

Laporan Penelitian

Implementasi Konsep Virtualisasi dalam Pengembangan Aplikasi Pemantauan dan Pengendalian Sumber Daya Komputer



disusun oleh:

Wilfridus Bambang Triadi Handaya, S.T., M.Cs

Bernard Renaldy Suteja, S.Kom., M.Kom

Fakultas Teknologi Informasi

Universitas Kristen Maranatha

Bandung

2009

LEMBAR IDENTITAS

1. Judul Penelitian: Implementasi Konsep Virtualisasi dalam Pengembangan Aplikasi Pemantauan dan Pengendalian Sumber Daya Komputer
2. Ketua/Penanggung Jawab Pelaksana Kegiatan Penelitian:
Nama (lengkap dengan gelar) : Wilfridus Bambang Triadi H, S.T., M.Cs.
NIK : 720248
Jabatan Akademik / Golongan : Asisten Ahli / III A
Fakultas / Jurusan : Universitas Kristen Maranatha
3. Jumlah Tim Peneliti : 2 orang
4. Lokasi Pelaksana Penelitian : Fakultas Teknologi Informasi
Universitas Kristen Maranatha
5. Lama Pelaksanaan : 2,5 bulan
6. Sumber Dana Penelitian : Universitas Kristen Maranatha
7. Biaya Penelitian : Rp. 7.595.000,-

Bandung, 5 Oktober 2009

Ketua/ Penanggung Jawab Pelaksana

Wilfridus Bambang Triadi H, S.T., M.Cs

Mengetahui

Dekan Fakultas Teknologi Informasi

Ketua LPPM

Radiant Victor Imbar, S.Kom., M.T.

Ir. Yusak Gunadi Santoso, MM.

LEMBAR PENGESAHAN

Judul Penelitian : Implementasi Konsep Virtualisasi dalam Pengembangan Aplikasi Pemantauan dan Pengendalian Sumber Daya Komputer

Peneliti : 1. Wilfridus Bambang Triadi Handaya, S.T., M.Cs.
2. Bernard Renaldy Suteja, S.Kom., M.Kom

Lokasi Pelaksana Penelitian: Fakultas Teknologi Informasi
Universitas Kristen Maranatha
Jl. Surya Sumantri no. 65
Bandung

Penelitian ini telah diselesaikan pada tanggal 5 Oktober 2009 sebagai salah satu perwujudan Tridharma Perguruan Tinggi Universitas Kristen Maranatha

Bandung, 5 Oktober 2009

Ketua Peneliti

Wilfridus Bambang Triadi Handaya, S.T., M.Cs

Dekan Fakultas Teknologi Informasi

Ketua LPPM

Radiant Victor Imbar, S.Kom., M.T.

Ir. Yusak Gunadi Santoso, MM.

DAFTAR ISI

LEMBAR IDENTITAS	ii
LEMBAR PENGESAHAN	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	ix
INTISARI	x
ABSTRACT	xi

I. Pendahuluan

1.1. Latar Belakang Masalah	1
1.2. Batasan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
1.5. Metode Penelitian	4

II. Landasan Teori

2.1. Konsep Remote Desktop.....	5
2.2. Konsep Virtualisasi	7
2.2.1. Pendekatan Virtualisasi	8
2.2.2. Tujuan Implementasi Virtualisasi	10

III. Analisis dan Perancangan Sistem

3.1. Analisis Sistem.....	12
3.2. Analisis Kebutuhan Aplikasi	14
3.2.1. Perspektif Perangkat Lunak	14
3.2.2. Fungsi Perangkat Lunak.....	15
3.2.3. Karakteristik Pengguna	15
3.2.4. Batasan-batasan.....	15
3.2.5. Kebutuhan Antarmuka Pengguna	16

3.2.6. Kebutuhan Antarmuka Perangkat Lunak.....	17
3.3. Analisis Fungsionalitas Aplikasi.....	18
3.3.1. Perancangan Arsitektur Perangkat Lunak	20
3.3.2. Data Flow Diagram (DFD) Level 0	22
3.3.3. Data Flow Diagram (DFD) Level 1	24
3.3.4. Data Flow Diagram (DFD) Level 2 dari Proses Inisialisasi Koneksi	26
3.3.5. Data Flow Diagram (DFD) Level 2 dari Proses Informasi Sistem.....	28
3.3.6. Data Flow Diagram (DFD) Level 2 dari Proses Akses Media Penyimpan	29
3.3.7. Data Flow Diagram (DFD) Level 2 dari Proses Kendali TaskManager.....	31
3.3.8. Data Flow Diagram (DFD) Level 2 dari Proses Perekaman Keyboard.....	33
3.3.9. Data Flow Diagram (DFD) Level 2 dari Proses Perekaman Tampilan Layar.....	35
3.3.10. Data Flow Diagram (DFD) Level 2 dari Proses Kendali Mouse & Keyboard.....	37

IV. Implementasi Sistem

4.1. Implementasi Sistem.....	39
4.2. Implementasi Aplikasi Pengendali	39
4.2.1. Pembahasan Fitur Inisialisasi Koneksi.....	39
4.2.2. Pembahasan Fitur Informasi Sistem.....	42
4.2.3. Pembahasan Fitur Akses Media Penyimpan	43
4.2.4. Pembahasan Fitur Kendali Task Manager	45
4.2.5. Pembahasan Fitur Perekaman Keyboard	47
4.2.6. Pembahasan Fitur Perekaman Tampilan Layar.....	48
4.2.7. Pembahasan Fitur Kendali Mouse & Keyboard.....	49

4.3. Implementasi Aplikasi Agen_Kendali	51
4.3.1. Fungsi Inisialisasi Koneksi	51
4.3.2. Fungsi Pendeteksi Aktifitas Keyboard.....	53
4.3.3. Fungsi Kendali Mouse & Keyboard	55
4.3.4. Fungsi Administrasi Sistem	57
V. Analisis Hasil	
5.1. Analisis Hasil	58
VI. Kesimpulan dan Saran	
6.1. Kesimpulan	71
6.2. Saran	72
DAFTAR PUSTAKA	73
LAMPIRAN	xii

DAFTAR GAMBAR

Gambar 2.1	Arsitektur Konseptual - Adaptasi Protokol Remote Desktop	6
Gambar 2.2	Arsitektur Platform Virtualization.....	9
Gambar 3.1	Arsitektur Sistem	13
Gambar 3.2	Perancangan Arsitektur Perangkat Lunak	21
Gambar 3.3	DFD Level 0 Remote Desktop	23
Gambar 3.4	DFD Level 1 Remote Desktop	25
Gambar 3.5	DFD Level 2 Proses Inisialisasi Koneksi	27
Gambar 3.6	DFD Level 2 Proses Informasi Sistem	28
Gambar 3.7	DFD Level 2 Akses Media Penyimpan.....	30
Gambar 3.8	DFD Level 2 Kendali Task Manager.....	32
Gambar 3.9	DFD Level 2 Perekaman Keyboard	34
Gambar 3.10	DFD Level 2 Proses Perekaman Tampilan Layar	36
Gambar 3.11	DFD Level 2 Proses Kendali Mouse & Keyboard.....	38
Gambar 4.1	Visualisasi Fitur Inisialisasi Koneksi	40
Gambar 4.2	Visualisasi Fitur Informasi Sistem	42
Gambar 4.3	Visualisasi Fitur Akses Media Penyimpan.....	44
Gambar 4.4	Visualisasi Fitur Kendali Task Manager	46
Gambar 4.5	Visualisasi Fitur Perekaman Keyboard	47
Gambar 4.6	Visualisasi Fitur Perekaman Tampilan Layar	49
Gambar 4.7	Visualisasi Fitur Kendali Mouse & Keyboard	50
Gambar 4.8	Visualisasi Aplikasi Agen_Kendali.....	52
Gambar 4.9	Visualisasi Form Konfigurasi alamat IP dan Port.....	52
Gambar 5.1	Implementasi pada sistem operasi Windows 98 SE.....	64
Gambar 5.2	Implementasi pada sistem operasi Windows Server 2003.....	64

Gambar 5.3	Implementasi pada sistem operasi Windows Vista 32 bit.....	65
Gambar 5.4	Implementasi pada sistem operasi Windows XP Professional SP2.....	65
Gambar 5.5	Implementasi pada sistem operasi Windows Vista 64 bit.....	66
Gambar 5.6	Kondisi Sistem setelah aplikasi diaktifkan	67
Gambar 5.7	Visualisasi eksekusi aplikasi agen_kendali dengan wine ...	69
Gambar 5.8	Visualisasi eksekusi aplikasi pengendali dengan wine	69
Gambar 5.9	Visualisasi komunikasi aplikasi pengendali dengan agen_kendali	70

DAFTAR TABEL

Tabel 5.1	Hasil Ujicoba Implementasi Aplikasi Pengendali.....	59
Tabel 5.2	Hasil implementasi Aplikasi Agen_Kendali.....	60
		1
Tabel 5.3	Implementasi aplikasi <i>control agent</i> pada varian SO Windows.....	62

INTISARI

Sistem yang berfungsi untuk memantau aktifitas pengguna telah dirancang dan diimplementasikan dengan tujuan untuk melakukan pengawasan yang memudahkan administrator sistem untuk mendapatkan informasi tentang penggunaan komputer di laboratorium atau instansi, yang kemudian dapat melakukan suatu tindakan terhadap pengguna yang ternyata terbukti melanggar loyalitas terhadap tata cara dan aturan penggunaan komputer publik.

Sistem terbagi dalam dua (2) jenis implementasi, yaitu pada sisi pengendali serta pada sisi agen_kendali. Komputer yang terinstall aplikasi pengendali berfungsi sebagai pemantau dan pengontrol aktivitas yang dilakukan oleh pengguna, dengan tersedianya antarmuka yang berisi daftar fitur yang dapat diakses untuk pengendalian aplikasi. Sedangkan pada komputer yang menjalankan aplikasi agen_kendali berfungsi untuk menerima perintah dari aplikasi pengendali (administrator) untuk selanjutnya mengeksekusi fungsi-fungsi yang telah disiapkan sebelumnya sebagai layanan, seperti melakukan perekaman terhadap tampilan layar monitor. Kemudian terdapat tambahan fungsi-fungsi lain seperti mengendalikan penggunaan keyboard dan mouse, pengaksesan media penyimpan data baik itu pada harddisk, cdrom ataupun perangkat eksternal, pengendalian taskmanager, hingga mengakses informasi dari aktifitas penekanan keyboard.

Hasil implementasi yang dilakukan membuktikan bahwa efektifitas aplikasi dalam melakukan pemantauan dan pengamatan terhadap kinerja dari pengguna secara spesifik. Informasi yang dihasilkan dari aktifitas pemantauan ini, menjadi suatu bukti otentik yang diperlukan untuk melakukan suatu reaksi atas aksi dari kegiatan yang tidak seharusnya dilakukan pada komputer publik.

Kata kunci : *administrator, pengendali, Internet*

ABSTRACT

A system to monitor user activities has been designed and implemented with the purpose to perform monitoring and easing the work of the system's administrator in collecting information about computer usage in laboratories or agencies. This information is then can be used as the basis to perform necessary actions against users who have been proven to violate their loyalties to the rules and regulations of using public computers.

The system is divided into two implementations, which are the controller side and the agent-controlled side. One computer, installed with the controller application, will function to monitor and control activities performed by users. This controller side computer is equipped with interface to access the available features of the controller. The other computers, installed with the agent-controlled applications, will receive the commands from the controller computer (administrator) and then execute the functions corresponding to these commands, such as recording the monitor screen. Other additional features are added such as controlling the use of keyboard and mouse, accessing storage medias (hard disk, CDROM, or external devices), controlling of task manager, and also accessing information about keyboard usage.

The implemented system shows the effectiveness of the application to perform monitoring the specific activities of users. These information can be used as authentic proof to perform necessary reaction to the prohibited actions that should not used in public computers.

Keyword: *administrator, controller, Internet*

I. PENDAHULUAN

1.1. Latar Belakang Masalah

Latar belakang pemikiran dari dibentuknya sistem pemantauan jarak jauh adalah bagaimana membentuk sistem komunikasi *client/server* untuk melakukan pemantauan komputer *server* dari jarak jauh dengan menerapkan konsep pemrograman *client/server* serta memanfaatkan protokol TCP/IP sebagai sarana pengiriman data informasi atau isyarat kendali melalui jaringan komputer. Sistem ini dibuat supaya memudahkan bagi seorang administrator jaringan untuk mendapatkan informasi tentang penggunaan komputer di laboratorium atau instansi, yang kemudian dapat melakukan suatu tindakan terhadap pengguna yang ternyata terbukti melanggar loyalitas terhadap penggunaan komputer publik tersebut.

Peneliti tertarik untuk membangun suatu aplikasi yang dapat mengendalikan keyboard dan mouse, merekam tampilan layar ke dalam file, kemudian memantau aktifitas penggunaan suatu komputer dari pencatatan setiap penekanan tombol pada *keyboard*, sehingga administrator mendapat informasi bahkan bukti terhadap loyalitas penggunaan komputer publik tersebut. Aplikasi ini terbagi menjadi dua jenis, yaitu: aplikasi client untuk pengendali dan aplikasi server.

1.2. Batasan Masalah

Penulis membatasi dalam pengimplementasian program ini, yaitu:

1. Perangkat lunak yang dikembangkan berbasis *client/server*. Perlu diperhatikan *client* merupakan komputer yang menjalankan aplikasi pengendali dan *server* merupakan komputer yang menjalankan aplikasi agen_kendali serta menjadi tujuan dari pemantauan.
2. Aplikasi yang dikembangkan berfungsi untuk melakukan pemantauan layer monitor dari pengguna, mengendalikan keyboard dan mouse, serta memantau aktifitas penekanan tombol keyboard.
3. Perangkat lunak *remote desktop* dijalankan pada komputer dengan basis sistem operasi Microsoft Windows 98 Second Edition (SE), Windows XP, Windows Server 2000/2003, Windows Vista 32 bit dan 64 bit, dimana perangkat lunak menggunakan layanan dari *file library* Windows.
4. Perangkat lunak *remote desktop* memerlukan jaringan komputer lokal (LAN) untuk dapat berkomunikasi antara komputer pengendali dengan agen_kendali.
5. Sistem dibangun menggunakan perangkat pemrograman Microsoft Visual Basic 6.0. Untuk membantu dalam pengembangan aplikasi yang dibuat adalah melalui penggunaan teknik pemrograman Windows API (*Application Programming Interface*) yang merupakan seperangkat kumpulan fungsi-fungsi

sistem Microsoft Windows yang digunakan untuk mengatur tampilan dan kelakuan setiap elemen *windows*.

1.3. Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah sebagai berikut:

1. Membuat aplikasi untuk pengendalian komputer dari jarak jauh, yang akan diimplementasikan pada sistem operasi berbasis Microsoft Windows, sehingga administrator dapat melakukan proses *maintenance* dari jarak jauh (*remote desktop*).
2. Menciptakan sebuah aplikasi yang melakukan pengawasan terhadap kegiatan penggunaan suatu komputer.

