

Analisis Kinerja Voip Server pada Wireless Access Point

Effan Najwaini*¹, Ahmad Ashari²

¹Program Studi S2 Ilmu Komputer, FMIPA UGM, Yogyakarta

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *effan.najwaini@gmail.com, ashari@ugm.ac.id

Abstrak

Pada komunikasi VoIP (Voice Over IP) kualitas suara dipengaruhi oleh banyak faktor salah satunya yaitu kualitas server. Pemilihan platform PC atau server yang cocok (baik dari segi harga maupun kinerja) merupakan persoalan utama dalam membangun jaringan VoIP. Kinerja server yang jelek akan menurunkan kualitas suara atau bahkan tidak mampu untuk menghubungkan antar pengguna.

Dalam penelitian ini dilakukan pengujian terhadap kinerja wireless access point Linksys WRT54GL yang dimanfaatkan sebagai VoIP server. Pengujian dilakukan untuk mengetahui berapa banyak panggilan VoIP yang mampu dilayani oleh wireless access point sebagai VoIP server serta berapa waktu yang diperlukan oleh server tersebut untuk dapat memproses setiap sinyal SIP maupun paket RTP.

Berdasarkan hasil pengujian yang dilakukan, VoIP server pada wireless access point mampu melayani komunikasi VoIP dengan baik untuk jumlah panggilan yang sedikit sehingga layak diimplementasikan pada penggunaan skala kecil. Penggunaan metode Native Bridging dalam penanganan media yang dilakukan oleh server dapat meningkatkan jumlah panggilan yang mampu dilayani sebesar 3 hingga 7 kali dibandingkan dengan metode lainnya.

Kata kunci—VoIP, Asterisk, Access Point, WRT54GL, OpenWRT, Kinerja

Abstract

Voice quality on VoIP communication is caused by many factors, one of which is the quality of the server. Choosing PC platform or server which is suitable is the main issue in developing VoIP network. A bad server performance or not equivalent with the most of users will degrade the sound quality or even not able to connect between users.

The test carried out to the performance of the wireless access point Linksys WRT54GL which is used as a VoIP server. The test was carried out to determine how many VoIP calls which are able to be serviced by a wireless access point as a VoIP server and how long the server needs to be able to process every signal of SIP and RTP packet.

Based on the test result performed, the VoIP server on the wireless access point is able to serve VoIP communication well for a few calls number, so it is worth to be implemented on the use of small scale. The use of Native Bridging method in handling the media performed by the server can increase the number of calls that were able to be served about 3 to 7 times compared with other methods.

Keywords— VoIP, Asterisk, Access Point, WRT54GL, OpenWRT, Performance

1. PENDAHULUAN

Munculnya teknologi VoIP yang menawarkan berbagai kelebihan membuat berbagai perusahaan maupun lembaga mulai beralih menggunakan teknologi VoIP dibandingkan dengan telepon tradisional menggunakan PBX. Permasalahan utama dari implementasi teknologi VoIP yaitu masalah waktu *call setup* dan QoS (*Quality of Service*). Pada telepon analog menghasilkan kualitas suara yang jernih dan waktu *call setup* yang cukup singkat. Pada VoIP kualitas suara dipengaruhi oleh banyak faktor diantaranya yaitu: *bandwidth*, beban jaringan, *delay*, serta kinerja *server* sehingga perlu diperhitungkan jumlah pengguna, kualitas jaringan dan kualitas *server* untuk mendapatkan QoS yang baik [1]. Kinerja *server* yang jelek akan menurunkan kualitas atau bahkan tidak mampu untuk menghubungkan antar *user*. Selain masalah *hardware*, kinerja *server* juga ditentukan oleh *software* yang digunakan. *Software* untuk membangun PBX yang *open source* dan paling efisien serta paling banyak digunakan hingga saat ini yaitu Asterisk [2,3]. Munculnya OpenWrt memungkinkan perangkat *embedded system* seperti *access point* untuk di-*install* berbagai aplikasi termasuk Asterisk [3]. *Access point* yang di-*install* Asterisk dapat berfungsi sebagai VoIP *server*, sehingga tidak lagi diperlukan sebuah komputer untuk membangun sebuah jaringan VoIP. Hal ini akan menghemat dari segi biaya pengadaan komputer dan juga biaya operasional[3]. Tetapi, karena keterbatasan perangkat keras pada *access point* dapat membuat kinerjanya menjadi lebih rendah dibandingkan penggunaan komputer untuk *server* VoIP. Sampai saat ini belum ada penelitian mengenai analisis kinerja dari VoIP *server* pada *wireless access point* sehingga dirasakan perlu adanya penelitian untuk mengetahui kinerjanya. Pada penelitian ini akan dibuat sebuah *server* VoIP menggunakan Asterisk yang di-*install* pada *wireless access point* dengan *firmware* OpenWrt. Kemudian dari *server* tersebut akan dilakukan analisis kinerja sehingga dapat diketahui berapa banyak panggilan yang mampu dilayani. Sebagai perbandingan, akan dilakukan juga pengujian terhadap VoIP *server* Asterisk menggunakan PC dengan prosedur yang sama.

