OPTIMALISASI RADIUS SERVER SEBAGAI SISTEM OTENTIKASI DAN OTORISASI UNTUK PROSES LOGIN MULTI APLIKASI WEB BERBASIS PHP

Herman Yuliansyah
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan,
Jalan. Prof. Dr. Soepomo, Janturan, Umbul Harjo Yogyakarta 55164
e-mail: herman_syah@mti.ugm.ac.id

Abstrak

Protokol RADIUS merupakan suatu aturan yang mendukung berbagai mekanisme metode untuk mengirimkan data pengguna yang sensitif dari dan ke server otentikasi. Dalam konteks penganggulangan bencana memungkinkan mengembangkan banyak aplikasi sistem informasi dan teknologi informasi. Oleh karena pengembangan itu tidak di rancang untuk saling terintegrasi, maka memungkinkan terjadi duplikasi informasi pengguna sehingga dengan optimalisasi radius server untuk proses login multi aplikasi web berbasis PHP menjadi suatu solusi alternalif mempermudah pengelolaan pengguna pada banyak aplikasi yang terpisah.

Penelitian ini bertujuan mengimplementasikan sistem otentikasi dan otorisasi untuk proses login multi aplikasi web berbasis PHP dengan mengoptimalkan penggunaan dari radius server. Metodologi yang digunakan adalah studi pustaka, perancangan sistem yang meliputi perancangan arsitektural dan desain topologi jaringan, serta implementasi sistem. Sistem yang di buat meliputi 4 bagian, yaitu bagian yang pertama adalah konfigurasi radius server menggunakan aplikasi FreeRADIUS. Bagian yang kedua adalah konfigurasi web server sehingga dapat berkomunikasi dengan radius server. Bagian yang ketiga adalah pengembangan dan implementasi aplikasi pengelolaan pengguna berbasis web dengan bahasa pemrograman PHP dan basis data MySQL. Bagian yang keempat adalah pengujian sistem otorisasi dan otentikasi pengguna pada beberapa aplikasi yang berbeda.

Hasil dari penelitian ini adalah pengguna pada beberapa aplikasi web berbasis PHP dapat diintegrasikan pengelolaannya dengan membangun sistem otentikasi dan otorisasi dengan radius server menggunakan aplikasi FreeRADIUS. Proses optimalisasi radius server sebagai sistem otentikasi dan otorisasi ini dapat membuat pengguna hanya akan memiliki satu akun tunggal untuk beberapa aplikasi yang berbeda.

Kata kunci*: *Radius Server, Sistem Otentikasi dan Otorisasi, Pengelolaan Pengguna

1. PENDAHULUAN

Sistem informasi dan teknologi informasi (SI/TI) memiliki peran yang tidak kecil dalam membantu mempermudah proses komunikasi dan penyajian informasi untuk penanggulangan bencana alam. Dalam hal penganggulangan bencana tersebut memungkinkan melibatkan banyak pihak, sehingga memunculkan kemungkinan pula pengembangan SI/TI pada banyak instansi sesuai dengan kebutuhan dan jenis SI/TI dari pihak-pihak tersebut. Apabila pengelolaan pengguna aplikasi-aplikasi SI/TI tersebut tidak di rancang untuk saling terintegrasi satu dengan lainnya atau pengelolaan penggunanya masih dilakukan pada masing-masing aplikasi secara mandiri maka memungkinkan terjadinya duplikasi pengguna dan informasi antar aplikasi untuk pengguna yang membutuhkan menggunakan aplikasi pada beberapa instansi yang berbeda tersebut.

Hal ini juga dijelaskan oleh Roderick W. Smith dalam bukunya tentang otentikasi pengguna. Menurut Smith[1], salah satu masalah dengan jaringan komputer yang besar adalah bahwa otentikasi pengguna dapat menjadi masalah. Jika jaringan memiliki 100 komputer, dan jika beberapa pengguna perlu untuk dapat menggunakan komputer ini, mengelola account untuk para pengguna tesebut dapat menjadi pekerjaan yang membutuhkan waktu khusus.

