Jurnal SimanteC
Vol. 2, No. 2 Desember 2011 ISSN 2088-2130

PENERAPAN OTENTIKASI ENDPOINT BERBASIS PROTOKOL AAA PADA LAYANAN SERVER VOIP H.323 MENGGUNAKAN RADIUS SERVER

Yoga Dwitya Pramudita
Jurusan Teknik Informatika Fakultas Teknik Universitas Trunojoyo
Jl. Raya Telang PO BOX 2 Kamal, Bangkalan, Jawa Timur, 69162
Email : yoga@if.trunojoyo.ac.id

ABSTRAK

Dalam implentasi komunikasi VoIP (*Voice over Internet Protocol, IP Telephony*), salah satu standar yang digunakan adalah standar protokol H.323. Dimana standar ini terdiri dari komponen, protokol, dan prosedur yang menyediakan komunikasi multimedia (*videophone*) melalui jaringan *packet-based* (berbasis paket-paket data). Layanan bisa diakses menggunakan jaringan komputer baik *intranet* maupun *internet* dengan *gatekeeper* sebagai *server* layanan. Pengguna (*user*) menggunakan *endpoint* sebagai klien (penerima layanan komunikasi) untuk mengakses layanan voip yang disediakan oleh *gatekeeper.* Untuk mengantisipasi akses yang tidak diinginkan dari user yang tidak berhak dalam sistem komunikasi voip, maka perlu adanya protokol yang menangani otentikasi dan otorisasi pengguna. Proses ini digunakan untuk memastikan pengguna mempunyai hak akses dan hak guna terhadap *server*, dalam hal ini adalah *gatekeeper*. Salah satu protokol yang mengimplementasikan proses otentikasi adalah RADIUS. Selain bertanggung jawab melakukan otentikasi dan otorisasi, RADIUS juga bertanggung jawab melakukan proses akunting (AAA : *Authentication, Authorization and Accounting*), sehingga RADIUS juga menangani pencatatan *log* komunikasi pengguna layanan ini.

Kata kunci : Voip, H.323, Gatekeeper, Endpoint, RADIUS, AAA.

ABSTRACT

In implementations of VoIP communication (Voice over Internet Protocol, IP Telephony), one of the standards used are H.323 standard protocol. Where the standard is made up of components, protocols, and procedures that provide multimedia communications (videophone) over packet-based networks (based on data packets). Services can be accessed using either a computer network with the intranet or internet as a gatekeeper server service. Users use the endpoint as a client (receiver communication service) to access VoIP services provided by the gatekeeper. To anticipate unwanted access from unauthorized users in a VoIP communication system, it is necessary to handle the authentication protocol and user authorization. This process is used to ensure users have access rights and user rights to the server, in this case is the gatekeeper. One protocol that implements the authentication process is the RADIUS. In addition responsible for authentication and authorization, the RADIUS is also responsible for the accounting (AAA : Authentication, Authorization and Accounting), so the RADIUS also handles the communication log recording.

***Keywords** : Voip, H.323, Gatekeeper, Endpoint, RADIUS, AAA.*

PENDAHULUAN

Perkembangan teknologi jaringan komputer dewasa ini sangat pesat seiring dengan kebutuhan masyarakat akan layanan yang memanfaatkan jaringan komputer. Jaringan *Interconnected-networking* (Internet) merupakan salah satu bentuk nyata dari perkembangan jaringan lokal komputer yang saling terkoneksi satu sama lain pada area geografis berbeda di dunia, sehingga pertukaran data, informasi dan komunikasi akan lebih mudah dan efektif, salah satunya adalah komunikasi suara berbasis IP (VoIP).

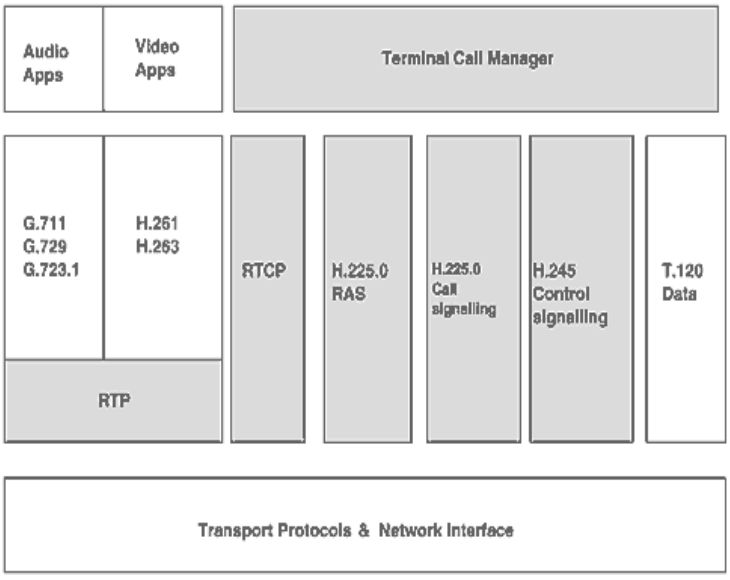
Komunikasi audio dibangun meng-gunakan teknologi *Voice Over Internet Protokol* (VoIP) , yaitu teknologi yang mampu melewatkan trafik suara berbentuk paket melalui jaringan IP (*Internet protocol*) [1]. Salah satu standar yang dipakai oleh VoIP adalah H.323 yang dikembangkan oleh ITU (*International Telecommunication Union*). Standar H.323 terdiri dari komponen, protokol, dan prosedur yang menyediakan komunikasi multimedia melalui jaringan *packet-based* (berbasis paket-paket data). Salah satu protokol yang masuk dalam standar H.323 adalah H.225.0 RAS (*Regristration, Admission, and Status*) digunakan untuk penanganan registrasi *endpoint*.

