

IMPLEMENTASI PORTABILITAS VOIP MENGGUNAKAN ASTERISK PADA OPENWRT

Ardian Bakti Prasetyo¹

¹Teknik Informatika Universitas Dian Nuswantoro Semarang

Email : 111201005197@mhs.dinus.ac.id

ABSTRAK

Perkembangan teknologi informasi saat ini tergolong sangat pesat seiring dengan terjadinya revolusi teknologi informasi. Demikian juga pada bidang komunikasi yang teknologi serta layanannya berkembang dengan pesat. Terciptanya teknologi VoIP memberikan alternatif kepada masyarakat dalam melakukan panggilan suara. Adanya teknologi baru dalam melakukan panggilan suara ini menjadikan masyarakat ingin mencoba mengaplikasikan VoIP di lingkungannya. Namun untuk dapat mengaplikasikan VoIP, masyarakat dihadapkan pada berbagai macam pilihan sulit, di antaranya adalah membeli *server* VoIP fisik yang telah jadi, berlangganan layanan VoIP pada *virtual server* atau membangun *server* VoIP sendiri. Apabila keputusan yang diambil adalah memiliki *server* fisik, maka harus dipertimbangkan ukuran serta spesifikasinya apakah berupa komputer *server* atau *embedded system*. Keputusan tersebut tergantung pada pilihan masing – masing individu. Namun apabila ingin membangun *server* VoIP tentu bukanlah hal yang mudah. Dibutuhkan instalasi serta konfigurasi yang panjang sebelum *server* siap untuk digunakan. Dengan menggunakan teknik duplikasi *firmware*, pengguna dapat melakukan *cloning firmware* dari *server* VoIP satu dengan yang lainnya. Sebagai contoh membangun *server* VoIP menggunakan sebuah *router* yang menggunakan sistem operasi OpenWRT dan Asterisk. Pengguna tidak perlu melakukan instalasi serta konfigurasi dari awal. Pengguna cukup melakukan *cloning firmware* maka *router* miliknya juga akan berfungsi menjadi *server* VoIP tanpa perlu melakukan instalasi dan konfigurasi.

Kata kunci : VoIP, OpenWRT, Asterisk, *embedded system*, duplikasi, *cloning firmware*.

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini tergolong sangat pesat seiring dengan terjadinya revolusi teknologi informasi. Demikian juga pada bidang komunikasi yang teknologi serta layanannya berkembang dengan pesat. Alternatif lain untuk melakukan panggilan suara antar pengguna seluler yaitu menggunakan VoIP. Adanya teknologi VoIP ini bukan sebagai pengganti teknologi GSM, tetapi untuk memberikan alternatif pengguna dalam melakukan panggilan suara [1]. VoIP bekerja dengan cara menggunakan jaringan *Internet Protocol* (IP) untuk mengirimkan data paket suara. Dengan begitu, VoIP dapat bekerja pada jaringan lokal (*intranet*) maupun internet. Syarat minimal untuk dapat menggunakan VoIP adalah harus ada satu buah *server* VoIP (dapat berupa PC atau akses poin dan *router*) dan dua buah klien. Untuk sisi *server*, pengguna dapat menggunakan *server* VoIP yang telah disediakan oleh berbagai *vendor* maupun dengan *server* VoIP milik pengguna. Untuk memiliki *server* VoIP ada dua alternatif, yang paling mudah adalah dengan membeli perangkat *server* yang sudah jadi (tinggal pakai) atau pengguna dapat membangun *server* VoIP sendiri [2].

Membangun sebuah *server* VoIP tentu harus didasari atas beberapa pertimbangan, seperti *hardware* (spesifikasi, ukuran fisik), sistem operasi (*opensource* atau berbayar), dan *software* yang dipakai untuk mengelola VoIP. Untuk sisi *hardware* dapat menggunakan akses poin atau *router* karena memiliki ukuran serta daya listrik yang tidak terlalu besar. Selain itu *router* maupun akses poin dapat dipakai secara terus menerus dibanding dengan komputer PC. Kemudian untuk sisi sistem operasi dapat menggunakan OpenWRT karena OpenWRT didesain untuk bekerja pada akses poin maupun *router* [3]. Untuk sisi *software* dapat menggunakan Asterisk, karena Asterisk merupakan bagian inti pada banyak *software* VoIP lain sehingga dengan Asterisk pengguna dapat menekan penggunaan sumber daya pada *hardware* server karena *server* VoIP yang digunakan bukan berupa komputer PC melainkan akses poin atau *router* [4].

