

## RANCANG BANGUN APLIKASI REMOTE UNTUK ADMINISTRASI LAN

**Ryan Aniestyo Hermoko, Fathul Wahid, M. Andri Setiawan**

*Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia*

*Jl. Kaliurang Km. 14 Yogyakarta 55501*

*Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148*

*E-mail: fathulwahid@fti.uui.ac.id, andri@fti.uui.ac.id*

### ABSTRAK

Penggunaan LAN (Local Area Network) sangat diperlukan dalam sebuah ruang lingkup kerja, misalnya di perkantoran. Kebutuhan pertukaran data secara cepat dan simultan dalam jumlah besar adalah salah satu faktor yang menyebabkan pentingnya peran jaringan komputer dalam sebuah ruang lingkup kerja. Tidak jarang dalam sebuah jaringan komputer, suatu sumber daya dipakai secara bersama-sama untuk menghemat biaya dan sumber daya yang terbatas. Untuk itu diperlukan sebuah aplikasi yang dapat digunakan untuk mengontrol komputer dari jarak jauh dalam sebuah LAN, dan mengawasi setiap aktivitas yang dilakukan karyawan di komputernya masing-masing agar tidak terjadi penyalahgunaan komputer dan sumber daya kantor lainnya.

Penelitian ini mencoba mengembangkan solusi yang lebih mudah dan sederhana dalam melakukan pengawasan yang dilakukan seorang administrator dalam sebuah LAN perkantoran untuk melakukan pengawasan dan pengontrolan terhadap komputer lain yang terdapat di jaringannya.

Penelitian ini telah menghasilkan sebuah aplikasi yang membantu administrator jaringan untuk melakukan pengawasan dalam jaringan komputer, diantaranya adalah penghematan waktu dan tenaga. Pengembangan aplikasi ini juga berhasil menyempurnakan kekurangan aplikasi Pengendali PC Jarak Jauh pada LAN.

*Kata kunci: LAN, Remote, Remote Administrator, Pop Remote, Keylogger*

### 1. PENDAHULUAN

Tidak jarang dalam sebuah jaringan komputer, suatu sumber daya dipakai secara bersama-sama untuk menghemat biaya dan sumber daya yang terbatas. Misalnya sumber daya *proxy server* atau *gateway* yang digunakan bersama-sama untuk menghemat *IP Public* agar dapat melakukan *browsing* di internet. Setiap kegiatan yang dilakukan oleh sebuah komputer di dunia internet akan dikenali sebagai komputer *gateway*, bukan komputer yang melakukan koneksi internet. Dikarenakan memiliki sifat yang berbagi maka sebuah sumber daya yang dipakai bersama-sama seperti *gateway* tidak boleh dipakai secara sembarangan.

Dari contoh yang telah dikemukakan diatas, maka diperlukan sebuah aplikasi yang dapat digunakan untuk mengontrol komputer dari jarak jauh dalam sebuah LAN. Misal pada suatu kantor, LAN dapat digunakan untuk mengawasi

setiap aktivitas yang dilakukan karyawan di komputernya masing-masing agar tidak terjadi penyalahgunaan komputer dan sumber daya kantor lainnya.

Penelitian ini mencoba mengembangkan solusi yang lebih mudah dan sederhana dalam melakukan pengawasan yang dilakukan seorang *administrator* dalam sebuah LAN perkantoran untuk melakukan pengawasan dan pengontrolan terhadap komputer lain yang terdapat di jaringannya.

## 2. RUMUSAN MASALAH

Bagaimana merancang dan mengimplementasikan suatu aplikasi perangkat lunak yang dapat mengendalikan, memantau tampilan *desktop*, dan memantau ketukan *keyboard* komputer lain dari jarak jauh.

## 3. BATASAN MASALAH

Untuk menghindari meluasnya ruang lingkup yang dibahas pada penelitian ini, maka diberikan batasan masalah sebagai berikut:

1. Percobaan terhadap aplikasi ini hanya akan dilakukan dalam sebuah jaringan LAN.
2. Aplikasi ini terbagi 2, yaitu aplikasi *client* dan aplikasi *server*.
3. Aplikasi dijalankan dalam sistem operasi Windows.
4. Perangkat lunak yang akan digunakan untuk membangun aplikasi ini adalah Borland Delphi 7.
5. Antarmuka perangkat lunak akan dikembangkan dalam bahasa Inggris karena alasan banyaknya istilah asing yang bila dialih-bahasakan ke bahasa Indonesia akan menimbulkan kerancuan.
6. Ketidakstabilan dan kegagalan aplikasi yang disebabkan karena cacat ataupun *bug* yang terdapat pada perangkat keras, *compiler*, dan sistem operasi merupakan kesalahan yang terlepas dari pengembangan aplikasi.