Dalam tulisan ini pengamanan sumber daya (*VoIP server*) dilakukan dengan melibatkan protokol tambahan yaitu protokol RADIUS (*Remote Authentication Dial In User Service*) [4] yang khusus menangani proses AAA (*authentication, authorization and accounting*) dan sebagai proses tambahan diluar proses registrasi pada gatekeeper.

Dengan pemisahan dari gatekeeper diharapkan beban penanganan proses otentikasi dan akunting bisa di lakukan dengan lebih baik.

Arsitektur H.323

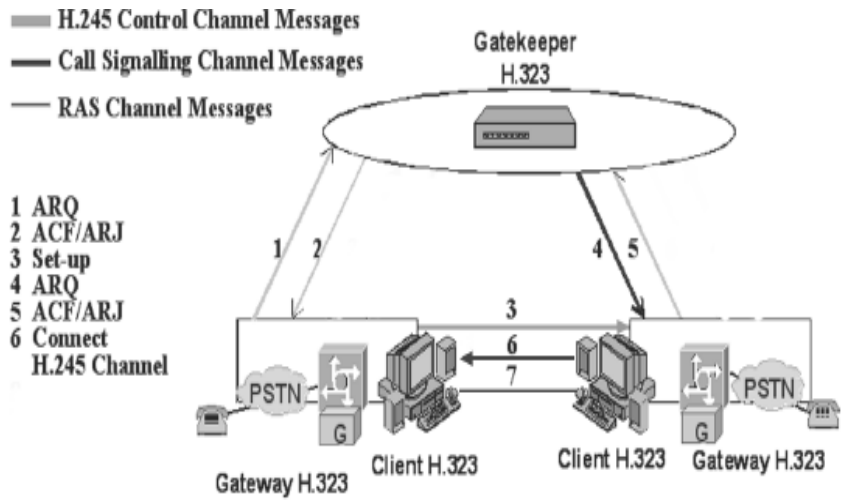
Dalam menyediakan layanan komunikasi, *VoIP server* (*Gatekeeper*) bertanggung jawab untuk mengatur jalannya komunikasi antar *endpoint*, baik itu dari segi user *Registration, Admission and Status* (H.225.0 RAS), C*all signaling* (H.225.0), *Control signaling* (H.245).



Gambar. 1 Arsitektur Protokol H.323

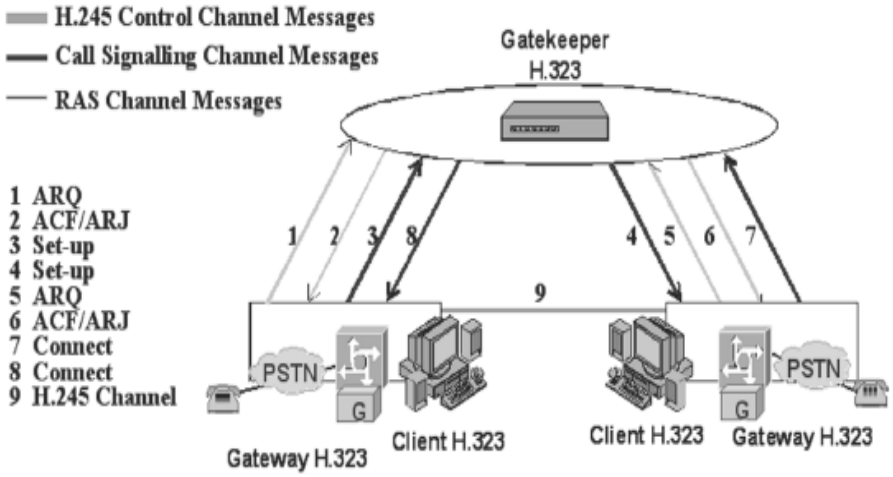
H.323 mempunyai tiga model signaling [1] yaitu :

- *Direct Signalling* (Pensinyalan langsung) Hanya pesan H.225.0 RAS yang dilewatkan melalui *gatekeeper* ketika akan terjadi pertukaran pesan antar dua *endpoint*. Pada gambar dibawah poin 1 dan 2 serta 4 dan 5 menunjukan bahwa proses *admission* saja yang dirouting oleh gatekeeper, sedangkan *call signalling* dan *call channel* langsung dikirimkan antar *endpoint* tanpa di-*routing* melalui *gatekeeper.*.



