Keamanan data Pada Jaringan Sensor Nirkabel Berbasis Linux Menggunakan SSH Protokol

Willy Permana Putra[1], A Sumarudin[2]

[1]*Jurusan Teknik Informatika, Politeknik Indramayu, Jalan Lohbener lama no 8 Lohbener Indramayu 45252*
E-mail : putranunuk@gmail.com
[2]Jurusan Teknik Informatika, Politeknik Indramayu, Jalan Lohbener lama no 8 Lohbener Indramayu 45252
E-mail: asumarudin@gmail.com

ABSTRAK

Jaringan sensor nirkabel hal yang terkait dengan keamanan data yang dikirim dari klien ke *gateway* merupakan masalah yang penting selama ini dirumuskan, dikarenakan media transmisi yang digunakan media nirkabel. Dalam hal ini, ssh merupakan teknik keamanan data yang ditawarkan untuk menjadi salahsatu solusi keamanan dalam pengiriman data jaringan sensor nirkabel. SSH merupakan teknik keamanan data dengan *key public* yang di *generate* di sisi klien dan *server*. Hasil dari enkripsi yang dihasilkan cukup memberikan hasil enkripsi yang baik.Dengan tidak memberikan beban jaringan yang berat.Penelitian ini menggunakan mote dari jaringan sensor nirkabel berbasis linux Ubuntu 12.04 LTS non GUI, dengan komunikasi menggunakan wifi (IEEE 801.11 w/g/n).Dari hasil penelitian data ter*enkripsi* dengan baik dan delay dan troughput yang tidak terlalu berat bagi jaringan sensor nirkabel. Dengan perbandingan delay0,481 msuntuk jaringan tanpa ssh dan 0,029 mS dengan ssh. Sedangkan untuk *troughput* didapat± 2,082 kbps tanpa ssh dan ± 2,229 kbps menggunakan jaringan dengan ssh. Dari hasil ini jaringan sensor nirkabel menggunakan ssh memberikan solusi keamanan data dengan delay dan troughput yang baik.

Kata Kunci : *Keamanan data,* Protokol SSH, Jaringan Sensor Nirkabel, Public Key, linux.

PENDAHULUAN

Di era jaman sekarang ini perkembagan internet sudah semakin canggih sehingga arus internet membawa dampak perubahan yang cukup signifikan baik dari yang memanfaatkan jasa layanan internet maupun yang melakukan berbagai hal komunikasi misalnya bertukar informasi, bertukar data, transaksi data dan lain-lain. Perkembangan teknologi itu sendiri antara lain adalah **WSN** (*jaringan sensor nirkabel*) perkembanga teknologi ini sangat *efisien* baik dari power maupun keakuratan datanya [1]. Salah satu perkembagan teknologi *jaringan sensor nirkabel* sering digunakan dalam beberapa aplikasi diantaranya pegukuran kekuatan jembatan, bidang pertanian dalam peoses informasi suhu, *ph* dan kelembaban dari suatu lahan. Salah satu peroses yang perlu dikhawatirkan adalah proses pengiriman data mulai dari *Node* sampai ke *gateway*. Seiring dengan kemajuan saat ini kebutuhan dalam keamanan data dalam bertukar informasi sangat diperlukan, tranfer data antar pegguna internet harus diberi suatu pengamanan agar data bisa diterima oleh yang berhak menerima.Dengan kemajuan sistem yang sekarang ini kejahatan bisa terjadi dimana saja, bahkan di dunia maya sekalipun.Dengan adanya kejahatan-kejahatan ini pengguna semakin tidak aman lagi dalam melakukan bertransaksi, maka diperlukannya solusi yang bisa membantu agar data dan informasi yang kita kirimkan menjadi lebih aman dan bisa sampai pada tujuan. Salah satu solusi yang ditawarkan adalah dengan meggunakan metode enkripsi data yang dimana dalam proses pengiriman data asli dirubah menjadi data *unicode* [2]. Dalam makalah ini dimana akan dibahas bagaimana proses pegiriman data melalui salah satu enkripsi moderen yaitu *Secure Shell (SSH)* dengan jaringan *jaringan sensor nirkabel*.