Membangun sebuah *server* VoIP memiliki banyak tantangan. Hal yang paling sering menjadi kendala adalah masalah konfigurasi *server* tersebut. Berbeda dengan *server* yang memang didesain untuk VoIP (tinggal pakai), *server* yang dibangun dari awal harus melalui konfigurasi yang cukup panjang, di antaranya adalah instalasi sistem operasi, instalasi *software* pendukung VoIP (Asterisk), dll. Tentu akan sangat membantu apabila konfigurasi tersebut dapat disimpan dan kemudian digunakan pada perangkatan *server* VoIP lain sehingga pengguna hanya perlu mengubah sedikit konfigurasi yang telah ada. Hal tersebut dapat dilakukan pada Asterisk dan sistem operasi OpenWRT [3].

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah mengimplementasikan teknologi VoIP agar dapat *portable* menggunakan Asterisk pada OpenWRT sehingga memudahkan pengguna dalam mengaplikasikan teknologi VoIP.

1.3 Batasan Masalah

Agar penelitian ini tidak melebar dan keluar dari bahasan serta mencapai tujuan yang diharapkan, maka penulis membatasi permasalahan penelitian sebagai berikut:

1. Server VoIP yang digunakan adalah *router* Huawei HG553 dan TP-Link MR3020 dengan menggunakan *firmware* OpenWRT dan Asterisk.
2. Penelitian membahas implementasi teknologi VoIP *portable* pada jaringan lokal (WLAN) dengan kondisi server VoIP terhubung langsung pada kedua klien menggunakan WLAN.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah memberikan memudahkan pengguna yang ingin mengaplikasikan teknologi VoIP dengan mengaplikasikan teknologi VoIP *portable* menggunakan Asterisk pada OpenWRT.

1.5 Manfaat Penelitian

Manfaat dari penelitian adalah:

1. Penelitian ini diharapkan dapat memudahkan pengguna dalam mengimplementasikan teknologi VoIP menggunakan server VoIP yang *portable*.
2. Penelitian ini diharapkan dapat menjadi alternatif untuk pengguna yang ingin mengaplikasikan VoIP namun tidak ingin bergantung pada *dedicated server* yang telah disediakan oleh *vendor* tertentu.
3. Penelitian ini diharapkan dapat memberikan gambaran bahwa konfigurasi yang telah ada pada *server* VoIP tertentu dapat diduplikasi dan dapat diaplikasikan pada perangkat lain sehingga memudahkan pengguna dapat dengan mudah mengimplementasikan teknologi VoIP.

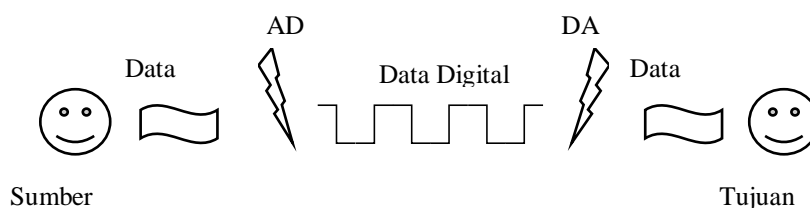
2. TINJAUAN PUSTAKA

2.1 Voice over Internet protocol (VoIP)

VoIP (*Voice over Internet Protocol*) merupakan teknologi yang mampu mengirimkan data suara dan video melalui jaringan internet *protocol* (IP). Berbeda dengan teknologi GSM, VoIP tidak menggunakan teknik *circuit switching* tetapi menggunakan teknik *packet switching*. VoIP memanfaatkan infrastruktur internet yang sudah ada untuk berkomunikasi seperti layaknya menggunakan telepon biasa namun dengan biaya yang lebih rendah untuk berkomunikasi dengan pengguna VoIP lainnya kapan saja dan dimana saja [5].