1.4. Manfaat Penelitian

Penelitian ini bertujuan untuk menghasilkan beberapa manfaat yaitu :

1. Menciptakan sebuah sistem *client/server* dimana *server* dapat melakukan pemantauan jarak jauh terhadap *client*, dengan cara menciptakan suatu aplikasi yang berbasis *client/server* yang berfungsi untuk merekam tampilan layar monitor dari pengguna, kemudian mengendalikan penggunaan mouse dan keyboard, serta memantau semua aktifitas penekanan tombol *keyboard*, kemudian memiliki fasilitas untuk administrasi sistem dari jarak jauh, berupa shutdown dan restart komputer client.

2. Sistem dapat digunakan oleh administrator jaringan untuk mendapatkan informasi mengenai penggunaan komputer di laboratorium yang diharapkan agar ada kestabilan kinerja dari sistem yang ada, serta terdapat adanya mekanisme penanggulangan terhadap ancaman-ancaman keamanan yang dapat mengganggu kinerja sistem secara keseluruhan yang mungkin terjadi.

1.5. Metode Penelitian

Penulis dalam penyusunan penelitian ini menggunakan metode penelitian laboratorium dengan langkah-langkah sebagai berikut:

1. Studi pustaka, dilakukan dengan cara mencari dan mempelajari literatur, jurnal, buku ataupun sumber yang ada hubungannya dengan obyek yang diteliti.
2. Pembuatan perangkat lunak, langkah-langkah yang digunakan adalah :
 - a. Mempelajari konsep dasar protokol TCP/IP, prinsip pemrograman *Client/Server*, konsep pemrograman winsock (*windows socket*) dan winAPI (*Windows Application Programing Interface*) yang digunakan pada Microsoft Visual Basic 6.0.
 - b. Mempelajari, merancang dan menerapkan permasalahan yang ada ke dalam bahasa pemrograman.
3. Pengujian di laboratorium

4. Pembuatan dokumentasi

II. LANDASAN TEORI

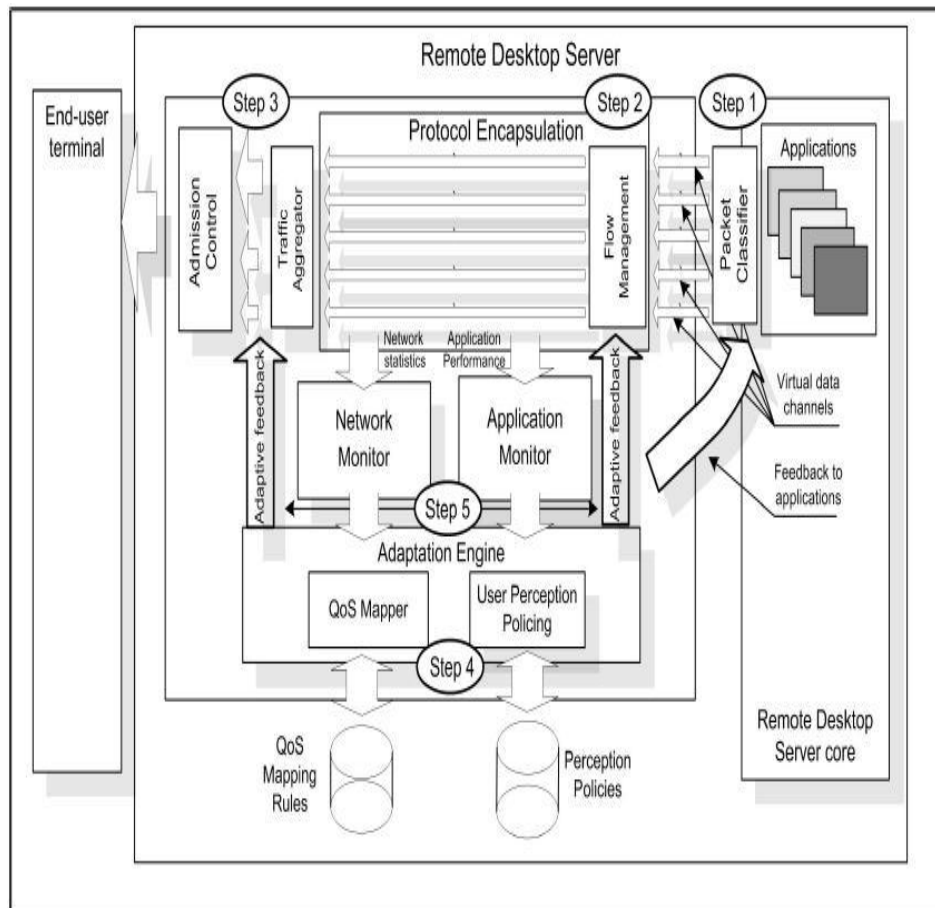
2.1. Konsep Remote Desktop

Konsep *remote desktop* adalah merupakan bentuk implementasi suatu protocol atau tata aturan yang digunakan oleh suatu *client* untuk memvirtualisasi tampilan layer monitor dari pengguna. Mekanisme kerjanya adalah melakukan enkapsulasi tampilan secara kontinu dari suatu aplikasi yang aktif pada komputer yang dipantau, dengan mengirimkan berbagai sumber daya dari *server* ke *client*, untuk selanjutnya terjadi interaksi berupa pengiriman perintah-perintah eksekusi ke komputer *server*. Perintah-perintah tersebut diberikan dalam bentuk masukan dari keyboard ataupun mouse. Berbagai contoh aplikasi remote desktop yang terdapat dalam ranah Teknologi Informasi adalah VNC, Microsoft Terminal Service, LogMein, dan lain sebagainya. Masing-masing aplikasi memiliki lisensi penggunaan tersendiri, baik itu secara komersil, gratis, sumber tertutup, maupun sumber terbuka.

Protokol dari remote desktop didesain dan dioptimasi agar dapat bekerja secara maksimal terhadap aplikasi yang berorientasi terhadap data pada suatu jaringan komputer local, dengan berbagai keadaan penunjang seperti misalnya spesifikasi perangkat keras yang rendah, atau dapat pula akses interkoneksi tersebut melalui jaringan komputer dengan lebar jalur data (*bandwidth*) yang rendah (Lubonski dkk, 2006).

Implementasi penerapan aplikasi remote desktop saat ini sudah banyak digunakan dalam keseharian pengguna di bidang teknologi informasi. Seperti halnya aplikasi Radmin, salah satu aplikasi yang sudah berada pada tahap melegenda, karena keberadaannya yang sudah lama dan banyak digunakan oleh administrator untuk mengawasi laboratorium. Kemudian terdapat beberapa aplikasi lain yang dapat pula melakukan komunikasi dalam bentuk pengiriman berkas audio dan video. Secara garis besar, aplikasi berada pada dua sisi, baik digunakan dalam bidang pengawasan penggunaan komputer, maupun dapat pula digunakan untuk melakukan penyadapan serta pengendalian sumber daya

komputer tanpa izin dari pengguna yang berhak. Hal ini diperdebatkan pula dalam berbagai forum, karena pengguna harus melihat dari sisi yang ada, serta pilihan yang begitu terbuka dalam implementasinya.



Gambar 2.1. Arsitektur Konseptual – Adaptasi Protokol Remote Desktop

Gambar 2.1 memperlihatkan visualisasi arsitektur konseptual dari suatu protocol Remote Desktop dimana komponen-komponen yang terdapat di dalamnya didesain untuk menjamin bahwa layanan-layanan bekerja dengan baik untuk data-data penting. Seperti misalnya pengendalian koneksi atau tindakan administrasi untuk aplikasi. Sebagai tambahan, maka didesain suatu komponen yang berada pada sistem yang sama, dan memungkinkan untuk dapat saling

mengintegrasikan kinerja dalam menghadapi suatu permasalahan. Secara umum, konfigurasi dari sistem remote desktop adalah mencakup hal-hal berikut:

- Pengguna yang berada pada sisi pengendali (*client*) bekerja dengan mengakses jaringan melalui aplikasi remote desktop khusus *client* yang sebelumnya telah terinstall pada komputer yang digunakan.
- Dimungkinkan juga untuk melakukan interkoneksi pada area yang lebih luas seperti Wide Area Network, antara kantor cabang dengan kantor pusat.
- *Server* yang berfungsi sebagai remote desktop akan bekerja lebih baik jika ditempatkan pada bagian pusat data, atau dengan kata lain memiliki lebar jalur koneksi (*bandwidth*) yang maksimal.

Kemampuan optimal dari sistem akan lebih efisien apabila ditempatkan pada bagian atau sisi *server*. Hal ini dikarenakan bahwa *server* yang menyediakan layanan yang diminta dari sisi *client*, sehingga mekanisme inti yang digunakan sebagai basis akan berinteraksi sesuai dengan perintah yang diterima. Pengaturan mekanisme tersebut mencakup pula hal-hal berikut ini:

- Pengawasan pengguna.
- Pemetaan antara kebutuhan pengguna dengan kualitas dari layanan yang diberikan.
- Terciptanya mekanisme baru yang mengatur mengenai korelasi antara satu dengan lainnya dalam hal pembagian traffic lalu lintas pertukaran data antara *client* dan *server*.

2.2. Konsep Virtualisasi

Kebutuhan akan komputasi yang semakin tinggi, membuat pengguna menghadapi masalah penyediaan sarana dan prasarana pendukung sehingga semua layanan dapat dihadirkan secara maksimal. Virtualisasi saat ini menjadi

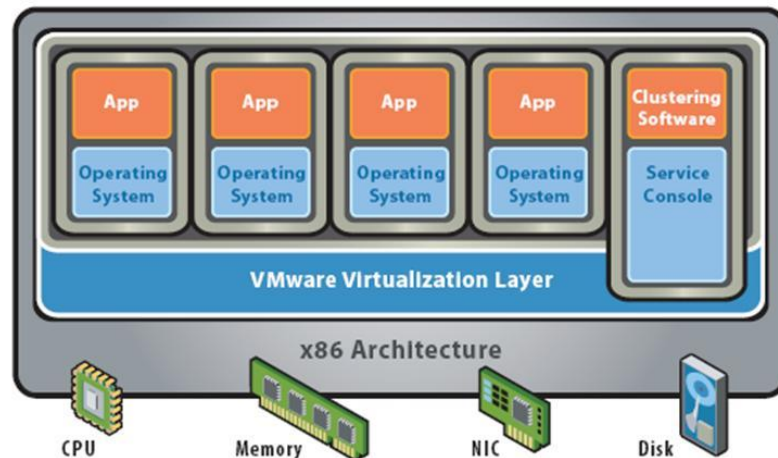
alternatif yang tepat untuk menciptakan suatu efisiensi terhadap investasi sumber daya, dimana terdapat suatu *framework* yang mengkombinasikan dan melakukan pembagian sumber daya yang tersedia untuk menyediakan satu atau lebih lingkungan sistem secara transparan pada komputer utama (*host*). Bentuk implementasinya adalah menghadirkan berbagai komputer logika dengan sumber daya independen, pada satu komputer utama secara fisik, yang dikendalikan oleh Virtual Machine Monitor (VMM) atau yang lebih dikenal juga sebagai Hypervisor (Holmqvist dkk, 2009). Untuk itu dibutuhkan dukungan spesifikasi yang baik pada perangkat keras (ruang lingkup utamanya adalah media penyimpan/*harddisk*, memori utama/RAM, dan prosesor), sistem operasi dan aplikasi virtualisasi dari pihak ketiga.

2.2.1. Pendekatan virtualisasi

Konsep virtualisasi sendiri dapat dilihat dari dua model pendekatan, yaitu *platform virtualization* dan *application virtualization*. Pengertian dari *platform virtualization* adalah menyediakan simulasi komputer secara logika, terdiri dari suatu komputer secara lengkap, terpisah dari komputer utama, sehingga memungkinkan untuk melakukan instalasi sistem operasi baru dan interaksi pengguna secara utuh dengan sistem. Keunggulan dari pendekatan ini adalah sisi fleksibilitas, dimana dapat menjalankan sistem operasi berbasis RISC sebagai mesin virtual, pada host berbasis Intel (RedHat, 2006). Untuk visualisasinya dapat dilihat pada gambar 2.2. Selanjutnya, beberapa tipe dari pendekatan ini adalah sebagai berikut:

- Full virtualization, sistem pada mesin logika mensimulasikan semua perangkat keras sehingga tidak dapat mengakses perangkat keras pada mesin utama. Pada bagian ini, performansi menjadi lebih rendah. Contoh: VMware Workstation, QEMU
- Paravirtualization, sistem pada mesin logika dapat mengakses perangkat keras pada mesin utama sehingga tingkat performansi sistem menjadi lebih baik (Wannous dkk, 2009), Contoh: VMWare Workstation, XEN

- Operating system-level virtualization, menghadirkan lingkungan sistem logika baru yang terisolasi dan terlihat seperti terpisah dari sistem operasi utama. Bagian ini juga sering disebut sebagai lightweight virtualization. Contoh: OpenVZ, chroot



Gambar 2.2. Arsitektur Platform Virtualization (Wroblewski dkk, 2009)

Pendekatan yang kedua adalah *application virtualization*, yaitu dimana sistem operasi mendukung suatu fungsi atau fitur yang terenkapsulasi dalam berkas pustaka agar dapat mengakses aplikasi yang memerlukan suatu spesifikasi khusus dari sistem yang berasal dari versi sebelumnya atau lingkungan yang berbeda, dengan disertai perbaikan serta optimasi yang disesuaikan. Aktifitas tersebut dilakukan apabila aplikasi berjalan pada modus yang tidak sesuai dengan kondisi aslinya. Implementasi metode ini salah satunya terdapat pada sistem operasi modern seperti Microsoft Windows dan Linux. Sebagai contoh, Windows Vista mendukung eksekusi aplikasi yang dibangun pada versi Windows terdahulu.

Keunggulan dari pendekatan ini adalah dapat mengimplementasikan prinsip keamanan dengan akses terbatas sehingga tidak memerlukan hak akses yang tinggi untuk mengeksekusi suatu aplikasi. kemudian memungkinkan mengeksekusi aplikasi pada lingkungan yang berbeda, misalkan menggunakan WINE pada sistem operasi Linux membuat aplikasi yang dibangun pada sistem berbasis Windows dapat dijalankan. Akan tetapi, pendekatan ini juga memiliki

kelemahan yaitu tidak semua aplikasi dapat dilakukan virtualisasi, karena kebutuhan akan sistem yang berbeda satu dengan yang lain.