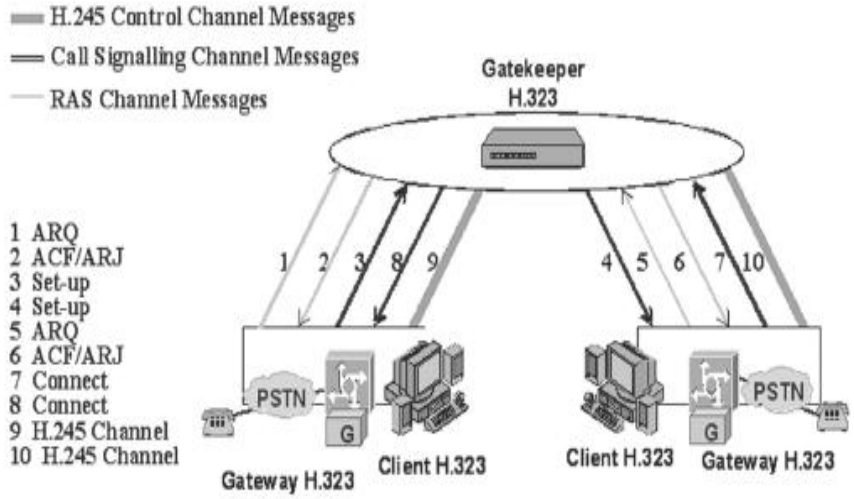
Gambar 2 Pensinyalan panggilan langsung

- *Gatekeeper-Routed Call Signalling* Dengan *signaling* model ini H.225.0 RAS dan *Call Signalling* dilewatkan melalui *Gatekeeper*. Dan gambar dibawah menunjukan bahwa baik itu *admission* dan *call signaling* di-*routing* melalui *gatekeeper*.



Gambar 3 Pensinyalan panggilan melalui *gatekeeper*

- *Gatekeeper-routed* H.245 *control channel Message*, H.225.0 RAS and H.225.0. Baik *call signaling* dan *conference control* dilewatkan melalui *gatekeeper* dan gambar berikut menunjukan bahwa semua sinyal di-*routing* melalui *gatekeepeer*.



Gambar 4 Pensinyalan panggilan dan *data stream* di-*routing* melalui *gatekeeper*

Penanganan registrasi user pada arsitektur H.323 dilayani oleh sub protokol H.225.0 RAS.

Protokol ini digunakan untuk komunikasi antara H.323 *gatekeeper* dan *endpoint*-nya, atau antar *gatekeeper* satu dengan *gatekeeper* lain, selain itu endpoint menggunakan RAS untuk :

- Melakukan proses registrasi *endpoint* terhadap *gatekeeper.*
- Meminta persetujuan *gatekeeper* untuk memakai sistem sumber daya komunikasi baik *codec audio* maupun *video.*
- Mendapatkan alamat *remote* dari *endpoint* yang ditetapkan
- Mengumpulkan informasi terbaru tentang sumber daya yang dipakai setelah proses panggilan berakhir.

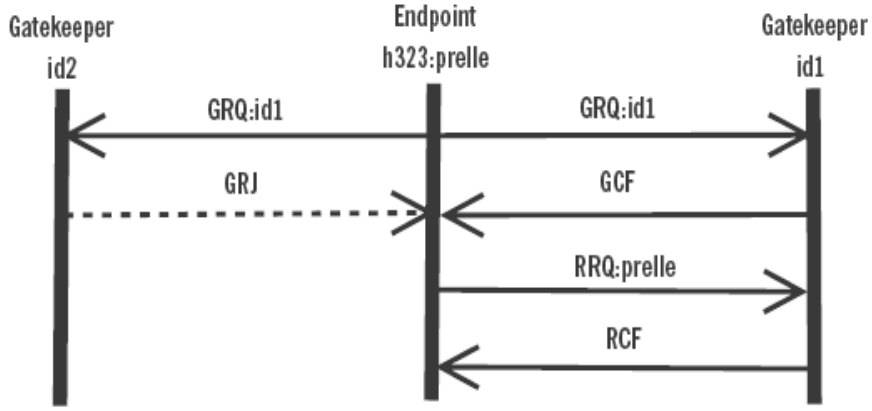
Secara umum RAS melayani mekanisme untuk otentikasi *endpoint* dan *Call Authorization*. *Gatekeeper* menggunakan beberapa model RAS untuk berkomunikasi dengan *endpoint* atau *gatekeeper* lain diantaranya:

- GRQ (*Gatekeeper Request*) yang dijawab dengan GCF (*Gatekeeper Confirm*) atau GRJ (*Gatekeeper Reject*).
- RRQ (*Registration Request*) yang dijawab dengan RCF (*Registration Confirm*) atau RRJ (*Request Reject*).
- ARQ (*Admission Request*) yang dijawab dengan ACF (*Admission Confirm*) atau ARJ (*Admission Reject*).