2. METODE PENELITIAN

2.1 Peralatan

Peralatan yang digunakan pada penelitian ini yaitu :

1. Access Point Linksys WRT54GL

Linksys WRT54G merupakan perangkat yang berfungsi sebagai *wireless access point* yang memungkinkan perangkat *wireless* terhubung ke jaringan *wired*. Selain sebagai *wireless access point*, perangkat ini juga dapat sekaligus berfungsi sebagai *router*. Pada penelitian ini, perangkat Linksys WRT54GL akan dimanfaatkan sebagai VoIP *server*.

2. Personal Computer (PC)

Komputer digunakan sebagai VoIP *server* untuk membandingkan kinerjanya terhadap perangkat *wireless access point* yang digunakan untuk VoIP *server*. Komputer ini akan menggunakan sistem operasi Trixbox 2.6 dengan Asterisk 1.4 sebagai *software server* VoIP

3. Notebook dan Handphone

Pada penelitian ini digunakan 7 buah *notebook* dan 2 buah *handphone* dimana 3 buah *notebook* sebagai *notebook* utama yang berfungsi sebagai *call generator*, *client* VoIP dan untuk melakukan *remote SSH* ke *access point* sedangkan *handphone* dan *notebook* lainnya hanya sebagai *client* VoIP. Ketiga *notebook* ini digunakan untuk melakukan tes kinerja terhadap *server* VoIP di WRT54GL dan komputer *server*.

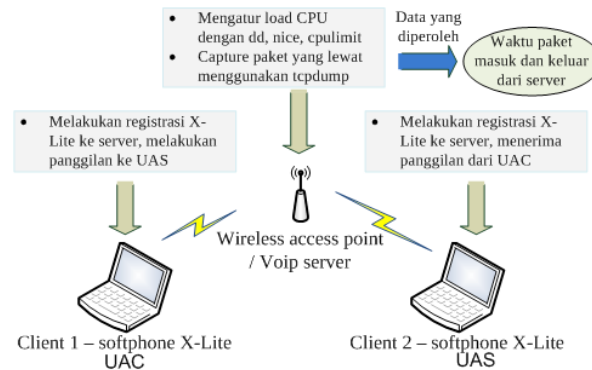
Sedangkan lunak yang digunakan yaitu:

- **OpenWRT** versi Kamikaze 8.09.2 sebagai sistem operasi / *firmware* pada perangkat Linksys WRT54GL. OpenWRT Kamikaze 8.09.2 berbasis linux dengan kernel 2.4.
- **Toolchain** merupakan program yang berguna untuk meng-*compile source code* OpenWRT dalam bentuk *file binary*. Toolchain akan melakukan *cross-compilation* ke platform perangkat *wireless access point* yang digunakan.
- **Asterisk** versi 1.4 sebagai *software server* VoIP yang berfungsi menangani berbagai pensinyalan VoIP. *Software* ini yang digunakan untuk membuat server VoIP baik itu pada perangkat WRT54GL maupun PC.
- **SIPp** merupakan aplikasi *open source* yang berfungsi untuk membangkitkan panggilan dalam sinyal SIP. SIPp digunakan untuk keperluan tes kinerja pada sebuah *server* VoIP. Pengguna aplikasi ini dapat membuat sebuah skenario pengetesan dalam bahasa XML yang nantinya akan dibaca dan dieksekusi oleh aplikasi SIPp.
- **TCPdump** dan **Wireshark** merupakan aplikasi *packet analyzer* yang berfungsi untuk menangkap dan menampilkan paket yang lewat pada suatu *interface* jaringan. TCPdump dijalankan melalui *command line* yang tidak mempunyai tampilan GUI, sedangkan wireshark dijalankan dalam lingkungan desktop yang mempunyai tampilan GUI. TCPdump digunakan untuk menangkap paket yang lewat pada interface VoIP *server* yang selanjutnya ditampilkan dan dianalisa melalui wireshark.
- **X-Lite dan Eyebeam** merupakan *softphone* yang menggunakan pensinyalan SIP. X-lite merupakan *software proprietary* gratis yang mendukung banyak *codec* suara, misalnya GSM, G.711 ulaw, G711 alaw, iLBC, Speex dan lainnya. Sedangkan Eyebeam merupakan versi berbayar dari X-Lite yang terdapat beberapa fitur tambahan, salah satunya yaitu dapat mengaktifkan lebih dari satu SIP *account* sekaligus. Kedua *Software* ini akan digunakan untuk menganalisa waktu proses dari sinyal SIP, waktu proses transfer media (RTP) di *server* dan menganalisa kualitas suaranya.
- **sar** dan **top** merupakan aplikasi pada linux yang berguna untuk menampilkan *load* prosesor dan memori. *Software* ini digunakan untuk menganalisa pengaruh kenaikan jumlah panggilan terhadap penggunaan prosesor dan memori pada *server* VoIP.
- **dd** merupakan perintah linux yang berguna untuk meng-*convert* dan meng-*copy file*. Dalam penelitian ini perintah ini digunakan untuk membuat prosesor sibuk bekerja, sehingga *load processor* dapat mencapai 100%.
- **cpulimit** merupakan aplikasi linux yang berguna untuk membatasi penggunaan prosesor dari suatu proses.
- **nice** merupakan aplikasi linux yang berguna untuk mengubah prioritas dari suatu proses yang sedang berjalan.

