

IMPLEMENTASI REMOTING SERVER LINUX BERBASIS SHORT MESSAGE SERVICE

Vishnu Valentino, S.Kom; Bernard Renaldy Suteja, S.Kom, M.Kom; Wilfridus Bambang, S.T
Staf Pengajar Fakultas Teknologi Informasi, UK. Maranatha Bandung
vishnuvalentino@yahoo.com; bernardjogja@gmail.com; wilfridus.bambang@eng.maranatha.edu

ABSTRACT

To increase the flexibility and mobility of system administration, a number of methods have been doing used including long range server machine control methods such as SSH (Secure Shell), Telnet (Teletype Network), etc. However, since these methods rely on the connection to internet, it is not really useful when there is no internet connection around.

The other methods that have been invented for this purpose with a flexible environment are the use of Kannel WAP and SMS gateway applications. However, these applications have a drawback, since they also implemented on some spesific technologies such as smart phone or PDA. In this paper, an application which uses only GSM (Global System of Mobile Communication), and allows system administration to be cunducted everywhere and easily is proposed.

Keywords: SMS, Kannel, GSM

1. Pendahuluan

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor dan RAM (*Random Access Memory*) yang besar, juga dilengkapi dengan sistem operasi khusus. Mesin *server* umumnya dikelola oleh seorang *administrator* sistem komputer, karena sebuah *server* memerlukan pemeliharaan khusus untuk operasinya.

1.1 Latar Belakang

Seorang administrator sistem harus mampu untuk menangani setiap permasalahan yang terjadi pada server dalam waktu 7 x 24 jam, karena umumnya *server* (dalam perusahaan besar) bersifat waktu nyata (*realtime*), khususnya perusahaan yang bergerak dalam bidang-bidang perbankan sehingga diperlukan penanganan khusus dan secepatnya untuk menghadapi setiap permasalahan yang terjadi. Server juga memerlukan pemeliharaan ekstra, seperti waktu berkala untuk memperbaharui sistem *server* dengan paket-paket terbaru sebagai menutup celah sekuriti.

Tingginya tingkat mobilitas seorang administrator menjadi sebuah kendala ketika *server* yang dikelolanya mengalami suatu masalah. Umumnya ketika sebuah *server* tidak bisa memberikan pelayanan semestinya dan administrator tidak ada di tempat kerjanya, notifikasi diperoleh melalui telepon rekan kerja atau SMS. Jika permasalahan tersebut memerlukan penanganan dini dan di lokasi seorang administrator tidak terdapat jaringan internet untuk melakukan pengendalian *server* akan menjadi sebuah masalah yang cukup serius.

1.2 Rumusan Permasalahan

Berdasarkan latar belakang permasalahan di atas, maka dapat dirumuskan untuk:

- Membuat suatu aplikasi yang menjadi sebuah solusi untuk masalah tersebut, sehingga dapat meminimalisasi kemungkinan administrator kehilangan kendali ketika *server* yang dikelolanya mengalami gangguan.
- Dapat mengetahui setiap saat ketika *server* yang dikelolanya mengalami gangguan.

1.3 Tujuan Permasalahan

Mempermudah kinerja seorang administrator sehingga dapat bekerja dimana saja sebagai tuntutan mobilitas, khususnya dapat mengatasi permasalahan *server* dengan sesegera mungkin, serta dapat melakukan pemeliharaan secara berkala untuk *server* yang dikelolanya.

1.4 Batasan Permasalahan

Batasan permasalahan yang dibuat mencakup:

- Sistem operasi yang digunakan untuk mesin *server* dan komputer pengontrol adalah Linux Debian Etch 4.0 dengan kernel 2.6.18.
- Perintah yang digunakan untuk melakukan pengendalian *server* dilakukan berdasarkan kebijakan administrator, mengingat bahwa SMS memiliki batasan 160 karakter per SMS.
- Server* dapat dikendalikan selama terdapat jaringan telepon seluler GSM dan kondisi jaringan antara *server* dan komputer pengontrol tidak terdapat masalah.

