APLIKASI MONITORING DAN CONTROLLING SERVER MENGGUNAKAN PROTOCOL ICMP (INTERNET CONTROL MESSAGE PROTOCOL) DAN SSH (SECURE SHELL) BERBASIS WEBSITE

Redo Yanuar Pratama[1], Mira Orisa[2], FX Ariwibisono[3]
Program Studi Teknik Informatika S1, Fakultas Teknologi Industri
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia
redoyanuar16@gmail.com

ABSTRAK

Sistem monitoring server sangat dibutuhkan bagi seorang administrator. Sebab memastika setiap server dalam kondisi yang baik merupakan tugas utama dari administrator, karena sifat server yang harus dalam kondisi *online* atau hidup setiap Sistem monitoring yang saat ini digunakan ialah dengan cara mengecek setiap server yang dikelolanya. Dengan cara tersebut maka administrator dapat mengetahui kondisi fisik serta keadaan server dalam keadaan *up* atau *down*.

Dengan menggunakan *protocoi* ICMP *(Internet Control Message Protocol)* untuk menjalankan fitur monitoring server secara *real time,* serta menggunakan *protocol* SSH *(Secure Shell)* untuk melakukan kontrol *server* dan juga melakukan *transfer file* ke *server.* Dalam Aplikasi monitoring dan kontrol server yang diterapkan dan diuji cobakan dalam sebuah jaringan lokal di Laboratorium Jaringan Komputer Tektnik Informatika S-1 ITN Malang.

Kegiatan monitoring dapat dilakukan melalui aplikasi ini adalah memantau kondisi *real time* dari *server* yang dikelola dalam kondisi *up* atau *down* serta, apabila server dalam keadaan *down* maka administrator akan memperoleh sebuah notifikasi. Fitur kontrol *server* yang dapat dilakukan dalam aplikasi ini ialah mematikan atau menjalankan ulang *server,* mengolah *service* yang bejalan diserver, serta fitur untuk melakukan pengiriman *file* kesebuah *server.*

Kata kunci : *Network monitoring, Remote Server, ICMP, SSH*

1. PENDAHULUAN

Sistem monitoring jaringan sangat dibutuhkan bagi seorang administrator dan juga bagi instansi yang bertugas untuk mengolah sebuah jaringan internet. Sistem monitoring jaringan ini sangatlah penting karena setiap saat perkembangan kebutuhan akan jaringan komputer sangatlah tinggi sehingga setiap saat ada kemungkinan untuk dilakukan penambahan sebuah perangkat jaringan atau *server* yang di *install* pada sebuah jaringan, dengan adanya kemungkinan penambahan perangkat jaringan atau *server* maka tanggung jawab administrator untuk pemeliharaan dan kelola jaringan komputer *server* menjadi bertambah. Maka dengan adanya sistem monitoring seorang administrator jaringan dapat terbantukan untuk bisa selalu memantau kondisi suatu jaringan yang dikelolanya sebab administrator dapat memantau seluruh perangkat jaringan atau *server* yang dikelola secara *real time*. Seorang administrator dituntut untuk bisa mengatasi segala permasalahan yang mungkin terjadi pada jaringan *server* tersebut. Sistem *network* monitoring yang saat ini digunakan ialah sistem monitoring secara manual yang dilakukan langsung oleh administrator dengan cara mengecek setiap server yang dikelolanya. Dengan cara tersebut maka administrator dapat mengetahui kondisi serta keadaan *server* dalam keadaan *online* atau *offline.* Monitoring dan kontrol terhadap suatu *server* sangatlah diperlukan dikarenakan sifat *server* yang harus terus berjalan dan berfungsi setiap saat serta dapat memeberikan layanan *service* yang maksimal bagi user. Dari penggunaan sistem monitoring manual tersebut maka administrator tidak dapat mengetahui kondisi *link* interkoneksi jaringan pada *server* tersebut. Serta apabila diperlukan *control* terhadap suatu server pada jaringan maka administrator perlu menghafal tiap-tiap *ip address* yang digunakan pada tiap perangkat jaringan komputer selain itu administrator juga diharuskan menghafal alamat *ip address, port, user* dan *password* dari *server* tersebut.

