

MANAJEMEN PENGELOLAAN LAN DENGAN *REMOTE SYSTEM APPLICATION*

Irwan Pribadi¹, Mukhammad Andri Setiawan²

CISCO Networking Academy Program, Informatics Department, Faculty of Industrial Technology,
Islamic University of Indonesia, Kampus Terpadu UII, Jl Kaliurang Km 14.5 Yogyakarta. 55501

Phone (0274) 895007, 895287 ext 122, 150; Fax. (0274) 895007 ext 148

E-mail: ¹free_badi@yahoo.com, ²andri@fti.uui.ac.id

Abstrak

Perkembangan teknologi jaringan pada saat ini sangatlah pesat. Hampir seluruh segi kehidupan tidak lepas dari teknologi jaringan ini. Local Area Network (LAN) merupakan salah satu bentuk dari teknologi jaringan komputer yang perkembangannya sangat pesat dan bahkan telah digunakan hampir di seluruh kehidupan kita pada berbagai bidang. Proses manualisasi sangatlah menghambat seorang admin tersebut dimana waktu akan terbuang percuma jika melakukan pengawasan client dengan mendatangi komputer masing-masing client tersebut. Kemudahan bagi seorang admin untuk melakukan pengaturan atau pengontrolan bagi masing-masing client pada jaringan LAN sangatlah diperlukan. Aplikasi remote sistem pada jaringan LAN merupakan sebuah solusi untuk bagi seorang admin guna menyelesaikan masalah client tanpa harus mendatangi komputer client tersebut.

Kata Kunci: Remote Sistem, Administrasi LAN,

1. Pendahuluan

Teknologi jaringan komputer dan jaringan internet pada saat ini telah merasuk hampir di seluruh segi kehidupan, sehingga sangat sulit menemukan bidang yang belum tersentuh oleh teknologi jaringan komputer. Local Area Network (LAN) merupakan salah satu bentuk dari teknologi jaringan komputer, dengan menghubungkan dua atau lebih komputer dalam suatu lingkup jaringan [1].

Manajemen dan diagnosis dalam LAN merupakan bagian penting untuk menjamin keberlangsungan konektivitas dalam jaringan [2][3] [5]. Pemantauan dari satu server terhadap beberapa client dalam sebuah jaringan LAN sangat diperlukan oleh seorang administrator mengingat kemungkinan lambatnya penanganan apabila dilakukan pengecekan dari masing-masing komputer secara manual. Kebutuhan ini akan semakin meningkat ketika jaringan semakin meluas, dan dibutuhkan penanganan yang semakin kompleks.

Dengan menggunakan Aplikasi Remote Sistem pada Jaringan Local Area Network (LAN) maka akan mempermudah penanganan masalah dan memungkinkan administrator dalam pengendalian jarak jauh (*meremote*) suatu client.

2. Tujuan Penelitian

Penelitian ini bertujuan untuk memahami proses peremotan client oleh seorang admin sebagai sarana untuk memudahkan admin dalam mengontrol clien-client dalam sebuah jaringan LAN yang menjadi tanggung jawabnya.

3. Landasan Teori

Remote artinya pengendali, maksudnya ialah mengendalikan sesuatu dari jauh dengan menggunakan alat tertentu. Meremote sesuatu pada

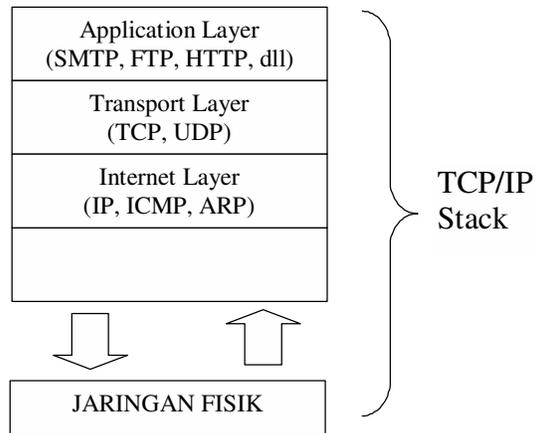
dasarnya ialah mengendalikan sesuatu itu dari jarak tertentu. Pengendali (yang melakukan *remoting*) berada dalam dua titik dengan target yang saling berhubungan.