2. Landasan Teori

2.1 Linux

Linux adalah sistem operasi bebas (*open source*) yang dibuat pertama kali oleh Linus Torvalds, dari Universitas Helsinki di Finlandia pada tahun 1991. Dalam perkembangannya, saat ini sistem operasi Linux telah digunakan pada banyak tipe arsitektur komputer dengan berbagai macam bentuk implementasi pada sisi *server* dan client.

2.2 Web Server: Apache

Apache merupakan perangkat lunak bersifat gratis dan *open source* yang banyak digunakan oleh para administrator untuk *web server* yang mereka kelola. Berikut adalah beberapa kelebihan *web server* Apache :

- Perangkat lunak yang gratis (*freeware*).
- Kemudahan dalam proses instalasi.

2.3 DNS (Domain Name System)

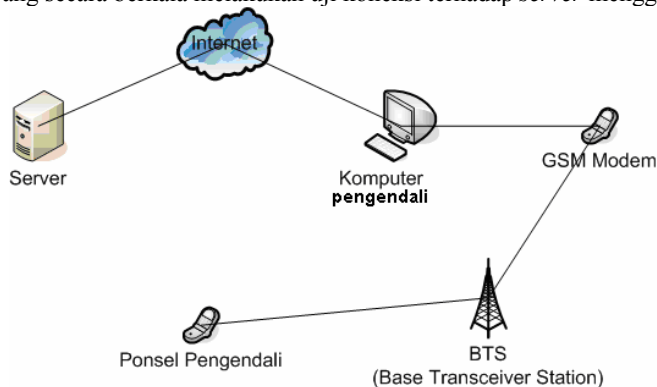
DNS (*Domain Name System*) banyak digunakan dalam kegiatan sehari-hari, khususnya untuk kegiatan yang berhubungan dengan Internet. Fungsi utama dari sebuah sistem DNS adalah menerjemahkan nama-nama host (*hostname*) menjadi nomor IP (*IP address*) ataupun sebaliknya, sehingga nama tersebut mudah diingat oleh pengguna Internet.

2.4 KANNEL (Opensource WAP dan SMS Gateway)

Kannel adalah WAP dan SMS *Gateway*. WAP adalah kependekan dari *Wireless Application Protocol*, adalah kumpulan perangkat, bahasa dan infrastruktur untuk mengimplementasikan layanan untuk telepon seluler. Kannel dapat pula melayani sebagai SMS *Gateway* untuk jaringan GSM. Hampir seluruh telepon seluler GSM dapat melakukan pengiriman dan penerimaan pesan SMS.

3. Analisa dan Pemodelan

Arsitektur topologi yang digunakan dapat dilihat pada Gambar 2 dibawah ini dimana penulis menggunakan 2 buah komputer yaitu sebuah komputer *server* dan sebuah komputer pengendali yang menggunakan sistem operasi Debian Etch 4.0. Alasan penulis menggunakan topologi ini ialah penulis ingin selalu mengetahui keadaan *server* yang dikelolanya apakah masih dalam keadaan terkoneksi dengan jaringan atau terputus dari jaringan, dimana pekerjaan tersebut dilakukan oleh komputer pengendali yang secara berkala melakukan uji koneksi terhadap *server* menggunakan perintah ping.



Gambar 1. Topologi yang digunakan

Semua status koneksi antara komputer pengendali dengan *server*, baik itu dalam kondisi tersambung ataupun terputus, maka sistem akan mengirimkan pesan berupa SMS ke telepon seluler *administrator* sebagai ponsel pengendali. Ponsel pengendali mengirimkan sebuah perintah *shell* yang kemudian dikirimkan kepada nomor ponsel yang dijadikan sebagai GSM modem melalui perantara suatu BTS. Pada komputer pengendali terdapat sebuah aplikasi SMS *Gateway* yaitu Kannel. Ketika SMS diterima oleh ponsel yang dijadikan GSM modem yang terdapat pada komputer pengendali, Kannel akan mengambil alih pesan tersebut untuk kemudian meneruskan isi dari pesan kepada komputer *server* melalui halaman web.