2.2.2. Tujuan implementasi virtualisasi

Virtualisasi dilakukan didasari oleh beberapa tujuan yang ingin dicapai. Berikut adalah penjelasan lengkap mengenai hal tersebut:

a. Konsolidasi server

Virtualisasi membuat penurunan signifikan jumlah komputer yang digunakan secara fisik. Suatu organisasi dapat melakukan penghematan besar dari sisi biaya, ruang, dan sumber daya listrik, dimana pada satu komputer server saat ini dapat menjalankan sejumlah mesin logika, dimana masing-masing bersifat independen, baik itu perangkat keras dan perangkat lunak, dalam kurun waktu yang bersamaan.

b. Pemulihan lebih baik ketika terjadi bencana

Kondisi dimana mesin virtual secara fisik pada media penyimpan adalah dalam bentuk berkas dan direktori tunggal, membuat proses penanganannya menjadi lebih mudah. Administrator server atau *host*, dapat menjalankan mekanisme pembuatan *backup* secara cepat dibandingkan dengan proses yang sama pada komputer secara fisik. Dengan kemudahan tersebut, kontinuitas pembuatan *backup* dapat lebih terjadwal, dan dapat dipadukan dengan penggunaan aplikasi kompresi seperti misalkan winrar untuk menghemat ukuran berkas atau direktori. Apabila terjadi kerusakan pada suatu sistem mesin logika, maka proses pemulihan hanya melakukan penghapusan semua berkas yang terhubung dan langsung melakukan ekstraksi data dari berkas terkompresi cadangan.

Jumlah komputer server yang secara fisik menjadi sedikit, membuat proses evakuasi atau pemindahan ke lokasi aman jika terjadi bencana menjadi lebih cepat dan dapat dilakukan secara individu ataupun kelompok kecil.

c. Kemudahan ujicoba aplikasi ataupun sistem baru

Ujicoba suatu aplikasi atau sistem baru membutuhkan spesifikasi komputer atau server tertentu dapat dilakukan melalui interaksi dengan lingkungan mesin logika, yang terpisah dari sistem utama. Adanya lingkungan baru yang independen tersebut membuat proses ujicoba dapat berlangsung dengan aman, optimal serta bersifat keseluruhan aspek, karena tidak melakukan interupsi terhadap suatu layanan, kemudian dapat disesuaikan fitur kebutuhannya, misalkan dieksekusi pada berbagai karakteristik dan tipe sistem operasi yang berbeda, sehingga apabila terjadi kesalahan atau gangguan terhadap sistem logika, tidak akan mempengaruhi sistem yang lain.

III. ANALISIS DAN PERANCANGAN SISTEM

Perancangan perangkat lunak terbagi atas: *Data Flow Diagram* (DFD), diagram yang menjelaskan mengenai aliran data dan proses yang terlibat di dalamnya. Analisis sistem digunakan untuk mempelajari lebih rinci mengenai tiap bagian yang ada dalam perancangan.

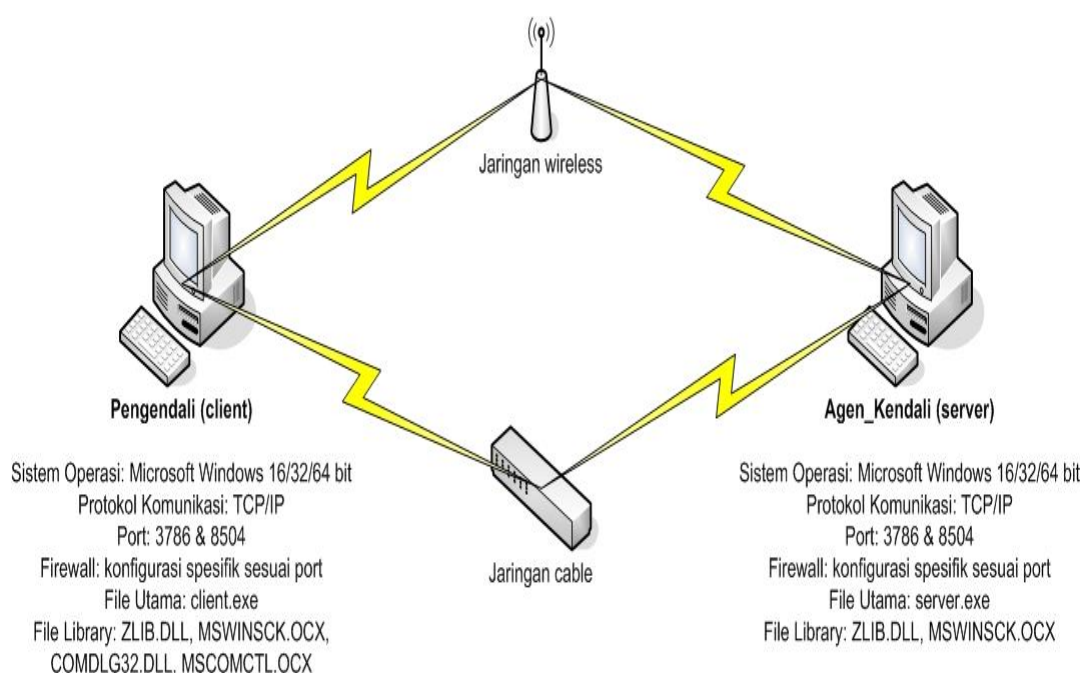
3.1. Analisis Sistem

Implementasi pemrograman *client/server* yang dibuat ini, aplikasi yang digunakan sebagai modelnya dibagi menjadi dua bagian yaitu pengendali dan agen_kendali. Pengendali adalah program yang mengirim permintaan *connect* dan meminta layanan dari agen_kendali, serta akan menunggu (*listening*) permintaan dari pengendali, untuk selanjutnya memberikan layanan. Selain itu, komputer agen_kendali mengumpulkan informasi dari *workstation*, sedangkan pengendali mengendalikan sebagian besar tugas dari agen_kendali.

Kondisi tersebut melatarbelakangi perancangan suatu perangkat lunak pemantauan jarak jauh yang membantu seorang administrator jaringan untuk dapat dengan mudah memantau aktifitas dari pengguna yang aktif pada sisi komputer agen_kendali, dengan cara mengakses rekaman tampilan layar monitor, kemudian mencatat setiap kejadian penekanan tombol pada *keyboard* dan *window* yang aktif yang dikenai aksi dari *keyboard*, sehingga administrator dapat menindaklanjuti jika terjadi penyalahgunaan.

Sistem ini terdiri dari dua buah aplikasi, yaitu aplikasi pengendali dan aplikasi *agen_kendali*. Aplikasi pengendali dapat dijalankan dari workstation manapun dalam sebuah LAN, digunakan untuk melakukan pengawasan dan pengendalian terhadap semua *workstation* dalam LAN yang telah aktif menjalankan aplikasi *agen_kendali*. Aplikasi pengawasan ini berjalan secara *background* dan menyembunyikan dirinya sehingga tidak terlihat pada *window task manager*.

Jalur komunikasi antar aplikasi menggunakan protokol TCP/IP, dimana komunikasi dilakukan setelah proses inialisasi nomor IP antar komputer serta pengenalan port-port yang digunakan oleh masing-masing proses. Dalam bentuk transfer data dari sisi *agen_kendali* berupa tampilan gambar dalam format BMP ke sisi pengendali. Untuk visualisasi dari arsitektur sistem dari aplikasi, dapat dilihat pada gambar 3.1.



Gambar 3.1. Arsitektur Sistem.

Port yang digunakan sebagai media interkoneksi antara aplikasi pengendali dengan agen_kendali harus diberikan akses di dalam *rule* firewall yang aktif di kedua sistem. Pengguna dapat memilih untuk menonaktifkan firewall seperti halnya pada konfigurasi standar Microsoft Windows XP dan versi setelahnya, atau dengan memilih untuk memberikan kebijakan *permit/allow* untuk mengizinkan adanya komunikasi pada port-port yang digunakan oleh perangkat lunak dalam penelitian ini.

3.2. Analisis Kebutuhan Aplikasi

Kebutuhan perangkat lunak pemantauan jarak jauh (*remote desktop*) akan dijelaskan berikut ini.

3.2.1. Perspektif Perangkat Lunak

Aplikasi *Remote desktop* ini berfungsi untuk melakukan pengawasan dan perekaman tampilan layar monitor yang merupakan aktifitas dari suatu komputer dalam jaringan komputer lokal, untuk selanjutnya dari sisi pengintai memperoleh informasi jelas apa saja yang telah dilakukan oleh seorang pengguna pada komputer tersebut. Penggunaannya dalam area kantor ataupun laboratorium secara jelas akan dapat membuat administrator memperoleh informasi-informasi penting, serta dapat pula memonitor perilaku seseorang dalam beraktifitas menggunakan sarana komputer publik.

3.2.2. Fungsi Perangkat Lunak

Perangkat lunak pemantauan jarak jauh (*remote desktop*) akan bekerja secara *background process* yang akan melakukan perekaman terhadap tampilan layar monitor dari sistem target, yang merupakan pencatatan aktifitas yang dilakukan oleh pengguna pada sistem tersebut ke dalam file grafis dengan format BMP. Kegiatan penyadapan akan secara otomatis dilakukan apabila komputer diaktifkan karena terdapat mekanisme yang menjalankan aplikasi dieksekusi saat sistem bekerja.

3.2.3. Karakteristik Pengguna

Pengguna perangkat lunak pemantauan jarak jauh (*remote desktop*) dalam hal ini terbagi menjadi dua, yaitu pengendali dan agen_kendali.

Agen_kendali merupakan pengguna yang bertindak sebagai target dimana dirinya tidak mengetahui perangkat lunak *remote desktop* berjalan pada sistem komputernya.

Pengendali merupakan pengguna perangkat lunak yang memantau aktifitas komputer yang terhubung pada komputer dalam suatu jaringan.

3.2.4. Batasan-batasan

Batasan-batasan yang digunakan dalam mengembangkan perangkat lunak pemantauan jarak jauh (*remote desktop*) ini adalah sebagai berikut :

6. Perangkat lunak yang dikembangkan berbasis *client/server*. Perlu diperhatikan *client* merupakan komputer yang menjalankan aplikasi pengendali dan *server* merupakan komputer yang menjalankan aplikasi agen_kendali serta menjadi tujuan dari pemantauan.
7. Aplikasi yang dikembangkan berfungsi untuk melakukan pemantauan layer monitor dari pengguna, mengendalikan keyboard dan mouse, serta memantau aktifitas penekanan tombol keyboard.
8. Perangkat lunak *remote desktop* dijalankan pada komputer dengan basis sistem operasi Microsoft Windows 98 SE, Windows ME, Windows XP, Windows Server 2000/2003, Windows Vista 32 bit dan 64 bit, dimana perangkat lunak menggunakan layanan dari *file library* Windows.
9. Perangkat lunak *remote desktop* memerlukan jaringan komputer lokal (LAN) untuk dapat berkomunikasi antara komputer pengendali dengan agen_kendali.
10. Sistem dibangun menggunakan perangkat pemrograman Microsoft Visual Basic 6.0. Untuk membantu dalam pengembangan aplikasi yang dibuat adalah melalui penggunaan teknik pemrograman Windows API (*Application Programming Interface*) yang merupakan seperangkat kumpulan fungsi-fungsi sistem Microsoft Windows yang digunakan untuk mengatur tampilan dan kelakuan setiap elemen *windows*.

3.2.5. Kebutuhan Antarmuka Pengguna

Interaksi dari pengguna terjadi pada sisi klien, dimana sistem akan secara kontinu memonitor pergerakan tampilan layar monitor yang menggambarkan

aktifitas yang dilakukan. Aplikasi akan bekerja secara *stealth* dan tidak akan terlihat pada pengecekan di daftar program yang aktif. Hal ini dilakukan agar pengguna tidak mengetahui adanya sistem yang melakukan proses pemantauan.

Komputer *pengendali* yang dirancang untuk melakukan pengawasan, dengan fungsi-fungsi yang telah ditentukan sebelumnya akan menginisialisasi berbagai macam perintah yang diinginkan untuk dikerjakan pada sisi *agen_kendali*.

3.2.6. Kebutuhan Antarmuka Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan oleh aplikasi pemantauan jarak jauh (*remote desktop*) adalah :

- Sistem operasi Microsoft Windows 98, 98 Second Edition (SE), Millenium Edition (ME), XP, Server 2000/2003, Vista 32 bit dan 64 bit. Penggunaan *file library* dari sistem operasi yang terdapat pada lokasi direktori **\Windows\System** dan **\Windows\System32** dalam operasional dari perangkat lunak menjadi basis atau dasar yang utama. Fitur ini menjadi pendukung implementasi metode pemrograman API.

Contoh *library* yang digunakan adalah **MSWINSCK.OCX** yang memberikan fasilitas dalam pemrograman windows socket. Kemudian pPenggunaan *file library* tambahan yang bersifat *thirdparty product* seperti **ZLIB.DLL** yang digunakan untuk fungsi kompresi dari aliran data gambar.

3.3. Analisis Fungsionalitas Aplikasi

Tahap perancangan aplikasi pemantauan jarak jauh (*remote desktop*) dimulai dengan menginventarisasi semua kejadian pada mesin *pengendali* yang dikendalikan dari *agen_kendali*, kemudian menuliskannya sebagai suatu daftar kejadian (*Event List*). Untuk setiap kejadian ditentukan segala informasi yang terlibat, ditentukan alirannya (sumber dan tujuan) dan digambarkan dalam bentuk DFD. Tetapi akan dijelaskan fungsi-fungsi yang terdapat di dalamnya. Berikut fungsi-fungsi yang terdapat pada aplikasi ini adalah:

a. Fungsi Pengendali Utama

Fungsi ini mencakup inisialisasi program (*remote desktop*) ke sistem yang diinginkan. Untuk fungsi-fungsi yang tercakup di dalamnya adalah untuk pengecekan IP komputer *agen_kendali* serta port yang telah ditentukan untuk melakukan koneksi antara *agen_kendali* dengan *pengendali*. Konfigurasi dari protokol dan nilai *port* yang digunakan untuk melakukan komunikasi dengan aplikasi *server*. Selain itu terdapat pula prosedur untuk melakukan pembuatan *file* konfigurasi yang ditetapkan untuk menyimpan data IP *pengendali* dan port.

b. Fungsi Informasi Sistem

Form ini akan menampilkan beberapa informasi penting dari komputer *agen_kendali* yang dipantau, seperti spesifikasi komputer, memori, serta pengguna dari komputer *agen_kendali*.

c. Fungsi Akses Media Penyimpan

Fitur berikut ini digunakan untuk melakukan eksplorasi harddisk aktif dari komputer *agen_kendali*, sehingga pihak yang melakukan pemantauan dapat mengakses secara penuh sumberdaya yang ada. Beberapa fitur penting yang tersedia adalah upload dan download berkas dan penghapusan (*delete*).

d. Fungsi Kendali Task Manager

Fitur ini digunakan oleh pengendali untuk melakukan kendali terhadap tray dari sistem windows pada komputer *agen_kendali*. Kemudian terdapat pula akses untuk melakukan administrasi reboot, shutdown, maupun logoff.

e. Fungsi Perekaman Keyboard

Fungsi ini dikembangkan untuk digunakan mencatat semua aksi dari penekanan pada tombol *keyboard* pada komputer *agen_kendali*. Pendeteksian tombol yang ditekan pada *keyboard* dilakukan secara berulang dengan waktu perekaman ke *file* log setiap 10 detik. Perekaman tombol disertai dengan merekam pada window mana yang dikenai aksi penekanan tombol tersebut, disertai data waktu pada saat jendela tersebut dikenai aksi.

f. Fungsi Perekaman Tampilan Layar

Form ini digunakan ketika pengguna pada sisi komputer pengendali akan melakukan pemantauan terhadap layar monitor pada komputer *agen_kendali*. Kemudian terdapat pula fitur untuk melakukan perekaman layar dalam bentuk berkas BMP yang akan disimpan pada komputer *pengendali*.