## 4. PROTOKOL TCP/IP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah protokol standar yang digunakan di dunia internet. Dalam protokol TCP/IP, terdapat tiga protokol di level yang lebih tinggi lagi, yaitu TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), dan ICMP (*Internet Control Message Protocol*) (S'to, 2004).

TCP adalah protokol dalam dunia jaringan komputer yang digunakan untuk melakukan komunikasi pertukaran data secara *reliable*. Metode komunikasi TCP mengutamakan terbentuknya sebuah ikatan antara dua buah komputer terlebih dahulu, baru setelah itu komunikasi data dapat dilakukan. Hal ini lebih dikenal dengan istilah *connection oriented*. Sebuah komputer yang hendak mengirim data ke komputer lain melalui protokol TCP harus memastikan terlebih dahulu bahwa komputer tersebut dapat dijangkau dan tidak terdapat masalah dalam koneksi tersebut, baru kemudian data dikirimkan (S'to, 2004).

Hal ini tidak terdapat dalam protokol UDP, dimana data yang dikirim tidak tergantung dari konektifitas antara kedua komputer. Istilah yang sering digunakan adalah *connectionless*. Sedangkan ICMP biasanya digunakan untuk melakukan

pemeriksaan koneksi antar dua komputer dengan mengirim pesan ke komputer tersebut, dan menunggu balasan dari komputer tersebut. Pesan yang dikirim disebut *echo* dan balasannya disebut *echo-reply*. Contoh dari penggunaan protokol ICMP ini adalah aplikasi ping.

## 5. PERANCANGAN SISTEM

Aplikasi ini terdiri dari dua bagian, yaitu *client* dan *server*, dan dikembangkan dengan menggunakan bantuan *compiler* Borland Delphi 7 dengan bantuan fungsi-fungsi dari Windows API dan registry Windows. Untuk implementasi jaringan, digunakan protokol TCP agar dapat terbangun komunikasi dua arah yang *established* antara aplikasi *client* dan *server*.

### 5.1 Analisis Kebutuhan Sistem

Aplikasi yang akan dibangun terdiri dari dua aplikasi. Yaitu aplikasi server dan aplikasi client. Aplikasi client dijalankan di komputer *remote*, yaitu komputer yang dipantau dan dikontrol. Sedangkan aplikasi server dijalankan oleh admin jaringan tersebut.

Untuk mempermudah pembuatan aplikasi, maka diperlukan mekanisme pemecahan permasalahan menjadi bagian-bagian yang lebih kecil sehingga mempermudah dalam pengembangan dan pengecekan kesalahan.

Pembuatan kedua aplikasi ini dibagi dalam beberapa bagian sesuai tugas dan peranan komponen-komponen tersebut.

1. Komponen untuk menangani pencarian *port* yang digunakan oleh server socket.
2. Komponen untuk menangani koneksi utama antara *client* dan *server*.
3. Komponen untuk menerima dan mengirim *file*.
4. Komponen untuk menerima hasil *screen capture* dari *server*.
5. Komponen untuk menangani tampilan antar muka.

### 5.2 Analisis Kebutuhan Input

Masukan dari aplikasi ini adalah alamat IP atau *hostname* komputer *server* yang akan dikontrol dan dipantau aktifitasnya. Sedangkan untuk proses-proses setelah koneksi terbentuk, masukan yang dibutuhkan adalah sebagai berikut:

1. Untuk membaca ukuran kapasitas tiap *drive*, yang dibutuhkan adalah nama *drive* tersebut yang dilambangkan dengan huruf kapital.
2. Untuk menutup *window* sebuah aplikasi yang berjalan di *server*, yang dibutuhkan adalah *caption window* tersebut.
3. Untuk menampilkan isi dari sebuah direktori di server, yang dibutuhkan adalah alamat lengkap *path* direktori tersebut.
4. Untuk melakukan fungsi-fungsi standar windows explorer di komputer *server*, yang dibutuhkan adalah nama *file* atau *folder* beserta *path*-nya.