2.2 Prosedur pengumpulan data

- Waktu proses dari masing – masing sinyal SIP

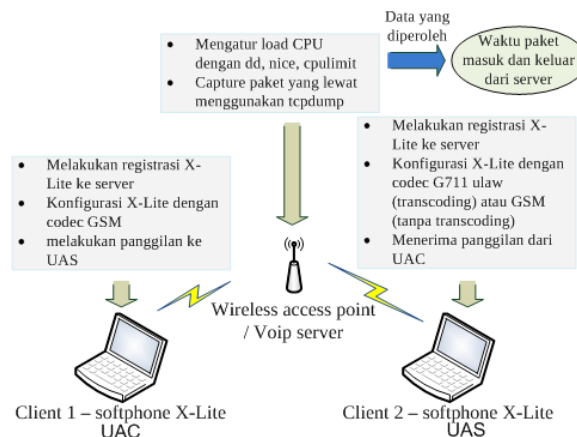
Pengujian dilakukan pada saat *load* CPU 0%, 25%, 50%, 75% dan 100%. Untuk mengatur *load* CPU digunakan tool dd, nice dan cpulimit. Pengujian dilakukan untuk proses pembangunan/pemutusan hubungan. Client menggunakan softphone X-Lite yang kemudian melakukan panggilan voice ke client yang lain. Pengambilan data dengan men-capture paket data yang masuk dan keluar di VoIP server menggunakan tcpdump, sehingga diperoleh data waktu paket masuk ke server dan waktu paket tersebut keluar dari server untuk diteruskan ke client. Untuk mendapatkan waktu proses dilakukan dengan mengurangi waktu keluar dan waktu masuk paket tersebut. Sinyal SIP yang diukur meliputi invite, ringing, ok, bye. Setiap pengambilan data diulang sampai 30 kali. Prosedur pengambilan data dapat dilihat pada Gambar 1.



Gambar 1 Prosedur pengambilan data untuk waktu proses sinyal SIP

- Waktu proses dari paket suara (RTP)

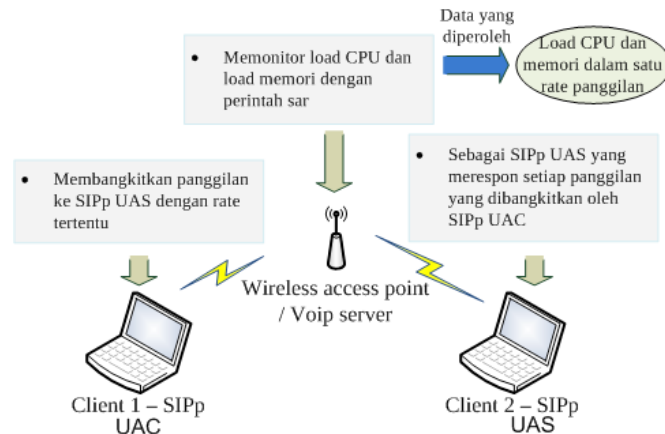
Pengujian dilakukan pada saat *load* CPU 0%, 25%, 50%, 75% dan 100%. Untuk mengatur *load* CPU digunakan tool *dd*, *nice* dan *cpulimit*. *Client* menggunakan *softphone X-Lite* yang kemudian dilakukan registrasi akun ke *server*. Pengambilan data dilakukan untuk 3 metode *forward* RTP yaitu *Generic Bridging* (ada proses *transcoding*) dan *P2P Bridging* (tanpa adanya proses *transcoding*). Pada pengambilan data dengan proses *transcoding*, *client* pertama dikonfigurasi menggunakan *codec* GSM, sedangkan *client* kedua dikonfigurasi menggunakan *codec* G711 ulaw. Dengan adanya perbedaan *codec* ini, *server* akan melakukan proses *transcoding*. Untuk pengambilan data tanpa proses *transcoding* (*P2P Bridging*), kedua *client* dikonfigurasi menggunakan *codec* yang sama yaitu GSM atau G711 ulaw. Pengambilan data dengan men-*capture* paket data yang masuk dan keluar di *VoIP server* menggunakan *tcpdump* seperti yang ditunjukkan pada Gambar 2, sehingga diperoleh data waktu paket masuk ke *server* dan waktu paket tersebut keluar dari *server* untuk diteruskan ke *client*. Untuk mendapatkan waktu proses dilakukan dengan mengurangi waktu keluar dan waktu masuk paket tersebut. Setiap pengambilan data dilakukan untuk komunikasi selama 30 detik.