2. LANDASAN TEORI

2.1 Pengertian Secure Shell (SSH)

Pada awalnya SSH dikembangkan oleh Tatu Yl nen di Helsinki University of Technology. SSH memberikan alternatif yang secure terhadap *remote session* tradisional dan *file transfer protocol* seperti telnet dan relogin.Protokol SSH mendukung otentikasi terhadap *remote host*, yang dengan demikian meminimalkan ancaman pemalsuan identitas *client* lewat *IP address spoofing* maupun manipulasi DNS. SSH adalah

program yang memungkinkan anda untuk bisa login ke sistem *remote/server* dengan memiliki *autentifikasi public-key*. SSH merupakan paket program yang digunakan untuk memasukan data sebagai pengganti aplikasi yang tidak aman seperti *rlogin, rsh dan rcp*. SSH menggunakan *public-key cryptography* untuk mengenkripsi komunikasi antara dua host, demikian pula untuk *autentikasi* pemakai [2].

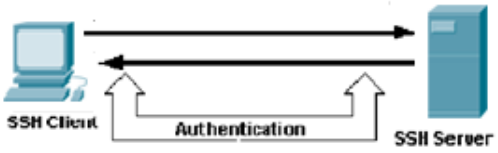
2.1.1 Dasar-dasar Protokol Secure Shell

a. User Authentication

Gambar 1:*Authentication Client* ke Server

Otentikasi, juga disebut sebagai identitas pengguna, adalah sarana yang wajib diketahui oleh sistem agar dapat diverifikasi oleh sistem/server.Banyak metode otentikasi saat ini digunakan, mulai dari *password*, *user name* [2].

b. Host Authentication



Gambar 2: Authentication Host Server to Client

Kunci *host* digunakan oleh server untuk membuktikan identitasnya kepada klien dan untuk memverifikasi "Dikenal"tuan rumah/server. Kunci *public* digambarkan sebagai identitas atau sandi untuk bisa masuk kedalam sistem server [2].

c. Data Encryption & Integrity

Gambar 3: Data Encryption & Integrity

Enkripsi, kadang-kadang disebut sebagai privasi, berarti bahwa data yang dilindungi dari pengungkapan seorang calon penyerang "*sniffing*" atau mencoba untuk menangkap data.*Blok cipher* adalah yang paling umum digunakan dalam bentuk algoritma kunci simetrik (misalnya *DES, 3DES*, *Blowfish*, *AES*, dan *Twofish*). Fungsi ini biasanya digunakan data yang mau dikirim dan umumnya melibatkan beberapa pola dalam proses untuk mengamankan data. Data yang terkirim akan "Terenkripsi" dan tidak dapat dikembalikan tanpa kunci bersama. Sedangkan untuk *Integritydata* menjamin bahwa data yang dikirim dari satu *client* ke server berjalan aman [2].

2.1.2 Cara Kerja Jaringan Secure Shell (SSH)

Klien mencoba mengakses server dengan cara membuka *post* 22 dan melakukan koneksi ke server dengan melakukan *autentifikasi public-key*, berikut langkah-lagkah koneksinya [3]

- Langkah 1 :*Client* melakukan koneksi ke *port* 22 pada server.
- Langkah 2 :*Client* dan server setuju untuk menggunakan sesi SSH tertentu.
- Langkah 3 :*Client* meminta *public key* dan *host* key milik server.
- Langkah 4 : *Client* dan server menyetujui algoritma enkripsi yang akan dipakai
- Langkah 5 :*Client* membentuk suatu *session key* yang didapat dari *client* dan mengenkripsinya menggunakan *public key* milik server.
- Langkah 6 : Server men-*decrypt session key* yang didapat dari *client*, meng-*re-encrypt*-nya dengan *public key* milik *client*, dan mengirimkannya kembali ke *client* untuk verifikasi.
- Langkah 7 : Pemakai mengotentikasi dirinya ke server di dalam aliran data terenkripsi dalam *session key* tersebut.