### 5.3 Analisis Proses

1. Pengidentifikasian komputer *server*.

2. Aplikasi *client* mengirim perintah pada aplikasi *server* yang menunggu perintah.
3. Aplikasi *server* melakukan proses yang diperintahkan.
4. Aplikasi *server* mengirimkan hasilnya pada aplikasi *client*.

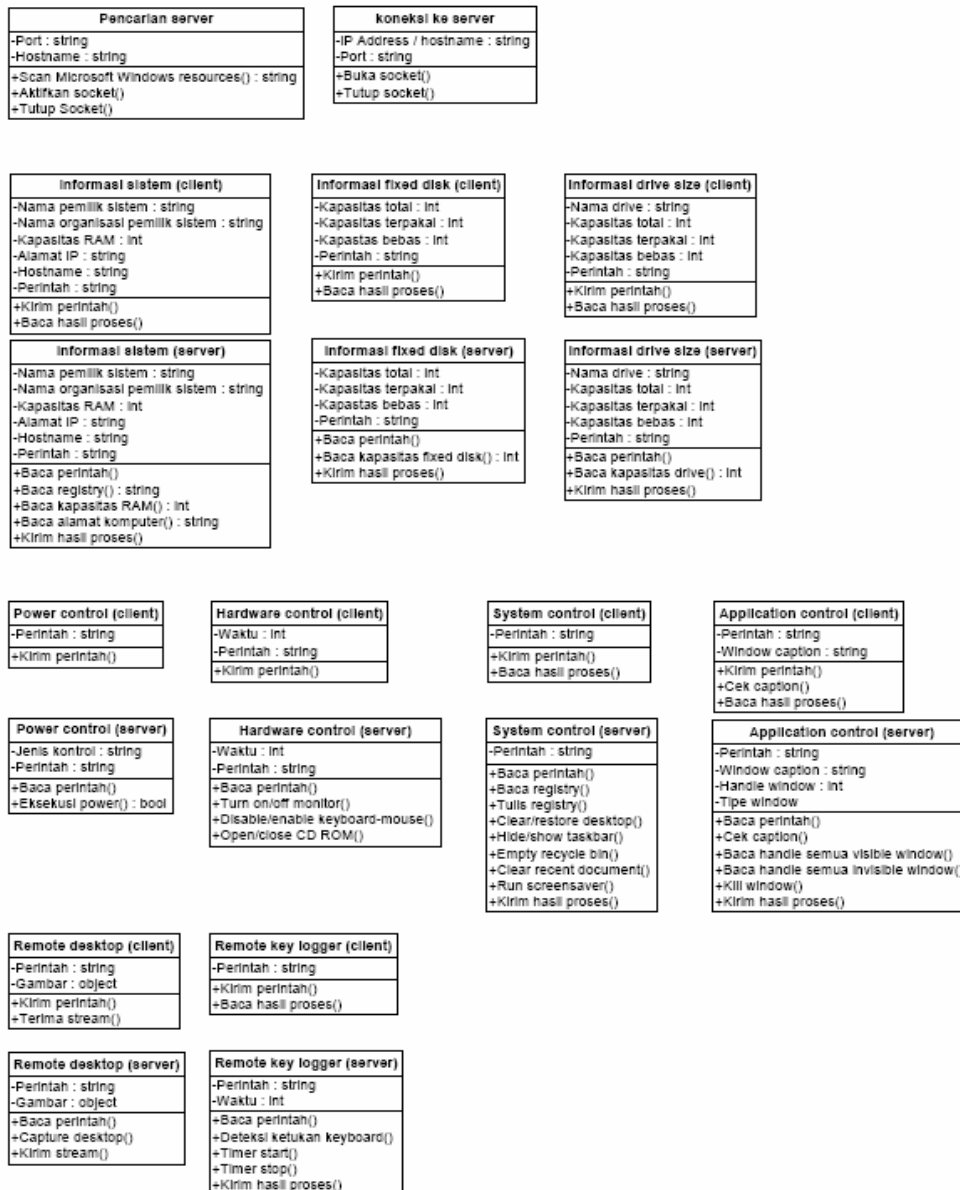
#### 5.4 Analisis Output

Output yang dapat diberikan oleh sistem antara lain:

1. Daftar semua komputer yang telah menjalankan aplikasi *server*.
2. Informasi sistem, yaitu berisi informasi versi sistem operasi, pemilik dan organisasi, RAM yang digunakan, *hostname* dan alamat IP, ukuran *disk* beserta informasi ukuran ruang yang tersedia, yang sudah digunakan, dan yang masih belum digunakan.
3. Kontrol, yaitu terdiri dari pengontrol *power* (*shut down*, *restart*, *stand by*, dan *log off*), pengontrolan *hardware* (*monitor*, *keyboard*, *mouse*, dan *CD ROM*), pengontrol sistem (menon-aktifkan *Task Manager*, *Display Properties* dan *System Properties*, membersihkan *desktop*, menyembunyikan *taskbar*, mengosongkan *recycle bin*, membersihkan *recent document*, dan menjalankan *screen saver*), Pengontrol aplikasi (menampilkan aplikasi yang tersembunyi dan yang tidak tersembunyi, beserta fasilitas untuk menghentikan aplikasi tersebut).
4. Mata-mata, yaitu terdiri dari fasilitas untuk memantau tampilan monitor komputer *server*, dan ketukan *keyboard* yang dilakukan di komputer *remote*.
5. Chat, yaitu terdiri dari aplikasi untuk *chatting* dengan komputer *server*, dan pengiriman pesan *pop-up* (*warning*, *info*, dan *stop*).
6. Explorer, yaitu aplikasi untuk *browsing* komputer *server*, beserta fasilitas untuk membuat dan menghapus *folder*, *download* dan *upload file* ke komputer *remote*, mengeksekusi atau membuka *file* di komputer *remote* dengan aplikasi *default-nya*, dan fungsi-fungsi standar lain (*cut*, *copy*, *paste*, *delete*, dan *rename*).
7. Pencatatan setiap aktifitas yang sudah dilakukan dalam sebuah *log file* yang berupa *file* teks.
8. Penyimpanan dan memasukkan daftar komputer *remote* ke dalam sebuah file teks.

#### 5.5 Perancangan Diagram Kelas

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan obyek beserta hubungannya satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Beberapa *class diagram* untuk aplikasi yang akan dibuat ditunjukkan dalam Gambar 1 (Dharwiyanti, S., dan Wahono, R.S., 2006).



Gambar 1. Class diagram

## 6. HASIL DAN PEMBAHASAN

### 6.1 Asumsi-asumsi yang digunakan

Dalam pengembangan aplikasi ini, terdapat beberapa asumsi yang diberikan sebelumnya, antara lain:

1. Aplikasi ini akan digunakan di lingkungan LAN dalam perkantoran dan sejenisnya. Diasumsikan aplikasi *server* telah terpasang dan berjalan di setiap *host*. Sedangkan aplikasi *client* terpasang di komputer admin, atau dimanapun

administrator jaringan ingin menjalankannya, asalkan masih dalam jangkauan jaringan tersebut.

2. Aplikasi ini menggunakan protokol TCP dengan memanfaatkan *port* 23000 untuk pencarian *server*, *port* 23001 untuk pengiriman dan penerimaan perintah berupa teks, *port* 23099 untuk pengiriman *file*, *port* 3250 untuk pengiriman *screen shot desktop*. Diasumsikan *port-port* tersebut belum digunakan.
3. Aplikasi ini dibatasi hanya dapat melakukan koneksi satu-satu. Artinya satu *server* hanya dapat melayani satu *client*, dan satu *client* hanya dapat mengakses satu *server*.

## 6.2 Batasan pengembangan aplikasi

Batasan-batasan yang diberikan dalam pengembangan perangkat lunak ini adalah sebagai berikut:

1. Aplikasi ini berjalan pada sistem operasi Microsoft Windows.
2. Aplikasi ini tidak dirancang untuk mengakses komputer yang berada dibalik NAT (*Network Address Translation*).
3. Tidak terdapat penanganan kesalahan apabila *port* 23000, 23001, 23099 atau 3250 sudah digunakan oleh aplikasi lain.
4. Aplikasi *client* hanya dapat mengakses sebuah *server* pada satu waktu karena keterbatasan sumber daya.
5. Aplikasi *server* hanya dapat diakses oleh satu *client* pada satu waktu karena alasan keterbatasan sumber daya.
6. Fitur *remote key logger* dan *remote desktop* bila dijalankan bersamaan tidak akan berjalan optimal karena pada saat komputer *server* melakukan *screen shot*, ia tidak dapat melakukan pengecekan tombol apa yang ditekan pada *keyboard*. Untuk itu akan digunakan sebuah *message box* peringatan.
7. Tidak terdapat fasilitas penggunaan *password* untuk menjalankan, menghentikan, atau mematikan *server*.
8. Tidak terdapat fasilitas penggunaan *password* sebagai upaya pencegahan penggunaan oleh pihak lain yang sama-sama ingin memanfaatkan aplikasi *server*.
9. Tidak terdapat penanganan *permission user* yang *login*, terhadap *file* atau *folder* yang berhak diakses.
10. Tidak ada fasilitas pengontrolan *mouse* pada fitur *remote desktop*.