Gambar 2 Prosedur pengambilan data untuk waktu proses paket RTP

- Maksimum panggilan perdetik

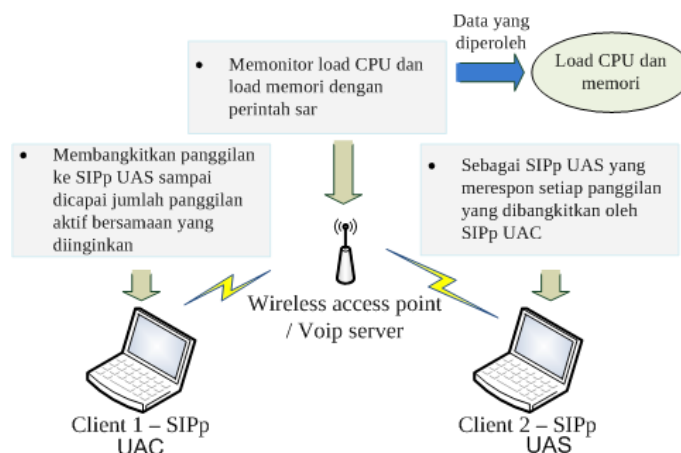
Pengambilan data dilakukan dengan membangkitkan panggilan menggunakan SIPp. Skenario SIPp dibuat tanpa adanya paket suara yang ditransfer, hanya pembangunan dan pemutusan hubungan (sinyal SIP). Pengambilan data dilakukan dengan menaikkan jumlah rate panggilan perdetik pada SIPp sampai penggunaan CPU server mencapai 100% atau panggilan sudah tidak dapat diproses dengan baik oleh *server*. Data *load* CPU *server* dan *load* memori pada saat *rate* panggilan tertentu didapat dengan menggunakan perintah *sar* pada *server*. Data *load* CPU dan memori yang didapat merupakan rata – rata selama 10 detik setelah pembangkitan panggilan oleh *call generator* SIPp. Prosedur pengambilan data untuk maksimum panggilan perdetik ditunjukkan pada Gambar 3.



Gambar 3 Prosedur pengambilan data untuk maksimum panggilan perdetik

- Maksimum panggilan bersamaan

Pengambilan data dilakukan dengan membangkitkan panggilan menggunakan SIPp. Selain membangkitkan sinyal SIP, juga akan dibangkitkan paket – paket RTP dengan *codec* G.711 ulaw dan GSM. SIPp diatur agar membangkitkan panggilan sampai dicapai jumlah tertentu panggilan bersamaan yang aktif. Setiap sesi panggilan yang dibangkitkan, masing – masing panggilan tidak langsung terputus, tetapi mengirim paket RTP sampai dicapai jumlah panggilan bersamaan yang diinginkan. Data *load CPU server* dan *load memori* pada saat *rate* panggilan tertentu didapat dengan menggunakan perintah sar pada *server*. Data *load CPU* dan memori yang didapat merupakan rata – rata selama 10 detik setelah jumlah panggilan bersamaan yang aktif tercapai. Pengambilan data dilakukan untuk 3 metode penanganan RTP paket yaitu Generic Bridging (ada proses *transcoding*), Packet2Packet Bridging (tanpa adanya proses *transcoding*), dan SIP Native Bridge (RTP tanpa melewati *server*). Untuk proses dengan *transcoding*, digunakan *codec* G.711 dan GSM, sedangkan untuk proses tanpa *transcoding* dan tanpa melewati *server*, kedua *generator (user agent)* menggunakan *codec* yang sama, yaitu G.711 atau GSM. Proses pengambilan data untuk panggilan bersamaan ditunjukkan pada Gambar 4.