Setelah proses diatas selesai maka dalam hal ini koneksi sudah terhubung antara *Client* dan Server

2.2 Jaringan Sensor nirkabel *(Jaringan sensor nirkabel)*

Wirelless Sensor Network (WSN) terdiri dari sejumlah besar *node sensor* yang padat

[4].Banyak aplikasi berbasis WSN bermuncul menggunakan metode ini.Disamping *low-power* juga keakuratan datanya.Dibandingkan dengan jaringan nirkabel konvensional, *Wirelless Sensor Network*(WSN) memiliki beberapa karakteristik unik termasuk *bandwidth* transmisi yang terbatas, kemampuan perhitungan terbatas *node* individu, dan pasokan energi yang terbatas.*Wirelless Sensor Network*(WSN) juga dapat memiliki beberapa fitur menarik termasuk *self-organisasi*, topologi jaringan yang dinamis, dan *multi-hop routing*, yang penting bagi banyak aplikasi dunia nyata [5].Dalam teknologi *Wirelless Sensor Network* terdapat berbagai isu dalam desain jaringan.Terdapat tiga teknik traditional dalam melakukan analisis performance dari *Wirelless Sensor Network* diantaranya metode analisis, komputer simulasi dan pengukurang perangkat aplikasi (*physical measurement*).Dimana terdapat banyak masalah dalam sensor *network* adalah daya yang terbatas, tidak tersentralisasi kolaborasi data, dan *fault tolerence* membutuhkan algoritma kompleks dalam analisis. [6].*Jaringan sensor nirkabel* merupakan *systemwireless* dengan *system ad-hoc mode*. Prinsip *ad-hoc network* adalah *point to point* koneksi, koneksi adalah dua *client* secara langsung.

Gambar 4: *Connection Point to point*
Sumber :ItrainOnline MMTK
www.itrainonline.org

3. METODE PENELITIAN

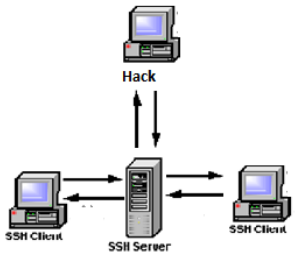
3.1 Bahan Penelitian

Bahan penelitian ini menggunakan bahan utama dari beberapa dokumen yang terkait dengan *Wirelless Sensor Network* (WSN) dan *Secure Shell* (SSH). Sumber data meliputi diantarannya:

- Review Documentation, yaitu meninjau dokumen yang telah ada dan berkaitan dengan permasalahan
- Uji coba, yaitu dengan menguji program pengiriman paket data yang menggunakan *Secure Shell* (SSH) dan data yang tidak menggunakan *Secure Shell* (SSH)

3.2 Alat Penelitian

Penelitian dilakukan dengan cara memperaktekan beberapa *client* dan satu server yang sudah menggunakan protokol *Wirelless Sensor Network*(WSN)dan mengamati hasil dari proses. Oleh karena itu dalam pengamatan yang akan dilakukan adalah



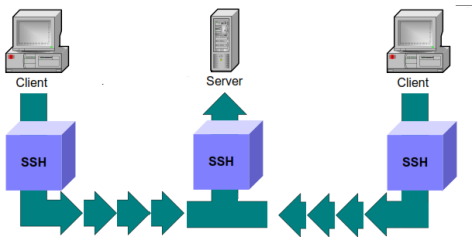
Gambar 5: prosesi pengiriman data dan *Authentication*

- Prosesi pengiriman data dari client ke server
- Grafik prosesi pengiriman data dari client ke server
- Pengecekan data *Authentication* dari client ke server atau pun sebaliknya

4. HASIL DAN PEMBAHASAN

4.1 Desain dalam percobaan

Desain ini mencoba dalam 2 klien ke 1 server, yang sudah menggunakan protokol *Wirelless Sensor Network*(WSN) dimana server menerima inputan secara bersamaan dan si *client*mengirimkan data bersamaan dalam waktu detik sekali kirim data.



