasi *web* yang memudahkan pembuatan basis data pada konfigurasi RADIUS, melihat status *server*, menambah user, serta banyak lagi fungsi lain, seperti diperlihatkan dalam Gambar 7, 8 dan 9.

Gambar 7. Pengecekan status *server* dengan DaloRADIUS



Gambar 8. Input user baru FreeRADIUS melalui DaloRADIUS



Gambar 9. Tes user FreeRADIUS melalui DaloRADIUS

Agar RADIUS dapat bekerja dengan MySQL, ada beberapa pengaturan yang harus ditambahkan. Pengaturan pertama pada file `/etc/freeradius/sql.conf` perlu didaftarkan username dan password MySQL serta nama basis data yang disediakan untuk RADIUS, agar FreeRADIUS diberikan izin untuk mengakses basis data tersebut.

File `/etc/freeradius/clients.conf`, perlu diisi dengan daftar NAS (*Networks Address* Server) yang boleh mengakses *server* RADIUS, jika NAS dan RADIUS berada pada satu *server* data diisikan dengan data localhost. File Selanjutnya adalah `/etc/freeradius/sites-enable/default.conf`. Pada file ini perlu diatur media penyimpanan data RADIUS, yaitu dengan menghilangkan tanda "#" sebelum teks "sql" pada isi file tersebut. Terakhir adalah file `/etc/freeradius/radiusd.conf`, baris 504, ubah kata status proxy_request menjadi no dan pada akhir bagian modules ditambahkan baris berikut.

```
pap {
 authtype = md5
 auto_header = yes
 }
```

B. Pemasangan HTTPS

Protokol HTTPS dapat dihasilkan dari *web server* apache yang dilengkapi dengan modul SSL berupa sertifikat[17, 18, 16]. Pertama perlu dibuat kunci untuk sertifikat dengan perintah

```
#openssl genrsa -des3 -out server.key
    1024
```

Kemudian buat sertifikat dengan perintah

```
#openssl req-new-key server.key --out
    server.csr
#openssl x509 -req -days 365 -in
    server.csr -signkey server.key -
    out server.crt
```

Perlu dibuat sertifikat "insecure" untuk memudahkan pengaktifan sertifikat.

```
#openssl rsa -in server.key -out
    server.key.insecure
#mv server.key server.key.secure
#mv server.key.insecure server.key
```

Baris-baris perintah di atas akan menghasilkan file-file sebagai berikut :

- `server.crt` : sertifikat yang dihasilkan oleh *server*
- `server.csr` : Permintaan penandatanganan sertifikat *server*
- `server.key` : kunci pribadi (*Private*) *server*, yang tidak memerlukan *password* ketika memulai apache
- `server.key.secure` : kunci pribadi *server*, yang memerlukan *password* ketika memulai apache.

Setelah sertifikat telah terbentuk, sertifikat ini perlu dimasukkan pada Apache sebagai modul tambahan. Pertama pindahkan sertifikat dan kunci agar mudah diambil.

```
#mv server.crt /etc/apache2/ssl/certs/
#mv server.key /etc/apache2/ssl/keys/
```

Setelah selesai, modul SSL diaktifkan dengan perintah.

```
#a2enmod ssl
```

Dan dibuatkan satu situs khusus untuk akses HTTPS (HTTP lewat SSL):

```
#cp /etc/apache2/sites-available/
    default /etc/apache2/sites-
    available/ssl
```

Edit berkas `/etc/apache2/sites-available/ssl`, ubah 3 baris teratas menjadi seperti *snippet* berikut ini:

```
NameVirtualHost *:443
VirtualHost *:443
 ServerAdmin webmaster@localhost
 SSLEngine on
 SSLCertificateFile /etc/apache2/ssl/
     cert/server.crt
 SSLCertificateKeyFile /etc/apache2/
     ssl/keys/server.key
 DocumentRoot /var/secure/
 #baris selanjutnya biarkan saja .....
```