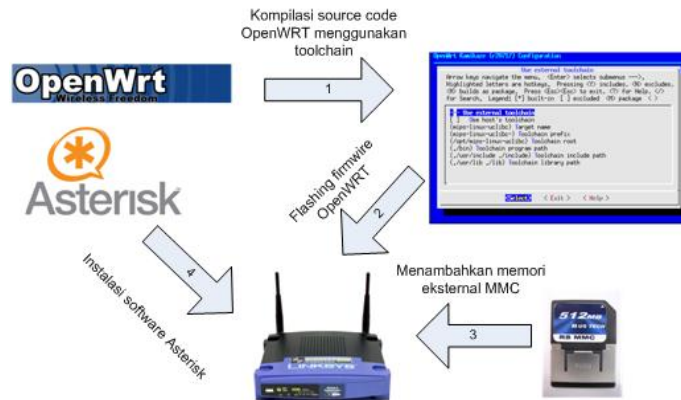
Gambar 4 Prosedur pengambilan data untuk maksimum panggilan bersamaan

- QoS

Pengukuran QoS dilakukan untuk ketiga metode penanganan media RTP oleh server. Pada pengujian ini hitung nilai delay, jitter dan packet loss dari setiap komunikasi pengujian yang dilakukan. Pengujian dilakukan dengan membangkitkan panggilan dan kemudian menangkap data yang ditransfer menggunakan tcpdump atau wireshark.

2.3 Pembuatan server VoIP pada wireless access point

Pada penelitian ini perangkat Linksys WRT54GL akan dibuat mampu untuk menjalankan fungsi sebagai *wireless access point*, serta sekaligus sebagai VoIP server dengan menggunakan *software* asterisk. Tahapan proses pembangunan server VoIP ditunjukkan pada Gambar 5.

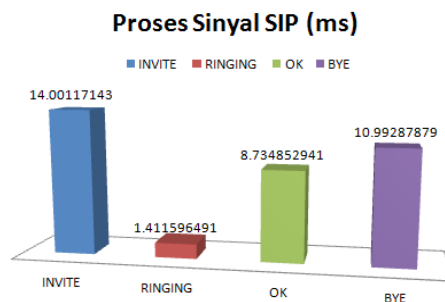


Gambar 5 Proses pembangunan server VoIP pada wireless access point [4]

3. HASIL DAN PEMBAHASAN

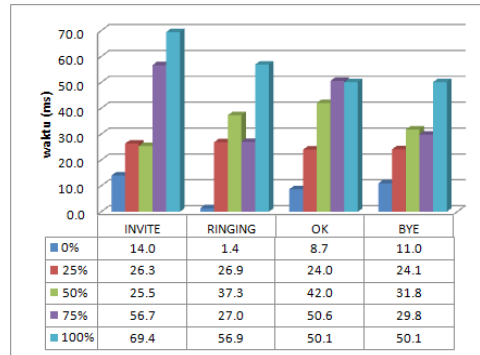
3.1 Waktu Proses Sinyal SIP

Waktu proses sinyal SIP pada *server* Asterisk diperangkat *wireless access point* pada saat kondisi prosesor idle (*utilization* 0%) dapat dilihat pada grafik Gambar 6. Dari grafik tersebut terlihat bahwa sinyal INVITE memerlukan waktu yang paling lama untuk diteruskan yaitu sebesar 14 ms. Ketika menerima sinyal INVITE, *server* Asterisk melakukan banyak proses diantaranya yaitu autentikasi pemanggil, pengecekan *dial plan*, pencarian alamat IP tujuan serta penentuan *codec* yang didukung oleh akun tersebut. Banyaknya proses ini membuat sinyal “invite” diproses paling lama dibandingkan sinyal SIP lainnya.



Gambar 6 Proses Sinyal SIP saat prosesor idle

Percobaan dengan menaikkan penggunaan prosesor membuat waktu proses dari sinyal SIP cenderung meningkat. Pada penggunaan prosesor yang mencapai 100%, server masih mampu melayani panggilan masuk dengan waktu proses sinyal “invite” rata-rata 69,4 ms. Gambar 7 menunjukkan data waktu proses masing-masing sinyal pada saat penggunaan prosesor 0%, 25%, 50%, 75% dan 100%.