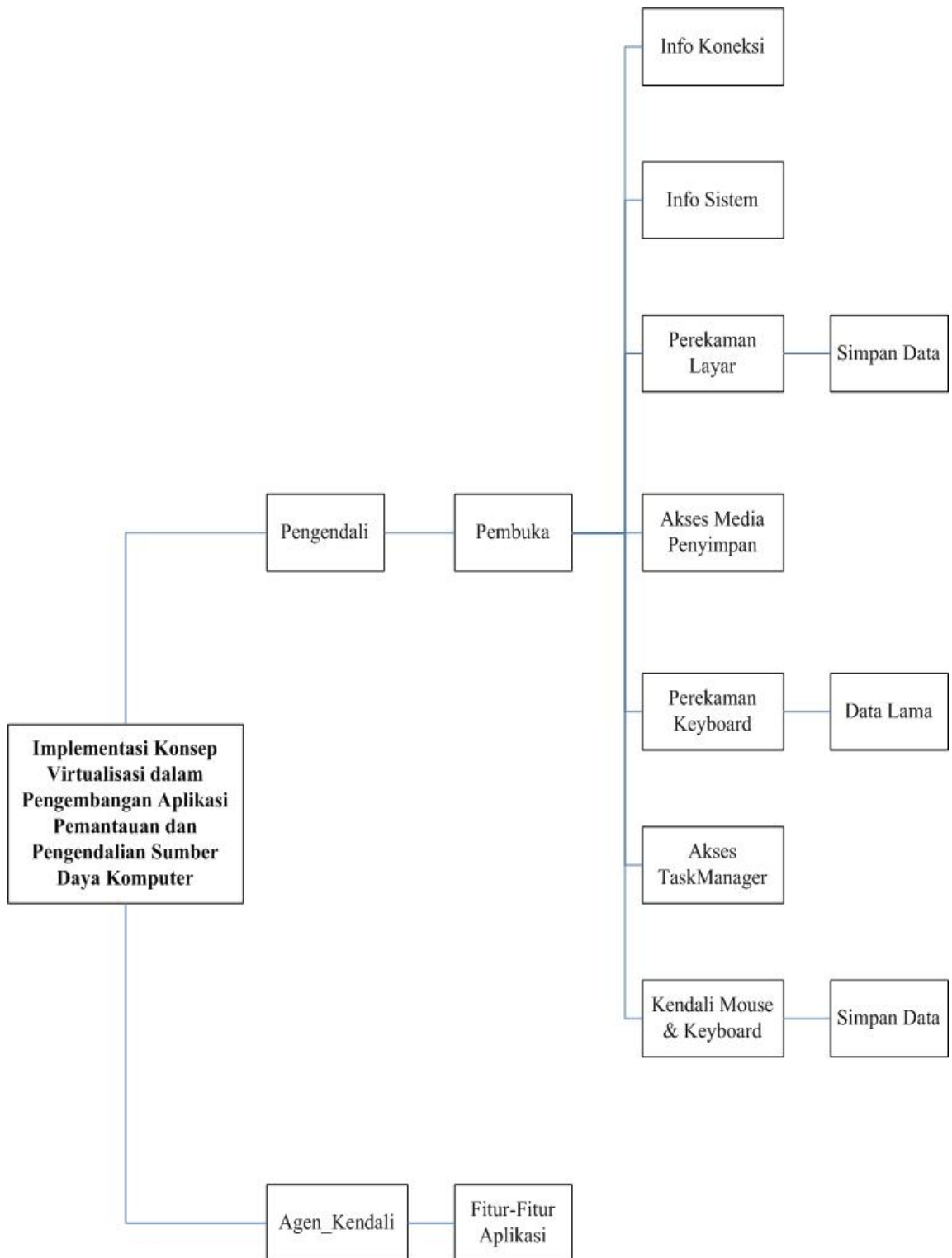
g. Fungsi Kendali Mouse & Keyboard

Fitur ini berfungsi untuk melakukan pengendalian komputer, baik pada sisi keyboard ataupun mouse yang aktif pada komputer agen_kendali. Opsi tersebut dapat pula dikonfigurasi sesuai dengan kebutuhan. Tampilan yang muncul pada sisi *pengendali* dapat diatur sesuai dengan kebutuhan kinerja yang diharapkan selama melakukan pemantauan, serta terdapat pula fasilitas untuk melakukan konfigurasi skala layar yang diakses oleh pengendali.

Perangkat lunak pemantauan jarak jauh (*remote desktop*) di sisi agen_kendali juga menyamarkan daftar nama aplikasi dirinya dari *task manager* dengan menginisialisasi aplikasi aktif tersebut dengan nama yang familiar dengan layanan aktif pada Windows seperti service, akan diganti dengan servic3. Cara ini efektif untuk menghindari pengguna mengetahui adanya aplikasi tambahan yang berjalan dan menghindari pengguna tersebut menghentikan kerjanya. Akan tetapi, setiap aplikasi yang berjalan tetap mempunyai identitas (ID) dan nama proses seperti yang terlihat pada tab Processes di *task manager* pada sistem Microsoft Windows XP, Windows Server 2000/2003, Windows Vista 32 bit dan 64 bit.

3.3.1. Perancangan Arsitektur Perangkat Lunak

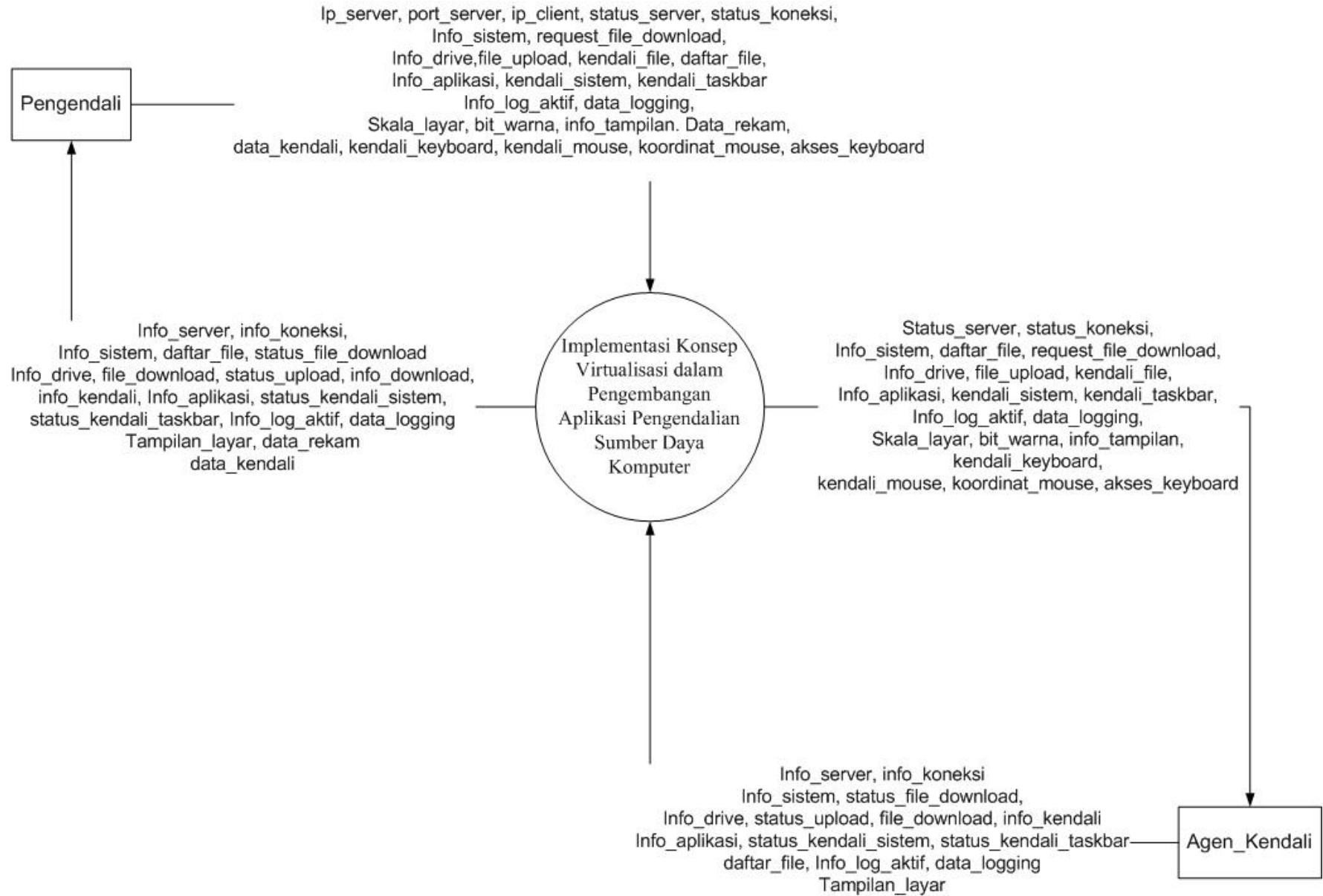
Arsitektur perancangan perangkat lunak menggunakan pohon hirarki, yang menjelaskan struktur halaman perangkat lunak akan divisualisasikan pada gambar 3.3. Sedangkan untuk desain antarmuka, dapat dilihat pada bagian lampiran 1.



Gambar 3.2. Perancangan Arsitektur Perangkat Lunak.

3.3.2. Data Flow Diagram (DFD) Level 0

DFD level 0 ini akan menjelaskan proses yang terjadi dalam perangkat lunak pemantauan jarak jauh (*remote desktop*) secara global beserta entitas yang terlibat di dalamnya. Seperti terlihat pada gambar 3.3, terdapat sebuah proses global, yaitu *remote desktop*; terdapat entitas pengendali, dan terdapat entitas *agen_kendali..*



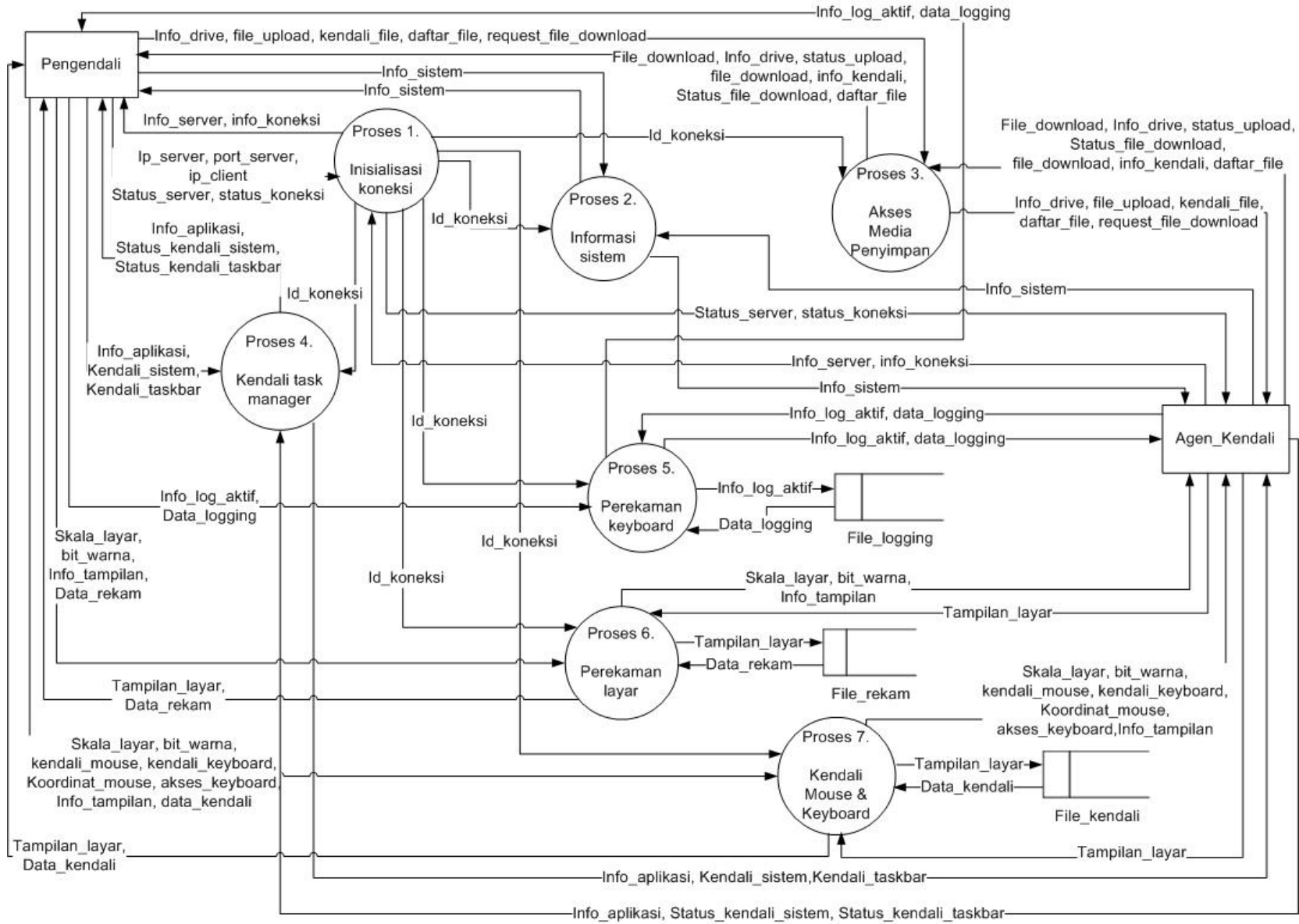
Gambar 3.3. DFD Level 0 *Remote desktop*

Sistem akan bekerja untuk memonitor secara keseluruhan aktifitas dari komputer target, dengan berbagai fitur aplikasi yang akan dijelaskan pada bagian

3.3.3. DFD level 1.

3.3.3. Data Flow Diagram (DFD) level 1

DFD level 1 yang divisualisasikan pada gambar 3.4 merupakan penurunan dari proses pada level 0, yang akan menjelaskan mekanisme proses dari aplikasi yang ada pada level 0 secara lebih detil mengenai proses dan aliran datanya.