Enable situs SSL yang baru dibuat tadi kemudian ubah situs *default* supaya tidak berbenturan dengan situs SSL.

```
# a2ensite ssl
# nano /etc/apache2/sites-available/
    default
```

Ubah 2 baris teratas berkas `/etc/apache2/sites-available/default` menjadi seperti *snippet* berikut ini:

```
NameVirtualHost *:80
VirtualHost *:80
#baris berikutnya biarkan saja.....
```

Dengan cara tersebut akses HTTP dan HTTPS akan dipisahkan baik *port* kerjanya ataupun letak file-file aplikasi *web* yang dijalankan. Apache juga harus mendengarkan *port* 443 untuk menerima permintaan HTTP, untuk itu ubah isi berkas `/etc/apache2/ports.conf` menjadi:

```
Listen 80
Listen 443
```

Simpan berkas tersebut kemudian *reload* Apache:

```
#/etc/init.d/apache2 force-reload
```

Untuk mencoba HTTPS bisa digunakan perintah

```
#nmap –A localhostgrep Apache
```

C. Konfigurasi OpenVPN

Setelah FreeRADIUS sudah dapat berjalan pada *server*, langkah selanjutnya adalah menambahkan aplikasi *OpenVPN* yang akan menangani *tunneling* pada jaringan publik sehingga dapat mengakses jaringan lokal. Agar kedua aplikasi ini dapat bekerja sama perlu ditambahkan suatu *plug-in* yang bernama **radiusplugin**, *plug-in* ini nantinya akan menjadi penghubung antara aplikasi *OpenVPN* dengan FreeRADIUS. Radiusplugin bisa diunduh di alamat http://www.nongnu.org/radiusplugin/radiusplugin_v2.0c.tar.gz.

Selain paket OpenVPN ada paket lain yang harus di pasang agar OpenVPN dan radiusplugin dapat berjalan pada Ubuntu, paket tersebut adalah libgcrypt11-dev dan g++**.** Instalasi radiusplugin dilakukan dengan menjalankan perintah `make` pada folder radiusplugin. Proses instalasi radiusplugin adalah file `radiusplugin.cnf` dan `radiusplugin.so`, kedua file ini perlu dipindahkan ke folder openVPN. File `radiusplugin.cnf` perlu diubah, terutama pada password untuk mengakses radius dan alamat *server* radius dilihat dari letak file `radiusplugin.cnf`.

Langkah selanjutnya yang perlu dilakukan adalah pembuatan *Public Key Infrastructure*(PKI), yang berfungsi sebagai enkripsi data dan otentikasi klien. Untuk membuat PKI ini sudah tersedia *easy-rsa* yang telah disertakan oleh *OpenVPN*:

```
#cp –a /usr/share/doc/OpenVPN/examples
    /easy-rsa /etc
#cd /etc/easy-rsa/2.0/
```

Pembuatan PKI dapat dilakukan dengan menjalankan beris perintah berikut ini pada folder `2.0/`

```
#source ./vars
#./clean-all
#./build-ca
#./build-key-server server
#./build-dh
```

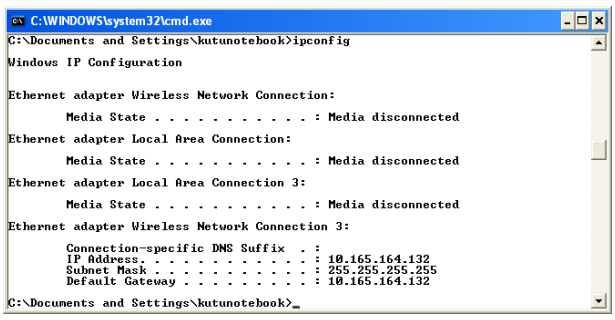
Hasil dari perintah di atas adalah file `dh1024.pem`, `ca.crt`, *`server`*`.crt` dan *`server`*`.key` yang terletak di *folder* `keys`, *folder* `keys` perlu dipindahkan ke folder `OpenVPN`. File `ca.crt` tidak diperlukan *server* dan harus diberikan ke klien sebagai public key.