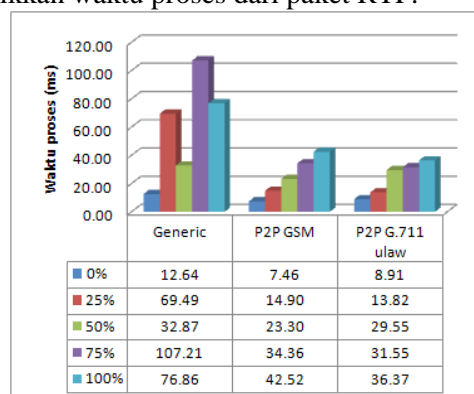
Gambar 7 Grafik waktu proses sinyal SIP

Pengujian waktu proses sinyal SIP menggunakan PC mendapatkan hasil 201,5 ms untuk sinyal INVITE dimana nilai ini jauh lebih besar dibandingkan waktu proses pada wireless access point yaitu selama 14 ms. Pada pengujian PC digunakan OS Trixbox yang dibuat khusus untuk membuat server VoIP menggunakan Asterisk. Asterisk pada Trixbox dibuat dengan fitur dan modul yang sangat lengkap, sedangkan Asterisk pada OpenWRT dibuat dengan fitur dan modul yang minim untuk menghemat penggunaan memori flash pada perangkat. Dial plan pada Trixbox lebih kompleks dan secara default melibatkan tiga buah file konfigurasi dial plan, sedangkan pada OpenWRT dial plan lebih sederhana dan hanya dibuat pada satu file konfigurasi. Banyaknya fitur yang didukung oleh server Asterisk dan kompleksnya dial plan pada Trixbox membuat waktu proses dari sinyal INVITE menjadi lebih lama.

3.2 Waktu proses paket RTP

Pengujian waktu proses dari paket RTP pada metode Generic Bridging pada saat prosesor idle yaitu 12,64 ms sedangkan pada metode Packet2Packet Bridging didapatkan sebesar 7,46 ms untuk komunikasi esame codec GSM dan 8,91 ms untuk codec G.711 ulaw. Pada metode Generic Bridging, server Asterisk melakukan proses transcoding untuk menyamakan codec dari kedua client yang berkomunikasi sehingga memerlukan waktu yang lebih lama, sedangkan pada metode Packet2Packet Bridging, paket RTP masih melewati server tetapi server tidak melakukan proses transcoding karena kedua client yang berkomunikasi sudah menggunakan codec yang sama.

Gambar 8 menunjukkan lamanya waktu proses pada masing-masing metode dengan utilization processor yang ditingkatkan. Hasil yang didapatkan waktu proses tertinggi untuk metode Generic Bridging yaitu 107,21 ms sedangkan pada metode Packet2Packet Bridging dengan codec GSM 42,52 ms dan 36,37 ms pada codec G.711 ulaw. Kenaikan penggunaan prosesor cenderung akan menaikkan waktu proses dari paket RTP.

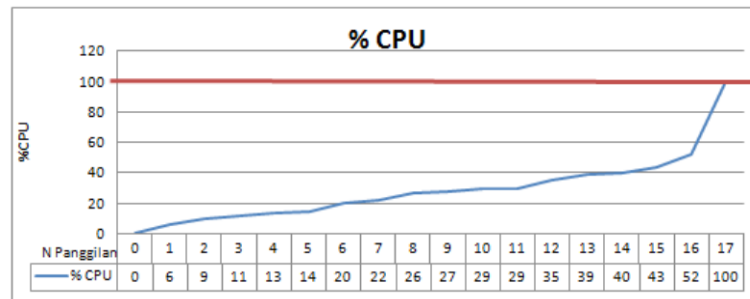


Gambar 8 Grafik waktu proses paket RTP

3.3 Maksimum panggilan perdetik

Grafik pada Gambar 9 menunjukkan besarnya penggunaan prosesor terhadap kenaikan dari rate kedatangan panggilan. Pada rate panggilan sebesar 15 panggilan perdetik penggunaan

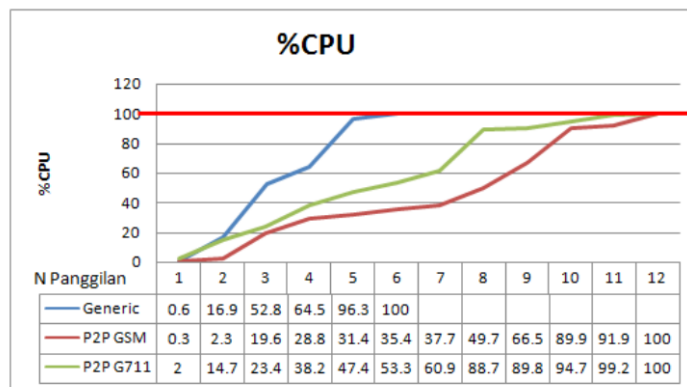
prosesor tidak lebih dari 50%, tetapi jika panggilan dinaikan menjadi 17 panggilan perdetik membuat penggunaan prosesor mencapai 100% dan terjadi banyaknya panggilan yang tidak mampu dilayani oleh server.