RADIUS server kebanyakan digunakan sebagai sistem otentikasi, otorisasi dan akunting pada beberapa perangkat jaringan, diantaranya seperti wireless hotsport baik di universitas, fasilitas umum maupun diperkantoran. Oleh karena itu, melihat kebutuhan dari pengelolaan pengguna sebagai suatu integrasi beberapa aplikasi yang berbeda maka dengan optimalisasi radius server sebagai sistem otentikasi dan otorisasi untuk proses login multi aplikasi web berbasis PHP dapat menjadi suatu solusi alternalif mempermudah pengelolaan pengguna pada banyak aplikasi web berbasis PHP yang terpisah.

Tujuan dari penelitian ini adalah untuk mengimplementasikan sistem otentikasi dan otorisasi untuk proses login multi aplikasi web berbasis PHP dengan mengoptimalkan penggunaan dari radius server. Ruang lingkup dari sistem ini adalah aplikasi yang akan diintegrasikan ke sistem otentikasi dan otorisasi akan saling terbuka dan mengijinkan untuk saling berkomunikasi.

2. TINJAUAN PUSTAKA

a. Deskripsi RADIUS

Menurut Hassel[2], *Remote Access dial up user service* (RADIUS), awalnya dikembangkan oleh Livingston Enterprises, adalah sebuah protokol *access-control* yang memverifikasi dan mengotentikasi pengguna umumnya berdasarkan pada metode *challenge/response*. Sementara RADIUS memiliki tempat yang menonjol di antara penyedia layanan Internet, hal itu juga termasuk dalam lingkungan di mana otentikasi terpusat, pengatur otorisasi, dan rinci *accounting* pengguna baik yang diperlukan atau diinginkan.
Menurut Rigney[3], beberapa fitur kunci dari RADIUS adalah:

1. **Model *Client*/*Server*.**

   Sebuah *Network Access Server* (NAS) beroperasi sebagai RADIUS klien. Klien bertanggung jawab untuk menyampaikan informasi pengguna ke RADIUS server yang ditunjuk, dan kemudian bekerja untuk mengembalikan respon.

   RADIUS server bertanggung jawab untuk menerima permintaan koneksi pengguna, melakukan otentikasi pengguna, dan kemudian mengembalikan semua informasi konfigurasi yang diperlukan bagi klien untuk memberikan layanan kepada pengguna.

2. ***Network Security***

   Transaksi antara klien dan RADIUS server dikonfirmasi melalui penggunaan *shared secret*, yang tidak pernah dikirim melalui jaringan. Selain itu, setiap pengguna akan mengirimkan *password* yang telah dienkripsi antara klien dan RADIUS server, untuk menghilangkan kemungkinan bahwa seseorang mengintip di satu jaringan yang tidak aman dapat dengan menentukan *password* penggunanya.

3. ***Flexible Authentification Mechanism***

   RADIUS server dapat mendukung berbagai metode untuk otentikasi pengguna. Ketika disediakan dengan *username* dan *password* asli yang diberikan oleh pengguna, dapat mendukung PPP PAP atau CHAP, UNIX login, dan mekanisme otentikasi lainnya.

4. ***Extensible Protocol***

   Semua transaksi yang terdiri dari panjang variabel *Attribute-Length-Value 3-tuples*. Nilai atribut baru dapat ditambahkan tanpa mengganggu implementasi protokol yang ada.

b. Metode Otentikasi

RADIUS mendukung berbagai mekanisme protokol yang berbeda untuk mengirimkan data pengguna tertentu sensitif dari dan ke server otentikasi. Dua metode yang paling umum adalah *Password Authentication Protocol* (PAP) dan *Challenge-Handshake Authentication Protocol* (CHAP). RADIUS juga memungkinkan atribut lainnya dan metode yang dikembangkan oleh vendor, termasuk dukungan untuk fitur-fitur khusus untuk Windows NT, Windows 2000, dan sistem operasi jaringan lainnya yang populer dan layanan direktori[2]. Bagian berikut ini mengeksplorasi dua metode yang paling umum secara lebih rinci.