Untuk menjalankan fasilitas OpenVPN, harus dibuat file *`server`*`.cnf` yang berisi konfigurasi *server* VPN yang dibuat, seperti pengaturan nomor port, protokol yang digunakan, serta metode otentikasi dan enkripsi yang ingin digunakan. File *`server`*`.cnf` harus memanggil file-file radiusplugin dan PKI. File ini harus diletakkan pada folder `OpenVPN`. Menjalankan OpenVPN dapat dilakukan dengan perintah.

```
#/etc/init.d/openvpn start
```

D. Pengujian OpenVPN

Pengujian aplikasi OpenVPN dapat dilakukan melalui komputer klien dengan menambahkan program OpenVPN GUI yang dapat didownload di http://OpenVPN.se. Agar program tersebut dapat berjalan perlu dibuat konfigurasi klien yang sesuai dengan konfigurasi di *server*, serta file ca.crt yang bisa diambil di *server* VPN. Sebelum memulai sambungan ke *server* OpenVPN koneksi internet dari komputer pengguna perlu dicek (Gambar 10).

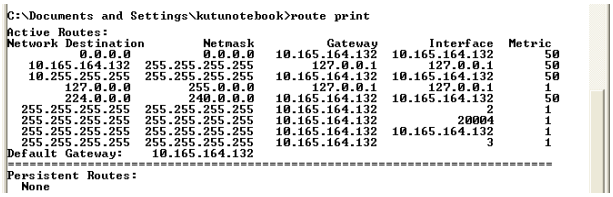


Gambar 10. Konfigurasi IP pada klien dengan jaringan Indosat IM3

Selain alamat IP perlu diketahui juga tabel *routing* dari computer yang telah terhubung ke internet tersebut, seperti diperlihatkan dalam Gambar 11.

Tanda jika OpenVPN GUI sudah berjalan adalah ikon pada toolbar windows berupa dua layar yang jika berwarna merah (lihat Gambar 12). Hal ini menandakan *OpenVPN* GUI sudah berjalan tetapi belum tersambung pada *server* VPN manapun.

Klik kanan pada ikon tersebut, lalu pilih *connect* untuk memulai sambungan (Gambar 13). Akan muncul kotak dialog yang menanyakan *username* dan *password*



Gambar 11. Tabel *routing* klien sebelum sambungan VPN

Gambar 12. Ikon yang menandakan OpenVPN GUI sedang aktif

(Gambar 14), isikan sesuai *username password* yang ada pada FreeRADIUS. Setelah proses penyambungan selesai ikon pada *toolbar* akan berubah menjadi hijau yang menandakan sambungan VPN telah terbentuk (Gambar 15). Bisa dilihat pula alamat IP yang diberikan oleh *server OpenVPN*.

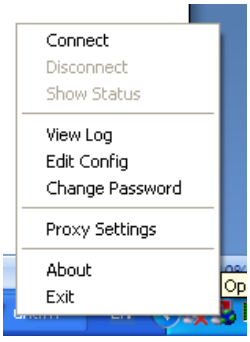
Aplikasi-aplikasi lokal lain, seperti Remote Desktop dan juga File Sharing, bisa diakses dengan memanfaatkan VPN ini (Gambar 16).

Pada komputer yang terhubung VPN telah ditambahkan tabel *routing*. Tabel routing memungkinkan komputer tersebut dapat menghubungi semua komputer dalam jaringan yang ada di tabel *routing*, untuk lebih jelasnya dapat dilihat pada tabel *routing* pada Gambar *17*.