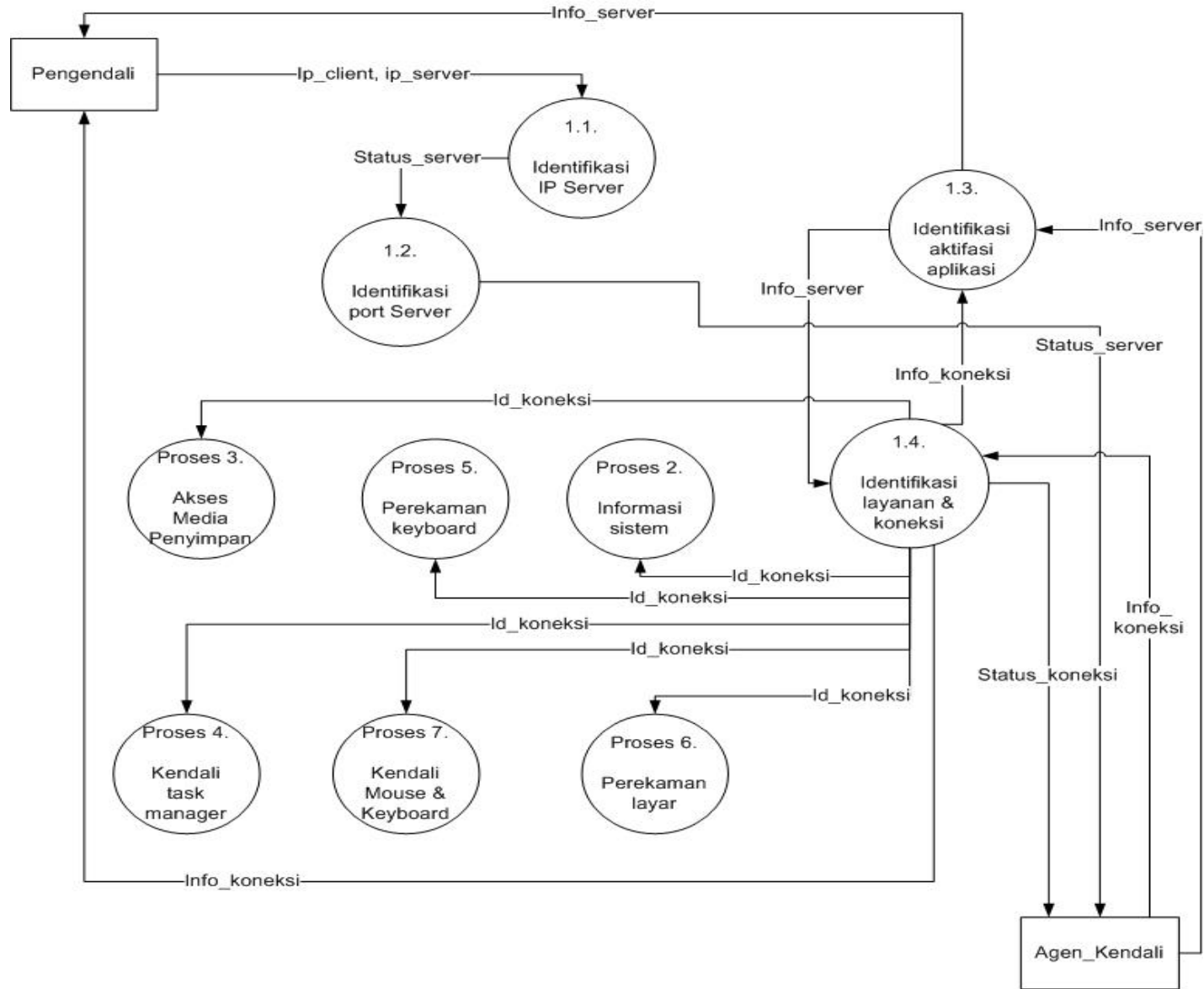


Gambar 3.4. DFD Level 1 *Remote desktop*

Penjelasan lengkap mengenai masing-masing proses dapat dilihat pada sub-bab pembahasan level 2 dari proses yang berjalan.

3.3.4. Data Flow Diagram (DFD) Level 2 dari Proses Inisialisasi Koneksi

Pada DFD level 2 proses Inisialisasi Koneksi (terlihat pada gambar 3.5) menjelaskan secara lebih lengkap proses yang digambarkan pada DFD level 1.

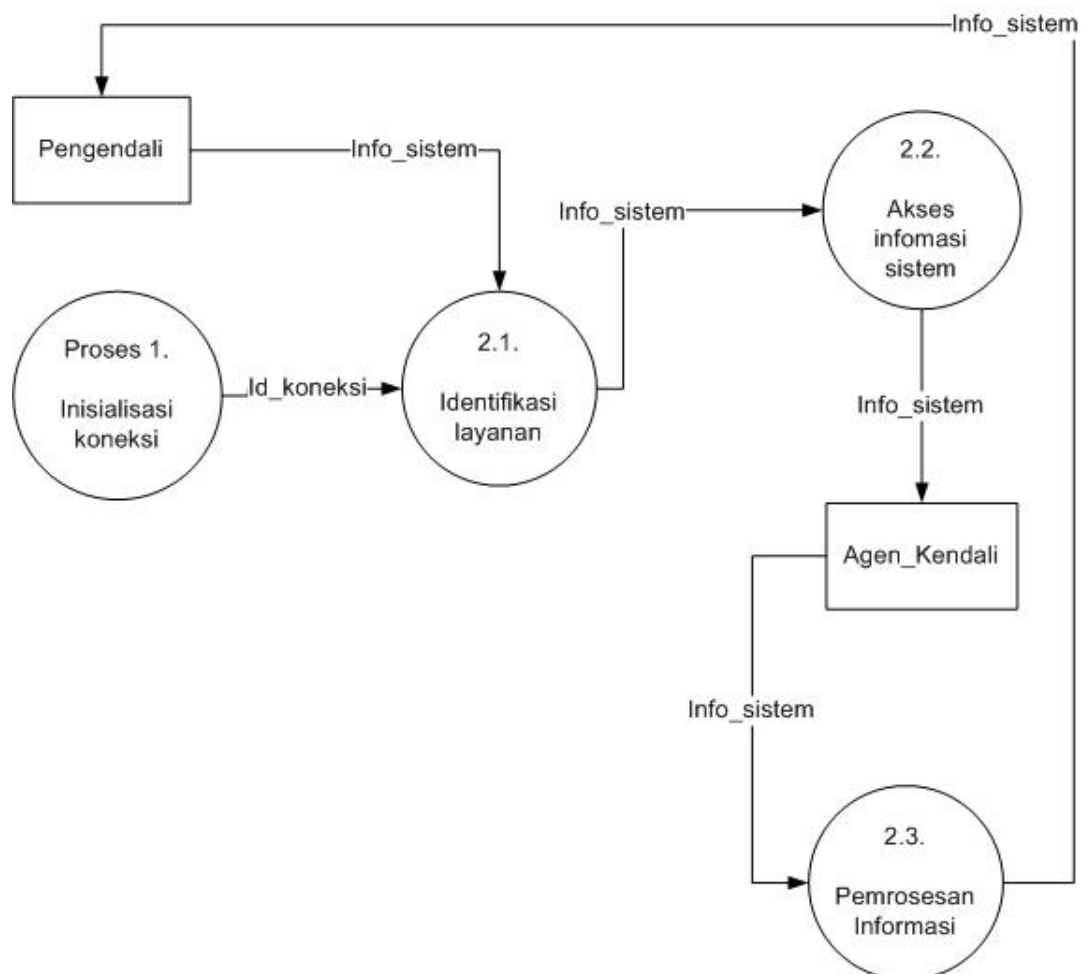


Gambar 3.5. DFD Level 2 Proses Inisialisasi Koneksi

Proses 1.1 menjelaskan mengenai inisialisasi koneksi lebih pada pengaturan akses koneksi dan diskoneksi dari komputer agen_kendali ke pengendali. Mekanisme pengenalannya melalui nomor IP komputer yang dimasukkan oleh pengguna dari sisi agen_kendali untuk selanjutnya melakukan inisialisasi port-port yang telah ditentukan sebelumnya agar segera siap menjalankan *request* yang dikirimkan.

3.3.5. Data Flow Diagram (DFD) Level 2 dari Proses Informasi Sistem

Visualisasi DFD level 2 proses Informasi Sistem terlihat pada gambar 3.6.

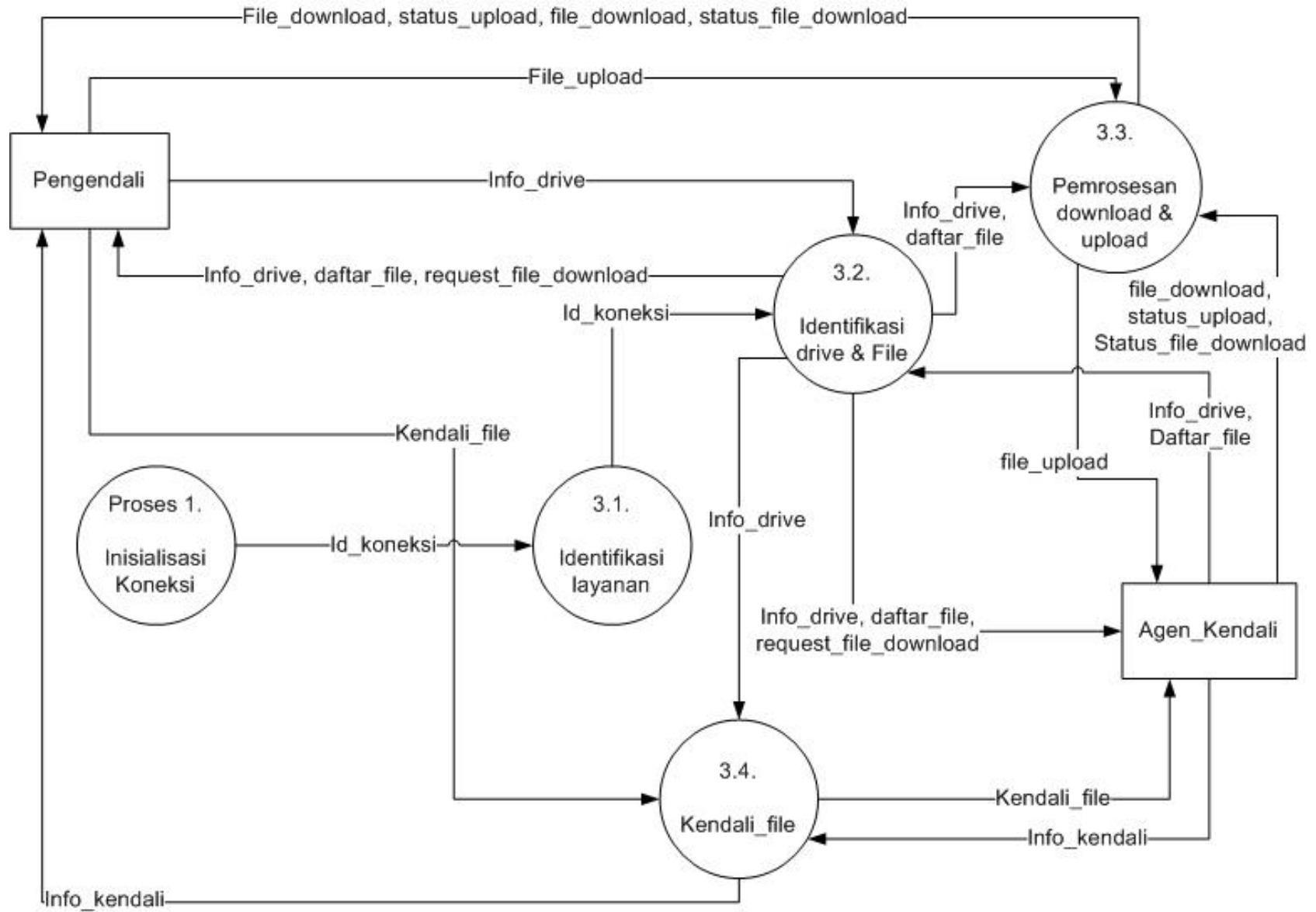


Gambar 3.6. DFD Level 2 Proses Informasi Sistem

Proses 1.2 menjelaskan mengenai fungsi Informasi Sistem yang akan menampilkan beberapa informasi penting dari komputer agen_kendali yang dipantau, seperti spesifikasi komputer, memori, serta pengguna dari komputer agen_kendali.

3.3.6. Data Flow Diagram (DFD) Level 2 dari Proses Akses Media Penyimpan

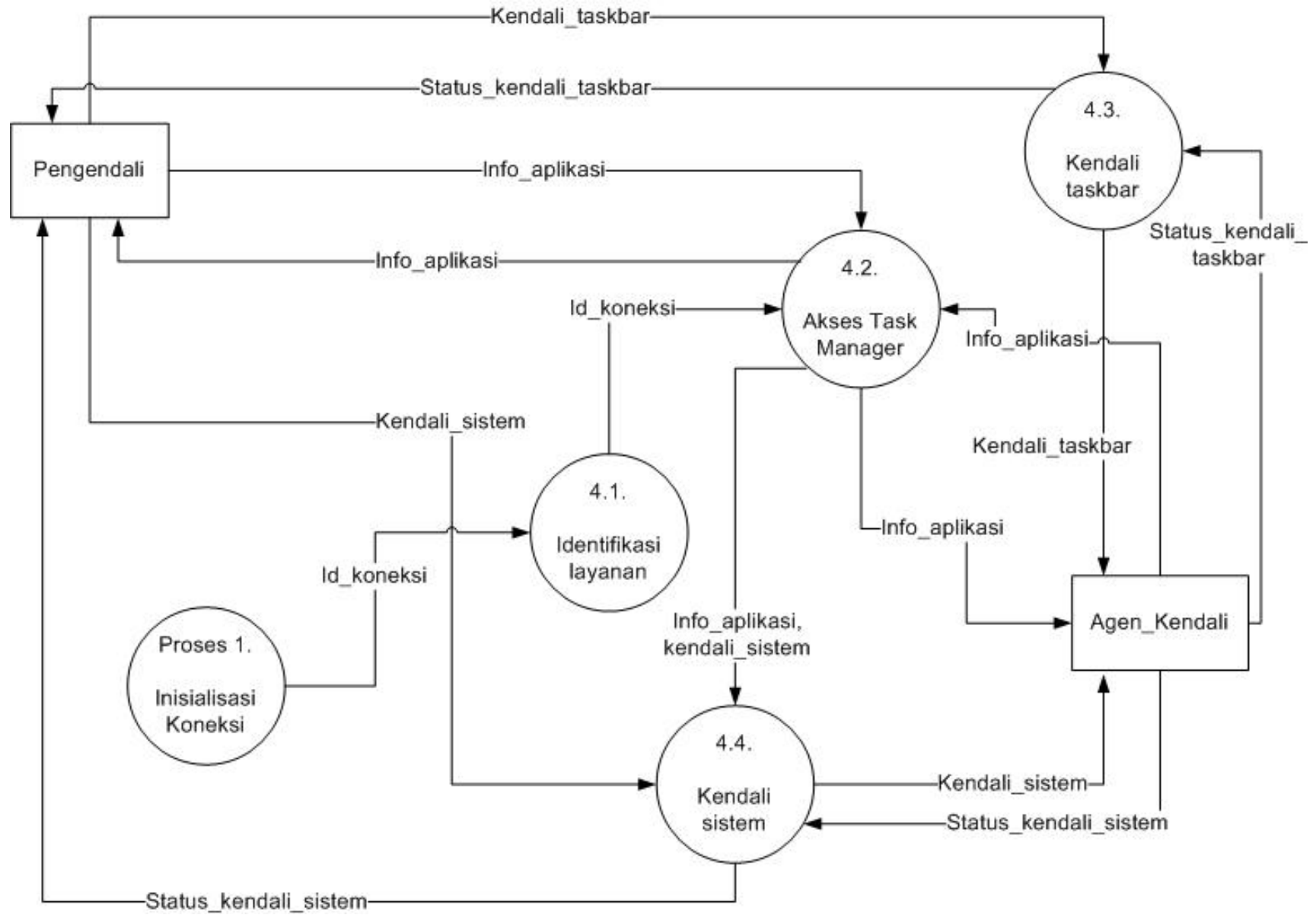
Proses 1.3 yaitu Akses Media Penyimpan ialah digunakan untuk memberikan administrator hak melakukan pengaksesan media penyimpanan, baik itu *harddisk*, *cdrom*, maupun perangkat tambahan lain yang aktif dari komputer yang menjadi target pengawasan. Fungsi berikut ini berbeda dengan konsep *sharing* dari *netbios* karena sistem akan langsung mengakses *resource* menggunakan prosedur *win32API*. Kemampuan yang disertakan adalah untuk melakukan pengiriman dan pengambilan *file*, mengeksekusi file, serta penghapusan *file*. Konsep yang digunakan adalah sama seperti aplikasi Windows Explorer, akan tetapi *resource* yang dikendalikan adalah berasal dari komputer lain. Untuk visualisasinya dapat dilihat pada gambar 3.7.