Gambar 6: Desain trafer data

4.1.2 Pembuatan SSH *Injeck*

Dalam proses *injeck*ssh ada beberapa hal yang harus di setujui antara *client* dengan server yaitu proses permintaan ssh ID *(Public-Key*). Dalam proses pembuatan *public key* yang perlu dibuat adalah :

ssh-*keygen -t rsa { digunakan untuk membuat public key}*
ssh-copy-id -i ~/.ssh/id_rsa.pub server@118.97.196.163 {digunakan untuk memasukan public key yang sudah dibuat ke dalam server)
sshserver@118.97.196.163 {Prosesi masuk ke dalam ssh server}

Ini dimaksudkan untuk mengidentifikasi si *client* agar dapat di terima oleh server.Hasil dari prosesi ini dapat dilihat pada gambar 8.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ubuntu.
Your public key has been saved in ubuntu.pub.
The key fingerprint is:
13:3d:8a:2b:a0:28:38:80:41:1b:02:62:03:91:df:e8 ubuntu@ubuntu-sa
```

Gambar 7: Proses pembuata *public key*

Setelah proses pembuatan *public key* berhasil maka selanjutnya adalah proses pemasukan *public key* pertama. Ini *client* memasukan *public key* yang diminta oleh si server dan si server merespon untuk *Authentication,* seperti yang ditunjukan pada gambar 8.