Pada komputer *server* terdapat sebuah halaman php yang berisi fungsi *exec* untuk melakukan eksekusi perintah pada sistem lokal. Penulis menggunakan metode GET dimana isi dari variabelnya diambil dari isi pesan sms yang diteruskan oleh kannel kepada komputer pengendali. Untuk balasan (*reply*) kepada ponsel pengendali, hasil eksekusi perintah pada komputer *server* akan diambil dan dikirimkan kepada ponsel pengendali. Mengingat batasan karakter SMS adalah 160 karakter per-SMS, maka pesan yang memiliki karakter lebih dari 160 akan otomatis terpotong, namun konfigurasi dapat dirubah.

4. Hasil dan Pembahasan

4.1 Konfigurasi BIND (Berkeley Internet Name Domain)

Langkah pertama yaitu melakukan konfigurasi alamat IP pada berkas `/etc/network/interfaces`, sebagai berikut:

```

auto lo
iface lo inet loopback
allow-hotplug eth0
iface eth0 inet static
address 10.10.10.10
netmask 255.255.255.0
  
```

```
gateway 10.10.10.1
broadcast 10.10.10.255
auto eth0
```

- a. Setelah itu pengguna harus melakukan konfigurasi terhadap berkas `/etc/hosts`.

```
127.0.0.1 localhost dua
10.10.10.10 msggateway.org v4L
```

- b. Pengguna juga harus melakukan konfigurasi pada berkas `/etc/resolv.conf`.

```
domain msggateway.org
nameserver 10.10.10.10
```

- c. Pengguna melakukan instalasi BIND (*Berkeley Internet Name Domain*) `apt-get install bind9`.

- d. Lakukan konfigurasi lokal terhadap `server` BIND, pada berkas `/etc/bind/named.conf.local`.

```
zone "msggateway.org" {
    type master;
    file "/etc/bind/db.msggateway.org"; };
zone "10.10.10.in-addr.arpa" {
    type master;
    file "/etc/bind/10.10.10.reverse"; };
```

- e. Berkas zona yang telah ditulis diatas yaitu `/etc/bind/db.msggateway.org` dan `/etc/bind/10.10.10.reverse` harus dibuat. Berikut adalah isi dari `db.msggateway.org` pada komputer penulis.

```
@ IN SOA msggateway.org. root.msggateway.org. (
    1 ; serial
    604800 ; refresh
    86400 ; retry
    2419200 ; expire
    604800) ; minimum
@ IN NS dns.msggateway.org.
www IN A 10.10.10.10
dns IN A 10.10.10.10
```

- f. Sedangkan untuk isi dari berkas `10.10.10.reverse` pada komputer penulis adalah sebagai berikut.

```
@ IN SOA msggateway.org. root.msggateway.org. (
    1 ; serial
    604800 ; refresh
    86400 ; retry
    2419200 ; expire
    604800) ; minimum
IN NS dns.msggateway.org.
1 IN PTR www.msggateway.org.
1 IN PTR dns.msggateway.org.
```