Pada penelitian ini dikembangkan system monitoring dengan judul *"Aplikasi Monitoring Dan Controlling Server Menggunakan Protocol ICMP (Internet Control Message Protocol) Dan SSH (Secure Shell) Berbasis Website"*. Dengan memanfaatkan *protocol* ICMP *(Internet Control Message Protocol)* yang berjalan pada layer 3 dalam susunan *Osi Layer* [1], dengan *protocol* ICMP *(Internet Control Message Protocol)* ini setiap *device* jaringan yang terhubung dapat berkomunikasi dengan *protocol* ICMP *(Internet Control Message Protocol)* ini maka akan mampu mengetahui keadaan dari perangkat jaringan seperti *server* apakah dalam keadaan *up* atau dalam keadaan *down*. Selain itu dalam penelitian kali ini penulis juga akan merancang sebuah sistem kontrol *server* dengan memanfaatkan protocol *SSH (Secure Shell).* Untuk melakukan kegian kontrol terhadap server tersebut. Dengan menggunakan protocol *SSH (Secure Shell)* ini

maka administrator tidak perlu lagi untuk menghampiri komputer *server* untuk melakukan kontrol terhadap *server*, karena *protocol* SSH *(Secure Shell)* dirancang untuk melakukan kendali (*remote)* server. Maka dengan *protocol SSH (Secure Shell)* ini administrator dapat melakukan kontrolling terhadap server dari jarak jauh [2]. Melalui satu jaringan dengan menggunakan *platform website*.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Salah satu permasalahan pada jaringan komputer adalah kurang stabilnya kecepatan *transfer* pada sebuah jaringan karena besarnya trafic data. Seorang administrator jaringan dituntut untuk selalu siap pengecekan terhadap kondisi dari jaringan yang dikelolanya [3]. Sistem monitoring digunakan oleh administrator untuk mempermudah dalam melakukan pekerjaan teknis dalam melakukan pemantauan jaringan secara rutin. [4] karena pada sebuah jaringan, itu tidak bebas dari masalah keamanan[5] selain memantau jaringan administrator juga dituntut untuk melakukan kelola *server*, karena Umumnya server dijalankan secara terus meneru untuk 24 jam per hari untuk memberikan layanan atau *service* komunikasi pada sebuah jaringan komputer. [6]

2.2 *Internet Control Message Protocol* (ICMP)

PING (*Packe internet Gopher*) merupakan salah satu fitur yang berfungsi untuk melakukan pengecekan komunikasi antar *host* pada jaringan, fitur PING ini berjalan pada protokol ICMP (*Internet Control Message Protocol*) *protocol* PING (*Packe internet Gopher*) akan mengirimkan *echo request messages* kepada ip address *host* yang mejadi tujuan dan meminta balasan dari *host* tujuan.[2]

2.3 *Secure Shell* (SSH)

SSH merupaka protokol jaringan yang memungkinkan untuk melakukan komunikasi pertukaran data melalui saluran aman antara dua *device* jaringan. SSH selain itu SSH juga mendukung fitur untuk pengiriman berkas atau *file* melalui SFTP (*secure File Transfer Protocol*) atau SCP *(Secure Copy)*. SSH menggunakan model komunikasi *client-server*. Standart *port* dari ssh adalah port 22 yang telah ditetapkan sebagai jalur untuk server SSH. Sebuah klien SSH biasanya digunakan untuk membangun koneksi antara klient dengan *server* SSH supaya dapat dikendalikan via kendali jarak jauh. [7]

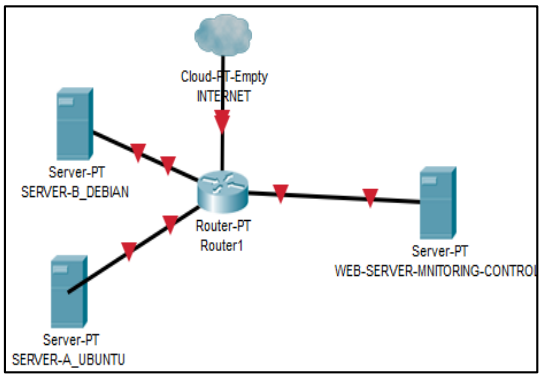
3. METODE PENELITIAN

3.1 Analisis dan kbutuhan Sistem

Melihat dari kegiatan monitoring yang sangat penting karena kebutuhan akan jaringan komputer dan juga jaringan *server* yang terus berkembang. Serta kebutuhann untuk kontrol *server* yang sangat penting bagi administrator, untuk melakukan control administrator harus menghafal atrbut dari *server* yang diperlukan untuk melakukan kontrol terhadap *server*.