```
The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.
ECDSA key fingerprint is 4e:03:1f:99:64:c5:a0:4b:b5:96:38:2d:30:b0:fe:c1.
Are you sure you want to continue connecting (yes/no)? yes
```

Gambar 8: Authentication SSH

4.2 Hasil dari pengujian

Dalam proses ini yang akan diteliti yaitu ada 3 tahap yaitu Prosesi pengiriman data, Grafik dari prosesi pengiriman, dan hasil *sniffing*. Dari ketiga pengamatan ini nantinya didapatkan hasil perbedaan antara prosesi pengiriman data yang menggunakan *Secure Shell* (SSH) dan data yang tidak menggunakan *Secure Shell* (SSH).

4.2.1 Prosesi Pengiriman Data

Dalam proses pengiriman data, data yang akan dikirim menggunakan *scp* yang sudah disediakan oleh ssh, dalam proses pengiriman ini data dikirim setiap detik. Data yang dikirim per detik akan dicatat dan dikimkan ke server kemudian dari hasil laporan itu ada catatan bahwa data dikim perdetik berhasil dikim ke server. Prosesi pengiriman data dapat dilakukan dengan cara menuliskan sintak yang ada pada *bash sell* dan sintak ini dijalankan sesui dengan perintah. Semua prosesi pengiriman data ini menggunakan perintah *bash sell* kenapa menggunakan *bash sell* dikarenakan gampang untuk perintahnya serta bisa dijalnkan dari luar atau dijalankan dengan *remote*.

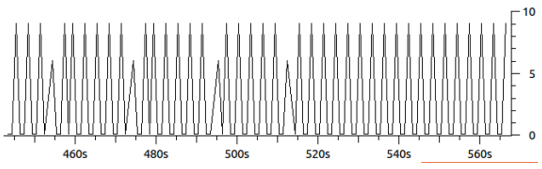
```
#!/bin/bash
i=0;
while [ $i -le 5 ];
let i=$i+1;
do
echo "Kirim Data ke - $i";
sleep 1;
echo "File 1 "$(date)
echo "Pengiriman Data ke - $i, File ke 1 " "terkirim pada" ,$(date) >> data.txt
scp data.txt server@118.97.196.163:/home/server/hasil-data
```

Hasil pengiriman data dapat dilihat pada gambar 9, dari gambar 9 menunjukan data dikirim per detik dan data yang kirim dicatat berdasarkan waktu.

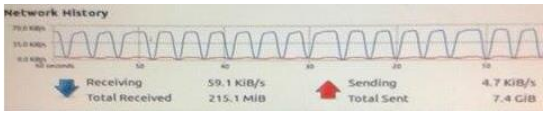
```
Kirim Data ke - 234
File 1 Sun May 11 08:25:45 WIB 2014
data.txt                                      100%   18KB  18.4KB/s   00:00
Kirim Data ke - 235
File 1 Sun May 11 08:25:48 WIB 2014
data.txt                                      100%   18KB  18.5KB/s   00:00
Kirim Data ke - 236
File 1 Sun May 11 08:25:51 WIB 2014
data.txt                                      100%   19KB  18.6KB/s   00:00
Kirim Data ke - 237
File 1 Sun May 11 08:25:53 WIB 2014
data.txt                                      100%   19KB  18.6KB/s   00:00
Kirim Data ke - 238
File 1 Sun May 11 08:25:56 WIB 2014
data.txt                                      100%   19KB  18.7KB/s   00:00
Kirim Data ke - 239
File 1 Sun May 11 08:25:59 WIB 2014
data.txt                                      100%   19KB  18.8KB/s   00:00
```

Gambar 9: Tranfer data clien to server

4.2.2 Grafik dan delay, latency, troughput



(a)



(b)

Gambar 10: Grafik Tranfer data clien to server
(a). Gambar berdasarkan koneksi *bandwidth* pengiriman data.
(b). Gambar tranferdata dari sisi server

Dari hasil grafikmenunjukan bahwa data yang dikirim dari *client* ke server stabil.Sehingga data yang dikirim perdetiknya tidak mengalami perubahan.Baik dari segi bobot maupun transfer data.

(a). Delay

Delay merupakan penundaan waktu suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik yang lain yang menjadi tujuannya [7].

Untuk perhitungan *Delay*

$$Delay(\text{sec})Tx = \frac{\text{Time between first and last paket}}{Jumlah\ Paket}$$