4.2 Berkas PHP Pengontrol

Untuk melakukan pengendalian `server`, di sini penulis menggunakan skrip php dengan fungsi `exec`, yaitu fungsi untuk melakukan eksekusi perintah melalui php. Fungsi `exec` ini tidak dapat dijalankan jika opsi `Safe_Mode = On` yang terdapat pada `/etc/php5/apache2/php.ini`. Pada komputer yang penulis gunakan, berkas ini diberi nama `perintah.php`.

```
<? $ip = getenv('REMOTE_ADDR'); if ($ip!="10.10.10.10")
    echo "<center><h1>Not Allowed, You Are Not Authorized To View This Page</h1><br>-admin-</center>";
else{
$search=array ("<script[^\>]*?>.*?</script>'si","<[\\\/!]*?<[^\>]*?>'si","([\r\n])[\s]+",
    "'&(quot|#34);'i","'&(amp|#38);'i","'&(lt|#60);'i","'&(gt|#62);'i",
    "'&(nbsp|#160);'i","'&(iexcl|#161);'i","'&(cent|#162);'i","'&(pound|#163);'i",
    "'&(copy|#169);'i","'&#(\d+);'e");
$replace = array ("","","\1","\","&","<",">"," ", chr(161),chr(162),chr(163),chr(169),"chr(\1)");
if($_GET['perintah']=="format"){
exec($_GET['perintah'],$simpan);
if(substr($_GET['perintah'],0,2)=="cp"){
    $veril = split(" ",$_GET['perintah']); happy1 = split("/",$_GET['perintah']);
    for ($i=0; $i < sizeof($happy1); $i++) {
        if(preg_match("/\./i",$happy1[$i])) { $tempat1 = $happy1[$i]; }
    }
    echo "copy file $veril[1] ke $veril[2] telah berhasil";
}
if(substr($_GET['perintah'],5,2)=="cp"){
    $veril2 = split(" ",$_GET['perintah']); $happy2 = split("/",$_GET['perintah']);
    for ($i=0; $i < sizeof($happy2); $i++) {
        if(preg_match("/\./i",$happy2[$i])){ $tempat2 = $happy2[$i]; }
    }
    echo "copy file $veril2[2] sebagai root ke $veril2[3] telah berhasil";
}
if(substr($_GET['perintah'],0,2)=="rm"){
    $veril = split(" ",$_GET['perintah']); $happy1 = split("/",$_GET['perintah']);
    for ($i=0; $i < sizeof($happy1); $i++) {
        if(preg_match("/\./i",$happy1[$i])){ $tempat1 = $happy1[$i]; }
    }
    echo "Hapus file/folder $veril[1] telah berhasil";
}
if(substr($_GET['perintah'],5,2)=="rm"){
    $veril2 = split(" ",$_GET['perintah']); $happy2 = split("/",$_GET['perintah']);
    for ($i=0; $i < sizeof($happy2); $i++) {
```

```

        if(preg_match("/\./i",$happy2[$i])){ $tempat2 = $happy2[$i];}
        echo "hapus file/folder $veri2[2] sebagai root telah berhasil";}
for ($i=0; $i < sizeof($simpan); $i++) { echo("<br>$simpan[$i]");} ?>

```

Agar berkas perintah.php dapat dieksekusi, maka rubah sifat berkas menjadi dapat dieksekusi dengan menggunakan perintah **chmod +x perintah.php**.

Konfigurasi berkas httpd.conf yang terdapat pada **/etc/apache2/httpd.conf**

```

<Directory "/home/valentino/web/perintah">
    Deny from all
    AllowOverride
    Allow from 10.10.10.11
</Directory>

```

Skrip di atas mengarahkan agar sistem hanya mengijinkan koneksi dari alamat IP 10.10.10.11 untuk mengakses direktori **/home/valentino/web/perintah** melalui web.

4.3 Instalasi dan Konfigurasi Kannel SMS Gateway

Lakukan instalasi kannel menggunakan berkas sumber yang telah disediakan oleh Debian yang terdapat pada paket DVD yang disertakan. Ketikkan perintah **apt-get install kannel**. Selanjutnya adalah melakukan konfigurasi pada **/etc/kannel/kannel.conf**. Disini penulis menggunakan ponsel Sony Ericsson T616 sebagai GSM modem.

```

group = core
admin-port = 13000
admin-password = bar
access-log = "/var/log/kannel/beareraccess.log"
white-list = http://www.smsgateway.org/perintah/white.txt
smsbox-port = 13001
box-deny-ip = "* * * *"
box-allow-ip = "127.0.0.1;10.10.10.11"
store-file = "/var/log/kannel/kannel.store"
group = smsbox
bearerbox-host = smsgateway.org
sendsms-port = 13013
log-file = "/var/log/kannel/smsbox.log"
access-log = "/var/log/kannel/smsaccess.log"
group = sendsms-user
username = v4l
password = v4l
group = modems
id = ericsson
name = T616
detect-string = "T616"
init-string = AT+CNMI=2,3,0,0,0
group = smsc
smc = at
modemtype = ericsson
device = /dev/ttyUSB0
pin = 1111
speed = 57600
sms-center = +62855000000
group = sms-service
keyword = default
get-url = "http://www.smsgateway.org/perintah/perintah.php?perintah=%a"
max-messages = 1

```

Keterangan berkas konfigurasi kannel.conf:

- group = core adalah konfigurasi mutlak yang harus ada dalam konfigurasi kannel.
- admin-port adalah *port bearerbox* untuk eksekusi perintah berbasis HTTP. Umumnya nilai dari *port* di atas 1023.
- admin-password digunakan untuk kata kunci administrasi perintah HTTP.
- access-log adalah berkas dimana informasi mengenai sms yang diterima maupun dikirim akan tercatat.
- white-list adalah daftar nomor telepon yang diijinkan untuk mengakses modem/gateway.
- smsbox-port digunakan untuk mengatasi trafik SMS masuk dan keluar.
- box-deny-ip dan box-allow-ip adalah alamat IP yang diijinkan dan ditolak untuk mengakses kannel.
- store-file adalah tempat penyimpanan ketika kannel menerima SMS, akan disimpan sampai ditangani oleh sistem.
- group = smsbox merupakan perintah yang harus dicantumkan jika hendak menggunakan smsbox.
- Bearerbox-host adalah lokasi *host* dimana bearerbox tersimpan.
- sendsms-port adalah *port* yang digunakan untuk melakukan pengiriman atau penerimaan SMS.
- log-file dan access log adalah berkas yang akan menyimpan akses ke kannel.
- group = sendsms-user merupakan grup standar untuk nama pengguna dan kata kunci dalam mengakses kannel.
- username dan password adalah nama pengguna dan kata kunci yang didefinisikan untuk mengakses kannel.

- group = modems adalah konfigurasi standar jika hendak mendefinisikan ponsel yang dijadikan modem.
 - id = ericsson adalah tipe dari handphone yang akan dijadikan modem. Ini disesuaikan dengan tipe perangkat.
 - name adalah nama konfigurasi modem yang didefinisikan. Digunakan pada pencatatan perubahan sistem (log).
 - detect-string didefinisikan mendeteksi string ketika kannel mengakses ponsel. Umumnya diisi dengan tipe ponsel.
 - init-string, inialisasi string agar kannel mengenali penggunaan perintah AT.
 - group=smc merupakan konfigurasi standar jika hendak menggunakan SMSC (SMS-center).
 - smc digunakan mengidentifikasi tipe dari sms center. Umumnya smc dikonfigurasi menjadi "at" atau automatic.
 - device, variabel ini dimaksudkan agar sistem mengetahui bagaimana ponsel terkoneksi dengan komputer, seperti /dev/USB0. Diketahui dengan membuka berkas catatan (log), menggunakan perintah **tail -f /var/log/messages**.
 - pin adalah nomor pin kartu GSM yang pengguna gunakan sehingga kannel dapat mengakses kartu GSM.
 - Speed, kecepatan transfer data ponsel. Kecepatan ini bervariasi tergantung tipe ponsel
 - sms-center, mendefinisikan nomor sms center dari penyedia layanan telepon GSM yang digunakan. Penulis menggunakan layanan Indosat M3 yang memiliki nomor sms center +62855000000.
 - group = sms-service, digunakan untuk layanan pesan yang masuk ke dalam sistem.
 - max-messages = 1, sistem akan mengirimkan balasan berdasarkan query.
- Dikarenakan sistem kannel akan mengakses ponsel yang terkoneksi dengan komputer, maka hak akses dari /dev/ttyUSB0 (letak koneksi ponsel pada komputer penulis) harus dirubah dengan perintah: **chmod 666 /dev/ttyUSB0**.

Secara standar, kannel akan menjalankan daemon WAPBOX yang terdapat pada **/etc/default/kannel**, dimana daemon tersebut tidak akan digunakan, namun disini akan digunakan SMSBOX, sehingga pengguna harus menghapus tanda komentar '#' sebelum baris **START_SMSBOX = 1**. Berikut adalah konfigurasi pada komputer penulis.

```
#START_WAPBOX=1
START_SMSBOX=1
```

Setelah seluruh konfigurasi berhasil dilakukan, maka pengguna dapat melakukan pengetesan koneksi menggunakan perintah **bearerbox -lokasi/konfigurasi/kannel.conf/** tersimpan.

