

REMOTE REGISTRY EDITOR

Bambang Kelana Simpony

AMIK BSI Tasikmalaya

Jalan Tanuwijaya No.4 Empangsari, Tawang, Tasikmalaya 46113 Indonesia

bambang.bky@bsi.ac.id

Abstract - Permissions setting on a computer is necessary. This is an effort that is not easy to change the system configuration or settings changed by the user. With a network of computers, of course, permissions setting do not need to be done one by one manually. Because in a computer network course there are many collection of computers connected together. Permissions setting so that the system can use the client-server applications that access restrictions can be done effectively. As the implementation of client-server applications can be created using Visual Basic 6.0. This language has been able to access the socket on the Windows operating system, named Winsock API that supports TCP / IP. This protocol is widely used because of the reliability of client-server application programming. The application is divided into two main applications, namely the client and server program name with the name of the Receiver Sender program. Receiver function receives instructions restriction of access rights Sender and send reports to the Sender process execution. While Sender function to send instructions restrictions permissions via the Registry to the Receiver. And after the test, the application can block important features available in the Windows operating system. So it is expected that these applications can help in permissions setting on a computer network.

Keywords: Registry, Winsock API, TCP/IP, Socket, Protocol, Networking

Abstrak - Pengaturan hak akses di suatu komputer sangat diperlukan. Hal ini merupakan salah satu upaya agar konfigurasi sistem tidak mudah dirubah atau diganti pengaturannya oleh pengguna. Dengan adanya jaringan komputer, tentunya pengaturan hak akses tidak perlu dilakukan satu-satu secara manual. Karena dalam suatu jaringan komputer pastinya terdapat banyak kumpulan komputer yang saling terhubung. Sehingga pengaturan hak akses bisa menggunakan sistem aplikasi *client-server* agar pembatasan hak akses bisa dilakukan dengan efektif. Sebagai implementasi aplikasi *client-server* dapat dibuat dengan menggunakan bahasa pemrograman Visual Basic 6.0. Bahasa ini sudah bisa mengakses socket pada sistem operasi Windows yang diberi nama Winsock API yang sudah mendukung protokol TCP/IP. Protokol ini banyak dipakai karena kehandalannya dalam pemrograman aplikasi *client-server*. Aplikasi ini terbagi menjadi dua aplikasi utama, yaitu client dengan nama program *Receiver* dan *Server* dengan nama program *Sender*. *Receiver* berfungsi menerima instruksi pembatasan hak akses dari *Sender* dan mengirimkan laporan hasil proses eksekusinya ke *Sender*. Sedangkan *Sender* berfungsi untuk mengirim instruksi pembatasan hak akses melalui *Registry* ke *Receiver*. Dan setelah di uji, aplikasi ini dapat memblokir fitur-fitur penting yang ada di sistem operasi Windows. Sehingga diharapkan dengan adanya aplikasi ini dapat membantu dalam pengaturan hak akses di suatu jaringan komputer.

Kata Kunci : Registry, Winsock API, TCP/IP, Soket, Protokol, Jaringan

PENDAHULUAN

Perkembangan teknologi jaringan komputer meningkat dengan cepat, hal ini terlihat pada era tahun 80-an jaringan komputer masih merupakan teka-teki yang ingin dijawab oleh kalangan akademisi, dan pada tahun 1988 jaringan komputer mulai digunakan di universitas-universitas, perusahaan - perusahaan, sekarang memasuki era millennium ini terutama *world wide internet* telah menjadi realita sehari-hari jutaan manusia di muka bumi ini.

Salah satu manfaat yang bisa kita rasakan dengan adanya teknologi jaringan komputer adalah semakin lancarnya kegiatan komputerisasi, yang dulunya suatu model komputer tunggal yang melayani seluruh tugas-tugas komputasi suatu organisasi kini telah diganti dengan sekumpulan komputer yang terpisah-pisah akan tetapi saling berhubungan dalam melaksanakan tugasnya.

Sekumpulan komputer tersebut biasanya selalu dipantau oleh seorang admin untuk mengawasi kondisi komputer agar bisa bekerja secara normal baik dari segi perangkat kerasnya (*hardware*) maupun pada perangkat lunaknya (*software*).