Gambar 9 Grafik waktu proses paket RTP

3.4 Maksimum panggilan bersamaan

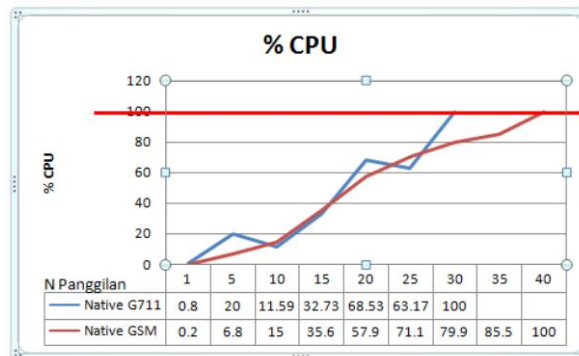
Pengujian panggilan bersamaan dilakukan terhadap tiga metode pengiriman media, yaitu *Generic Bridging*, *P2P Bridging* dan *Native Bridging*. ada pengujian dengan metode *Generic Bridging*, server hanya mampu melayani kurang dari 6 panggilan secara bersamaan, sedangkan pada metode *P2P Bridging* mampu melayani kurang dari 12 panggilan secara bersamaan. Hasil pengukuran maksimum panggilan bersamaan ditunjukkan pada Gambar 10.



Gambar 10 Hasil pengujian panggilan bersamaan metode *Generic Bridging* dan *P2P Bridging*

Penggunaan memori pada pengujian dengan dua metode ini dapat dilihat pada Pada grafik tersebut tidak dapat disimpulkan pengaruh penggunaan metode terhadap penggunaan prosesor. Grafik penggunaan memori terlihat acak sehingga susah untuk disimpulkan pengaruhnya baik terhadap metode yang digunakan maupun terhadap besarnya panggilan bersamaan yang dibangkitkan. Pada saat penggunaan prosesor telah mencapai 100%, memori yang tersisa masih sekitar 300 – 400 KB.

Pada pengujian *Native Bridging*, perangkat *wireless access point* hanya berfungsi mem-forward frame sebagaimana komunikasi jaringan *wireless infrastructure*. Paket RTP tidak masuk ke server Asterisk sehingga penggunaan metode ini dapat meningkatkan jumlah panggilan bersamaan yang mampu dilayani oleh perangkat *wireless access point*. Gambar 11 menunjukkan hasil pengujian pada metode *Native Bridging* dengan codec GSM dan codec *G.711 ulaw*. Pada metode *Native Bridging* dengan codec GSM mampu melayani panggilan bersamaan hingga 40 panggilan, sedangkan untuk codec *G.711* mampu melayani panggilan bersamaan hingga 30 panggilan.



Gambar 11 Hasil pengujian panggilan bersamaan metode *Native Bridging*

3.5 QoS

Pengujian QoS dilakukan terhadap tiga parameter QoS yaitu *delay*, *jitter* dan *packet-loss*. Dari hasil *delay* dan *packet-loss* yang didapat kemudian dihitung nilai MOS dengan menggunakan pendekatan E-Model dan R factor. Pengujian ini bertujuan untuk mengetahui kualitas dari komunikasi suara yang mampu dilayani oleh *server* VoIP pada setiap metode penanganan media yang digunakan.

- **Generic Bridging**

Perhitungan MOS dilakukan dengan menggunakan R factor pada persamaan. Perhitungan R factor ini melibatkan *delay* dan *packet-loss* yang telah didapat sebelumnya. Tabel 1 menunjukkan hasil MOS yang didapat melalui perhitungan. Hasil paling rendah pada saat jumlah panggilan bersamaan sebanyak 5. Pada saat nilai panggilan 5 didapat nilai MOS 3.1 dan 3.3 yang dikategorikan kurang baik. Untuk *codec* GSM nilai MOS maksimum yaitu 3,4 sehingga tidak dapat dikategorikan baik pada penggunaan *codec* GSM.

Tabel 1 MOS untuk Generic Bridging

MOS Generic		
N Panggilan	GSM	G.711
1	3.437225193	4.398802762
2	3.406453021	4.389198911
3	3.375580838	4.372240679
4	3.361570803	4.320876785
5	3.12534723	3.398756789

- **Packet2Packet Bridging**

Hasil MOS pada metode *packet2packet bridging* dapat dilihat pada Tabel 2. MOS pada metode *packet2packet bridging* dikategorikan buruk pada saat jumlah panggilan bersamaan diatas 10 panggilan untuk kedua *codec*.

Tabel 2 MOS untuk P2P Bridging

MOS P2P Bridging		
N Panggilan	GSM	G.711
1	3.444626051	4.405324989
2	3.415045245	4.402117565
3	3.427569636	4.397726713
4	3.425309713	4.392307409
5	3.420274453	4.388181114
6	3.415125748	4.3799178
7	3.398042823	4.372387639
8	3.38729382	4.363473088
9	3.307350909	4.362121927
10	3.117274184	3.548311571
11	1	1.051167186

- ***Native Bridging***

Pada metode native bridging, nilai MOS dikategorikan buruk pada saat jumlah panggilan diatas 35 panggilan untuk *codec* GSM dan diatas 25 panggilan untuk *codec* G.711. Hasil lengkap perhitungan MOS untuk metode native bridging dapat dilihat pada Tabel 3.