$$Delay(\text{sec})Tx = \frac{49{,}065}{104} = 0{,}471$$

Berdasarkan perhitungan *Delay* yang telah dilakukan selama 50 detik, maka perhitungan delay selama 100, 150, 200 detik, dengan cara yang sama didapatkan hasil perolehan, seperti pada tabel 1.

Tabel 1.Hasil pengujian *Delay* menurut *SoftwareWireshark*

Dari hasil tabel 1 diperoleh nilai rata-rata *delay* pada saat peroses dengan waktu 50 detik, 100 detik, 150 detik, 200 detik sebesar 0,481 sec. Dari besaran nilai rata-rata *delay* yang terjadi masih terlihat bagus. Dan untuk terkecil 0,029 dengan menggunakan ssh

(b). Throughput

Dalam menghitung *throughput* berdasarkan hasil pengujianmenurut *software wireshark.*[7]. Untuk menghitung *throughput*

$$Delay(\text{sec})Tx = \frac{\text{Average Bytes/sec}}{\text{Between first and last packet (sec)}}$$

$$Delay(\text{sec})Tx = \frac{193{,}237}{49{,}065} = 3{,}938$$

Tabel 2. Hasil Pengujian *Throughput* menurut *Software Wireshark*

Waktu	Average Bytes/sec		Between first and last packet (sec)		Throughput (kbps)	
Non SSH SSH	Non SSH	SSH	Non SSH	SSH	Non SSH	SSH
50	193,237	6115,219	49,065	47,553	3,938	4,063
100	201,926	6011,885	99,146	98,291	2,036	2,054
150	199,289	5976,680	148,219	111,081	1,344	1,794
200	200,486	5913,723	197,829	199,324	1,013	1,005

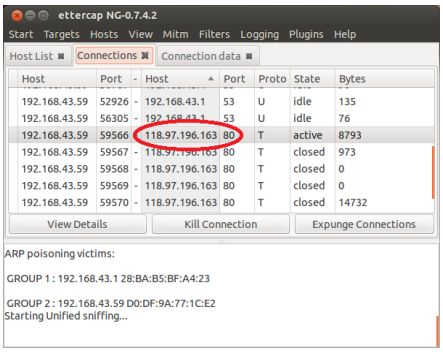
Berdasarkan perhitungan *Throughput*yang telah dilakukan selama 50 detik, maka perhitungan delay selama 100, 150, 200 detik, dengan cara yang sama didapatkan hasil perolehan, seperti pada tabel 2.Nilai yang terbesar yaitu saat pertama yaknik 3,938 dan nilai *Throughput*yang terkecil 1,013.Dan dari data ssh lah yang mengalami besar nilai yaitu 4,063. Data hasil pengujian di atas, membuktikan bahwa semaki n lama waktu melakukan, maka nilai*throughput*yang dihasilkan akan semakin rendah. Hal ini disebabkan karena jumlah paket dari jenis trafik yang dikirimkan semakin banyak juga.

4.2.3 Hasil dari *sniffing* data

Dalam proses ini akan dibahan mengenai sisi keamanan data dengan menggunakan *Secure Shell* (SSH) yang diterapkan pada *Wirelless Sensor Network*(WSN), terlihat perbedaan yang jelas bagaimana data yang dikirim menggunakan *Secure Shell* (SSH) dengan yang tidak menggunakan *Secure Shell* (SSH). Dalam gambar 11 sampai gambar 14 menunjukan data keamanan, dimana prosesi pengiriman menggunakan *Secure Shell* (SSH) terlihat bahwa *Authentication* sangat rumit sekali untuk dipahami sedangkan yang tidak menggunakan

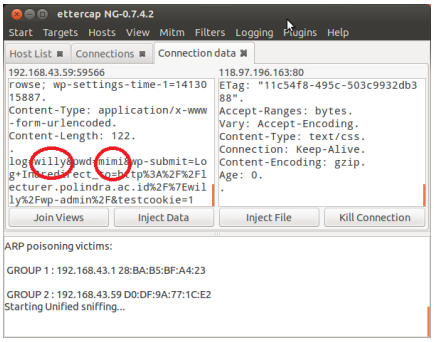
Waktu	Paket		Between first and last packet (sec)		Delay (sec)	
Non SSH SSH	Non SSH	SSH	Non SSH	SSH	Non SSH	SSH
50	104	1633	49,065	47,553	0,471	0,029
100	206	3320	99,146	98,291	0,481	0.029
150	308	3731	148,219	111,081	0,481	0,029
200	411	6606	197,829	199,324	0,481	0,030

Secure Shell (SSH) terlihat jelas *user* dan *password*-nya.



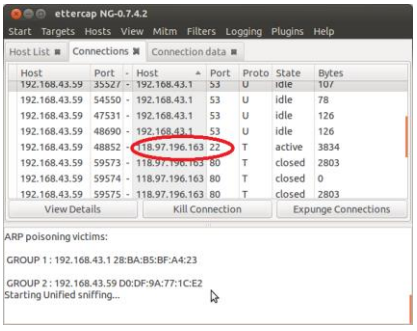
Gambar 11: Koneksi *port* 80

Terlihat dengan jelas koneksi *port* yang dipakai yakni tidak menggunakan *Secure Shell* (SSH) yaitu menggunakan *port* 80.



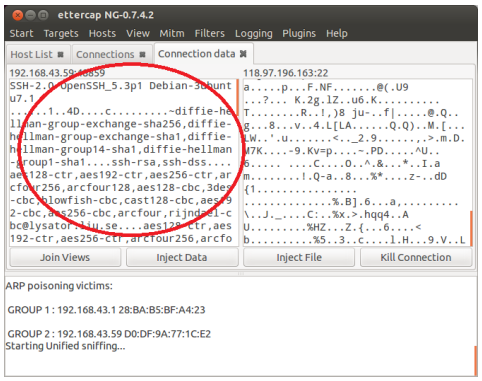
Gambar 12 : Pengecekan Password tanpa ssh

Dari hasil pengamatan yang terjadi didapatkan hasil bahwa ketika *user* melakukan *log in* terlihat jelas pada *user* dan *password*