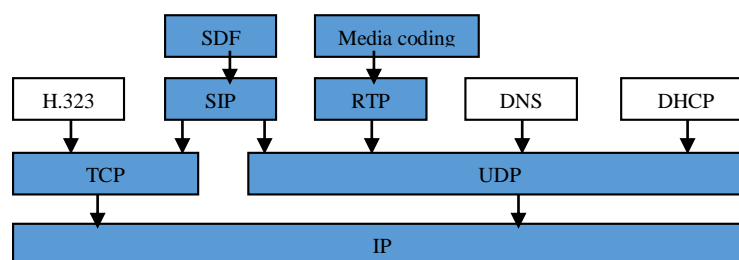
Teknik dasar VoIP adalah melakukan percakapan telepon dengan menggunakan jalur komunikasi data pada suatu jaringan, sehingga teknologi memungkinkan untuk melakukan komunikasi suara menggunakan jaringan yang berbasis *internet protocol* (IP) untuk berjalan di atas infrastruktur jaringan *packet network*. Sedangkan jaringan yang digunakan dapat berupa internet atau intranet. VoIP bekerja dengan cara mengubah suara menjadi format digital tertentu yang dapat dikirimkan melalui jaringan *internet protocol* (IP). Standarisasi protokol komunikasi pada teknologi VoIP adalah SIP (*Session Initiation Protocol*) dan H.323. Namun, dalam perkembangannya SIP lebih banyak digunakan oleh pengembang teknologi VoIP [5].

Pada dasarnya VoIP bekerja dengan cara mengubah data analog ke dalam bentuk data digital. Kemudian data digital tersebut akan dipecah ke dalam paket – paket kecil. Paket – paket kecil tersebut kemudian di kirim ke tujuan. Setelah sampai di tujuan paket – paket data tersebut akan diubah kembali ke dalam bentuk data analog. Untuk lebih jelasnya akan dijelaskan dalam bentuk gambar sebagai berikut [5].



Gambar 2.1 Cara kerja VoIP [5]

Sedangkan protokol – protokol yang digunakan oleh VoIP adalah sebagai berikut.



Gambar 2.2 Internet multimedia protocol stack [5]

Pada Gambar 2.2, protocol yang diarsis merupakan protocol yang digunakan pada VoIP berbasis SIP. Dapat disimpulkan bahwa VoIP menggunakan TCP dan UDP sebagai transport layer-nya.

2.2 Asterisk

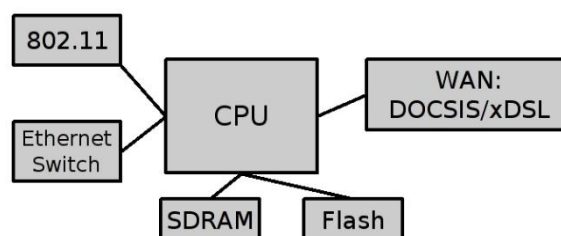
Asterisk merupakan perangkat lunak yang bersifat *open source*. Asterisk dikembangkan dan disponsori oleh digium. Asterisk pada dasarnya adalah sebuah *software telephony* yang memungkinkan pengembang untuk mengembangkan fitur-fitur lain yang tidak terdapat pada telepon analog pada umumnya, seperti IVR, *Teleconference* dan lainnya. Asterisk biasanya digunakan untuk membangun sebuah server mini telekomunikasi VoIP. Alasan pemilihan Asterisk dalam penelitian ini dikarenakan satu – satunya paket modul VoIP yang tersedia pada OpenWRT hingga saat penelitian ini dibuat adalah Asterisk [6] [7]. sebagian besar software VoIP server yang ada menggunakan Asterisk sebagai intinya sehingga tidak diragukan lagi kemampuannya. Selain itu Asterisk dapat berjalan pada semua sistem operasi berbasis unix sehingga akan mudah diaplikasikan di banyak perangkat [8].

Asterisk dapat berjalan pada OpenWRT karena asterisk merupakan paket modul dalam OpenWRT. Agar Asterisk dapat bekerja dalam OpenWRT dibutuhkan paket – paket penunjang lain. Paket – paket tersebut adalah libopenssl, zlib, libncurses, terminfo, libpopt, libpthread. Paket – paket tersebut merupakan paket yang diperlukan untuk fungsi dasar Asterisk. Fungsi dasar Asterisk dalam penelitian ini merupakan panggilan suara melalui IP [7].