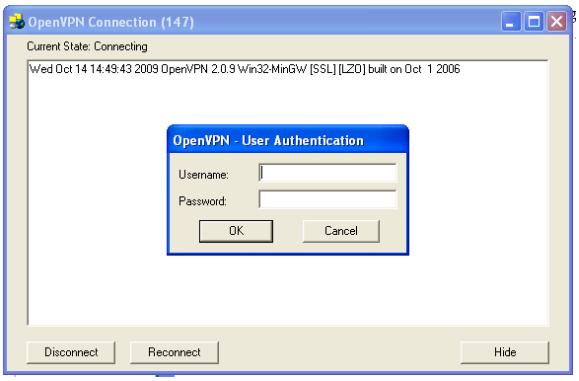
E. Konfigurasi dan Pengujian Glype Proxy

Agar Glype Proxy dapat berjalan, yang perlu dilakukan hanya meletakkan folder Glype Proxy tersebut pada `/var/secure`, yang merupakan folder untuk akses HTTPS. Pada implementasi ini, alamat *server web* proxy adalah sift.undip.ac.id, karena itu untuk mengaksesnya bisa dilakukan dengan mengetikan alamat https://sift.undip.ac.id, di browser.

Pengujian terhadap Glype Proxy dapat dilakukan dengan mengakses alamat *server* Glype Proxy. Saat akses untuk pertama kali akan muncul pesan bahwa sambungan tidak dipercaya, hal ini dikarenakan sertifikat yang dihasilkan *server* menanyakannya terlebih dahulu pada pengguna (Gambar 18). Pilih *Add Exception* untuk melanjutkan proses pemasangan sertifikat.

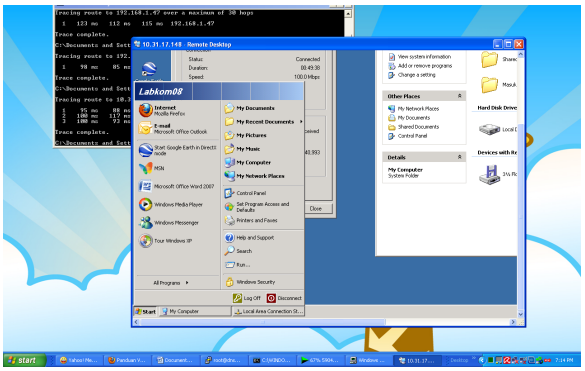


Gambar 13. Memulai sambungan ke OpenVPN

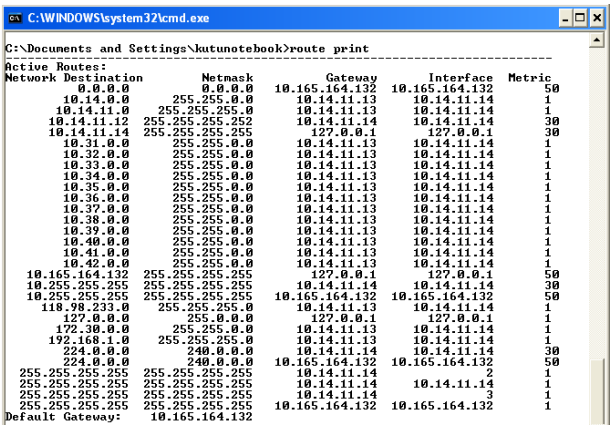


Gambar 14. Otentikasi Pada OpenVPN GUI

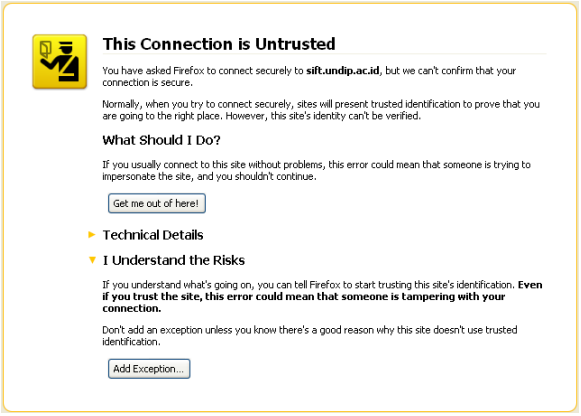
Gambar 15. *Ikon* OpenVPN GUI yang berubah setelah tersambung