3.2 Topologi Jaringan

Pada gambar 3.1 menjelaskan mengenai design dan penerapan topologi jaringan yang digunakan untuk uji coba aplikasi pada jaringan lokal. Topologi yang digunakan adalah topologi star.

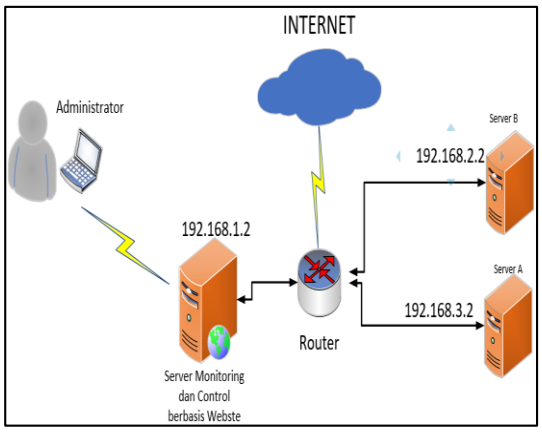


Gambar 1 Topologi Jaringan

Pada gambar 1 adalah design *topologi* yang akan diterapkan p ada pengujian monitoring dan kontrol *server* dalam jaringan local. Pada *topologi* tersebut terdapat 2 buah *server* berbeda yang tiap server akan diisi sistem operasi Ubuntu dan Debian. Kedua server tersebut adalah *server* yang akan dijadikan sebagai objek server yang akan dimonitoring serta dapat dikontrol. Kedua server tersebut terhubung dengan sebut *router*, *router* pada design *topologi* ini akan bertindak sebagai *konsentrator*. *Ether* 3 dan *ether* 4 pada *router* akan terhubung dengan kedua *server*, selanjutnya *router* akan mem*broadcast* ip yang akan diterima oleh server-A dan server-B. sedangkan pada *ether* 1 di *router* akan terhubung dengan jaringan *internet*, disini *router* juga sebagai sumber internet dalam jaringan local ini. Selanjutnya pada *ether* 2 di *router* akan terhubung dengan web *server* Monitoring dan Kontrolling.

3.3 Diagram Blok

Pada gambar 3.2 menjelaskan mengenai blog diagram dari aplikasi ini.

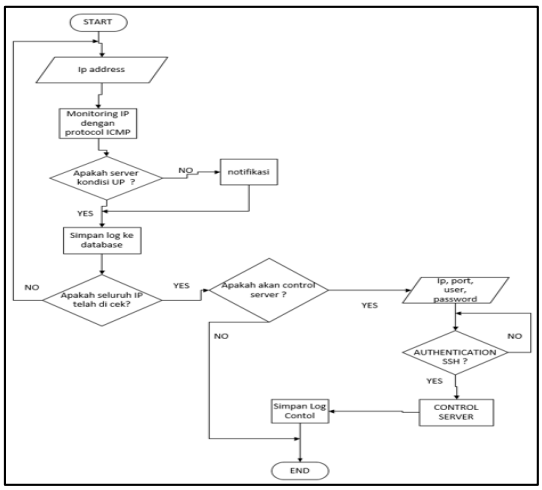


Gambar 2. Diagram blok

Pada pembuatan sistem monitoring dan kontrol *server* sumber internet akan masuk ke sebuah *router* melalui *ether1*, *router* sebagai sumber internet juga akan menyediakan *service* DHCP *server*, Lalu *server* akan mendapatkan sumber internet. Semua *server* tersebut akan terhubung pada satu *router*. Fungsi *router* pada topologi ini adalah sebagai konsentrator. Maka dari design topologi tersebut server-A akan mendapat ip DHCP dari *router* dengan asumsi ip 192.168.3.2 ditopologi ini server-A akan menggunakan sistem operasi ubuntu *server* dan server-B akan mendapat ip DHCP dari *router* dengan asumsi ip 192.168.2.2 sedangkan server-B akan menggunakan sistem operasi *Debian*. lalu pada *server* monitoring dan kontrol akan mendapat ip DHCP dari *router* dengan asumsi ip 192.168.2.1 yang didalam *server* tersebut terdapat sebuah aplikasi berbasis website yang digunakan untuk melakukan kerja sebagai monitoring serta kontrolling terhadap *server* yang dikelolahnya. Terakhir administrator dapat mengakses website monitoring dan kontrol *server* melalui web *browser*.