Namun terkadang adanya perubahan *setting*-an pada komputer tanpa sepengetahuan admin menyebabkan kegiatan yang berhubungan dengan proses komputerisasi bisa terganggu.

Pengubahan *setting*-an dan gangguan-gangguan pada komputer dapat dikurangi dengan cara membatasi hak akses terhadap penggunaan pada setiap komputer yang ada dalam suatu jaringan, maka dari itu dilakukanlah penelitian untuk membuat sebuah program yang dirancang untuk seorang admin agar dapat memelihara kondisi komputer-komputer yang ditanganinya tetap dalam kondisi yang baik, sehingga kegiatan komputerisasi bisa dilakukan lebih efektif dan efisien.

Untuk mendukung pembuatan program tersebut menggunakan Microsoft Visual Basic 6 karena sudah

terdapat objek yang dinamakan WinSock (*Windows Socket*) yang merupakan komponen utama pembuatan program dalam penelitian ini.

Pada penelitian ini dibatasi pembahasan pada masalah berikut:

1. Memulai komunikasi antara aplikasi *Sender* dengan *Receiver*.
2. Proses pengiriman instruksi dari *Sender* ke *Receiver*.
3. Cara mengakhiri komunikasi aplikasi *Sender* dan *Receiver*.

Tujuan dari penelitian yang hendak dicapai adalah memberikan kemudahan bagi admin jaringan dalam mengelola dan memelihara kondisi komputer nya agar selalu dalam kondisi optimal. Hasil penelitian ini bisa diterapkan oleh admin jaringan untuk mengamankan seperangkat komputer dari segi *software*, sehingga dapat terhindar dari perubahan sistem yang tidak perlu dan tanpa sepengetahuan admin.

METODE PENELITIAN

Metode penelitian yang dilakukan yaitu menganalisa permasalahan yang telah didefinisikan dan dikelompokkan dalam tahap awal serta mengumpulkan dan menganalisa kebutuhan pendukung berupa *software* dan *hardware* untuk memecahkan masalah tersebut secara keseluruhan dengan rinci. Selanjutnya tahapan desain, dalam tahap ini hasil analisis dibuat dalam struktur program, dan rancangan antar muka keseluruhan secara rinci. Selanjutnya adalah mengaplikasikan pemodelan sistem (*design*) ke dalam bahasa pemrograman dengan *source code/syntax* yang sesuai dan bahasa yang digunakan dalam penelitian ini yaitu menggunakan bahasa pemrograman Visual Basic. Selanjutnya tahap dokumentasi yaitu melakukan dokumentasi/pengumpulan dokumen untuk semua bahan, data-data, modul-modul yang telah selesai dikerjakan dari tahap awal sampai dengan tahap akhir pengerjaan yang berhubungan dengan program yang dikerjakan.

PEMBAHASAN

Registry Editor pada sistem operasi Windows merupakan pusat dari segala konfigurasi sistem operasi baik itu perangkat lunak maupun perangkat keras. Pengubahan pengaturan di *registry* bisa mengakibatkan sistem tidak berjalan sebagaimana mestinya. Apalagi jika komputer tersebut berada dalam jumlah yang cukup banyak, maka admin harus bisa mengawasi dan menjaga komputer tersebut agar selalu berjalan sebagaimana mestinya.

Dalam penelitian ini dibuatlah aplikasi yang bisa mengubah pengaturan *registry* dari jarak jauh, sehingga pengaturan *registry* komputer bisa dilakukan secara efektif dan efisien. Aplikasi ini harus bisa menulis ke *registry* yang berisi pengaturan-pengaturan penting sistem.

Tentunya pada implementasi aplikasi ini harus memiliki lebih dari satu socket, yang mana socket yang satu digunakan untuk *listening* dan yang satu lagi untuk koneksi program.

Komunikasi socket ini membutuhkan sebuah port untuk berkomunikasi, layaknya sebuah gerbang untuk lalu lintas agar si A dapat pergi ketempat si B. *Port* ditentukan dengan sebuah angka. Dari sekian banyak jumlah *port* yang disediakan oleh windows, yaitu sebanyak 65536 *port*. *Port* 3030 digunakan sebagai jalur komunikasi. Karena *port* ini tidak digunakan secara internal oleh sistem Windows. Pada saat akan melakukan koneksi, objek *winsock* ini harus di-*load* terlebih dahulu.