Gambar 11: Koneksi *port* 22

Terlihat dengan jelas koneksi *port* yang dipakai *Secure Shell* (SSH) yaitu menggunakan *port* 22.



Gambar 14 : Pengecekan Password dengan ssh

Dari hasil pengamatan yang terjadi didapatkan hasil bahwa ketika *user* melakukan *log in* tidak terlihat jelas pada *user* dan *password.*Dari yang terjadi pada hasil menunjukan data yang sulit untuk dipahami.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Terlihat jelas bahwa dari hasil percobaan menunjukan transfer data pada jaringan *Wirelless Sensor Network* (WSN) dengan menggunakan SSH protokol terlihat sangat aman begitu juga dilihat dari nilai *delay* dan *throughput*, nilai rata rata untuk delay 0,481 (non ssh) 0,029 (ssh) dan untuk *throughput*sebesar ± 2,082 kbps (non ssh) ± 2,229 kbps (ssh).Dan untuk keamanan lebih aman menggunakan SSH dibandingkan dengan yang tidak menggunakan SSH protokol dan dalam percobaan menunjukan untuk prosesi pengiriman berjalan dengan normal tanpa adanya gangguan baik dari prosesi pengiriman maupun data yang hilang.Data-data yang dikirimkan dari *client* ke server meunjukan kesetabilan dalam prosesi pengiriman dan dilihat dari segi keamanan sangat aman dikarenakan kunci *public key* diubah oleh *Secure Shell* (SSH).

5.2 Saran

Dalam penelitian ini masih banyak mengalami kendala ketika alamat server berubah maka harus melakukan Injeck ssh kembali sehingga membutuhkan konfigurasi kembali untuk mengkoneksikan antara *client* dangan server.

6. UCAPAN TERIMA KASIH

Terimakasih kepada dirjen dikti melalui hibah penelitian yang telah membiayai penilitian ini, Meskipun penulis telah berusaha sekuat tenaga dan mencurahkan segala pikiran yang ada untuk dapat menyajikan penelitian ini sebaik-baiknya, namun hasil yang penulis capai masih jauh dari kesempurnaan.Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penulisan ini.Penulis juga tidak lupa mengucapkan terima kasih kepada :Heny Rahmawati sebagai istri dari Willy Permana Putra.

DAFTAR PUSTAKA

[1] K. Eitaro, O. Tomoyuki and K. Yoshiaki, "*Secure Decentralized Data Transfer against Node Capture Attacks for Jaringan sensor nirkabel*s, " in Proc. 05207366 IEEE SRDS, pp. 54–63.

[2] VanDyke Software, "*An Overview of the Secure Shell (SSH),* " in White Paper. 2008.

[3] Wicaksono Aloysius S And Katon Glagah Seto S , "*Telnet dan SSH*," in Jurusan Teknik Elektro FT UGM, Yogyakarta. 2009.

[4] Ian Downard. "*SIMULATING SENSOR NETWORKS IN NS-2*". Naval Research Laboratory, Washington DC.

[5] Yunjiao Xue, dkk. "*Performance Evaluation of NS-2 Simulator for Jaringan sensor nirkabels*", Department of Electrical and Computer Engineering, Faculty of Engineering, The University of Western Ontario, London, ON, Canada.

[6] Gilbert Chen, dkk. "*SENSE: A JARINGAN SENSOR NIRKABEL SIMULATOR*". Department of Computer Science, Rensselaer Polytechnic Institute. 2004.

[7] N gurah,Anak Agung. 2012. "*Analisis Dan Implementasi IPTV Dengan Menggunakan MediaWebcam*"Jurusan lmu Komputer.Fakultas Matematika dan Ilmu Pengetahuan Alam,Universitas Udayana

[8] Rayhan Yuvandra, M. Zulfin. "ANALISIS KINERJA TRAFIK VIDEO CHATTING PADA SISTEM CLIENT-CLIENT DENGAN APLIKASI WIRESHARK". Jurnal SINGUDA ENSIKOM Vol 3 September 2013. Medan. 2013.

HAK CIPTA