Sebuah H.323 *endpoint* harus melakukan registrasi terhadap *gatekeeper* untuk bisa melakukan panggilan terhadap *endpoint* lain yang juga terdaftar pada *gatekeeper* yang sama [1]. Ada dua kemungkinan untuk sebuah *endpoint* menemukan dan kemudian melakukan registrasi terhadap *gatekeeper* seperti pada gambar 2, diantaranya:

- *Multicast discovery* dimana *endpoint* mengirim sebuah sinyal *gatekeeper request* (GRQ) ke alamat *multicast* (224.0.1.41 default alamat *multicast gatekeeper*) dengan *port* (1718 *default port* yang digunakan). *Gatekeeper* penerima kemudian bertanggung jawab memberikan konfirmasi terhadap *endpoint* (GCF) apakah diterima atau diabaikan.
- *Configuration* dimana *endpoint* dikonfigurasi secara manual untuk mengetahui alamat dari *gatekeeper*. Pada kondisi ini GRQ tidak dibutuhkan lagi sehingga proses registrasi hanya melibatkan sinyal *registration request* (RRQ) dan *registration confirm* (RCF) apakah *endpoint* diterima atau diabaikan,

tetapi ada juga beberapa produk *endpoint* yang membutuhkan protokol GRQ dan GCF untuk melakukan registrasi dengan menggunakan alamat *unicast* (satu pengirim dan satu penerima).



Gambar 5 Proses Registrasi antara *endpoint* dan *gatekeeper* secara *discovery*

RADIUS Authentication

Remote Authentication Dial In User Service (RADIUS) [4] adalah sebuah protokol yang menangani AAA (*authentication, authorization and accounting*) untuk *network acces* atau *IP mobility*, dan digunakan melayani proses lokal atau *roaming*. RADIUS *server* berfungsi untuk menangani otentikasi, admisi dan akunting user, sebelum user bisa menggunakan layanan *network* yang disediakan, user akan diminta untuk memasukan informasi yang bisa dipercaya (seperti *username* dan *password*) yang kemudian diteruskan ke *Network Access Server* (NAS, dalam hal ini adalah *gatekeeper* atau MCU) untuk kemudian diproses apakah *user* tersebut layak mendapatkan akses layanan atau tidak. RADIUS *server* menggunakan skema PAP, CHAP dan EAP untuk melakukan otentikasi informasi dari user, jika diterima maka *server* akan memberitahu NAS bahwa *user* mempunyai hak akses dalam jaringan NAS.

RADIUS juga mengijinkan otentikasi terhadap *server* dengan parameter tambahan untuk diteruskan terhadap NAS, diantaranya seperti :

- IP *address* yang secara spesifik harus digunakan oleh *user*
- Alamat *pool* IP yang seharusnya digunakan oleh *user*
- Banyak *user* maksimum yang dilayani
- Daftar akses, antrian priorotas yang digunakan untuk mengatur akses *user*
- Parameter L2TP
- *Etc*.

Protokol RADIUS tidak mengirimkan informasi *password* dengan metode *cleartext* (meskipun skema PAP yang digunakan) antara RADIUS *server* dan NAS, tetapi secara tersembunyi ada proses lain seperti penggunaan enkripsi MD5 dan informasi tambahan berupa kata kunci yang disebut *shared secret*.

RADIUS juga bisa diguakan untuk proses akunting, sehingga NAS bisa menggunakan RADIUS untuk proses akunting paket-paket yang dikirimkan, seperti :

- *Session start user*
- *Session stop user*
- Total paket yang dikirim selama sesi
- *Volume* data yang dikirim selama sesi
- Alasan kenapa sesi berakhir

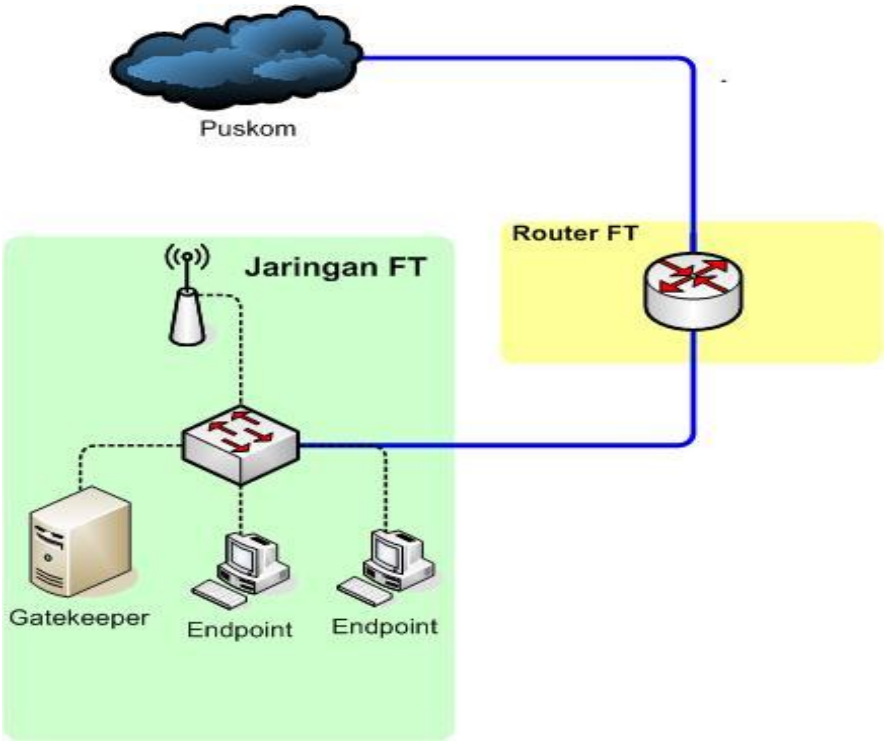
Tujuan akhir dari informasi yang didapatkan melalui proses akunting adalah *log*

komunikasi yang dilakukan oleh pengguna layanan.