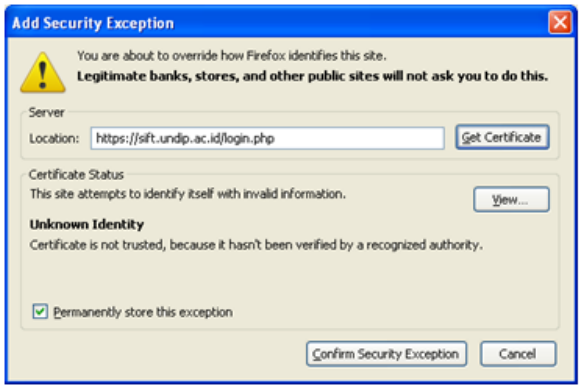
Gambar 16. *Remote Desktop* ke komputer lokal



Gambar 17. Tabel *routing* pada klien setelah sambungan dengan *server* OpenVPN terbentuk



Gambar 18. Halaman yang muncul saat akan mengakses Glype Proxy



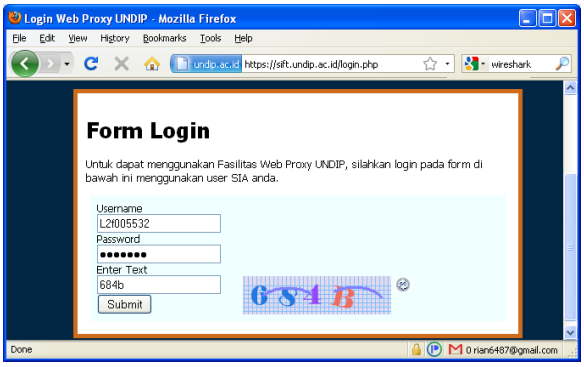
Gambar 19. Pemasangan Sertifikat secara manual

Pilih "*Get Certificate*" lalu "*Confirm Security Exception*" untuk menginstall sertifikat pada *browser*. Contoh diatas adalah untuk *browser* Mozilla Firefox versi 3.5 tiap *browser* memiliki halaman pesan yang berbeda tetapi pada intinya yang harus pengguna lakukan adalah menginstall sertifikat tersebut.

Gambar 19 di atas adalah form login yang diakses tanpa menggunakan protokol SSL, jika form tersebut telah di sumbit, data yang diisikan dapat terlihat di jaringan, di sini penulis menggunakan aplikasi *wireshark*[19] untuk melihat paket-paket data tersebut seperti ditunjukkan dalam Gambar 20.

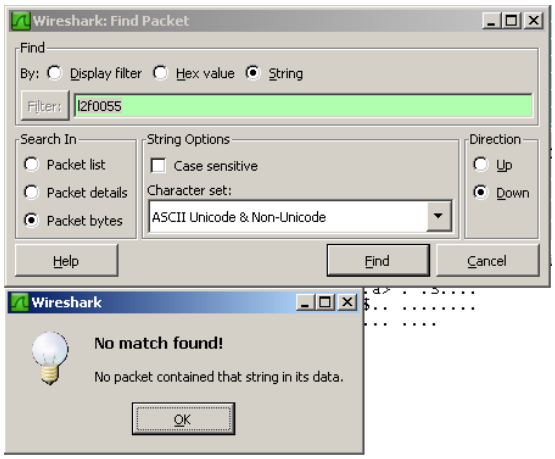
Gambar 20. Paket yang tertangkap oleh Wireshark

Saat paket yang dikirim dilihat melalui wireshark, terlihat bahwa data yang dikirim ke jaringan masih dapat terbaca. Hal ini menjadi suatu celah keamanan karena data *username/password* dapat dilihat. Berbeda jika *web proxy* tersebut dilengkapi dengan modul SSL di *web server*nya, data *username/password* tersebut tidak dapat terlihat di jaringan atasnya.



Gambar 21. Halaman *Login* Glype Proxy menggunakan *protocol* HTTPS

Contoh pada Gambar 21 di atas adalah halaman *login* yang dilengkapi dengan modul SSL, Tanda bahwa suatu halaman *web* dilengkapi dengan modul SSL di *web server*nya adalah adanya tanda gembok pada pojok bawah mesin pencari, alamat URL juga berubah menjadi berwarna biru, menandakan sertifikat yang diinstall kurang keabsahannya, jika sertifikat tersebut dinyatakan sah secara penuh oleh mesin pencari, maka URL tersebut akan berwarna hijau.