4.4 Skrip Bash untuk Menguji Koneksi Server

Penulis menggunakan skrip bash untuk melakukan pengecekan secara berkala terhadap *server* yang diintegrasikan dengan modul Perl yaitu libwww-perl. Berikut adalah isi dari berkas **/usr/local/sbin/koneksi.sh** pada komputer penulis.

```
hasil=$(ping -c 1 $1|head -n 2|tail -n 1)
hasil=${hasil:0:2}
if(($hasil != 64))
then
hidup=$(cat /usr/local/sbin/live.txt)
if (($hidup == 1))
then
echo 0 > /usr/local/sbin/live.txt
time=$(date +%H:%M:%S")
echo "Server not connected at $time"
GET -e 'http://127.0.0.1:13013/cgi-bin/sendsms?user=v4l&pass=v4l&from=v4l&to=%2b622270462032&text=Server%20is%20not%20Connected'
else
echo 'Server still not Connected'
fi
else
hidup=$(cat /usr/local/sbin/live.txt)
if (($hidup == 0))
then
echo 1 > /usr/local/sbin/live.txt
time=$(date +%H:%M:%S")
echo "The Server Now Connected at $time"
GET -e 'http://127.0.0.1:13013/cgi-bin/sendsms?user=v4l&pass=v4l&from=v4l&to=%2b622270462032&text=The%20Server%20Now%20Connected'
else
echo 'Server still Connected'
fi
fi
```

Penulis menggunakan metode ping untuk melakukan pengecekan terhadap *server*. Jika hasil ping adalah 64, maka skrip akan mengecek berkas live.txt, apakah bernilai sama dengan nol (0), jika nol, maka masukkan angka 1 terhadap berkas tersebut, lalu tuliskan "Server now connected at -waktu saat server terkoneksi—" dan kirim pesan kepada nomor ponsel +622270462032. Jika hasil ping masih menunjukkan nilai 64, skrip akan mengecek berkas live.txt apakah nilai dari berkas tersebut adalah tidak sama dengan nol (0), maka tuliskan pesan "server still connected" dan sistem tidak melakukan pengiriman pesan melalui sms.

4.5 Konfigurasi Crontab

Skrip Crontab berikut akan menjalankan perintah *bash* yang telah dibuat dan memasukkan catatan (log) hasil eksekusi perintah ke dalam berkas `/home/valentino/logmon.txt` setiap satu menit sekali. Isi dari konfigurasi crontab:

```
0-59 * * * * /usr/local/sbin/koneksi.sh 10.10.10.11 >> /home/valentino/logmon.txt
```

4.6 Implementasi

Beberapa interaksi melalui peralatan *mobile* (HP) untuk berkomunikasi antar *administrator* dengan *server* Linux.

- Untuk mendaftarkan No HP yang digunakan oleh *administrator* dilakukan cara sebagai berikut : **reg password** dan akan diberikan *response* sebagai berikut :



- Untuk melakukan unregistrasi No HP dilakukan cara sebagai berikut : **ureg password** dan akan diberikan *response* sebagai berikut:



- Server* akan memberikan informasi terhadap sesuatu yang terjadi dengan berkirim pesan ke administrator, seperti terjadi perubahan port, *server down* atau *server up*.



5. Penutup

Dari Implementasi Remoting Server Linux berbasis Short Message Service ini maka dapat disimpulkan bahwa:

- Dengan memanfaatkan implementasi *remoting server* Linux berbasis *short message service* maka pengelolaan *server* dapat dilakukan oleh seorang *administrator* tanpa terbatas ruang dan waktu.
- Pemanfaatan aplikasi ini dapat mempercepat seorang admin untuk mengambil keputusan.

6. Daftar Pustaka

- [1] Authentication, Authorization and Access Control, <http://httpd.apache.org/docs/2.0/howto/auth.html>, diakses terakhir tanggal 15 Juni 2007 21:30
- [2] Kannel project, <http://www.kannel.org>, diakses terakhir tanggal 15 Juni 2007 21:30
- [3] PHP: Taryan, http://72.14.235.104/search?q=cache:NT64m1gQRL0J:202.51.235.120/taryan/master/windows/Webserver/bukutam.u.rtf+apakah+_GET&hl=id&ct=clnk&cd=8&gl=id, diakses terakhir tanggal 15 Juni 2007 21:32.