Implementasi

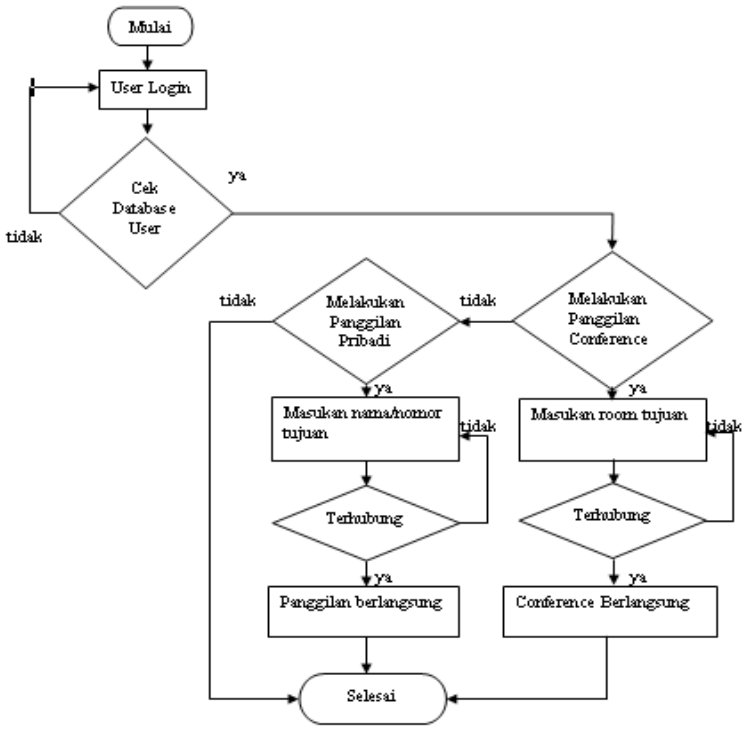
Dari uraian diatas bisa diketahu bagaimana *gatekeeper* dan *endpoint* bekerja dalam membangun dan menangani komunikasi, sedangkan mekanisme AAA ditangani oleh RADIUS.

Penerapan pada jaringan lokal [3] melibatkan beberapa perangkat berupa *gatekeeper* [2] (dilengkapi dengan layanan H.323 *gatetekeeper* gnugk, *web server*, DNS *server*, RADIUS *server*, *database server*), MCU *server* dan *endpoint* (komputer dengan menggunakan *softphone* yang mendukung protokol H.323).



Gambar 6 Topologi Jaringan Lokal

Sebelum *user* bisa melakukan panggilan, diharuskan dulu melakukan registrasi dengan memasukan *username*, *password* (nomor VoIP yang nantinya digunakan untuk nomor panggilan akan secara otomatis diaktifkan pada *endpoint* sesuai dengan *user database*) pada *softphone* yang digunakan.



Gambar 7 *Flowchart* proses panggilan

Proses Registrasi dan Panggilan

Proses yang terjadi pada saat user melakukan registrasi terhadap *gatekeeper* melibatkan proses pada RADIUS *server* (freeradius sebagai aplikasi RADIUS *server*). Sehingga yang menentukan layak atau tidaknya *user* untuk bisa mengakses NAS (*Network Access Server*) (*gatekeeper*) adalah *server* RADIUS dengan menggunakan skema SQL dan referensi data yang sudah tersimpan di *database server* (postgresql).

User mempunyai dua opsi panggilan, *private* dan *conference* [3]. Untuk panggilan private user dapat langsung men-*dial username* lain atau nomor VoIP *user* lain yang sudah melakukan proses registrasi pada *gatekeeper*, sedangkan untuk *conference* harus men-*dial room* yang akan digunakan untuk konferensi kemudian menunggu *user* lain yang akan terlibat dalam proses konferensi untuk men-*dial room* tujuan yang sama.

Proses Otentikasi

Sesi otentikasi atau yang lebih dikenal dengan protokol AAA ditangani langsung oleh aplikasi freeradius sebagai implementasi dari RADIUS *server*. Freeradius itu sendiri adalah paket software RADIUS berbasis *modular* dan berkinerja tinggi termasuk *server*, *client*, pustaka pengembangan dan utilitas tambahan untuk implementasi RADIUS [5]. Untuk memper-mudah proses pengelolaan otentikasi *user*, freeradius melibatkan *database server* dalam membangun skema otentikasi.