1. ***Password Authentication Protocol* (PAP)**

   Atribut *User-Password* adalah sinyal paket meminta ke RADIUS server di mana protokol PAP akan digunakan untuk transaksi tersebut. Sangat penting untuk di catat bahwa hanya pada kolom yang wajib dalam hal ini adalah kolom *User-Password*. Kolom *User-Name* tidak harus dimasukkan dalam paket *request*, dan sangat mungkin bahwa server RADIUS sepanjang rantai proxy akan mengubah nilai dalam kolom *User-Name*.

   Algoritma yang digunakan untuk menyembunyikan *user password* asli disusun oleh banyak elemen. Pertama, klien mendeteksi *identifier* dan *shared secret* untuk *original request* dan mendaftarkannya ke sebuah urutan MD5 hasing. *Pasword* asli dari klien diletakkan melalui proses XOR dan hasil yang berasal dari kedua urutan ini kemudian dimasukkan ke dalam kolom *User-Password*. RADIUS server menerima kemudian membalikkan prosedur untuk menentukan apakah akan mengotorisasi koneksi. Sifat dasar dari mekanisme *password-hidding* mencegah pengguna untuk menentukan jika waktu otentikasi gagal, kegagalan tersebut disebabkan oleh sandi yang salah atau *secret* yang tidak valid.

2. ***Challenge-Handshake Authentication Protocol* (CHAP)**

   CHAP didasarkan pada premis bahwa *password* tidak harus di kirim dalam paket di dalam jaringan. CHAP mengenkripsi secara dinamis meminta *user id* dan *password*. Klien kemudian menuju ke prosedur *logon* yang telah mendapat kunci dari peralatan klien RADIUS yang panjangnya minimal 16 oktet. Klien melakukan *hash* kunci dan mengirimkan kembali ID CHAP, respons CHAP, dan *username* ke klien RADIUS. Setelah menerima semua hal di atas, menempatkan kolom CHAP ID ke tempat-tempat yang sesuai pada atribut *CHAP-Password* dan kemudian mengirimkan respon. Nilai yang diperoleh awalnya ditempatkan di atribut *CHAP-Challenge* atau di *authenticator* sehingga server dapat dengan mudah mengakses nilai dalam rangka untuk otentikasi pengguna.

   Untuk mengotentikasi pengguna, RADIUS server menggunakan nilai *CHAP-Challenge*, ID CHAP, dan *password* pada rekaman pengguna tertentu dan menyerahkan kepada algoritma MD5 hashing lainnya. Hasil dari algoritma ini harus identik dengan nilai ditemukan pada atribut *CHAP-Password*. Jika tidak, server harus menolak permintaan tersebut, sebaliknya permintaan tersebut disetujui.

3. METODOLOGI PENELITIAN

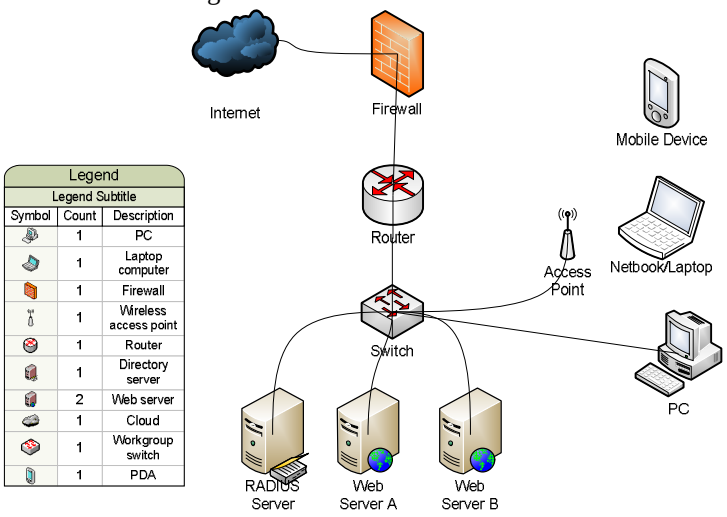
Perangkat keras yang digunakan dalam penelitian ini adalah PC server sebagai web server dan RADIUS server dengan spesifikasi Pentium 4 ke atas dan 1 GB DDR RAM. Perangkat lunak yang digunakan dalam penelitian ini adalah linux ubuntu, paket Apache, paket MySQL, paket PHP dan paket FreeRADIUS.
Jalan penelitian memberikan uraian mengenai langkah-langkah yang di tempuh dalam penelitian ini, yaitu:

a. Investigasi dan Definisi Permasalahan

Pada tahapan ini bertujuan untuk melakukan identifikasi kenyataan, harapan dan permasalahan yang ada sehingga menjadi latar belakang integrasi manajemen pengguna sistem informasi dan aplikasi. Beberapa subtahapan yang dilalui pada tahapan ini adalah observasi penelitian, yaitu tahapan untuk mengumpulkan data melalui pengamatan langsung di lapangan. Data yang diambil adalah proses/prosedur login aplikasi PHP berbasis web. Subtahapan berikutnya adalah Studi Pustaka, yaitu Tahapan ini untuk mengumpulkan sumber informasi dan referensi yang mendukung penelitian ini. Sumber informasi/referensi tersebut berupa referensi mengenai protokol RADIUS, pemrograman PHP dan sistem otentikasi dan otorisasi RADIUS server.

b. Perancangan Arsitektural Sistem

Pada tahapan ini menentukan arsitektur sistem. Model arsitektur sistem yang akan digunakan adalah model klien-server. Berikut adalah rancangan arsitektural sistem.



Gambar 1 Topologi Jaringan

Pada gambar 1 terlihat bahwa sistem terdiri dari firewall, internet gateway sebagai pintu akses ke jaringan publik, dan kumpulan server yang akan di bagun meliputi RADIUS server dan Web Server-Web Server yang akan terkoneksi dengan RADIUS server, serta akses klien dengan menggunakan media kabel dan media wi-fi yang menggambarkan penyederhanaan dari jaringan akses pada kenyataannya.

c. Implementasi Unit dan Pengujian

Pada tahapan ini dilakukan implementasi dan pengujicobaan hasil rancangan dalam bentuk laboratorium kecil yang merefleksikan gambaran yang ada berdasarkan hasil observasi lapangan.

d. Implementasi Sistem dan pengujian

Unit sistem yang sifatnya individual tersebut, diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa tujuan penelitian telah dicapai. Pada tahapan ini bertujuan untuk mengimplementasikan sistem otentikasi dan otorisasi untuk proses login multi aplikasi web berbasis PHP dengan mengoptimalkan penggunaan dari radius server

4. HASIL DAN PEMBAHASAN

a. Implementasi dan pengujian unit RADIUS server dengan FreeRADIUS

Alasan utama dipilihnya FreeRADIUS adalah FreeRADIUS merupakan salah satu server RADIUS yang bersifat non-komersil atau *freeware*. Selain itu juga dikarenakan FreeRADIUS dapat berjalan di berbagai sistem operasi, seperti linux, FreeBSD, OpenBSD, OSF/Unix, dan Solaris serta sudah mendukung beberapa *Access Point/Network Access Server* (NAS).
Pada tahap awal implementasi diasumsikan bahwa server RADIUS telah terinstalasi aplikasi Apache, PHP dan MySQL sebagai suatu aplikasi *web server*. Berikut ini adalah proses implemetasi yang dilakukan melalui repositori dengan sistem operasi linux ubuntu:

1. Tahap instalasi FreeRADIUS

Proses awal instalasi FreeRADIUS dilakukan dengan menginstalasi paket aplikasi FreeRADIUS, paket aplikasi freeradius-mysql supaya FreeRADIUS dapat mendukung data yang di simpan pada *database*

MySQL dan paket freeradius-utils sebagai perangkat utilitas dari aplikasi FreeRADIUS. Instalasi FreeRADIUS ini dilakukan dengan printah berikut menggunakan linux ubuntu:

 # apt-get install freeradius freeradius-mysql freeradius-utils

2. Konfigurasi FreeRADIUS

Ada 4 file secara umum yang harus dikonfigurasi, yaitu radiusd.conf, site-enable/default.conf, clients.conf dan sql.conf. File radiusd.conf merupakan file konfigurasi utama dari FreeRADIUS. File ini menentukan modul-modul yang akan digunakan dalam menjalankan layanan FreeRADIUS. Oleh karena layanan FreeRADIUS akan menjalankan modul sql, maka file ini perlu di konfigurasi supaya dapat menjalankan modul konfigurasi sql dengan menghilangkan tanda komentar (#) pada baris $INCLUDE sql.conf. File site-enable/default.conf merupakan file konfigurasi dari *virtual host* server RADIUS. Konfigurasi tersebut diantaranya, sebagai berikut:

```
authorize {
    preprocess
    chap
    mschap
    suffix
    eap
    sql
    expiration
    logintime
    pap
}
authenticate {
    Auth-Type PAP {
            pap
    }
    Auth-Type CHAP {
            chap
    }
    Auth-Type MS-CHAP {
            mschap
    }
    eap
}
preacct {
    preprocess
    acct_unique
    suffix
}
accounting {
    detail
    sql
    sql_log
}
session {
    radutmp
    sql
}
```

File client.conf merupakan file konfigurasi untuk memberikan izin RADIUS klien yang dapat berkomunikasi dan mendapatkan layanan dari RADIUS server. Konfigurasi tersebut diantaranya adalah memberikan *shared secret* dan identitas type NAS. Berikut ini konfigurasi yang diterapkan pada file ini.

```
client 192.168.0.2/32 {
    secret          = tester
    shortname               = pcku
    nastype         = other
}
```

File sql.conf merupakan file konfigurasi yang akan diterapkan supaya server RADIUS dapat terkoneksi dengan *database* MySQL. Konfigurasi yang terjadi pada file adalah pemberian nilai alamat host, *username*, *passoword* dan nama *database* yang digunakan oleh *database* server RADIUS.

3. Pengujian FreeRADIUS

Langkah awal dari pengujian ini adalah dengan menjalankan FreeRADIUS dalam mode debug. Langkah ini dilakukan dengan mengetikkan perintah "freeradius -Xxx" pada *command line* linux.

Pengujian unit FreeRADIUS dilakukan dengan 2 cara yaitu pengujian dari internal server RADIUS dan dari eksternal klien RADIUS.

Proses pengujian dari internal dilakukan dengan cara mengetikkan perintah "radtest user passwd radius-server[:port] nas-port-number secret [pphint] [nasname]". Perintah "radtest" merupakan suatu perintah untuk mengirimkan paket ke server RADIUS dan memperlihatkan hasil replay dari proses pengiriman paket tersebut.

Pengujian yang kedua dilakukan dari RADIUS klien dari luar RADIUS server. Pengujian unit dari eksternal RADIUS server dengan menggunakan aplikasi NTRadPing. NTRadPing merupakan tools pada sistem operasi windows yang ditujukan untuk melakukan testing instalasi dari RADIUS server.

b. Implementasi Web Server dan Paket Otentikasi ke RADIUS Server.

Untuk dapat membangun aplikasi PHP, dibutuhkan beberapa paket yang digunakan pada web server diantaranya adalah paket apache, paket PHP dan paket MySQL. Implementasi web server pada linux ubuntu dilakukan dengan perintah berikut:

#apt-get install apache2
#apt-get install mysql-server mysql-client
#apt-get install php5 libapache2-mod-php5
#apt-get install php5-mysql php5-curl php5-gd php5-idn php-pear php-db php5-imagick php5-imap php5-mcrypt php5-memcache php5-ming php5-ps php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
#a2enmod ssl
#a2ensite default-ssl
#/etc/init.d/apache restart
#/etc/init.d/mysql restart

Setelah proses instalasi web server selesai, proses selanjutnya adalah proses aktifasi paket otentikasi ke RADIUS server. Proses ini dilakukan dengan menambahkan modul untuk phpnya. Modul tersebut adalah php5-dev, php5-auth-pam, php5-radius, dan php-pear serta Auth_RADIUS untuk pear-nya.