Dan jika koneksi telah berakhir, hendaklah meng-*unload* objek yang tidak terpakai tersebut. Dengan adanya aplikasi ini maka sistem di komputer akan terus terjaga dan bisa berjalan dengan optimal.

Analisa Kebutuhan Sistem

Dalam penelitian ini dirancang dengan dua aplikasi utama yaitu aplikasi *Sender* dan aplikasi *Receiver*. Aplikasi *Receiver* harus di-*install* di semua komputer yang akan dibatasi hak akses

nya, sedangkan aplikasi *Sender* cukup di-*install* di satu komputer saja.

Aplikasi *Sender* harus bisa melakukan hal berikut ini, yaitu:

1. *Sender* harus bisa mengecek kartu jaringan yang ada di komputer apakah ter-*install* atau tidak.
2. IP Address di input secara manual, dan di aplikasi ini menggunakan format IP Address versi 4 (IPv4). Jika salah input, maka aplikasi akan menampilkan pesan error, lalu aplikasi akan tertutup otomatis.
3. Tombol "*Connect*" digunakan untuk mengkoneksikan ke aplikasi *Receiver*, jika *Receiver* belum aktif, maka *Sender* akan menampilkan pesan error dan langsung menutup aplikasi. Jika terkoneksi, maka tombol "*Connect*" akan berubah menjadi "*Disconnect*".
4. Selanjutnya tombol "*Enable*" dan "*Disable*" yang tadinya tidak aktif, maka akan aktif karena *Sender* sudah terkoneksi ke *Receiver*.
5. Selanjutnya memilih menu pembatasan hak akses yang akan di-*setting*, dan memilih tombol apakah "*Enable*" atau "*Disable*". Jika tidak memilih menu, maka akan tampil pesan error.
6. Jika instruksi berhasil di eksekusi oleh *Receiver*, maka akan tampil pesan bahwa pembatasan hak akses telah selesai diproses.
7. Jika akan mengakhiri koneksi dengan *Receiver*, maka cukup pilih tombol "*Disconnect*", dan aplikasi akan kembali seperti semula.

Aplikasi *Receiver* harus bisa melakukan hal berikut ini, yaitu:

1. Aplikasi bisa menampilkan IP Address dan Host Name yang sedang aktif, dan menampilkan informasi apakah ada koneksi *Sender* yang masuk.
2. *Receiver* harus bisa mengecek kartu jaringan yang ada di komputer apakah ter-*install* atau tidak.
3. *Receiver* harus bisa mendeteksi apakah *port* yang di-*setting* belum dipakai oleh aplikasi lain dan aplikasi *Receiver* ini tidak bisa diaktifkan lebih dari satu kali dalam waktu bersamaan.

4. Komputer *Receiver* harus bisa menerima instruksi yang dikirim *Sender* dan mengeksekusinya, selanjutnya mengirim laporan hasil eksekusi ke *Receiver*.

Rancangan program aplikasi dalam penelitian ini harus bisa melakukan proses berikut ini:

1. Sistem harus bisa input IP Address komputer yang akan diberi hak akses.
2. Sistem harus bisa mengecek apakah user salah input IP Address dan apakah *Receiver* sudah diaktifkan.
3. Sistem harus bisa menampilkan informasi apakah kedua aplikasi sudah terjadi koneksi.
4. Sistem harus bisa menentukan pengaturan hak akses mana yang akan dilakukan dan menampilkan hasil proses yang dikerjakan.

Berikut rancangan antar muka berdasarkan kebutuhan sistem:

Nama Program	Informasi Status Koneksi	Informasi Host Name dan IP Address yang aktif
	Tentang Program	

Gambar 3
Rancangan Form *Receiver*

Nama Program	Informasi Status Koneksi		Tombol Enable	Tombol Disable
	Input IP Address		Pilihan Hak Akses	
	Tombol Connect			
	Tentang Program	Tombol Close		