## 6.3 Implementasi

Penelitian ini akan diimplementasikan dalam sebuah perangkat lunak yang bernama Pop Remote. Pemilihan nama ini tidak didasarkan pada hal apapun, hanya dikarenakan kebiasaan penulis yang menggunakan kata 'pop' disetiap pengembangan perangkat lunak.

Secara umum, implementasi aplikasi ini adalah penggabungan beberapa fungsi yang dikelompokkan menjadi sebuah aplikasi baru yang dapat digunakan seorang admin jaringan untuk melakukan kontrol pada setiap komputer di jaringannya. Implementasi aplikasi ini ditampilkan dalam antarmuka dua aplikasi pendukung, yaitu *client* dan *server* yang dijabarkan sebagai berikut:

## 1. Aplikasi client

Aplikasi *client* memiliki beberapa *form*, yaitu sebuah *form* utama, lima *form* pendukung, sebuah *form about*, dan beberapa *form* pelengkap.

### a. Form utama

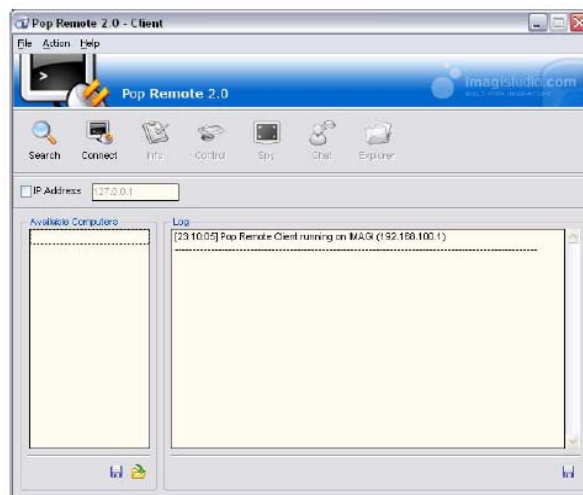
*Form* ini adalah *form* utama aplikasi *client*. Tampilan awal aplikasi *client*, sebelum melakukan aktifitas apapun termasuk pencarian komputer *server*, sehingga *log* pada aplikasi ini masih kosong, ditunjukkan pada Gambar 2.

### b. Form info

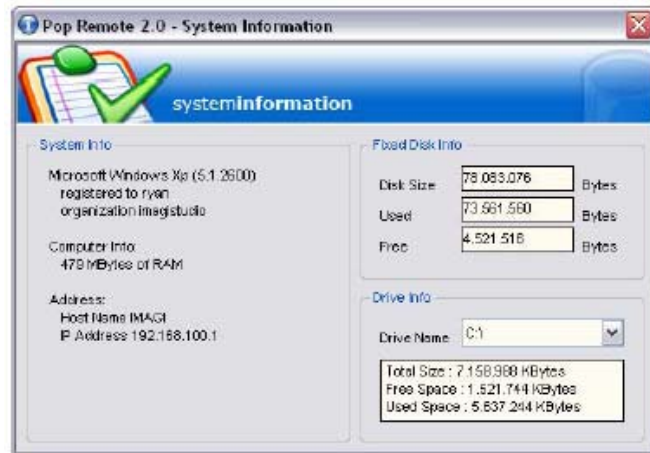
*Form* ini muncul sebagai *form modal*, yaitu *form* yang harus ditutup dahulu bila ingin menjalankan *form* utama. *Form* ini berisi info tentang komputer *remote*. Yaitu terdiri dari versi sistem operasi, *owner* dan *organization*, serta besarnya RAM yang dimiliki. Terdapat juga fasilitas melihat info besarnya kapasitas *harddisk* dan fasilitas melihat besarnya kapasitas tiap *drive* termasuk *removable drive*. *Form* info tampak pada Gambar 3.