Sebuah *remote* sistem juga merupakan dua buah titik yang saling berhubungan, dimana satu titik menjadi pengendali dan satu titik lainnya menjadi target. Titik target akan menjadi *passive* atau dalam istilah komunikasi disebut dengan status *listen*. Titik pengendali akan bersifat *active* dengan mengirimkan sebuah data atau lebih yang nantinya akan diterjemahkan oleh titik target sebagai proses/perintah yang harus dijalankan oleh target yang merupakan proses *remoting*.

3.1. TCP / IP

Inti dari TCP/IP (*Transmission Control Protocol/Internet Protocol*) yang merupakan kombinasi dari dua protokol yang bekerja bersama-sama untuk memberikan koneksi pada internet/intranet. IP dipakai untuk mendefinisikan dan mengirimkan datagram (unit data internet) dan menyatakan skema pengalamatan. TCP bertanggung jawab terhadap servis-servis level atas.

TCP/IP terdiri dari bagian-bagian tertentu dari komunikasi data. Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan interface jaringan, karena sebagian besar kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar TCP/IP dapat berjalan di atas interface jaringan tertentu, hanya perlu dilakukan perubahan pada protokol yang berhubungan dengan antarmuka jaringan saja. Sekumpulan protokol TCP/IP ini dimodelkan dengan empat layer TCP/IP [1], sebagaimana diperlihatkan Gambar 1.



Gambar 1. Layer TCP / IP

- **Network Interface Layer**
Bertanggung jawab mengirim dan menerima data dari media fisik. Media fisik ini berupa kabel, serat optik, atau gelombang radio. Protokol pada layer ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan sejenisnya.
- **Internet Layer**
Bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada layer ini terdapat tiga macam protocol, yaitu IP (*Internet Protocol*), ARP (*Address Resolution Protocol*) dan ICMP (*Internet Control Message Protocol*)
- **Transport Layer**
Bertanggung jawab untuk mengadakan komunikasi antar dua host/komputer. Kedua protocol tersebut ialah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*)
- **Application Layer**
Terletaknya semua aplikasi yang menggunakan protocol TCP / IP.

3.2. Socket

Adalah sebuah *resource* yang disediakan untuk melewati pengiriman dan penerimaan data pada komunikasi data. Pada O/S Microsoft Windows, *socket* tersedia pada library **winsock.dll**. Pemrograman *socket* adalah bagian dari pemrograman tingkat rendah. Terdapat tiga jenis koneksi *socket* yaitu:

- **Koneksi client**
Dimulai oleh *client* dan menghubungkan *socket client* lokal dengan sebuah *socket server* remote. *Socket client* harus menyatakan *server* yang akan dihubungi dengan memberikan nama *host* atau alamat IP dan *portnya*

- **Koneksi 'mendengarkan'**
Socket server bersifat pasif dan menunggu *client*. Jika *client* memberikan request baru *server* membuat *socket* baru khusus untuk koneksi tersebut dan 'mendengarkan' lagi. *Socket server* ini harus menyatakan port yang menyatakan servis yang diberikan
- **Koneksi server**
Koneksi yang diaktifkan oleh *server* pada saat *server* menerima request dari *client*

Tipe-tipe koneksi tersebut hanya penting untuk membangun hubungan antara *client* dan *server*. Setelah hubungan terbentuk, kedua sisi bebas untuk membuat *request* dan menerima data ke sisi lain.

3.3. Socket

Untuk dapat berkomunikasi pada protokol TCP/IP, diperlukanlah sebuah pintu yang sama pada titik-titik yang melakukan proses komunikasi. Pintu inilah yang disebut dengan port komunikasi. Port komunikasi sangat beragam nilainya, dimana jumlah port yang tersedia sangat banyak, dan agar dapat melakukan proses komunikasi data, protokol TCP/IP harus menunjuk sebuah port tertentu.