Gambar 4
Rancangan Form *Sender*

Sender

Error Warning: Connection is forcefully rejected

OK

Gambar 5
Rancangan Pesan *Error* jika *Receiver* tidak aktif

Sender

Error Warning: Authoritative answer: Host Not Found

OK

Gambar 6
Rancangan Pesan *Error* jika salah input IP Address

Sender

Error Warning: The attempt to connect timed out

OK

Gambar 7
Rancangan Pesan *Error* jika IP Address tidak ada di jaringan

Sender

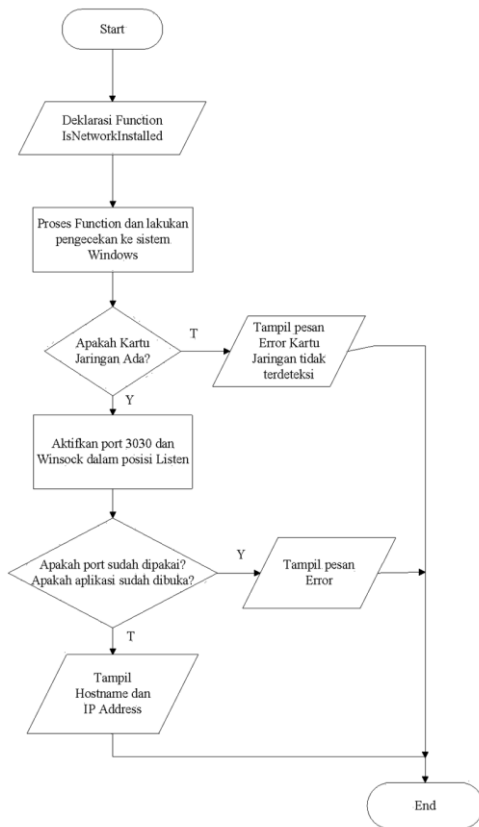
Report From Receiver : "Task Manager Can Not Be Opened"

OK

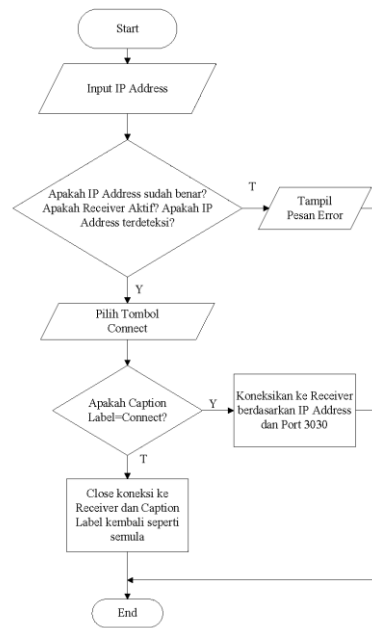
Gambar 8
Rancangan Pesan Jika *Sender* Berhasil Mengirim Instruksi Hak Akses

Program Flowchart

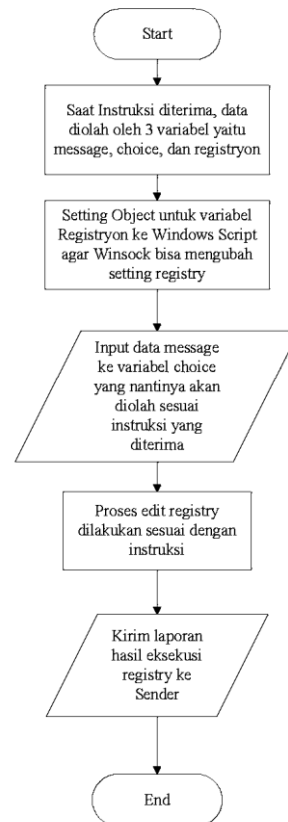
Berikut ini merupakan gambaran *flowchart* yang dijadikan acuan dalam pembuatan program aplikasi *Sender* dan *Receiver*:



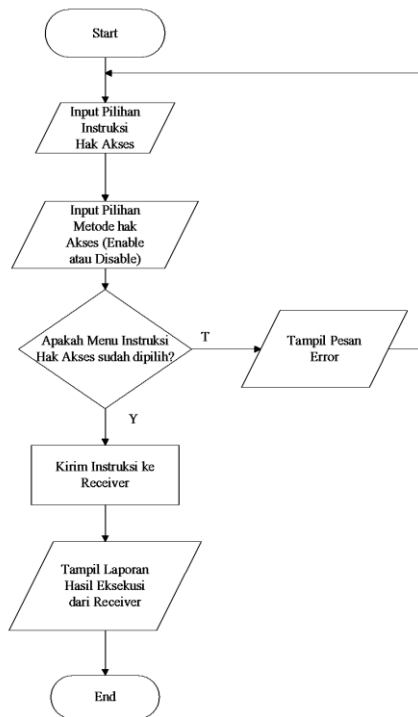
Gambar 9
Flowchart Aplikasi *Receiver*



Gambar 10
Proses Koneksi dari *Sender* ke *Receiver*



Gambar 11
Proses Saat *Receiver* Menerima Instruksi



Gambar 12
Proses Pengiriman Instruksi dari *Sender* ke *Receiver*

Dalam penerapannya, aplikasi ini membutuhkan sarana pendukung program baik itu dari segi software maupun hardware. Spesifikasi minimum perangkat keras pendukung program aplikasi ini adalah:

Processor :Intel Pentium 4 HT
(atau yang sudah mendukung Hyper Threading).

Harddisk :20 GB
RAM :256 MB
VGA :OnBoard (64 MB)
Jenis Koneksi :Boleh menggunakan LAN dengan jalur Wireless atau Kabel.
Monitor :CRT
Keyboard : Standar Keyboard 104 Keys
Mouse : PS/2+USB Compatible

Aplikasi ini dalam pembuatannya membutuhkan beberapa perangkat lunak diantaranya:

Sistem Operasi :Microsoft Windows XP SP2 (Spesifikasi Minimum)

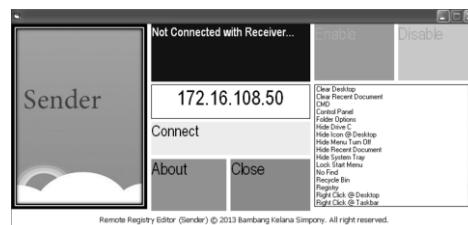
Bahasa Pemrograman :Microsoft Visual Basic 6.0
Komponen Utama :Winsock (Windows Socket)

Langkah awal untuk memulai menggunakan program ini yaitu dengan meng-*install* kedua aplikasi tersebut (*Receiver* di *install* di semua komputer yang akan dibatasi hak akses nya, sedangkan *Sender* di *install* cukup di satu komputer saja).

Setelah aplikasi di *install*, maka langkah selanjutnya adalah semua aplikasi *Receiver* harus dibuka supaya *Sender* bisa terkoneksi dengan *Receiver*. Cara koneksi nya adalah dengan input IP Address komputer yang akan diberi hak akses. Seperti yang ditampilkan di gambar berikut ini.

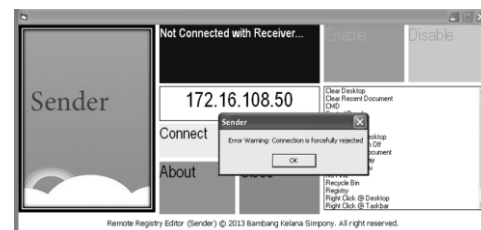


Gambar 13
Tampilan *Form Receiver*



Gambar 14
Tampilan form *Sender*

Jika *Receiver* belum berjalan, atau salah input IP Address maka akan terlihat seperti gambar berikut.



Gambar 15
Tampilan Error jika *Receiver* tidak aktif



Gambar 16
Tampilan *Error* jika Salah input IP Address

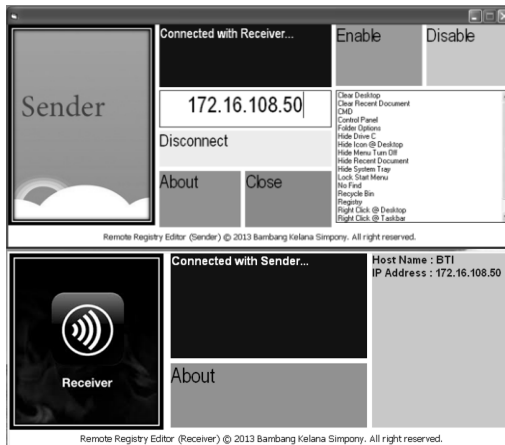


Gambar 19
Tampilan *Sender* Saat Menentukan Hak Akses



Gambar 17
Tampilan *Error* Jika IP Address Tidak Ada Di Jaringan

Sebagai bentuk keluarannya, setelah *Sender* bisa menghubungi *Receiver* maka pengiriman intruksi hak akses pun bisa dilakukan. Seperti yang ditampilkan pada gambar berikut.