### c. Form control

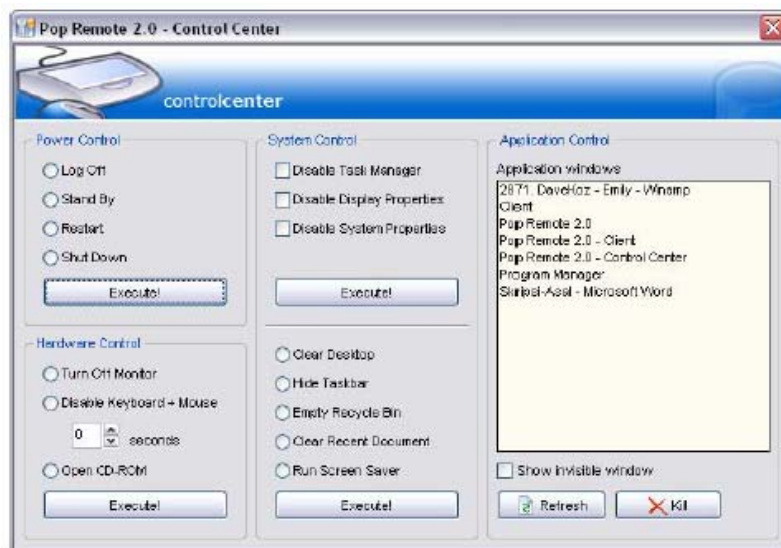
*Form* ini dibagi menjadi 4 bagian utama. Bagian pertama adalah Power Control yang berisi fasilitas untuk melakukan *Log Off*, *Stand By*, *Restart*, dan *Shutdown* pada komputer *server*. Bagian kedua adalah Hardware Control, yang berisi fasilitas untuk mematikan/menyalakan monitor, menonaktifkan *keyboard* dan *mouse* dalam selang waktu tertentu, dan membuka/tutup *CD-ROM drive*. Bagian ketiga adalah System Control, yang berisi fasilitas untuk melakukan pemblokiran terhadap Task Manager, Display Properties, dan System Properties. Juga terdapat fasilitas untuk membersihkan desktop dari semua *icon*, menyembunyikan *taskbar*, mengosongkan *recycle bin*, membersihkan daftar dokumen terakhir yang dibuka, dan menjalankan *screen saver*. Bagian keempat adalah Application Control yang berisi daftar aplikasi yang berjalan di komputer *server*, baik yang terlihat maupun yang tersembunyi, beserta fasilitas untuk menghentikan aplikasi tersebut.



Gambar 2. Form utama



Gambar 3. Form info



Gambar 4. Form Control

d. *Form spy*

*Form* ini terdiri dari dua bagian utama, yaitu Remote Desktop dan Remote KeyLogger. Remote Desktop digunakan untuk memantau aktifitas tampilan desktop komputer *server*. Sedangkan Remote KeyLogger digunakan untuk membaca ketukan *keyboard* yang terjadi di komputer *server*.

e. *Form chat*

*Form* ini digunakan untuk melakukan *chatting* dengan *user* komputer *server*. Selain itu *form* ini juga digunakan untuk mengirim pesan *pop-up* pada komputer *server*. *Form chat* tampak pada Gambar 5.

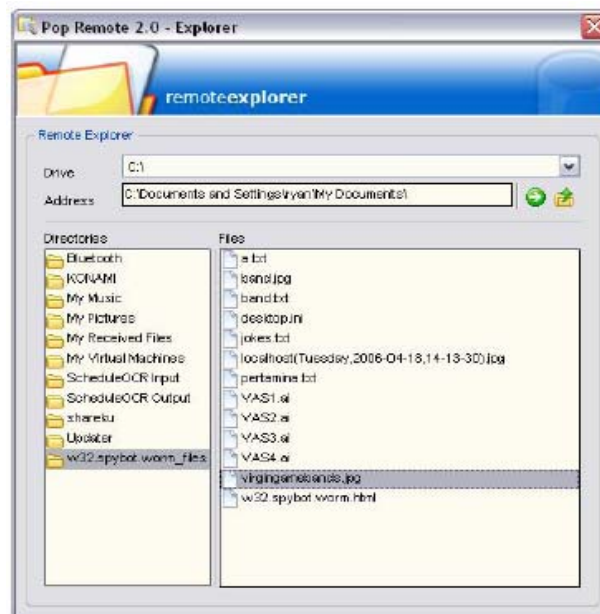


f. *Form explorer*

*Form* ini digunakan seperti layaknya Windows Explorer pada sistem operasi Windows. Digunakan untuk melakukan eksplorasi setiap media penyimpanan yang ada di komputer *server*, baik itu *fixed disk*, maupun *removable disk*. *Form* ini dapat digunakan untuk membuat dan menghapus direktori. Melakukan *copy*, *cut*, *paste*, *delete*, *rename* pada komputer *remote*. Fasilitas ini juga dapat melakukan *download* dan *upload* dari dan ke komputer *remote*. Selain itu dapat juga melakukan pengeksekusian *file* pada komputer *server* dengan aplikasi *default*-nya. *Form explorer* tampak pada Gambar 6.