Tabel 3 MOS pada Native Bridging

MOS Native Bridging		
N Panggilan	GSM	G.711
1	3.458702052	0.993871246
5	3.457942646	4.406609245
10	3.455893529	4.406404662
15	3.4560924	4.406766787
20	3.410738632	4.35066323
25	3.401141065	4.121954659
30	3.338951394	1.213340667
35	3.190827761	
40	1	

4. KESIMPULAN

Dari hasil penelitian dan pengujian kinerja yang dilakukan pada *wireless access point* Linksys WRT54GL yang dimanfaatkan sebagai VoIP *server* dapat diambil kesimpulan sebagai berikut :

- VoIP *server* pada *wireless access point* mampu melayani komunikasi VoIP dengan baik untuk jumlah panggilan yang sedikit sehingga layak diimplementasikan untuk penggunaan skala kecil.
- VoIP *server* pada *wireless access point* mampu melayani hingga 40 panggilan secara bersamaan dengan metode *Native Bridging*, 5 panggilan bersamaan pada metode *Generic Bridging* dan 11 panggilan bersamaan pada metode *P2P Bridging*. Tetapi dari segi kualitas suara, *server VoIP* hanya mampu melayani 35 panggilan secara bersamaan dengan kualitas yang tidak buruk pada metode *Native bridging* dengan *codec* GSM.
- Semakin banyak panggilan atau trafik yang dilayani oleh *server* atau *access point* akan semakin menurunkan kualitas suara dari panggilan VoIP yang melewati *server* atau *access point* tersebut.
- Penggunaan metode *Native Bridging* akan mengurangi fitur dari *server* VoIP misalnya *transcoding codec* tetapi akan menaikkan panggilan yang mampu dilayani secara bersamaan sebesar 3 hingga 7 kali dibandingkan penggunaan metode lainnya.
- Kinerja VoIP *server* pada *wireless access point* WRT54GL dalam melayani panggilan lebih ditentukan oleh kecepatan CPU yang digunakan pada *server* tersebut dibandingkan dengan kapasitas memori.
- Waktu penyambungan hubungan (Post-Dial Delay) pada VoIP *server* di *wireless access point* dengan OpenWRT lebih cepat dibandingkan dengan VoIP *server* di PC dengan OS Trixbox, hal ini dikarenakan VoIP *server* di OpenWRT dibuat dengan fitur dan modul yang minimalis untuk menghemat memori flash yang digunakan.
- Pada pengujian ini, waktu proses RTP untuk metode *Generic Bridging* dan *P2P Bridging* pada *wireless access point* jauh lebih besar hingga 150 kali dibandingkan waktu proses RTP di PC, sehingga komunikasi VoIP menggunakan *wireless access point* sebagai *server* akan meningkatkan *one way delay* dari paket RTP tersebut dan menurunkan kualitas suara (QoS) komunikasi.

5. SARAN

- Perlu dilakukan penelitian lebih lanjut terhadap pengaruh trafik data lainnya terhadap kualitas suara komunikasi VoIP yang dilayani.
- Perlu dilakukan penelitian lebih lanjut apakah penggunaan beberapa *wireless access point* sebagai VoIP *server* dengan sistem terdistribusi dapat meningkatkan jumlah panggilan yang mampu dilayani atau meningkatkan kualitas dari komunikasi VoIP.

DAFTAR PUSTAKA

- [1] Miller, M.A., 2005. *Implementing the VoIP Network A technical briefing series on VoIP and converged networks*, Longmont: DigiNet Corporation.
- [2] Zascpa, M., Sckalski, P., Sakowicz, B. & Mazur, P., 2009. Implementation of Cost-effective VoIP Network. *Proceeding of the 16th International Conference "Mixed Design of Integrated Circuits and Systems."*Lodz: Department of Microelectronics & Computer Science, Technical University of Lodz, pp. 4–7.
- [3] Tarrant, D. & Hunt, T., 2004. VOIP – Voice over IP Overview. , pp.1–2. Available at: <http://users.ecs.soton.ac.uk/dt302/guides/VOIP-Overview.pdf>.
- [4] Ardilestian, B.D., 2010. Implementasi Server VoIP pada Access Point Linksys WRT54GL Berbasiskan Sistem Operasi GNU/Linux Distro OpenWrt. Tesis. Universitas Gadjah Mada.