Tabel 1. Daftar Port Standar

Nilai Port	Standar Penggunaan Port
139	Koneksi
80	Web
21	FTP
8080, 3128	Proxy, Firewall
6667	IRC, Chatting
23	Telnet

3.4. Windows API

Windows API (*Application Programming Interface*) adalah sebuah pustaka berukuran sangat besar yang menyediakan ratusan fungsi standar untuk berbagai proses interaksi aplikasi yang dibuat dengan sistem operasi, dalam hal ini tentunya sistem operasinya Microsoft Windows. Fungsi-fungsi Windows API diletakkan pada beberapa file DLL, antara lain:

- *Gdi32.dll*, merupakan kumpulan fungsi GDI (graphics Device Interface), seperti TextOut, Ellipse, PlayMetaFile, SetViewportOrg, SetBkMode, DrawText, dan sebagainya
- *User32.dll*, merupakan kumpulan fungsi untuk antar muka pengguna (user interface), seperti penanganan mouse, cursor, ikon, keyboard, dan sebagainya
- *Kernel32.dll*, merupakan kumpulan fungsi untuk operasi level rendah, seperti manajemen memori, penanganan resource, multitasking, multithreading dan sebagainya

- *Lzexpand.exe*, merupakan kumpulan fungsi untuk menangani kompresi file dengan metode Lempel-Ziv
- *Mmsystem.dll*, merupakan kumpulan fungsi untuk penanganan untuk pemrograman multimedia

4. Desain

4.1. Desain dan Rancangan Aplikasi

Aplikasi Remote Sistem dibangun dari dua macam aplikasi yaitu aplikasi di sisi *server* dan aplikasi di sisi *client*.

4.1.1. Aplikasi Client

Merupakan aplikasi yang dijalankan di komputer *client* dengan status hide dan real time dengan tujuan *client* tidak mengetahui adanya aplikasi yang sedang berjalan dan memantau kegiatannya.

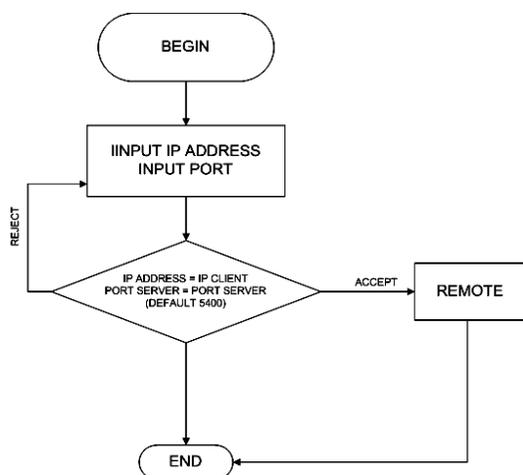
Aplikasi di sisi *client* bersifat listen atau mendengarkan admin, dimana *client* akan menerima pesan dari admin (defaultnya 5400) yang akan diterjemahkan menjadi perintah-perintah yang akan dilakukan oleh *client*.

4.1.2. Aplikasi Server

Merupakan aplikasi yang berjalan di sisi *server* yang memungkinkan seorang admin memberikan perintah yang akan dikirim berupa pesan melalui port (defaultnya 5400) kepada *client* dan akan diterjemahkan oleh *client* berupa intruksi atau proses yang akan dilakukan *client*.

5. Implementasi

Proses penyambungan program pengendali dari Aplikasi Remote Sistem dapat dilihat pada gambar 2. Aplikasi akan memastikan apakah aplikasi di sisi *server* tersambung dengan aplikasi di sisi *client* dengan melakukan pengecekan apakah port *client* dengan *server* sama dan pengecekan apakah *client* dalam network yang benar.



Gambar 2. Flowchart Aplikasi Remote Sistem

Server akan terhubung dengan *client* apabila IP ADDRESS telah terisi dengan alamat *client* yang diinginkan serta PORT *server* yang mempunyai nilai sama dengan PORT *client* (default 5400) dan kemudian dikirimkan Protocol Data Unit (PDU) berupa 'message' yang akan diterima oleh *client*. Apabila dua hal ini tidak tercapai maka *remoting* tidak akan dapat dilakukan oleh admin dan sistem akan meminta admin untuk melakukan pengecekan ulang terhadap IP ADDRESS dan PORT yang telah dimasukkan sampai kedua hal tersebut terpenuhi atau keluar dari aplikasi. Apabila IP ADDRESS dan PORT telah terpenuhi, maka *server* dapat melakukan proses peremotan terhadap *client* dan *client* berada dalam posisi listen.