3.4 Flowchart aplikasi monitoring dan control server.

Selanjutnya akan menjelaskan mengenai *flowchart* dari aplikasi monitoring dan control server, pada *flowchart* gambar 3 pertama kali aplikasi akan menjalanakan fungsi monitoring dengan menggunakan *protocol* ICMP. Sistem akan melakukan *ping* terhadapa semua *ip address* yang ada pada *database*. Proses monitoring dengan *protocol* ICMP ini akan berjalan pada *background* proses monitoring akan berjalan dengan *delay* estimasi 5 hingga 15 detik. Apabila sistem mendeteksi *server* dalam keadaan *down* maka system akan mengirimkan *notifikasi telegram* kepada administrator dengan isi pesan bahwa server dengan ip address tertentu sedang dalam keadaan *down*. Seluruh hasil dari monitoring dengan *protocol* ICMP akan di *record* ke dalam *database*. Hasil monitoring yang disimpan pada database meliputi data ip address, status dari ip tersebut *down* atau *up* serta waktu dari *down* atau *up*. Selanjutnya apabila ingin melakukan kontrol terhadap *server* maka aplikasi akan mengambil data atribut *server* yang disimpan didatabse meliputi *ip address*, *port*, *user*, *password*. Untuk atribut *ip address* merupakan ip dari *server* yang dituju untuk dilakukan kontrol, selanjutnya untuk atribut *port* merupakan *port* yang digunakan sebagai jalur untuk melakukan koneksi antara *server* dengan *client*. *port default* dari protocol SSH yaitu *port* 22. sedangkan untuk atribut *user* akan menggunakan user *default* yang dimiliki oleh tiap *server* linux yaitu *user root*. *User root* ini user yang memiliki hak akses full pada *server* linux, dan untuk atribut *password* akan diisi data *password* untuk *user root*. Apabila semua atribut yang dibutuhkan tersebut telah terpenuhi maka selajutnya proses *authentication* akan berjalan apabila proses *authentication* berhasil makan administrator dapat melakukan kontrol terhadap *server*.



Gambar 3 *flowchart* aplikasi

Pada flowchart gambar 3 pertama kali aplikasi akan menjalanakan fungsi monitoring dengan menggunakan *protocol* ICMP. Sistem akan melakukan *ping* terhadapa semua *ip address* yang ada pada *database*. Proses monitoring dengan *protocol* ICMP ini akan berjalan pada *background* proses monitoring akan berjalan dengan *delay* estimasi 5 hingga 15 detik. Apabila sistem mendeteksi *server* dalam keadaan *down* maka system akan mengirimkan *notifikasi telegram* kepada administrator dengan isi pesan bahwa server dengan ip address tertentu sedang dalam keadaan *down*. Seluruh hasil dari monitoring dengan *protocol* ICMP akan di *record* ke dalam *database*. Hasil monitoring yang disimpan pada database meliputi data ip address, status dari ip tersebut *down* atau *up* serta waktu dari *down* atau *up*. Selanjutnya apabila ingin melakukan kontrol terhadap *server* maka aplikasi akan mengambil data atribut *server* yang disimpan didatabse meliputi *ip address*, *port*, *user*, *password*. Untuk atribut *ip address* merupakan ip dari *server* yang dituju untuk dilakukan kontrol, selanjutnya untuk atribut *port* merupakan *port* yang digunakan sebagai jalur untuk melakukan koneksi

antara *server* dengan *client*. *port default* dari protocol SSH yaitu *port* 22. sedangkan untuk atribut *user* akan menggunakan user *default* yang dimiliki oleh tiap *server* linux yaitu *user root*. *User root* ini user yang memiliki hak akses full pada *server* linux, dan untuk atribut *password* akan diisi data *password* untuk *user root*. Apabila semua atribut yang dibutuhkan tersebut telah terpenuhi maka selajutnya proses *authentication* akan berjalan apabila proses *authentication* berhasil makan administrator dapat melakukan kontrol terhadap *server*.