Gambar 3.7. DFD Level 2 Proses Akses Media Penyimpan.

3.3.7. Data Flow Diagram (DFD) Level 2 dari Proses Kendali Task Manager

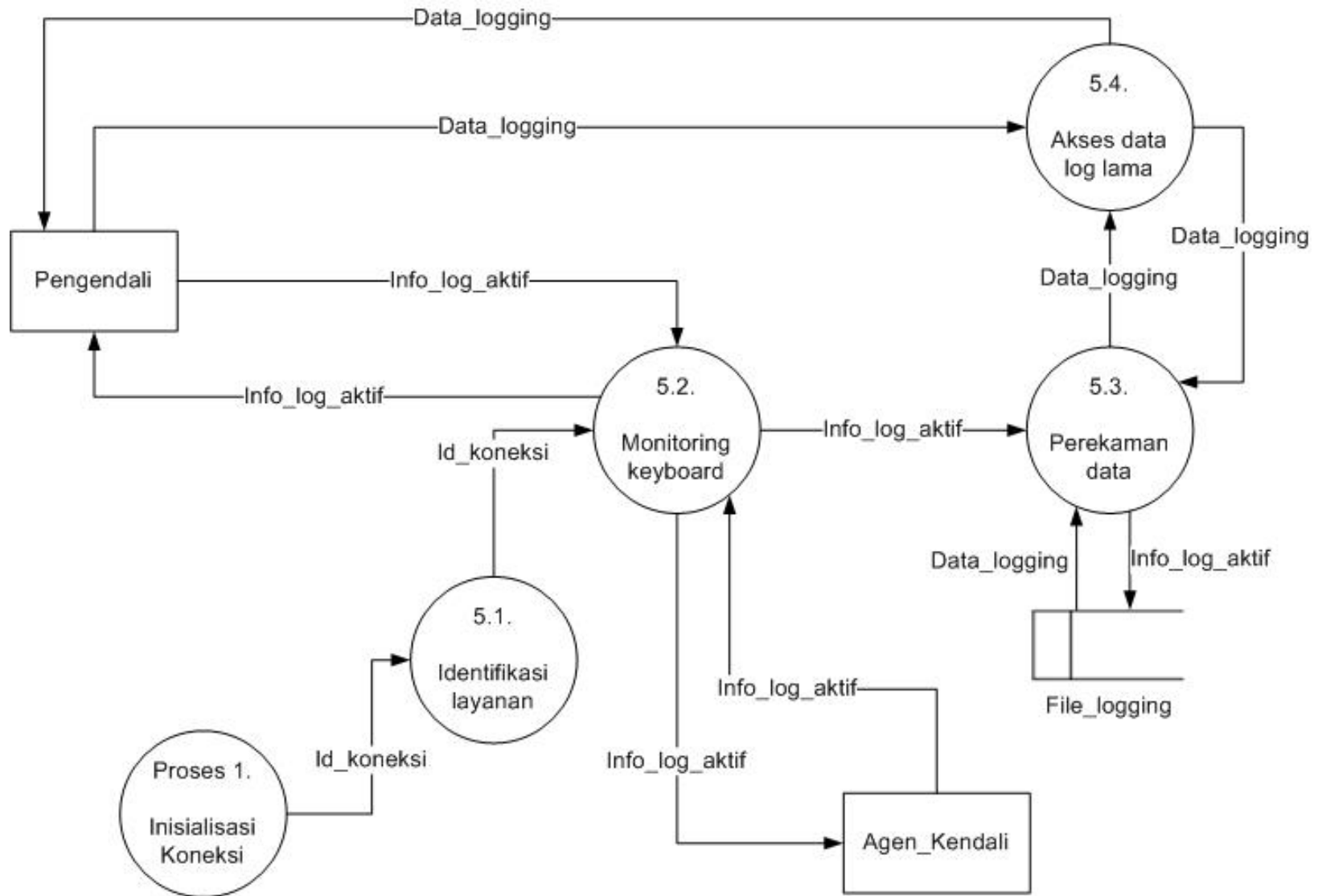
Proses 1.4 menjelaskan mengenai Proses Kendali Task Manager berfungsi untuk mengakses berbagai fitur terhadap tray dari sistem operasi Windows, maupun digunakan untuk melakukan penghentian akses berupa mengeksekusi perintah LogOff, Restart, maupun Shutdown. Untuk visualisasinya dapat dilihat pada gambar 3.8.



Gambar 3.8. DFD Level 2 Proses Kendali TaskManager

3.3.8. Data Flow Diagram (DFD) Level 2 dari Proses Perekaman keyboard

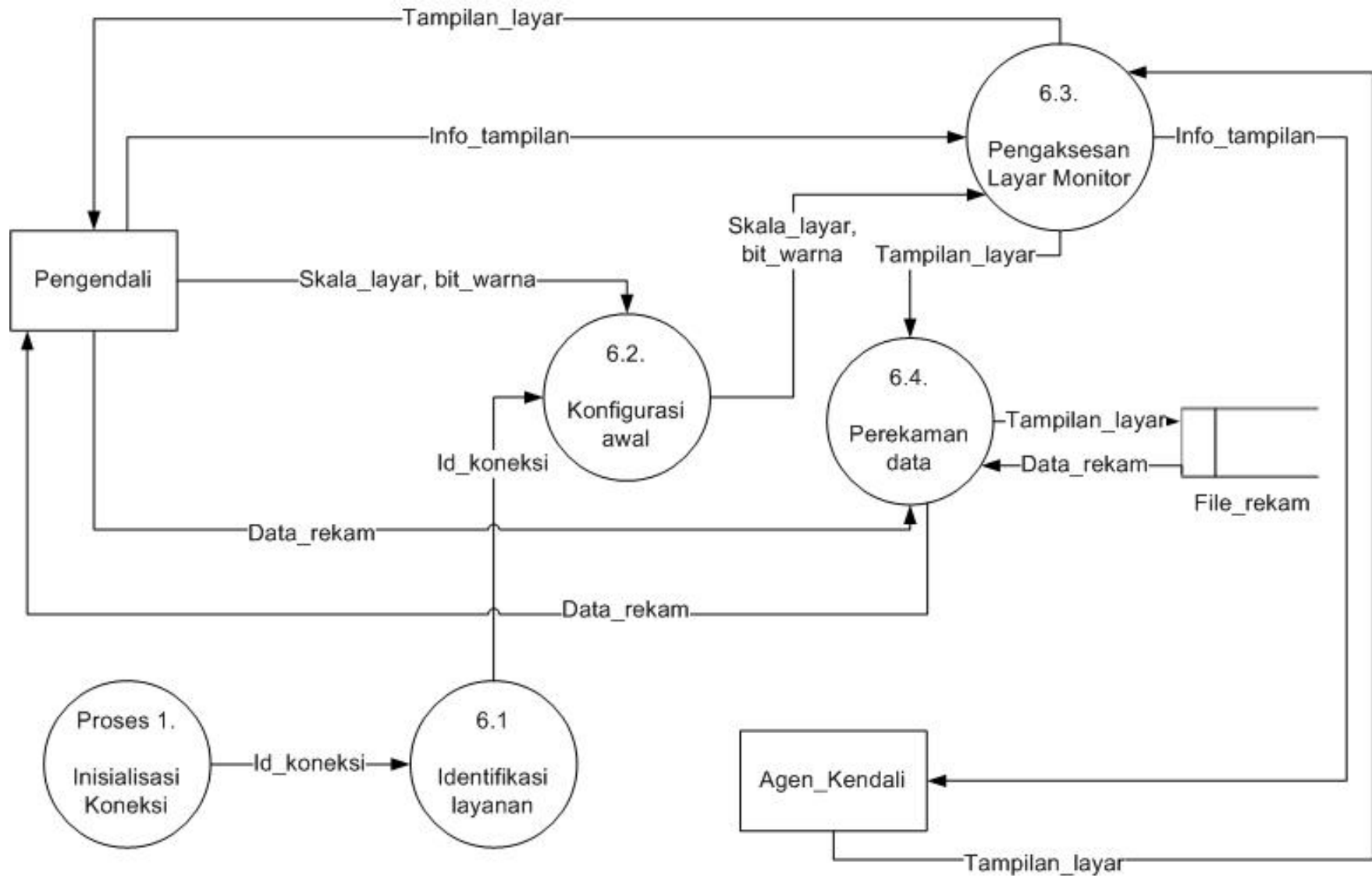
Proses 1.5 menjelaskan mengenai Proses Perekaman keyboard yang berfungsi untuk menampilkan hasil atau aktifitas penekanan keyboard dari pengguna pada komputer agen_kendali. Secara realtime akan dikirim data log tersebut, kemudian terdapat pula fungsi untuk memanggil berkas rekaman lama yang disimpan pada lokasi tersembunyi di komputer agen_kendali. Sehingga dari komputer pengendali dapat mengakses rekaman data tersebut. Untuk visualisasinya dapat dilihat pada gambar 3.9.



Gambar 3.9. DFD Level 2 Proses Perekaman Keyboard.

3.3.9. Data Flow Diagram (DFD) Level 2 dari Proses Perekaman Tampilan Layar

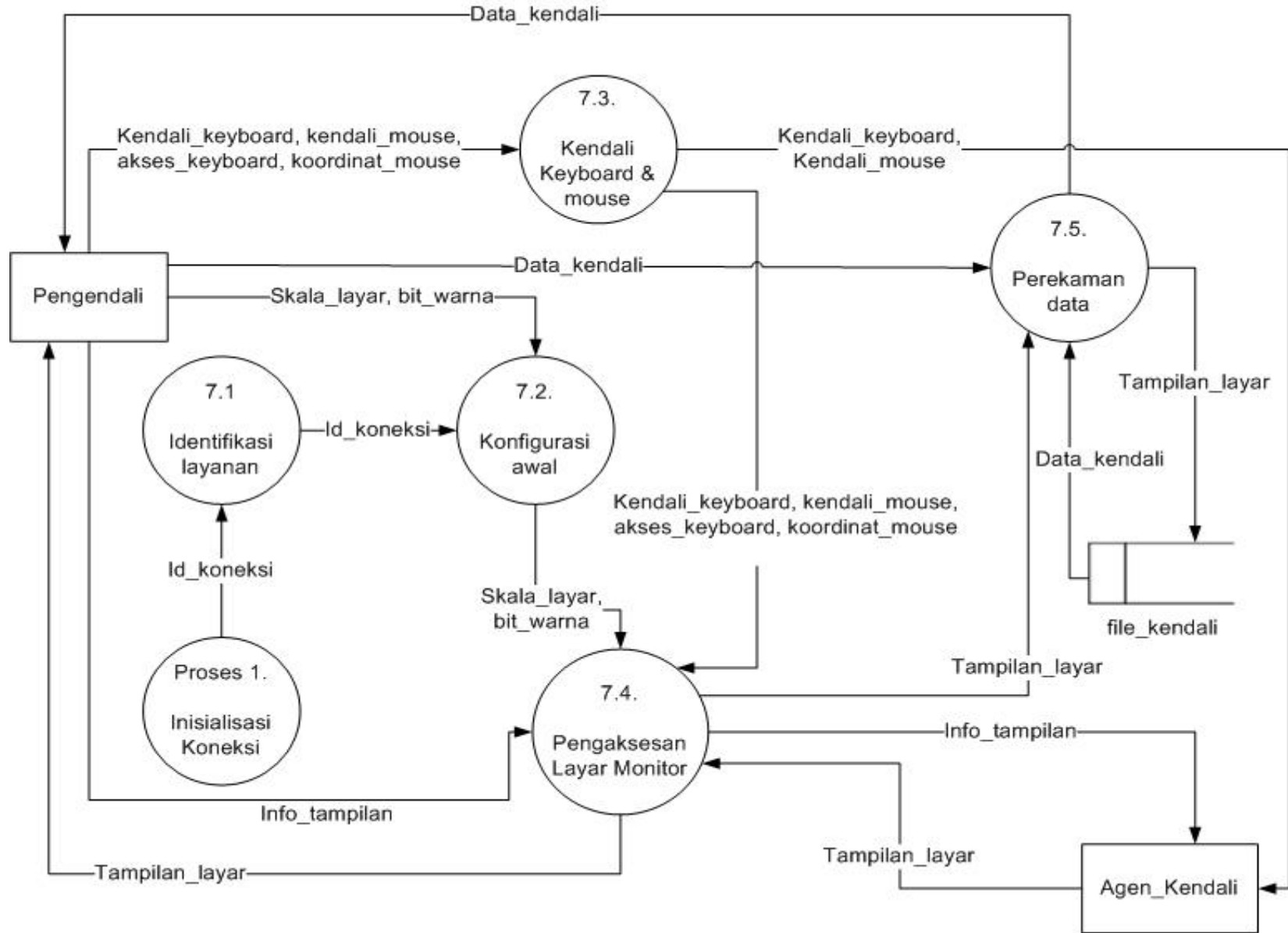
Proses 1.6 menjelaskan mengenai fungsi untuk melakukan perekaman tampilan layar, digunakan untuk mengawasi secara *soft* aplikasi-aplikasi apa saja yang sedang dijalankan oleh pengguna dari sisi *agen_kendali*. Metode pengawasan akan berdasar pada penekanan tombol kiri ataupun kanan dari *mouse*. Untuk visualisasinya dapat dilihat pada gambar 3.10.



Gambar 3.10. DFD Level 2 Proses Perekaman Tampilan layar.