2.3 OpenWRT

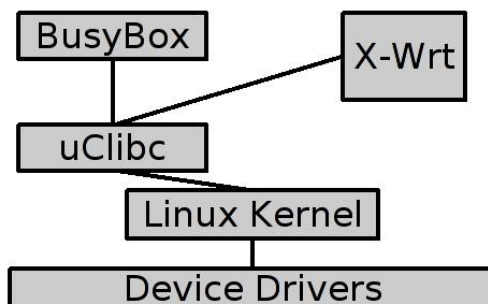
OpenWRT merupakan salah satu distribusi linux yang didesain untuk perangkat *embedded*. OpenWRT menyediakan *file system* dengan manajemen paket. Yang dimaksud paket adalah modul-modul dari aplikasi terpisah yang dapat diimplementasikan pada OpenWRT [3].

OpenWRT merupakan sistem operasi yang dirancang untuk perangkat *embedded*. Salah satu contoh penerapan sistem operasi OpenWRT adalah pada *router*. Berikut ini adalah skema dari *hardware* pada *wireless router* pada umumnya.



Gambar 2.3 Skema hardware pada wireless router [9]

Sedangkan untuk kedudukan OpenWRT sendiri pada sebuah perangkat *embedded* adalah sebagai berikut.



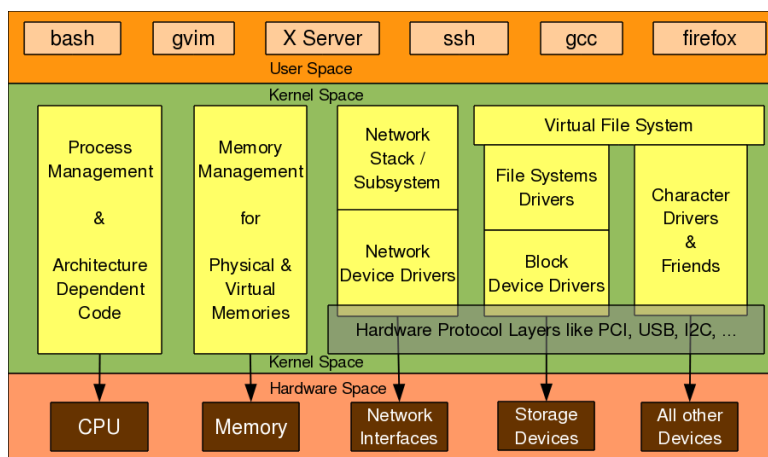
Gambar 2.4 Skema *software* OpenWRT pada *wireless router* [9]

OpenWRT menggunakan *tools* yang umumnya juga terdapat pada linux. *Tools* tersebut di antaranya uClibc, busybox, shell interpreter, abstraksi perangkat keras dan juga paket manager. Berikut adalah bagan arsitektur *software* OpenWRT [10].

UCI	IPKG	User Program
Busybox		
uClibc		
Linux Kernel		

Gambar 2.5 Arsitektur *software* pada OpenWRT

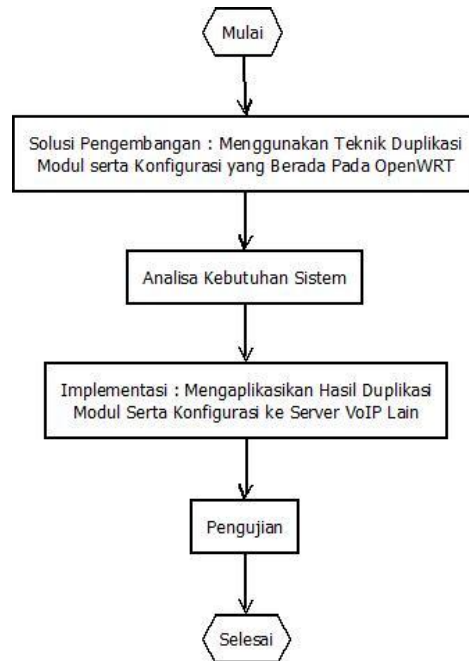
Pada penelitian ini paket – paket yang diinstal oleh pengguna berada pada User Program. Asterisk merupakan paket modul yang diinstal pada OpenWRT dan akan berada pada bagian User Program. Paket – paket yang membutuhkan akses ke *hardware* akan melewati Linux Kernel dan kemudian akan diteruskan ke *hardware* yang dituju. Berikut adalah bagan dari kernel linux.