4. HASIL DAN PEMBAHASAN

4.1 Tampilan Monitoring Server

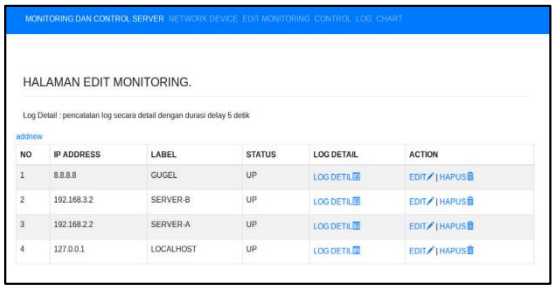
Pada halaman ini adalah halaman untuk melakukan monitoring server pada jaringan secara realtime dengan protocol ICMP.

Gambar 4 Halaman monitoring

Pada hasil pengujian Digambar 4 dilakukan pengujian semula *server* dengan ip 192.168.2.2 dan *server* dengan ip 192.168.3.2 terpantau dalam kondisi *up*. Lalu dilakukan pengujian dengan mematikan kedua *server* tersebut, dan hasil dari pengujian tersebut kondisi *server* denga nip 192.168.2.2 dan juga server denga nip 192.168.2 yang dimonitor melalui website berubah menjadi *down*. (icon berubah menjadi merah dengan tanda X).

4.2 Tampilan Menu edit monitoring

Tampilan dari menu Edit Monitoring

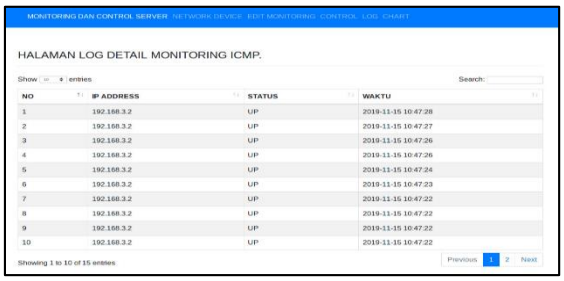


Gambar 5 Halaman menu edit monitoring

Pada menu ini adalah proses CRUD (*Create Read Update Delete)* apabila administrator ingin menambahkan server atau ingin melakukan edit dan delete. Serta terdapat satu fitur yaitu log detail yang berisi informasi detail seperti pada gambar di point no 4.3 dibawah.

4.3 Tampilan Log Detil

Fitur untuk melihar secara detail log monitoring

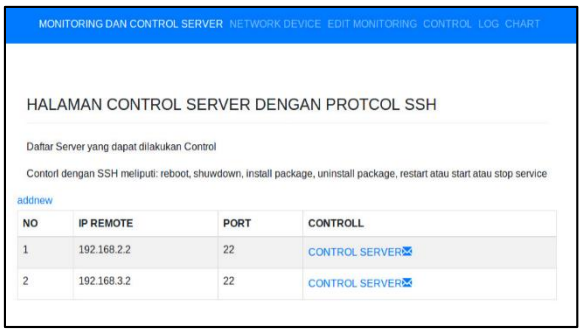


Gambar 6 Log Detil

Pada Gambar 6 Menampilkan informasi detail mengenai kapan waktu UP dan waktu DOWN dengan delay waktu 5 detik.

4.4 Tampilan Halaman Menu Control

Menu untuk melakukan control terhadap server dengan protocol *SSH.*

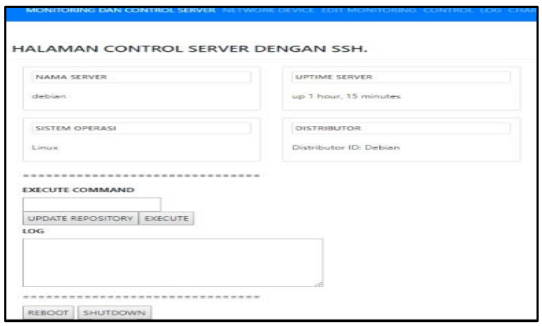


Gambar 7 halaman control server

Pada menu ini akan menampilkan server yang dapat control dengan protocol ssh. Jika admininstrator memilih untuk mengcontrol server dengan ip 192.168.2.2. maka halaman akan diarahkan ke alamat sshControl.php secara otomatis maka web akan melakukan tunnel terhadap ip tersebut dengan menggunakan protocol ssh.