3.3.10. Data Flow Diagram (DFD) Level 2 dari Proses Kendali Keyboard & Mouse

Fitur ini berfungsi untuk melakukan pengendalian komputer, baik pada sisi keyboard ataupun mouse yang aktif pada komputer target pemantauan. Opsi tersebut dapat pula dikonfigurasi sesuai dengan kebutuhan. Tampilan yang muncul pada sisi pengendali dapat diatur sesuai dengan kebutuhan kinerja yang diharapkan selama melakukan pemantauan, serta terdapat pula fasilitas untuk melakukan konfigurasi skala layar yang diakses oleh komputer pengendali Visualisasi dari proses tersebut dapat dilihat pada gambar 3.11.



Gambar 3.11. DFD Level 2 Proses Kendali Keyboard & Mouse.

IV. IMPLEMENTASI

4.1. Implementasi Sistem

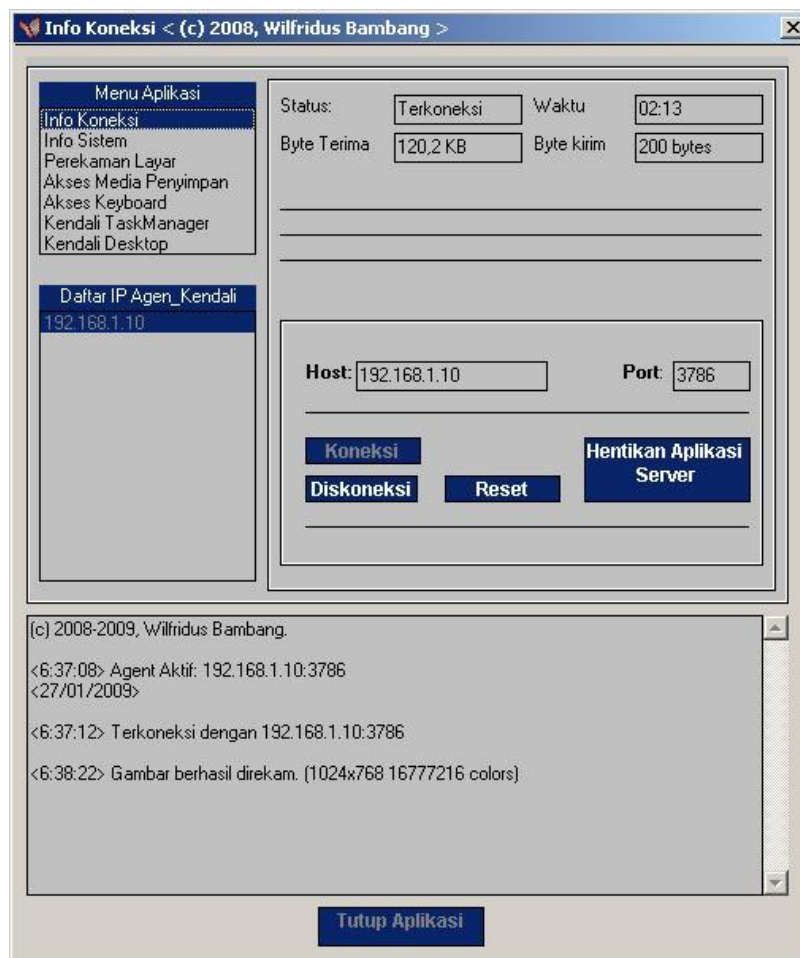
Perangkat lunak pengendalian layar ini dibangun menggunakan aplikasi pemrograman Microsoft Visual Basic 6.0 dengan memanfaatkan fasilitas komponen utama Microsoft Winsock Control untuk membangun koneksi, jalur komunikasi, serta kemampuan pengontrolan serta pengendalian dari jarak jauh (*remote*).

4.2. Implementasi Aplikasi Pengendali

Proses inisialisasi terjadi baik pada sisi pengendali maupun pada sisi agen_kendali. Inisialisasi digunakan untuk menciptakan kondisi yang diinginkan serta menentukan nilai awal dari socket telah terdefinisi. Perintah-perintah yang diberikan kepada sistem akan diletakkan pada prosedur *form_load*.

4.2.1. Pembahasan fitur Inisialisasi Koneksi

Proses inisialisasi yang terjadi adalah mengecek terlebih dahulu apakah aplikasi telah aktif sebelumnya atau tidak. Pengecekan dilakukan dengan perintah *if app.previnstance = true then end* ini akan mengurangi resiko kegagalan pada proses pemantauan.



Gambar 4.1. Visualisasi fitur Inisialisasi Koneksi.

Aplikasi selanjutnya akan memuat daftar fitur-fitur yang tersedia, dengan menempatkannya dalam listbox pada form frmMain. Untuk selanjutnya membaca konfigurasi dari agen_kendali yang diakses sebelumnya, yang tersimpan dalam file **REMOTECONTROL.INI**, yang terdapat di lokasi standar direktori instalasi sistem operasi Windows pada masing-masing komputer pengguna. Pada sistem yang penulis gunakan, direktori standar berada pada lokasi **C:\WINDOWS**. Fitur untuk membaca isi dari file konfigurasi tersebut disimpan dalam suatu modul bernama **basregistry**. Seperti yang terlihat pada kode berikut:


```

With frmMain
  .lstMain.AddItem "Info Koneksi"
  .lstMain.AddItem "Info Sistem"
  .lstMain.AddItem "Pemantauan"
  .lstMain.AddItem "Akses Harddisk"
  .lstMain.AddItem "Akses Keyboard"
  .lstMain.AddItem "Akses TaskManager"
  .lstMain.AddItem "Kendali Desktop"
  .txtHost.Text = ReadINI("RememberedSettings", "Host", "remotecontrol.ini")
  .txtPort.Text = ReadINI("RememberedSettings", "Port", "remotecontrol.ini")

  basDeclare.LastPath = basRegistry.ReadINI("Settings", "LastPath",
"remotecontrol.ini")
End With

Public Function ReadINI(Section, KeyName, FileName As String) As String
  Dim sRet As String
  If FileName = "" Then FileName = "Remote.ini"
  sRet = String(255, Chr(0))
  ReadINI = Left(sRet, GetPrivateProfileString(Section, KeyName, "", sRet,
Len(sRet), FileName))
End Function

```

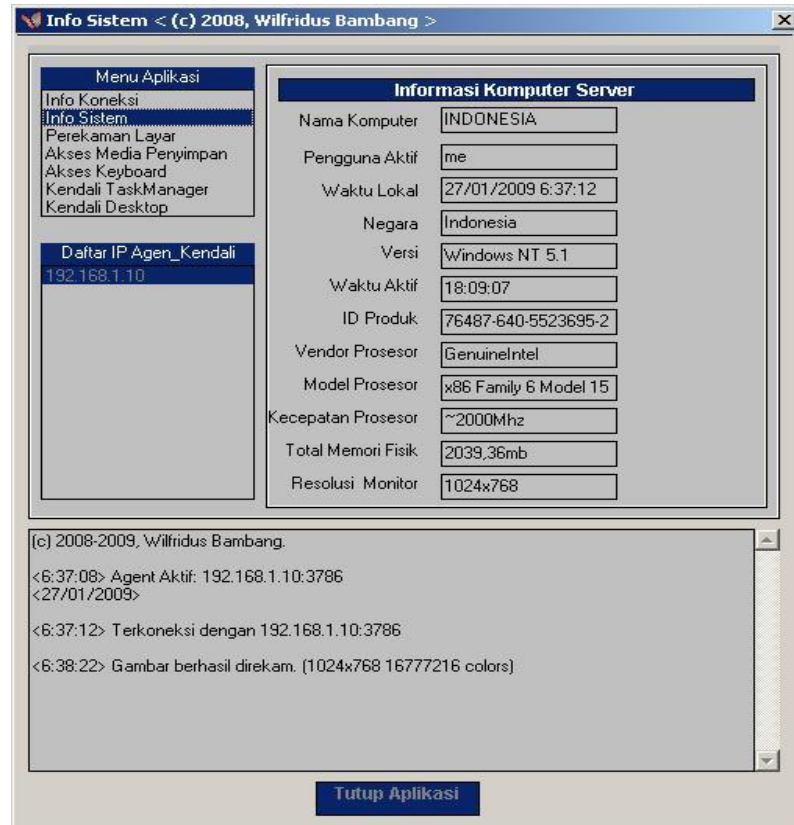
Konfigurasi protokol yang akan digunakan untuk komunikasi antara aplikasi pada sisi pengendali dengan aplikasi pada sisi agen_kendali, menggunakan protokol TCP melalui socket wskDeteksi sebagai sarana pengiriman permintaan layanan, pengesetan variabel nilai dari *port* sesuai dengan angka yang telah ditentukan sebelumnya yaitu 8504, sehingga socket siap untuk melakukan koneksitas yang diinginkan. Untuk socket wskDeteksi dikonfigurasi menggunakan index sebagai bantuan dalam menyimpan atau menuliskan nomor-nomor IP komputer agen_kendali pada *object* list pengguna.

```

wskDeteksi(0).Protocol = sckTCPProtocol
wskDeteksi(0).LocalPort = 8504
wskDeteksi(0).Bind wskDeteksi(0).LocalPort, wskDeteksi(0).LocalIP
wskDeteksi(0).Listen
SocketAkhir = 0

```

4.2.2. Pembahasan fitur Informasi Sistem



Gambar 4.2. Visualisasi fitur Informasi Sistem

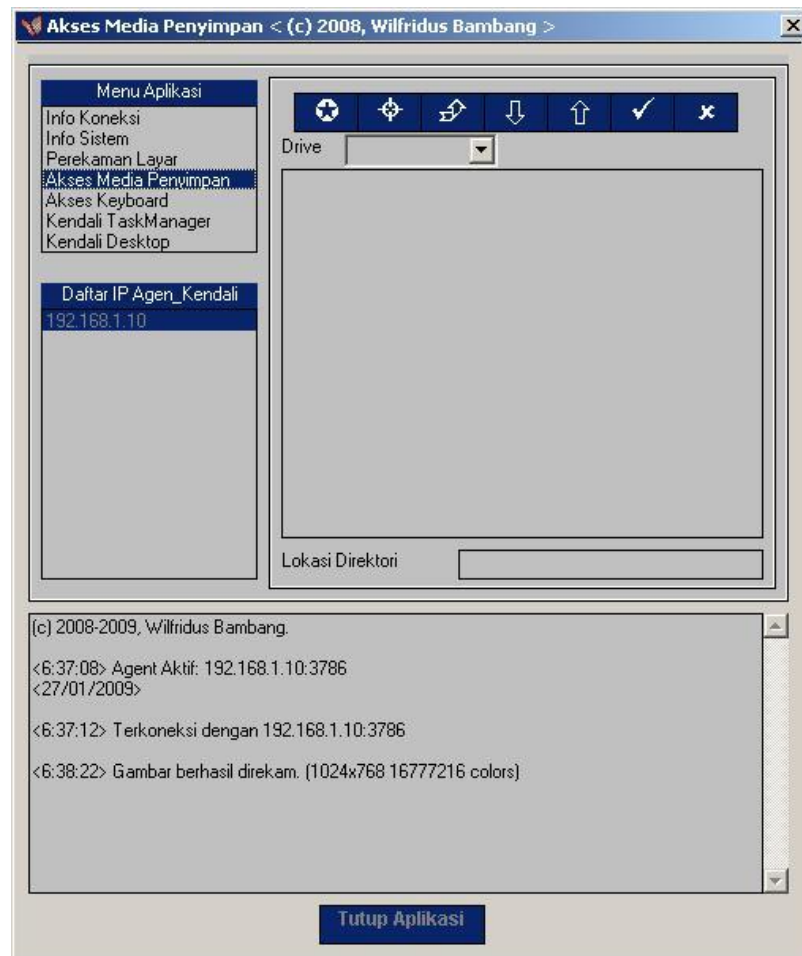
Gambar 4.2 memvisualisasikan fitur info sistem dimana pengguna pada sisi pengendali mendapatkan informasi-informasi penting dari sistem aktif pada sisi agen_kendali. Beberapa hal yang ditampilkan disini adalah:

- Nama komputer.
- Pengguna aktif
- Versi
- Total memori fisik.

Kebutuhan informasi akan disediakan oleh sistem, tetapi penting atau tidaknya informasi-informasi tersebut tergantung dari sisi pengguna dalam mencermati dan memproses suatu sistem. Karena informasi diatas sifatnya adalah informasi awal, dimana *level gathering information* dalam dunia sekuritas kompuer telah dilakukan dengan baik dan yang pasti secara tidak terdeteksi oleh target.

4.2.3. Pembahasan fitur Akses Media Penyimpan

Pengkaksesan sumber daya penyimpan data pada komputer agen_kendali juga dimungkinkan untuk dilakukan pada aplikasi ini. Menggunakan metode API dalam mengakses drive maupun direktori aktif, kemudian dapat pula melakukan beberapa eksekusi perintah seperti delete, upload, download, hingga mengeksekusi suatu file di komputer target menjadi pelengkap dari fitur utama dari bahasan ini, yaitu kemampuan menelusuri harddisk seperti halnya menggunakan aplikasi windows explorer, tetapi dari jarak jauh, dan juga dengan kemampuan akses yang tinggi, karena menggunakan akses administrator. Untuk visualisasi dari tampilan aplikasi, dapat dilihat pada gambar 4.3.