Gambar 2.6 Bagan kernel linux [11]

3. METODE PENELITIAN

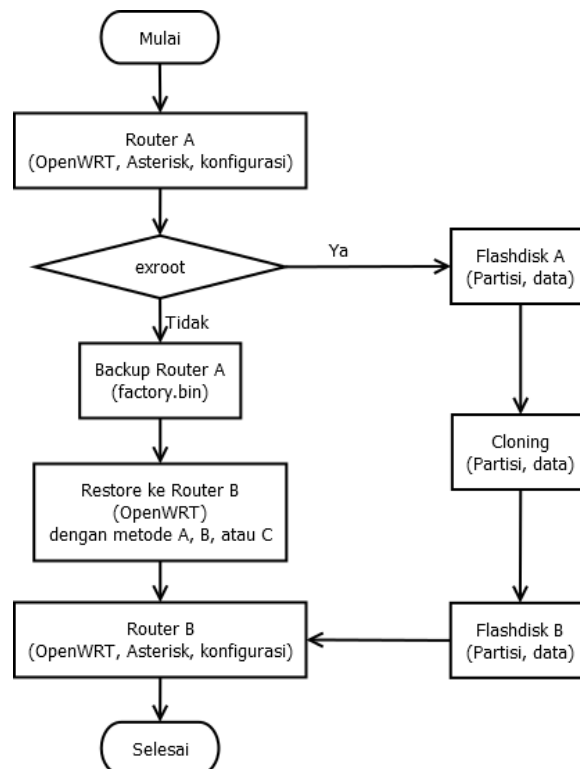
Penelitian ini menggunakan teknik duplikasi konfigurasi dari satu *server* VoIP ke *server* VoIP lain. Di bawah ini adalah kerangka penelitian VoIP *portable* menggunakan Asterisk pada sistem operasi OpenWRT.



Gambar 3.1 Kerangka Penelitian VoIP *portable*

3.1 Metode Penyelesaian Masalah

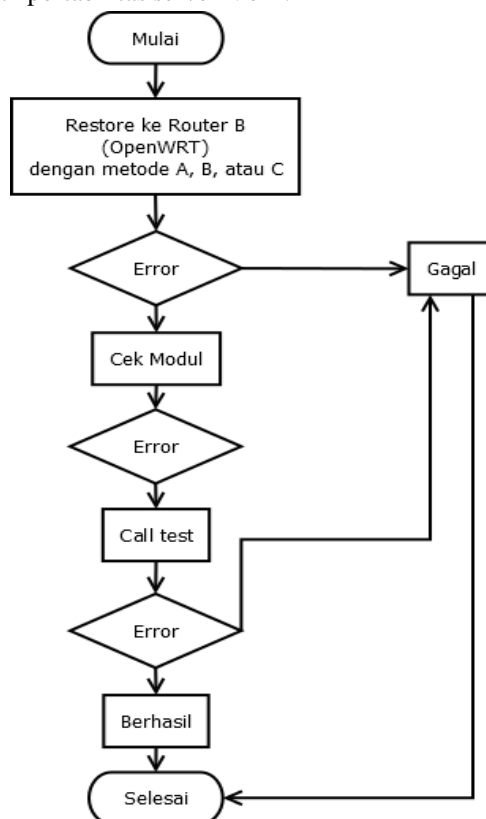
Pada penelitian ini metode yang digunakan untuk menerapkan VoIP *portable* adalah dengan membuat *backup* konfigurasi beserta paket – paket yang telah terinstal pada *server* VoIP. Kemudian file hasil *backup* tersebut akan diterapkan pada *server* VoIP yang baru. Selanjutnya *server* VoIP yang baru akan diuji kelayakannya sesuai dengan prosedur yang ada. Berikut adalah skema implementasi VoIP *portable*.