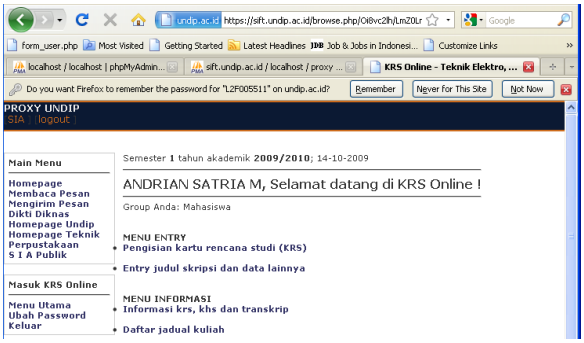
Gambar 22. Paket-paket yang berjalan pada sambungan HTTPS

Saat data *form* tersebut dikirim, paket data yang mengandung *username/password* tidak dapat ditemukan, jadi data aman dan tidak dapat diketahui isinya selain oleh *server* yang memiliki kunci dari data tersebut (Gambar 22).

Jika proses *login* berhasil, yang artinya *username/password* tersebut terdapat pada *server* basis data SIA, selanjutnya *web proxy* akan langsung membuka halaman *sia.undip.ac.id*. Dan jika telah masuk ke *web proxy* ini layanan informasi lokal seperti *sia.ft.undip.ac.id* ataupun layanan lokal lainnya dapat dibuka.

V. KESIMPULAN DAN SARAN

Dari analisa dan pembahasan dapat disimpulkan bahwa:



Gambar 23. Masuk ke SIA FT menggunakan Glype Proxy

1) Penggabungan kerja antara OpenVPN, OpenSSL, FreeRADIUS dan MySQL pada Ubuntu 8.10 dapat berjalan dengan baik, ditandai dengan berjalannya sistem VPN yang dibangun dengan sistem operasi dan aplikasi-aplikasi tersebut.
2) Aplikasi OpenVPN dapat digunakan untuk membangun *tunnel,* yang mampu menjembatani klien di tempat terpisah ke jaringan lokal UNDIP.
3) Dengan tergabung ke VPN, klien pada jaringan internet dapat menggunakan layanan lokal seperti aplikasi *web*, file sharing, dan remote desktop
4) Untuk dapat menjadi anggota VPN, klien harus memiliki sertifikat yang dikeluarkan *server* dan juga harus terdaftar pada *server* basis data.
5) Data yang dilewatkan pada tunnel yang dibentuk oleh OpenVPN tidak dapat diketahui isinya oleh pengguna di antara klien dan *server* OpenVPN yang bukan merupakan anggota tunnel tersebut.
6) Penggabungan kerja Apache dan OpenSSL pada Ubuntu 8.10 dapat berjalan dengan baik, ditandai dengan berjalannya sistem *web* proxy yang dibangun dengan sistem operasi dan aplikasi-aplikasi tersebut.
7) Dengan memanfaatkan Glype Proxy, aplikasi berbasis *web* yang terdapat di jaringan lokal dapat diakses dari internet, sehingga aplikasi SIA yang dimiliki UNDIP yang diletakkan pada jaringan lokal dapat digunakan oleh pengguna di internet.
8) Proses instalasi sertifikat pada browser pengguna tidak dapat dilakukan secara otomatis dikarenakan sertifikat tersebut tidak diakui keabsahannya oleh browser.
9) Data yang dikirimkan oleh klient ke *server* tidak dapat dilihat oleh pengguna internet lain karena adanya enkripsi terhadap data tersebut yang dihasilkan oleh OpenSSL.