Parameter yang dihasilkan dari freeradius yang nantinya diproses adalah:

- *User-Name* = h323id
- *CHAP-Password* = *password* h323
- *CHAP-Challenge* = *shared secret* yang telah dikonfigurasi secara manual
- *NAS-IP-Address* = alamat IP dari *server* (*Network Address Server*)
- *NAS-Identifier* = nama atau identitas dari *server* (diset pada client.conf menjadi ‘LabsiGK’)
- NAS-*Port-Type* = tipe *port* yang digunakan (selalu *virtual* karena penggunaan *port* diatur secara otomatis

oleh freeradius)

- *Service-Type* = tipe dari layanan yang sedang berlangsung (*Login-User*, *Call-Check* tergantung dari proses pengecekan layanan yang sedang berlangsung)
- *Framed*-IP-*Address* = alamat IP *endpoint* yang sedang melakukan proses AAA
- *Cisco-AVPair* = string yang digunakan oleh freeradius untuk mengirim data sinyal atau parameter protokol H.323 yang dibutuhkan seperti *h323id*, *alias E.164*, *call id* H323 dan parameter pendukung lainnya.
- *Calling-Station-Id = alias E.164 endpoint* tujuan
- *Called-Station-Id* = alias E.164 *endpoint* pemanggil
- *h323-conf-id* = digunakan sebagai id panggilan H.323
- *h323-gw-id* = id dari servis gnugk atau *gatekeeper*
- *h323-call-origin* = panggilan asal (berisi string *proxy* karena semua *call signal* dilewatkan melalui *gatekeeper*)
- *h323-call-type* = tipe panggilan (berisi string VoIP karena panggilan yang dilewatkan adalah panggilan VoIP)
- *h323-setup-time* = waktu pendirian kanal komunikasi
- *h323-remote-address* = alamat panggilan *remote* (jika panggilan konferensi dilakukan)
- *Acct-Delay-Time* = delay antara *setup time* dengan *accounting start time*
- *h323-return-code* = kode Q.931 yang digunakan untuk merepresentasikan penyebab pemutusan panggilan (jika bernilai 0 maka pemutusan panggilan tidak akan dilakukan)

UJICOBA & PEMBAHASAN

Sekenario ujicoba yang akan dilaksanakan yaitu sesi panggilan pribadi. Sebelum sesi panggilan tersebut dijalankan maka pastikan dilakukan pengecekan pada RADIUS *server*.

```
[root@server ~]# radiusd -x

Using deprecated naslist file.
Support for this will go away soon.
Module: Loaded expr
Module: Instantiated expr (expr)
Module: Loaded PAP
Module: Instantiated pap (pap)
Module: Loaded CHAP
Module: Instantiated chap (chap)
Module: Loaded SQL
...
...

rlm_sql (sql): Driver rlm_sql_postgresql
(module rlm_sql_postgresql) loaded and
linked
rlm_sql (sql): Attempting to
connect to gkradius@10.1.1.14:/voipdb
rlm_sql (sql): starting 0
...
...

Module: Instantiated sql (sql)
Module: Loaded Acct-Unique-Session-Id
Module: Instantiated acct_unique
(acct_unique)
Module: Instantiated detail (reply_log)
Initializing the thread pool...
Listening on authentication *:1812
Listening on accounting *:1813
Ready to process requests.
```

Gambar 8 proses *running* freeradius

Proses diatas menunjukan bahwa RADIUS *server* menggunakan database postgresql [3,6] dengan *database*-nya voipdb dan siap melakukan proses otentikasi dan akunting. MCU disini difungsikan sebagai *endpoint* untuk bisa melakukan proses registrasi pada *gatekeeper*.

```
rad_recv: Accounting-Request packet from
host 10.1.1.14:41612, id=92, length=47
        Acct-Status-Type = Accounting-On
        NAS-IP-Address = 10.1.1.14
        NAS-Identifier = "LabsiGK"
        NAS-Port-Type = Virtual
rlm_sql (sql): received Acct On/Off packet
rlm_sql (sql): Reserving sql socket id: 9
rlm_sql (sql): Released sql socket id: 9
Sending Accounting-Response of
id 92 to 10.1.1.14 port 41612
```

Gambar 9 proses deteksi MCU

Proses diatas menandakan bahwa MCU telah aktif dan RADIUS siap melakukan akunting jika terjadi komunikasi antar *endpoint* dengan MCU.

pada saat proses registrasi baik MCU maupun *endpoint*, maka RADIUS melakukan proses otentikasi dan dilanjutkan dengan persiapan melakukan proses akunting. Pada registrasi MCU proses otentikasi tidak melalui RADIUS, tetapi langsung pada sisi *gatekeeper* dengan metode otentikasi IP *address* pada seksi *FileIPAuth*. Sedangkan pada registrasi *endpoint*, RADIUS memegang kendali otentikasi dan proses akunting (jika terjadi panggilan pada *endpoint)*.