Gambar 18
Tampilan *Receiver* dan *Sender* saat terkoneksi

Selanjutnya memilih hak akses mana yang akan diatur, lalu memilih apakah akan diaktifkan (*Enable*) atau dinonaktifkan (*Disable*), jika berhasil maka akan menampilkan output seperti gambar berikut ini.

PENUTUP

Kesimpulan

Aplikasi *socket* yang menggunakan TCP memerlukan pertukaran data dua arah yang *valid*. Ini ditandai dengan adanya perintah *listen* dan *connect* sehingga menjamin reliabilitas antar aplikasi dalam berkomunikasi.

Walaupun aplikasi *socket* ini dicontohkan dalam *environment* Windows, bukan berarti aplikasi *socket* yang menggunakan *winsock* tidak dapat berkomunikasi dengan aplikasi *socket* berbasis Unix/Linux, karena komunikasi tetap dapat terjadi selama aplikasi tersebut menggunakan protokol jaringan yang sama.

Berdasarkan analisis perencanaan dan pembahasan yang telah dilakukan dalam pembuatan aplikasi ini, dapat diambil kesimpulan bahwa aplikasi ini sudah dapat digunakan untuk membatasi beberapa menu penting di Windows, diantaranya Command Prompt, Task Manager, Folder Options, Run.

Saran

Berbicara mengenai komunikasi jaringan, ternyata erat kaitannya dengan keamanan jaringan. Port yang terbuka dengan keluar masuknya data, sebenarnya juga merupakan celah keamanan yang dapat dimanfaatkan pihak-pihak tertentu yang berniat buruk.

Karena itu, pada saat mengembangkan sebuah aplikasi *socket*, maka perlu dipertimbangkan pula *issue* keamanan jaringan dan sistem komputer yang digunakan. Jika data yang dikirimkan merupakan data yang sensitif dan rahasia, maka dapat dilakukan

enkripsi data ataupun tindak pengamanan lainnya.

Dalam komunikasi data didalam jaringan, maka perlu dipertimbangkan pula besarnya data yang lalu-lalang pada jaringan, baik dengan menggunakan TCP maupun UDP. Keduanya harus dipersiapkan untuk mampu menangani data yang besar jika memang pengguna aplikasi *socket* tersebut sangat luas.

Dan beberapa saran yang hendaknya dikembangkan untuk aplikasi ini yaitu:

1. Pengaturan nomor port hendaknya di setting secara manual, agar tidak konflik dengan aplikasi lain.
2. Aplikasi *Receiver* hendaknya berjalan secara otomatis (kecuali disimpan di folder start up secara manual) dan proses nya di Task Manager di enkripsi sehingga tidak mudah di *close*.
3. Ditambahkan lagi fitur atau menu yang bisa di blok lewat registry.
4. *Setting*-an bisa di broadcast secara langsung untuk semua *Receiver* yang aktif.
5. Informasi restart hendaknya ditampilkan, karena ada beberapa *setting*-an yang tidak langsung berubah.
6. Sistem harus dapat mengecek IP Address komputer yang aktif atau tidak.

TCP/IP Untuk Membangun Aplikasi Administrator.
Yogyakarta: STMIK AMIKOM

Raharjo, Budi, & Imam Heryanto, Arif Haryono. 2010. Mudah Belajar Java Edisi Revisi. Bandung: Informatika.

Sopandi, Dede. 2008. Instalasi dan Konfigurasi Jaringan Komputer. Bandung: Informatika.

Viva, Vygory. 2008. Trik Pemrograman Jaringan dengan Visual Basic 6. Yogyakarta: Gava Media

REFERENSI

Anharku. 2009. Flowchart. Diambil dari :
<http://ilmukomputer.org/wp-content/uploads/2009/06/anharku-flowchart.pdf>

Masya, Fajar dan Andrew Fiade. 2011. Socket Programming. Yogyakarta: Graha Ilmu.

Nope'x. 2008. DNA Windows. Diambil dari:
<http://ilmukomputer.org/wp-content/uploads/2008/07/nopex-dna-windows.pdf>

Nugroho, Agus Sapto. 2011. Implementasi Winsock Berbasis