4.5 Tampilan Control Server

Halaman bagi administrator untuk melakukan control server dengan protocol SSH. Fungsi control hanya mendukung untuk malkukan *reboot, shutdown, restart service, install package, uninstall package.*



Gambar 8 Control Server

kontrol server akan menampilkan informasi tentang server yang dicontrol menggunakan protocol ssh. Fitur control meliputi fungsi *reboot, poweroff, restart service.* Hasil dari *excute command* akan ditampilakan dalam textarea..

4.6 Pengujian Aplikasi pada *web browser*

Pengujian aplikasi pada penelitian ini dengan dilakukannya pengujian kompabilitas *website* terhadap *web browser* bertujuan untuk mengetahui apakah halaman *website* yang dibuat dapat menampilkan keseluruhan data sesuai dengan perancangan

Tabel 1 Pengujian Aplikasi Pada *web browser*

No	Aspek Pengujian	Mozilla Firefox versi 68	Google Chrome versi 78
1	Monitoring *Server* secara *real time*	√	√
2	Menyajikan log monitoring *server*	√	√
3	Fungsi *create, read, update, delete*	√	√
4.	Fungsi kontrol *server* meliputi *update repository, install package, restart service, uninstall package, reboot, shutdown,* kirim file ke server	√	√
5	Log Kontrol *server*	√	√

Pada tahap pengujian fitur *website* pada beberapa *web browser* 100% berjalan pada ketiga *web browser*.

4.7 Pengujian Fitur monitoring dan kontrol server

Pengujian fitur monitoring dan kontrol *server* pada 2 jenis server yang terinstall sistem operasi yang berbeda, yaitu sistem operasi linux Debian dan linux ubuntu.

Table 2 pengujian fitur kontrol server

No	Fitur	Server	
		Debian	Ubuntu
1	Monitoring *server* secara *real time*	√	√
2	Menampilkan informasi server	√	√
3	fitur *update repository*	√	√
4	fitur *install* dan *uninstall package*	√	√
5	Fitur kelola *service* pada *server*	x	√
6	Fitur pengiriman *file*	√	√
7	Fitur *reboot* dan *poweroff*	√	√
8	*Notifikasi*	√	√

Terdapat fitur yang sedikit kurang sempurna karena pada aplikasi ini karena proses *get data stream* dari *server* debian dan ditampilkan dalam *website* memerlukan waktu sedikit lama. Serta pada saat ingin menggunakan fitur *update repository* diserver Debian diharuskan melakukan pembersihan *cache* pada *server* tersebut.

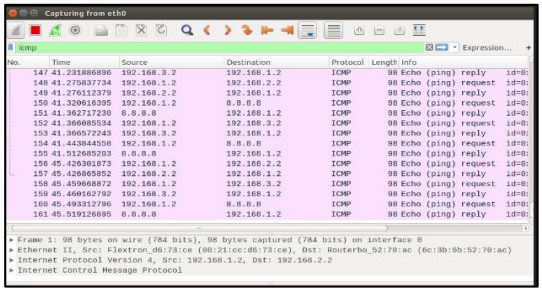
4.8 Pengujian Fitur Monitoring dengan protocol ICMP

Pengujian fitur monitoring *server* pada 2 jenis server yang terinstall sistem operasi yang berbeda, yaitu sistem operasi linux Debian dan linux ubuntu dengan menggunakan *protocol* ICMP *(Internet Control Message Protocol)*.

Table 3 Pengujian monitoring dengan ICMP

No	Sistem Operasi Server	Kondisi Server	
		UP	DOWN
1	Server Ubuntu	√	√
2	Server Debian	√	√
3	Log monitoring	√	√
4	Notifikasi telegram	√	√

Pada tahap pengujian fitur monitoring pada aplikasi ini berjalan dengan baik. Delay untuk monitoring server berkisar 5 detik.



Gambar 9 Pengujian *wireshark*

Pada gambar 9 adalah hasil pengujian dan pembuktian bahwa fitur monitoring pada aplikasi ini berjalan pada *protocol* yang sesuai ayitu ICMP *(Internet Control Message Protocol)*.