Gambar 5. Form chatting



Gambar 6. Form explorer

## 2. Aplikasi server

### a. Form options

Form ini muncul bila user menekan *hot-key* ALT+F1. Berisi pilihan untuk menjalankan, menghentikan, dan mematikan aplikasi *server*. Form option tampak pada Gambar 7.

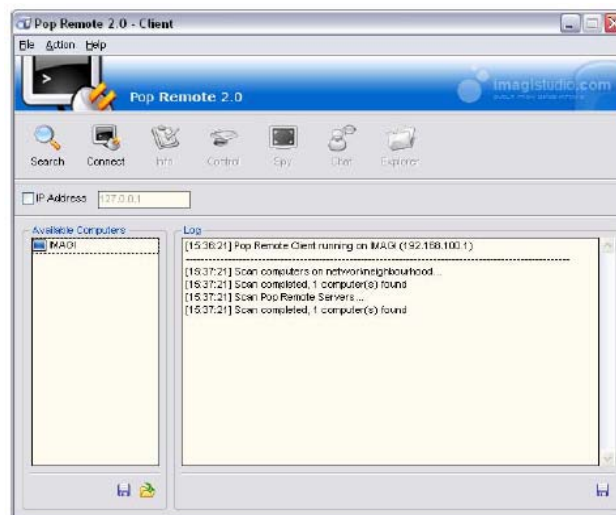
### b. Hasil pencarian server

Pada tahap implementasi, komputer yang mengeksekusi aplikasi *client* dan komputer yang mengeksekusi aplikasi *server* terhubung dalam satu LAN menggunakan kabel UTP dan konsentrator Hub. *Server* menunggu koneksi dari *client* pada *port* 23000 dan 23001.

Aplikasi *client* akan melakukan pencarian semua *resource* yang ada di Microsoft Windows Network, lalu melakukan pengecekan satu persatu di setiap *host*, apakah *port* 23000 terbuka. Bila tidak, maka akan dihapus dari daftar komputer target. Tampilan hasil pencarian server tampak pada Gambar 8.



Gambar 7. Form options



Gambar 8. Hasil pencarian server

Sedangkan *port* 23001 digunakan untuk mengirim dan menerima perintah dalam bentuk teks dari *server* ke *client* dan sebaliknya. Misalnya untuk mengirim dan menerima perintah untuk *chatting*. Pada implementasinya, *form* utama *server* dibuat tersembunyi dari *desktop*, *taskbar*, dan dari *task list* di Task Manager. Aplikasi *server* juga didaftarkan pada registry Windows pada key:

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

supaya berjalan otomatis bersamaan dengan *booting* sistem operasi Windows.

*Form chat* dan *form options* juga tersembunyi. Untuk *form chat*, akan otomatis muncul bila aplikasi *server* menerima perintah untuk memunculkan *form chat* ini dari aplikasi *client*.

Sedangkan pesan *pop-up* juga akan muncul di komputer *server* bila aplikasi *server* menerima perintah untuk menampilkan pesan *pop-up*. Pesan *pop-up* ini muncul dalam tiga macam jenis, yaitu *Warning Message*, *Info Message*, dan *Stop Message*. Tampilan dari pesan *pop-up* dapat dilihat pada Gambar 9.



Gambar 9. Pesan Pop-up

*Form options* muncul bila *user* di *server* menekan kombinasi tombol ALT+F1 yang sudah didaftarkan sebagai *Hot-Key*. *Form* ini berisi 3 opsi yang bisa dipilih, yaitu pilihan untuk menjalankan aplikasi, menghentikan aplikasi, atau menutup aplikasi. Setiap pilihan akan segera menyembunyikan kembali *form options*, kecuali pilihan Shut Down, yang akan menutup aplikasi secara keseluruhan.