Gambar 3.2 Implementasi VoIP *portable*

3.2 Pengujian Portabilitas VoIP Server

Pada penelitian ini pengujian portabilitas VoIP *server* terdiri dari tiga bagian. Bagian pertama adalah pengujian pada proses *restore* dari *router* sumber ke *router* tujuan. Hal yang digunakan sebagai indikator keberhasilan pada bagian ini adalah tidak adanya pesan *error* pada saat proses *restore* file *factory.bin*. Bagian kedua adalah pengecekan modul beserta konfigurasi pada *server* yang baru. Hal yang dijadikan indikator keberhasilan pada bagian ini adalah dengan adanya paket modul asterisk pada *server* yang baru. Bagian ketiga adalah pengujian fungsi panggilan suara melalui *server* VoIP yang baru. Hal yang digunakan sebagai indikator keberhasilan pada bagian ini adalah berhasilnya komunikasi suara antara dua klien VoIP. Di bawah ini adalah diagram alur dari proses pengujian portabilitas *server* VoIP.



Gambar 3.3 Bagan proses pengujian portabilitas *server* VoIP

4. IMPLEMENTASI

Implementasi dari penelitian ini adalah dengan melakukan backup dan restore konfigurasi server VoIP. Metode backup yang digunakan adalah dengan cara melakukan duplikasi partisi dari firmware (mtd5) dengan perintah "cat /dev/mtd5 > /tmp/factory.bin". Sedangkan metode restore yang diuji ada tiga cara. Cara pertama menggunakan web GUI OpenWRT (LuCI). Cara kedua dengan perintah "sysupgrade". Sedangkan cara ketiga adalah dengan menggunakan perintah "mtd write". Berikut adalah kondisi – kondisi yang diuji pada penelitian ini.

- Server VoIP router Huawei HG553 tanpa exroot metode restore LuCI
- Server VoIP router Huawei HG553 tanpa exroot metode restore sysupgrade
- Server VoIP router Huawei HG553 tanpa exroot metode restore mtd write
- Server VoIP router Huawei HG553 dengan exroot metode restore LuCI
- Server VoIP router Huawei HG553 dengan exroot metode restore sysupgrade
- Server VoIP router Huawei HG553 dengan exroot metode restore mtd write
- Server VoIP router TP-Link MR3020 tanpa exroot metode restore LuCI
- Server VoIP router TP-Link MR3020 tanpa exroot metode restore sysupgrade
- Server VoIP router TP-Link MR3020 tanpa exroot metode restore mtd write
- Server VoIP router TP-Link MR3020 dengan exroot metode restore LuCI
- Server VoIP router TP-Link MR3020 dengan exroot metode restore sysupgrade
- Server VoIP router TP-Link MR3020 dengan exroot metode restore mtd write
- Lintas Device dengan metode restore LuCI

- n. Lintas Device dengan metode restore sysupgrade
- o. Lintas Device dengan metode restore mtd write

5. HASIL

Dari penelitian pada bab sebelumnya diperoleh data sebagai berikut.

Tabel 5.1 Data hasil penelitian

No	Server VoIP	Metode yang digunakan	Exroot	Hasil	
				Restore	Panggilan suara
1	Huawei HG553	Menu restore backup LuCI	Tanpa exroot	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
2	Huawei HG553	Sysupgrade pada PuTTY	Tanpa exroot	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
3	Huawei HG553	Mtd write pada PuTTY	Tanpa exroot	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
4	Huawei HG553	Menu restore backup LuCI	Ya	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
5	Huawei HG553	Sysupgrade pada PuTTY	Ya	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
6	Huawei HG553	Mtd write pada PuTTY	Ya	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
7	TL-MR3020	Menu restore backup LuCI	Tanpa exroot	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
8	TL-MR3020	Sysupgrade pada PuTTY	Tanpa exroot	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
9	TL-MR3020	Mtd write pada PuTTY	Tanpa exroot	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
10	TL-MR3020	Menu restore backup LuCI	Ya	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
11	TL-MR3020	Sysupgrade pada PuTTY	Ya	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
12	TL-MR3020	Mtd write pada PuTTY	Ya	Tanpa pesan error (Berhasil)	Dua klien terhubung (Berhasil)
13	Lintas device	Menu restore backup LuCI	Tanpa exroot	Image file does not contain supported a format (Gagal)	Dua klien gagal terhubung (Gagal)
14	Lintas device	Sysupgrade pada PuTTY	Tanpa exroot	Image check failed (Gagal)	Dua klien gagal terhubung (Gagal)
15	Lintas device	Mtd write pada PuTTY	Tanpa exroot	Bad header, image check failed (Gagal)	Dua klien gagal terhubung (Gagal)

Dari data di atas dapat disimpulkan bahwa proses *backup* dan *restore* pada *device* yang memiliki spesifikasi sama memiliki tingkat keberhasilan 100%. Sebaliknya saat dicoba pada *device* yang memiliki spesifikasi berbeda selalu gagal.