Gambar proses berikut menunjukan bahwa RADIUS telah melakukan proses otentikasi terhadap *endpoint* dengan *username* "user8" dan *password* "user8" dan ketika dua data tersebut telah cocok dengan *user database* maka dilakukan replay data berupa RCF (*Request Confirm*) dengan menambahkan data alias yang sesuai dengan *database* yaitu "user8, 304100321" dimana alias yang berada setelah koma adalah no VoIP *user*, *h323-return-code* = 0 menunjukan bahwa RRQ diterima dengan protokol Q.931 dengan kode = 0.

```
rad_recv: Access-Request packet from
host 10.1.1.14:7883, id=133, length=136
        User-Name = "user8"
        CHAP-Password =
0xc96adcb2981bc413dc9b48fdd5f3c2ea60
        CHAP-Challenge = 0x46cd1216
        NAS-IP-Address = 10.1.1.14
        NAS-Identifier = "LabsiGK"
        NAS-Port-Type = Virtual
        Service-Type = Login-User
        Framed-IP-Address = 10.1.1.16
        Cisco-AVPair = "h323-ivr-
out=terminal-alias:user8,admin ok;"

.....
.....
.....
.....

rlm_sql (sql): Released sql socket id: 8
  rlm_chap: login attempt by "user8" with
CHAP password
  rlm_chap: Using clear text password user8
for user user8 authentication.
  rlm_chap: chap user user8 authenticated
succesfully
Login OK: [user8/<CHAP-Password>]
(from client LabsiGK port 0)
Sending
Access-Accept of id 133 to 10.1.1.14 port
7883
        h323-return-code =
"h323-return-code=0"
        Cisco-AVPair =
"h323-ivr-in=terminal-
alias:user8,304100321;"
```

Gambar 10 proses registrasi *endpoint*

Panggilan *point to point* melibatkan dua *user* dan *gatekeeper*, kedua *user* tersebut harus sudah teregistrasi dan tidak dalam kondisi sedang melakukan panggilan. Apabila salah satu *enpoint* dalm kondisi sedang dalam panggilan dengan *endpoint* lain maka *gatekeeper* akan mengirimkan pesan Q.931 pada saat *stop query* dilakukan dengan nomor kode 11 yang berarti bahwa *endpoint* tujuan sedang dalam proses panggilan.

Ketika proses panggilan terjadi *endpoint* meminta RADIUS melakukan pengecekan sebelum proses panggilan bisa dilaksanakan (ARQ), kemudian user tujuan akan melakukan *call check* (ACF) apakah user pemanggil

memang sudah diijinkan melakukan panggilan, kemudian proses *call setup* dan proses panggilan (H.245) bisa segera dilaksanakan.

Proses berikut menunjukan bahwa *user* dengan nama "yoga" dengan nomer panggilan "085645208049" akan melakukan panggilan ke nomer " 304100315", setelah RADIUS melakukan cek dan ternyata kedua *user* tersebut ada dan sudah terdaftar di *gatekeeper* maka proses panggilan bisa dimulai.

```
rad_recv: Access-Request packet from
host
10.1.1.14:9013, id=209, length=252
        User-Name = "yoga"
        CHAP-Password =
0xb9019670aff
       95a0b3f4392d90b902e2a25
        CHAP-Challenge = 0x4699ce4f
        NAS-IP-Address = 10.1.1.14
        NAS-Identifier = "LabsiGK"
        NAS-Port-Type = Virtual
        Service-Type = Login-User
        Framed-IP-Address =
10.1.1.15
        Calling-Station-Id =
"085645208049"
        Called-Station-Id =
"304100315"
        h323-conf-id = "h323-conf-
id=5B031E61 2AF91810 89400019
2142C456"
        h323-call-origin = "h323-
call-origin=originate"
        h323-call-type = "h323-call-
type=VoIP"
        h323-gw-id = "h323-gw-
id=LabsiGK"
  rlm_chap: Setting 'Auth-Type :=
CHAP'

……
……
……
……

rlm_chap: login attempt by "yoga"
with CHAP password
  rlm_chap: Using clear text
password
             yoga for user yoga
authentication.
  rlm_chap: chap user yoga
authenticated succesfully
```

Gambar 11 Proses *admission* (ARQ)

User tujuan juga melakukan proses *request call check* yang tujuannya untuk melakukan pengecekan apakah *user* pemanggil sudah terdaftar pada satu *gatekeeper* yang sama.

```
rad_recv: Access-Request packet from
host 10.1.1.14:9013, id=210,
length=250
        User-Name = "user2"
        CHAP-Password =
0xe3fd8be0e8215dafcd8cbd8277d3612b72
        CHAP-Challenge = 0x4699ce6b
        NAS-IP-Address = 10.1.1.14
        NAS-Identifier = "LabsiGK"
        NAS-Port-Type = Virtual
        Service-Type = Call-Check
        Framed-IP-Address = 10.1.1.107
        Calling-Station-Id =
"085645208049"
        Called-Station-Id =
"304100315"
        h323-conf-id = "h323-conf-
id=5B031E61 2AF91810 89400019
2142C456"
        h323-call-origin = "h323-call-
origin=answer"
        h323-call-type = "h323-call-
type=VoIP"
        h323-gw-id = "h323-gw-
id=LabsiGK"
  rlm_chap: Setting 'Auth-Type :=
CHAP'
……
……
……
……
rlm_chap: login attempt by "user2"
with CHAP password
  rlm_chap: Using clear text password
user2 for user user2 authentication.
  rlm_chap: chap user user2
authenticated succesfully
Login OK: [user2/<CHAP-Password>]
(from client LabsiGK port 0 cli
085645208049)
Sending Access-Accept of id 210 to
10.1.1.14 port 9013
        h323-return-code = "h323-
return-code=0"
```

Gambar 12 Proses *call check* RADIUS (ACF)