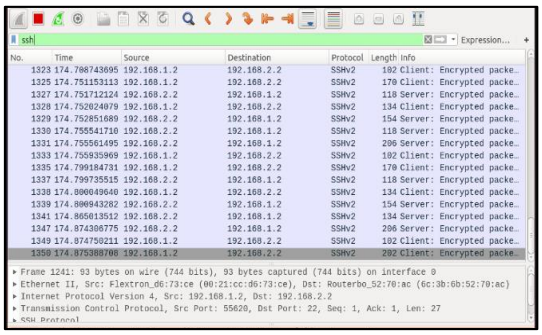
4.9 Pengujian Fitur Kontrol server dengan *protocol* SSH

Pengujian fitur kontrol *server* pada 2 jenis *server* yang terinstall sistem operasi yang berbeda, yaitu sistem operasi linux Debian dan linux ubuntu dengan menggunakan *protocol SSH (Secure Shell)*.

Table 4 Pengujian Kontorl server

No	Fitur	Server	
		Server Debian	Server Ubuntu
1	*Update Repository*	x	√
2	*Install Package*	√	√
3	*Uninstall Package*	√	√
4	Kelola *service* (*Start, Stop, Restart*)	√	√
5	Kirim File ke server	√	√
6	*Poweroff*	√	√
7	*reboot*	√	√
8	Log Kontrol server	√	√

Terdapat fitur yang sedikit kurang sempurna karena pada aplikasi ini karena proses *get data stream* dari *server* debian dan ditampilkan dalam *website* memerlukan waktu sedikit lama. Serta pada saat ingin menggunakan fitur *update repository* diserver Debian diharuskan melakukan pembersihan *cache* pada *server* tersebut.



Gambar 10 pengujian Wireshark

Pada gambar 10 adalah hasil pengujian dan pembuktian bahwa fitur kontrol *server* pada aplikasi ini berjalan pada *protocol* yang sesuai ayitu *SSH (Secure Shell)*.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berikut adalah kesimpulan dari proses pembuatan dan pengujian dari aplikasi "monitoring dan controlling server":

1. Monitoring server secara *real time* dapat diakses melalui halaman *website*. Pencatatan waktu log kondisi server *up* atau *down* dapat berjalan.
2. Notifikasi kepada melalui *telegram* apabila server dalam kondisi *down*.
3. Fungsional dari fitur kontrol server dapat berjalan dengan baik di linux ubuntu, namun kontol server di server debian kurang berfungsi secara optimal.
4. Pencatatan log kontrol *server* dapat diakses melalui website.

5.2 Saran

Pada aplikasi monitoring dan control server yang masih memiliki beberapa kekurangan. Maka untuk pengembangan selanjutnya dari aplikasi ini agar menjadi lebih baik lagi. Adapun saran untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Penyempurnaan fitur *update repository* yang dilakukan diserver debian
2. Fitur monitoring jaringan server untuk memantau kondisi *link* pada jaringan.
3. Penambahan fitur untuk control seperti fitur untuk melakukan konfigurasi server dan backup server melalui website.
4. Fitur control server dapat digunakan untuk beberapa jenis distro linux lainya seperti distro linux centos, RedHat

DAFTAR PUSTAKA

[1] Arifin, M. A. S., & Zulius, A. (2019). PERANCANGAN SISTEM KEAMANAN JARINGAN PADA UNIVERSITAS BINA INSAN LUBUKLINGGAU MENGGUNAKAN TEKNIK DEMILITARIZED ZONE (DMZ). Jusikom: Jurnal Sistem Komputer Musirawas, 4(1), 19-24.

[2] Jusuf, H. (2015). Penggunaan Secure Shell (SSH) Sebagai Sistem Komunikasi Aman Pada Web Ujian Online. Bina Insani ICT Journal (OLD), 2(2), 75-84.

[3] Faisol, A. (2017). Implementasi SMS Gateway Sebagai Sistem Monitoring Kinerja Jaringan Komputer. *Prosiding SENATEK 2015*, *1*(A), 503-507.

[4] Arifin (2017), implementasi monitoring jaringan menggunakan raseberry PI dengan notifikasi memanfaatkan SMTP

[5] Ariwibisono, F. X., Basuki, A., & Ramdani, F. (2018). Hybrid PRoPHET-Epidemic Routing Protocol for Optimizing Possibility of Sending Messages in Remote Fishermen Residential Area. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(2-3), 31-36.

[6] Rendy (2019), Implementasi Sequential Search Untuk Monitoring Log Client Pada Server Linux

[7] Sakti, B., Aziz, A., & Doewes, A. (2013). Uji Kelayakan Implementasi SSH sebagai Pengaman FTP Server dengan Penetration Testing. *ITSMART: Jurnal Teknologi dan Informasi*, *2*(1), 44-51.