c. Pengkodean Form Login dengan PHP

Form login dibangun dengan sebuah file html yang dilengkapi dengan beberapa sintaks PHP dan disimpan dengan ekstensi (.php). Berikut ini implementasi kode program dari form login menggunakan sistem otentikasi dan otorisasi RADIUS server.

```
<?php
$redirect_url = "https://192.168.0.2/login.php";
if (isNotEmpty($_POST["username"], $_POST["password"]) ) {
  $username = $_POST["username"];
  $password = $_POST["password"];
  $radserver = '192.168.0.1';
  $radport = 1812;
  $sharedsecret = 'testing123';
  $auth_type = 'pap';
  $radportacc = 1813;
  $starttime = time();
  $res = radius_auth_open();
  radius_add_server($res, $radserver, $radport, $sharedsecret, 3, 3);
  radius_create_request($res, RADIUS_ACCESS_REQUEST);
  radius_put_string($res, RADIUS_NAS_IDENTIFIER, isset($HTTP_HOST) ? $HTTP_HOST : 'localhost');
  radius_put_int($res, RADIUS_SERVICE_TYPE, RADIUS_FRAMED);
  radius_put_int($res, RADIUS_FRAMED_PROTOCOL, RADIUS_PPP);
  radius_put_string($res, RADIUS_CALLING_STATION_ID, isset($REMOTE_HOST) ? $REMOTE_HOST :
'127.0.0.1') == -1);
  radius_put_string($res, RADIUS_USER_NAME, $username);
  if ($auth_type == 'chap') {
        mt_srand(time());
        $chall = mt_rand();
        $chapval = pack('H*', md5(pack('Ca*',1 , $password . $chall)));
        $pass = pack('C', 1) . $chapval;
        radius_put_attr($res, RADIUS_CHAP_PASSWORD, $pass);
        radius_put_attr($res, RADIUS_CHAP_CHALLENGE, $chall);
  } else {
        radius_put_string($res, RADIUS_USER_PASSWORD, $username);}
  $req = radius_send_request($res);
  if($req == RADIUS_ACCESS_ACCEPT){
        if (!isset($REMOTE_ADDR)) $REMOTE_ADDR = '192.168.0.2';
        $resacc = radius_acct_open();
        radius_add_server($resacc, $radserver, $radportacc, $sharedsecret, 3, 3);
        radius_create_request($resacc, RADIUS_ACCOUNTING_REQUEST);
```

```
            radius_put_string($resacc, RADIUS_NAS_IDENTIFIER, isset($HTTP_HOST) ? $HTTP_HOST :
'localhost');
            radius_put_int($resacc, RADIUS_SERVICE_TYPE, RADIUS_FRAMED);
            radius_put_int($resacc, RADIUS_FRAMED_PROTOCOL, RADIUS_PPP);
            radius_put_string($resacc, RADIUS_CALLING_STATION_ID, isset($REMOTE_HOST) ?
$REMOTE_HOST : '127.0.0.1') == -1);
            radius_put_string($resacc, RADIUS_USER_NAME, $username);
            radius_put_addr($resacc, RADIUS_FRAMED_IP_ADDRESS, $REMOTE_ADDR);
            radius_put_int($resacc, RADIUS_ACCT_STATUS_TYPE, RADIUS_START);
            radius_put_int($resacc, RADIUS_ACCT_STATUS_TYPE, RADIUS_STOP);
            $sessionid = sprintf("%s:%d-%s", $REMOTE_ADDR, getmypid(), get_current_user());
            radius_put_string($resacc, RADIUS_ACCT_SESSION_ID, $sessionid);
            radius_put_int($resacc, RADIUS_ACCT_AUTHENTIC, RADIUS_AUTH_LOCAL);
            sleep(3);
            radius_put_int($resacc, RADIUS_ACCT_TERMINATE_CAUSE,
RADIUS_TERM_USER_REQUEST);
            radius_put_int($resacc, RADIUS_ACCT_SESSION_TIME, time() - $starttime);
            $reqacc = radius_send_request($resacc);
            radius_close($resacc);
            radius_close($res);
            header("Location: index.php");
  } else if($req == RADIUS_ACCESS_REJECT){
            radius_close($res);
            jsAlert("Radius Request rejected", $redirect_url);
  } else {
            radius_close($res);
            jsAlert("Unexpected return value:$req\n", $redirect_url);}
} else {
  jsAlert("maaf, user dan password harus di isi", $redirect_url);}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Login</title>
</head>
<body>
<form action="<?php echo $redirect_url;?>" method="post" encrypt="multipart/form-data">
<strong>Username</strong><input name="username" type="text" id="username"  size="20" />
<strong>Password</strong><input name="password" type="password" id="password"  size="20" />
<input name="login" type="submit" value="Login" /><input type="hidden" name="logon" value="1" />
</form>
</body>
</html>
```