6. KESIMPULAN DAN SARAN

Dari hasil penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa dengan metode *backup* dan *restore* yang telah dibahas pada penelitian ini dapat digunakan untuk mengimplementasikan teknologi VoIP agar dapat *portable* menggunakan Asterisk pada OpenWRT. Hal tersebut dapat memberikan alternatif untuk pengguna yang ingin mengaplikasikan VoIP namun tidak ingin bergantung pada *dedicated server* yang telah disediakan oleh *vendor* tertentu.

Saran untuk penelitian selanjutnya yaitu meneliti bagaimana mengimplementasikan teknologi VoIP agar dapat *portable* lintas *platform* menggunakan Asterisk pada sistem Operasi OpenWRT.

7. DAFTAR PUSTAKA

- [1] A. Arjona, "A Study of Mobile VoIP Performance in Wireless Broadband Networks," Helsinki University of Technology, Finland, 2009.
- [2] A. Bakre, "Intel VoIP over WLAN Architecture," in *WICON '06 Proceedings of The 2nd Annual International Workshop on Wireless Internet*, New York, 2006.
- [3] T. O. d. team, "OpenWrt Wireless Freedom," 2014. [Online]. Available: <https://openwrt.org/>. [Accessed 10 2 2014].
- [4] "Asterisk," Digium, Inc, 2014. [Online]. Available: <http://www.asterisk.org/get-started>. [Accessed 10 2 2014].
- [5] H. F. Y. Y. Domiko Fahdi Jaya Patih, "Analisa Perancangan Server VOIP (Voice Internet Protocol) Dengan Opensource Asterisk Dan VPN (Virtual Private Network) Sebagai Pengaman Jaringan Antar Client," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 1, pp. 42-48, 2012.
- [6] OpenWrt, "OpenWrt Wireless Freedom," 1 Januari 2012. [Online]. Available: [http://wiki.openwrt.org/doc/howto/voip.overview?s\[\]=kamailio](http://wiki.openwrt.org/doc/howto/voip.overview?s[]=kamailio). [Accessed 9 Maret 2014].
- [7] OpenWrt, "OpenWrt Wireless Freedom," 1 Januari 2014. [Online]. Available: <http://wiki.openwrt.org/doc/howto/voip.asterisk>. [Accessed 5 Maret 2014].
- [8] L. M. a. J. V. M. Russell Bryant, *Asterisk™: The Definitive Guide*, United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2013.
- [9] A. Sirotkin, "Linux Journal," 1 Agustus 2007. [Online]. Available: <http://www.linuxjournal.com/article/9691>. [Accessed 5 Maret 2014].
- [10] F. Fainelli, "The OpenWrt Embedded Development Framework," OpenWrt, 2008.
- [11] A. K. Pugaria, "Playing with Systems," 1 Februari 2013. [Online]. Available: <http://sysplay.in/blog/linux-device-drivers/2013/02/linux-device-drivers-for-your-girl-friend/>. [Accessed 2014 Maret 2014].
- [12] A. Syafari, "ilmukomputer.com," 2003. [Online]. Available: <http://www.ilmukomputer.com>. [Accessed 10 2 2014].
- [13] "4gamericas," Oktober 2012. [Online]. Available: <http://www.4gamericas.org/documents/4G%20Mobile%20Broadband%20Evolution-Rel%2010%20Rel%2011%20and%20Beyond%20October%202012.pdf>. [Accessed 11 2 2014].
- [14] S. R. K. Harun Mukhtar, "Penerapan Komunikasi Voice Over Internet Protocol (VOIP) Menggunakan Asterisk Session Initiation Protocol Pada Universitas Muhammadiyah Riau," *Jurnal Fasilkom V.1 No.1 Januari 2012*, vol. 1, pp. 1-7, 2012.
- [15] M. B. M. R. Claudio E. Palazzi, "An OpenWRT Solution for Future Wireless Homes," in *Multimedia and Expo (ICME), 2010 IEEE International Conference*, Suntec, 2010.