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

1) Sistem VPN yang yang telah digunakan dapat diterapkan pula untuk membuat jaringan virtual antar kampus yang terpisah jauh secara geografis, agar semua layanan lokal dapat dinikmati di seluruh UNDIP.
2) Sistem VPN dapat dikembangkan untuk menamankan suatu jaringan, sehingga jaringan tersebut hanya bisa diakses oleh anggota VPN. Hal ini bisa diterapkan di jaringan yang berisi perangkat jaringan yang penting.
3) Dapat dilakukan pembatasan *Upload* dan *Download* pada OpenVPN, yaitu dengan memanfaatkan fasilitas *bandwidth limiter* yang dimiliki FreeRADIUS.
4) Selain untuk mengakses sumber daya informasi lokal, system VPN dan *Web* Proxy ini dapat dikembangkan agar memungkinkan pengguna dari jaringan publik untuk mengakses internet dengan dikenali sebagai anggota jaringan UNDIP.

PUSTAKA

[1] "Wikipedia: Virtual Private Network." [Online]. Available: http://en.wikipedia.org/wiki/Virtual_private_network
[2] "Openmaniak: Open VPN Tutorial." [Online]. Available: http://openmaniak.com/openvpn.php
[3] Markus Feilner, *OpenVPN Building and Integrating Virtual Private Networks*. Birmingham: Packt Publishing, 2006.
[4] Hendra Wijaya, *Cisco ADSL Router, Pix Firewall, VPN*. Jakarta: Elex Media Komputindo, 2006.
[5] "Glype Proxy Script Official Website." [Online]. Available: http://www.glype.com
[6] "How Virtual Private Networks Work." [Online]. Available: http://computer.howstuffworks.com/vpn.htm
[7] "Connected: An Internet Encyclopedia." [Online]. Available: http://www.freesoft.org/CIE/Topics/index.htm
[8] "About.com: VPN Tunneling." [Online]. Available: http://compnetworking.about.com/od/vpn/a/vpn_tunneling.htm
[9] Hendra Wijaya, *Belajar Sendiri Cisco Router*. Jakarta: Elex Media Komputindo, 2001.
[10] "Wikipedia: Transport Layer Security." [Online]. Available: http://en.wikipedia.org/wiki/Transport_Layer_Security
[11] "OpenSSL Official Website." [Online]. Available: http://www.openssl.org
[12] "Wikipedia: HTTP Secure." [Online]. Available: http://en.wikipedia.org/wiki/HTTP_Secure
[13] "FreeRADIUS Official Website." [Online]. Available: http://freeradius.org/
[14] "Wikipedia: RADIUS." [Online]. Available: http://en.wikipedia.org/wiki/RADIUS
[15] "Wikipedia: Proxy Server." [Online]. Available: http://en.wikipedia.org/wiki/Proxy_server
[16] "uBuntu: Certificates and Security." [Online]. Available: https://help.ubuntu.com/9.04/serverguide/C/certificates-and-security.html
[17] "Generate and Create SSL Certificate in Linux Ubuntu." [Online]. Available: http://www.dhuha.net/en/content/computer/tutorial/generate-create-SSL-certificate-Linux-Ubuntu

[18] “Selfsign Certificate.” [Online]. Available: http://www.tc.umn.edu/~brams006/selfsign.html
[19] “Openmaniak: Wireshark.” [Online]. Available: http://openmaniak.com/wireshark.php

Adian Fatchur Rochim dilahirkan di Semarang, Indonesia, pada tahun 1973. Beliau mendapatkan gelar sarjana dari jurusan Teknik Elektro, Universitas Diponegoro, pada tahun 1997 dan gelar magister dari Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, pada tahun 2003. Saat ini, Beliau aktif menjadi dosen di jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, sejak tahun 1998. Bidang penelitian yang digeluti adalah jaringan komputer, mikroprosesor dan teknologi informasi.

Andrian Satria Martiyanto dilahirkan di Pemalang, Indonesia, pada tahun 1987. Beliau menjadi mahasiswa Strata-1 konsentrasi Komputer dan Informatika di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro sejak tahun 2005. Bidang keilmuan yang ditekuni adalah jaringan komputer dan teknologi informasi.