Kode program merupakan sebuah file php dengan nama login.php yang diatas terdiri dari sebuah form login dengan input username dan password serta form tersebut memiliki method post dan action pada file login.php itu sendiri. Ketika inputan username dan password dikirim maka akan menjalankan metode pengotorisasian dan pengotentikasian ke RADIUS server yang beralamat di IP Address '192.168.0.1'.

d. Pengujian Sistem Keseluruhan

Berikut ini hasil pengujian sistem dengan menggunakan tipe password yang berbeda dan metode otentikasi PAP dan CHAP.

Tabel 1 Data Hasil Pengujian Form Login Aplikasi Web Berbasis PHP Dengan Menggunakan Tipe Password dan Metode Otentikasi yang Beragam

No.	Username	Type Password	Metode Otentikasi PAP		Metode Otentikasi CHAP	
			Access-Accept	Access-Reject	Access-Accept	Access-Reject
1.	user1	User-Password	√		√	
2.	user2	Cleartext-Password	√		√	
3.	user3	Crypt-Password	√			√
4.	user4	MD5-Password	√			√
5.	user5	SHA1-Password		√		√
6.	user6	CHAP-Password		√		√

Dari tabel 1 menunjukkan hasil implementasi pengujian pada beberapa form login aplikasi web berbasis PHP sebagai RADIUS klien dari sebuah RADIUS server terlihat bahwa konfigurasi yang terapkan mampu

mengimplementasikan metode otentikasi PAP dengan type password dari pengguna yang diuji yaitu type password User-Password, Cleartext-password, Crypt-Password dan MD5-Password sedangkan untuk SHA1-Password dan CHAP-Password, akses type password tersebut ditolak. Dan pada metode otentikasi CHAP, tipe password yang diterima dari hasil konfigurasi tersebut adalah type password User-Password dan Cleartext-Password. Sedangkan tipe password yang ditolak adalah tipe password Crypt-Password, MD5-Password, SHA1-Password dan CHAP-Password.

5. KESIMPULAN

Melalui pengimplementasian optimalisasi RADIUS server sebagai sistem otentikasi dan otorisasi untuk proses login multi aplikasi web berbasis PHP, maka dapat diambil suatu kesimpulan bahwa pengguna pada beberapa aplikasi web berbasis PHP dapat diintegrasikan pengelolaannya dengan membangun sistem otentikasi dan otorisasi dengan radius server menggunakan aplikasi FreeRADIUS dan proses optimalisasi radius server sebagai sistem otentikasi dan otorisasi ini dapat membuat pengguna hanya akan memiliki satu akun tunggal untuk beberapa aplikasi yang berbeda.

DAFTAR PUSTAKA

[1] Smith, R.W., 2009, CompTIA Linux+ study guide. 1st ed. Indianapolis. Wiley Publishing.
[2] Hassel, J. 2002. RADIUS. Sebastopol. O'Reilly.
[3] Rigney, C,. Livingston, S.W., Merit, A.R., Daydreamer, W.S. 2000. Remote Authentication Dial In User Service (RADIUS). [Online] http://www.ietf.org/rfc/rfc2865.txt, Diakses pada tanggal 28 Juni 2010 11:24.