Pada fitur Remote Desktop dan Remote Keylogger, kedua fitur ini tidak dapat berjalan bersamaan dengan baik dan *real-time*, hal ini dikarenakan pada saat komputer mengambil tampilan *desktop*, CPU akan bekerja 100%, sehingga tidak dapat melakukan pembacaan *keyboard* pada 1/1000 detik tersebut.

Selain itu pada Remote Desktop, gambar yang ditampilkan juga tergantung dari resolusi *desktop* komputer *server*, sehingga bila lebih kecil dari resolusi *desktop* komputer *client*, seluruh layar akan terlihat. Namun bila sama dengan atau lebih besar, diperlukan *scroll bar* untuk dapat melihat seluruh tampilan.

## 7. SIMPULAN

Dari hasil analisis dan perbandingan yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Aplikasi Pop Remote yang telah dikembangkan memberikan kemudahan bagi administrator jaringan untuk mengelola jaringannya.
2. Aplikasi Pop Remote lebih menguntungkan dan hemat bila dibandingkan dengan pengontrolan jaringan secara manual.

3. Aplikasi Pop Remote lebih memberikan kemudahan dalam penggunaannya karena dirasa lebih *user-friendly* daripada aplikasi Pengendali PC Jarak Jauh pada LAN.
4. Aplikasi Pop Remote selain dapat dipakai sebagai alat kontrol juga dapat digunakan sebagai alat mata-mata, sehingga penyalahgunaan aplikasi ini menjadi tanggung jawab moral masing-masing penggunanya.
5. Fitur *remote key logger* yang ditawarkan merupakan fitur yang relatif baru, sehingga dapat bersaing dengan aplikasi-aplikasi sejenis di masyarakat.

## 8. SARAN

Untuk pengembangan aplikasi ini di kemudian hari, dan karena banyaknya kekurangan yang terdapat di sana-sini, maka berikut adalah saran-saran yang diberikan oleh penulis terhadap pengembangan berikutnya.

1. Perlunya dikembangkan aplikasi sejenis yang berjalan antar sistem operasi.
2. Perlunya fitur pengontrolan *mouse* pada fasilitas *remote desktop*.
3. Perlunya fitur *key logger* yang terus berjalan meskipun aplikasi *client* tidak sedang mengakses *server*.
4. Pencarian komputer target dilakukan dengan melakukan pembacaan seluruh *resource* di Microsoft Windows Network menyebabkan pencarian memakan waktu lama karena harus melakukan *resolve* di setiap *host*. Untuk itu, perlu dikembangkan mekanisme pencarian yang baru.
5. Perlu penggunaan *password* pada aplikasi *server* agar aplikasi ini tidak di hentikan oleh *user* di komputer yang dikontrol dan agar aplikasi *server* hanya dapat diakses oleh administrator jaringan, bukan oleh pemburu *trojan*.
6. Perlu penanganan agar aplikasi ini tidak terdeteksi sebagai *trojan*.
7. Perlu dikembangkan aplikasi *client* yang menjalankan beberapa fitur secara bersamaan, misalnya fitur *chatting* bersamaan dengan fitur *spy* atau dengan fitur lainnya.
8. Perlunya dikembangkan aplikasi *client* yang mampu mengakses banyak aplikasi *server* sekaligus pada satu waktu.

## PUSTAKA

- Chandraleka, H. (2004). *Keylogger dan Pemrogramannya*. Yogyakarta: ANDI.
- Dharwiyanti, S., dan Wahono R.S. (2006). *Pengantar Unified Modelling Language (UML)*, <http://ilmukomputer.com/umum/yanti/yanti-uml.zip>, diakses pada tanggal 23 April 2006.
- Martina, I. (2002). *Pemrograman Internet dengan Delphi*. Jakarta: PT Elex Media Komputindo.
- Wahana Komputer. (2003). *Tip dan Trik Pemrograman Delphi 7.0*. Yogyakarta: ANDI.
- Nugroho, W. (2002). *Tip dan Trik Pemrograman Delphi*. Jakarta: PT Elex Media Komputindo.
- Nugroho, A. (2005). *Rational Rose untuk Pemodelan Berorientasi Objek*. Bandung: Informatika Bandung.
- Pamitrapati, D., dan Siahaan, K. (2000). *Trik Pemrograman Delphi*. Jakarta: PT Elex Media Komputindo.
- S'to. (2004). *Seni Teknik Hacking I*. Jakarta: Jasakom Publisihing.