Segera setelah proses *call check* selesai dilakukan maka proses start akunting dilakukan oleh RADIUS *server*. Proses ini sebagai langkah awal memasukan data akunting kedalam tabel voipcall pada *database* voipdb.

```
Acct-Status-Type = Start
        NAS-IP-Address = 10.1.1.14
        NAS-Identifier = "LabsiGK"
        NAS-Port-Type = Virtual
        Service-Type = Login-User
        Acct-Session-Id =
"4699c68e00000001"
        User-Name = "yoga"
        Framed-IP-Address = 10.1.1.15
        Calling-Station-Id =
"085645208049"
        Called-Station-Id =
"304100315"
        h323-conf-id = "h323-conf-
id=5B031E61 2AF91810 89400019
2142C456"
        h323-gw-id = "h323-gw-
id=LabsiGK"
        h323-call-origin = "h323-
call-origin=proxy"
        h323-call-type = "h323-call-
type=VoIP"
        h323-setup-time = "h323-
setup-time=14:38:20.000 WIT Sun Jul
15 2007"
        h323-remote-address = "h323-
remote-address=10.1.1.107"
        Cisco-AVPair = "h323-ivr-
out=h323-call-id:5B031E61 2AF91810
893F0019 2142C456"
        Acct-Delay-Time = 0
……
……
……
……

rlm_sql_postgresql: affected rows = 1
rlm_sql (sql): Released sql socket
id: 6
Sending Accounting-Response of id 234
to 10.1.1.14 port 22013
```

Gambar 13 Proses *start accounting*

Proses akunting dimulai dengan penambahan baris baru dalam *database*, *rlm_sql_postgresql*: *affected rows = 1* menandakan bahwa baris dari tabel voipcall sebagai penyimpan data akunting sudah diisi dengan data awal panggilan.

```
Acct-Status-Type = Stop
        NAS-IP-Address = 10.1.1.14
        NAS-Identifier = "LabsiGK"
        NAS-Port-Type = Virtual
        Service-Type = Login-User
        Acct-Session-Id =
"4699c68e00000001"
        User-Name = "yoga"
        Framed-IP-Address = 10.1.1.15
        Acct-Session-Time = 33
        Calling-Station-Id =
"085645208049"
        Called-Station-Id =
"304100315"
        h323-conf-id = "h323-conf-
id=5B031E61 2AF91810 89400019
2142C456"
        h323-gw-id = "h323-gw-
id=LabsiGK"
        h323-call-origin = "h323-call-
origin=proxy"
        h323-call-type = "h323-call-
type=VoIP"
        h323-setup-time = "h323-setup-
time=14:38:20.000 WIT Sun Jul 15 2007"
        h323-connect-time = "h323-
connect-time=14:38:22.000 WIT Sun Jul
15 2007"
        h323-disconnect-time = "h323-
disconnect-time=14:38:55.000 WIT Sun
Jul 15 2007"
        h323-disconnect-cause = "h323-
disconnect-cause=10"
        h323-remote-address = "h323-
remote-address=10.1.1.107"
        Cisco-AVPair = "h323-ivr-
out=h323-call-id:5B031E61 2AF91810
893F0019 2142C456"
        Acct-Delay-Time = 0
```

Gambar 14 Proses *stop accounting*

Saat proses panggilan selesai dilakukan maka proses akunting dilakukan kembali dengan melakukan *update* terhadap tabel voipcall yang sebelumnya sudah terisi data pada saat proses *start accounting*.

KESIMPULAN

Berdasarkan ujicoba yang telah dilakukan pada sistem komunikasi *audio* dan *video conference* sebagaimana yang telah dijelaskan pada bab sebelumnya, maka dapat diambil kesimpulan bahwa proses otentikasi, otorisasi, dan akunting terhadap layanan *voip call* dengan menggunakan protokol H.323 bisa ditangani oleh

layanan pihak ketiga yang disediakan oleh freeradius sebagai *server* layanan yang mengimplementasikan protokol RADIUS.

Untuk pengembangan kedepan implementasi protokol otentikasi, otorisasi dan akunting bisa menggunakan skema lain, misal menggunakan skema *active directory* atau skema otentikasi lain yang nantinya bisa disesuai dengan implementasi di lapangan.

DAFTAR PUSTAKA

[1] Trans-Eoupe Research and Education Networking Association, march, 2004, *TERENA Report Ip Telephony Cookbook*

[2] Willamowius. Jan, 2007, *GNU Gatekeeper*, www.gnugk.org

[3] D.P. Yoga, September, 2007, Perancangan Dan Implementasi *Audio* Dan *Video Conference* Menggunakan Protokol H.323 Studi Kasus Universitas Trunojoyo, Tugas Akhir Teknik Informatika Universitas Trunojoyo

[4] Rigney C., Rubens A., Merit, Simpson W., Daydreamer,. Willens S, Livingston, *Remote Authentication Dial In User Service* (RADIUS), April, 1997, IETF Networking Group

[5] Cudbard-Bell Arran, November, 2011, *Wiki FreeRADIUS Documentation for the world's most popular RADIUS Server*, http://wiki.freeradius.org/

[6] The PostgreSQL Global Development Group, 1996-2003, *PostgreSQL 7.4.2 Documentation*, www.postgresql.org