Prosedur berikut menunjukkan sebuah contoh dari prosedur-prosedur yang terdapat dalam aplikasi *server* dimana terdapat 3 subjek utama yaitu : *scList*, *scBitM*, dan *scText* yang mempunyai masing-masing fungsi:

- scList*
menampilkan isi dari komputer *client* baik itu direktori, folder, maupun file yang dapat dilakukan proses manipulasi oleh admin baik itu copy, delete atau yang lainnya
- scBitM*
menampilkan file bitmap dalam komputer *client* dan dapat dimanipulasi oleh admin
- scText*
menampilkan file tulisan (*.txt) yang berada pada komputer *client*

```

Procedure TForm1.ClientSocket1Read(Sender:
TObject;
Socket: TCustomWinSocket);
var
Stringln : string;
Stream : TMemoryStream;
Terima : Integer;
begin
case StatusClient of
scKosong: //mencari data yang akan diterima
begin
Socket.ReceiveBuf(buffer, 5);
Stringln := Copy(Buffer, 1, 5);
if Stringln = 'TEXT!' then
StatusClient := scText
else if Stringln = 'BITM!' then
StatusClient := scBitM
else if Stringln = 'LIST!' then
StatusClient := scList
else if Stringln = 'ERROR' then
StatusClient := scSalah;
end;
scSalah: //menampilkan pesan
begin
ShowMessage(Socket.ReceiveText);
StatusClient := scKosong;
end;
scList: //baca daftar direktori
begin
ListBox1.Items.Text :=
Socket.ReceiveText;
StatusClient := scKosong;
end;
scText: //baca text file

```

```
begin
  with TFClientText.Create(Application) do
    begin
      Memol.Text := Socket.ReceiveText;
      Show;
    end;
    StatusClient := scKosong;
  end;
scBitM: //baca file Bitmap
  with TFClientBmp.Create(Application) do
    begin
      Stream := TMemoryStream.Create;
      Screen.Cursor := crHourGlass;
      try
        while True do
          begin
            Terima :=
Socket.ReceiveBuf(Buffer, sizeof(Buffer));
            if Terima <= 0 then
              break
            else
              Stream.Write(Buffer, Terima);
              // delay (200 miliseconds)
              Sleep(200);
            end;
            // reset and load the temporary file
            Stream.Position := 0;

Image1.Picture.Bitmap.LoadFromStream(Stream)
;
            finally
              Stream.Free;
              Screen.Cursor := crDefault;
            end;
            Show;
            StatusClient := scKosong;
          end;
        end; //case
      end;
    end;
```

Gambar 3. Flowchart Aplikasi Remote Sistem

Selain beberapa fungsi di atas, Aplikasi Remote Sistem ini juga dapat melakukan proses remote lain seperti:

- Reboot komputer *client*
- Open/Close CDROM *client*
- Chating
- General Information komputer *client*
- Dan implementasi lainnya yang membantu admin dalam mengontrol *client*

6. Kesimpulan

Dengan menggunakan Aplikasi Remote Sistem, didapatkan kemudahan bagi seorang administrator jaringan dalam manajemen komputer-komputer yang berada dalam jaringan tersebut.

Aplikasi ini dapat diimplementasikan di sebuah laboratorium dengan banyak komputer, departemen dalam sebuah organisasi atau perusahaan, dan lain sebagainya. Penggunaan aplikasi ini akan menghemat waktu dan tenaga yang dibutuhkan oleh seorang administrator untuk mengatur jaringan LAN.

Daftar Pustaka

- [1] Basalamah, A., et.al, *Applied Computer Internetworking: UNIX Integration to WAN*. 1995.
- [2] Masnfield, N., *Practical TCP/IP: Mendesain, Menggunakan, dan Troubleshooting Jaringan TCP/IP di Linux dan Windows*, Penerbit Andi, Yogyakarta. 2004.
- [3] Merz, M., Lamersdorf, W., *Generic Interfaces to Remote Applications in Open Systems*. Department of Computer Science, University of Hamburg. 1993.
- [4] Shetty, S., *Using WMI Scripting for System Administration*, POWER SOLUTIONS. 2004
- [5] Wyse Technology., *Wyse Thin Client Remote Access Solutions*. 2004.