Gambar 4.3. Visualisasi fitur akses media penyimpanan

Kode ini dibawah ini adalah bagaimana cara melakukan penelusuran harddisk, dengan secara spesifik mengakses direktori yang ada di dalamnya.

```

Function BrowseForFolder(Optional ByVal Title As String = "", Optional ByVal
RootDir As String = "", Optional ByVal StartDir As String = "", Optional Owner
As Form = Nothing, Optional ByVal NewStyle As Boolean = True, Optional ByVal
IncludeFiles As Boolean = False) As String
    Dim lpIDList As Long, lpIDList2 As Long, IDL As ITEMIDLIST
    Dim sBuffer As String, tBrowseInfo As BrowseInfo, r As Long

    If Len(RootDir) > 0 Then

        If PathIsDirectory(RootDir) Then
            SHParseDisplayName StrPtr(RootDir), ByVal 0&, lpIDList2, ByVal 0&,
ByVal 0&
            tBrowseInfo.pIDLRoot = lpIDList2
        Else
            r = SHGetSpecialFolderLocation(ByVal 0&, &H11, IDL) ‘ get “My
Computer” Folder
            If r = 0 Then tBrowseInfo.pIDLRoot = IDL.mkid.cb
        End If

    End If

    .....
    .....
    .....

End Function

```

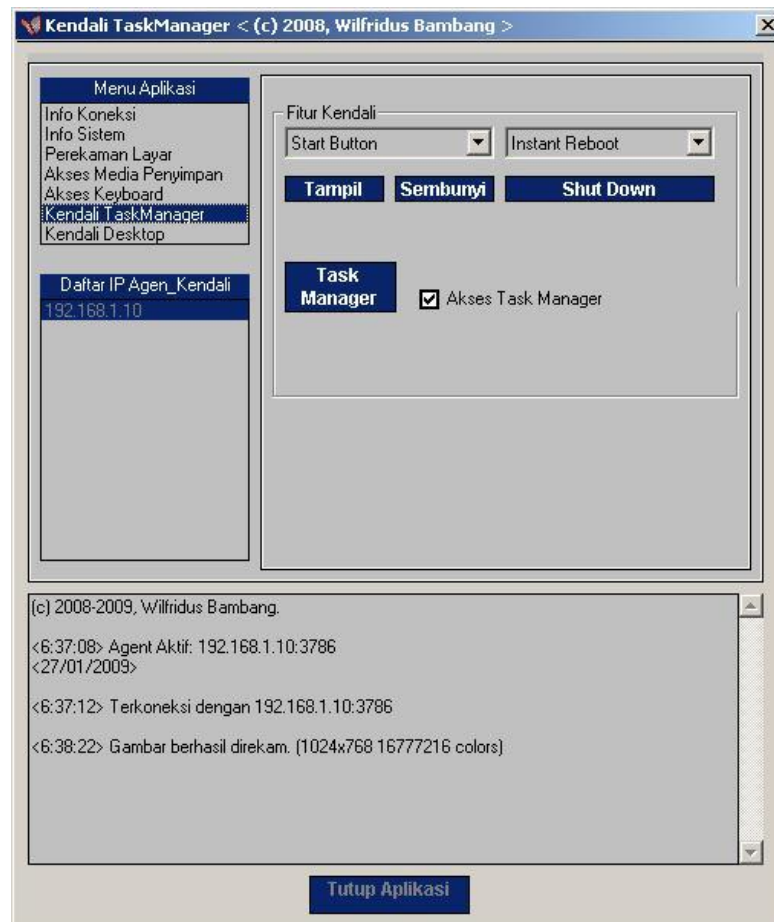
4.2.4. Pembahasan fitur Kendali TaskManager

Fitur yang terlihat pada gambar 4.4 adalah memberikan akses kepada pengguna di sisi pengendali dalam melakukan tindakan administrasi terhadap tray button maupun akses ke tingkat mematikan sistem yang berjalan aktif dengan memberikan pilihan perintah logoff, restart, maupun shutdown. Hal ini dimungkinkan dengan maksud memberi tanda kepada pengguna di sisi agen_kendali, bahwa komputer yang digunakannya adalah berada di bawah kendali dari administrator sebagai pihak yang berada di sisi pengendali. Berikut cuplikan kode pengiriman sinyal perintah ke agen_kendali untuk tray button maupun administrasi penghentian sistem.

```

Private Sub cmdShow_Click()
    Select Case Lcase(cmbHideShow.Text)
        Case "start button"
            Send "windows;show;startbutton;"
        Case "whole taskbar"
            Send "windows;show;taskbar;"
        Case "taskbar programs"
            Send "windows;show;taskbaritems;"
        .....
        .....
        .....
        Case "shutdown"
            Send "windows;power;shutdown;"
        Case "restart windows"
            Send "windows;power;restart;"
        Case "reboot computer"
            Send "windows;power;reboot;"
    End Select
End Sub

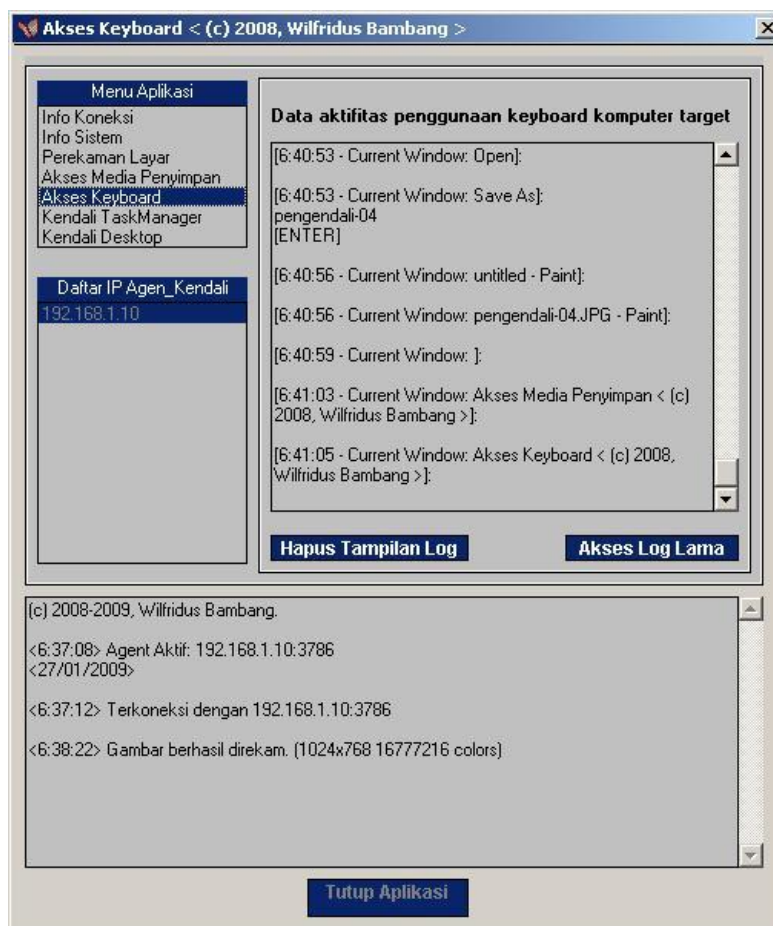
```



Gambar 4.4. Visualisasi fitur Kendali Task Manager

4.2.5. Pembahasan Fitur Perekaman Keyboard

Gambar 4.5 merupakan visualisasi dari tampilan pada sisi aplikasi pengendali, yang dapat pula mengakses data hasil aktifitas penekanan keyboard pada sisi agen_kendali, yang secara otomatis dikirimkan ke pengendali. Hal ini memungkinkan sisi pengendali memantau aktifitas yang dilakukan oleh pengguna di sisi agen_kendali, sehingga dapat diketahui maksud dan tujuan penggunaan komputer.



Gambar 4.5. Visualisasi Fitur Akses Keyboard.

Client dapat pula mengakses data-data hasil rekaman lama yang disimpan di suatu lokasi tersembunyi di komputer agen_kendali, yang dapat didownload dan dibaca karena sifatnya text. Hal ini juga memungkinkan pengendali mengetahui aktifitas yang dilakukan oleh pengguna di sisi agen_kendali sebelum dilakukan pemantauan.

4.2.6. Pembahasan fitur Perekaman Tampilan Layar

Salah satu fitur utama dari aplikasi pemantauan ini adalah melakukan perekaman tampilan layar yang sedang aktif di komputer yang sedang digunakan di sisi agen_kendali. Untuk visualisasinya dapat dilihat pada gambar 4.6. Dengan menggunakan file library **ZLIB.DLL** yang harus sebelumnya diinisialisasi atau disalin ke direktori sistem dari windows, maka sistem dapat bekerja secara optimal. Setiap satu layar monitor yang diakses dan direkam dalam fitur ini akan disimpan dengan nama preview.bmp, yang kemudian akan tampil pula ke sisi pengendali sebagai suatu berkas yang dapat disimpan. Tujuan akhir dari fitur ini adalah dapat digunakannya rekaman file tersebut sebagai bukti otentik suatu aktifitas pengguna.

```

Sub Preview(Data As String)
  Dim Temp() As String
  Dim ColorDepth As Long, PreviewWidth As Long, PreviewHeight As Long,
ByteArray() As Byte

  .....
  .....
  .....
  frmSpyView.PrevWidth = PreviewWidth
  frmSpyView.PrevHeight = PreviewHeight
  frmMain.imgScreenShot.Picture = LoadPicture(App.Path & "\preview.bmp")
End Sub

```




Gambar 4.6. Visualisasi fitur Perekaman Layar.

4.2.7. Pembahasan fitur Kendali Mouse & Keyboard

Pembahasan fungsi untuk mengontrol fitur kendali layar, seperti yang terlihat pada visualisasinya yang terlihat di gambar 4.7. Fitur ini akan mengirimkan konfigurasi yang dipilih oleh pengguna pada sisi pengendali, berupa skala dari layar, aktifasi mouse, aktifasi keyboard, hingga kedalaman warna dari screenshot yang muncul di monitor pengguna yang melakukan pengintaian. Kode yang dikirim ke sisi agen_kendali berupa kode rc.

```

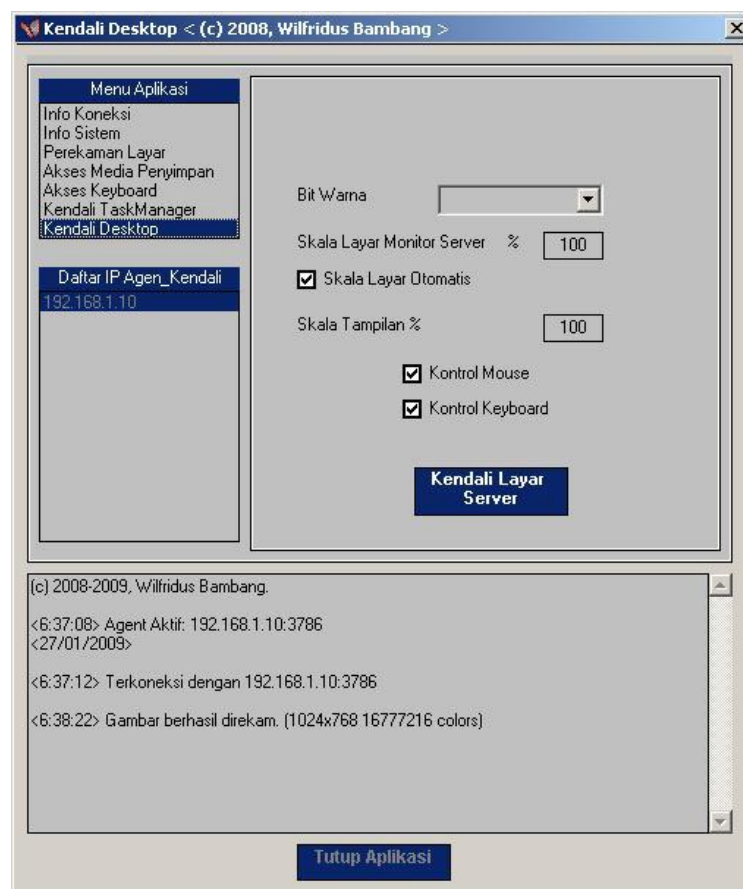
Private Sub cmdStartRemoteControl_Click()
    frmCapture.ClientScale = Val(txtViewScale.Text)

    Select Case chkControlMouse.value
    Case 0
        frmCapture.SendMouse = False
        frmCaptureTools.chkMouse.value = 0
    Case 1
        frmCapture.SendMouse = True
        frmCaptureTools.chkMouse.value = 1
    End Select

    .....
    .....
    .....

    Send "rc;" & Chr(3) & ";" & Chr(1) & Chr(0) &
    Val(txtHorizontalPartitions.Text) & Chr(0) & Val(txtVerticalPartitions.Text) &
    Chr(0) & Val(cmbColorDepth.Text) & Chr(0) & Val(txtHostScale.Text)
End Sub

```



Gambar 4.7. Visualisasi fitur kendali Mouse & Keyboard.

4.3. Implementasi Aplikasi Agen_Kendali

Implementasi aplikasi agen_kendali hanya membuat 2 form, yaitu form utama yang akan terlihat visualisasinya pada gambar 4.8 dan form yang akan meminta masukan alamat IP dari komputer pengendali serta port yang digunakan untuk jalur komunikasi, seperti yang terlihat pada gambar 4.9. Form isian alamat IP hanya akan tampil di layar apabila dalam pengecekan di awal ketika aplikasi dijalankan bahwa variabel penampung untuk alamat IP dan port masih kosong atau telah terhapus. Sistem kemudian akan meminta administrator untuk mengisikan data-data yang sesuai, untuk selanjutnya disimpan pada *file* bernama **LOG.INI** yang lokasi pada direktori sama dengan aplikasi agen_kendali. Jika pada waktu eksekusi aplikasi menemukan data yang diinginkan dengan membaca data dari *file* konfigurasi tersebut, maka fungsi utama dari aplikasi akan langsung dijalankan.

4.3.1. Fungsi Inisialisasi Koneksi

Aplikasi pada sisi pengendali yang berfungsi sebagai pemantau, diperlukan pula beberapa inisialisasi yang harus dilakukan agar dapat mengkoneksikan keberadaannya dengan komputer agen_kendali. Salah satu contoh yang dapat disebutkan adalah konsistensi dari penggunaan port. Pada perancangan kode sumbernya, digunakan banyak fungsi dalam WIN32API dalam pencapaian tujuan utama aplikasi ini dibangun. Visualisasi aplikasi agen_kendali dapat dilihat pada gambar 4.8.



The screenshot shows a window titled "Control Agent" with a sub-header "Informasi Koneksi". It contains several input fields: "IP Lokal" with the value "192.168.1.10", "Port Lokal" with the value "3786", "IP Client" (empty), and "Status:" (empty). Below these fields, there are two more input fields: "Byte Terima:" with the value "0b" and "Byte Kirim:" with the value "0b". A small icon with "<>" is positioned between the two "Byte" fields.

Gambar 4.8. Visualisasi aplikasi Server.



The screenshot shows a window titled "Alamat IP Pengendali". It contains three input fields: "Set IP Pengendali" (empty), "Port" (empty), and "Aktifasi Deteksi" (checkbox, unchecked). A "simpan" button is located at the bottom of the form.

Gambar 4.9. Visualisasi Form Konfigurasi alamat IP dan Port

Bagian agen_kendali ini adalah sifatnya menerima kode perintah yang dikirimkan oleh pengendali untuk selanjutnya dieksekusi sesuai kebutuhan. Kemudian pula aktivasi fitur-fitur yang disediakan juga dipengaruhi konfigurasi

yang dilakukan seperti pada bagian koneksi, perekaman, dan kendali layar. Semua fitur yang tampil pada menu aplikasi di sisi pengendali, akan mengirimkan kode-kode perintah secara spesifik yang dikenal oleh aplikasi penerima. Berikut cuplikan kode penerimaan perintah melalui socket yang aktif.

```

Private Sub wskInfo_DataArrival(ByVal bytesTotal As Long)
  Dim Data As String, Commands() As String, Temp() As String, Command As
String, Arguments As String
  wskInfo.PeekData Data
  If Right(Data, 2) <> "␣•" Then Exit Sub

  wskInfo.GetData Data
  Commands = Split(Data, "␣•")
  If UBound(Commands) < 2 Then Exit Sub

  TotalBytesReceived = TotalBytesReceived + Len(Data)

  .....
  .....
  .....

  Case "oldlogs"
    modProcess.ListOldLogs
  Case "settings"
    modProcess.SaveSettings Arguments
  Case "terminate"
    End
  End Select
NextCommand:
  Next i
End Sub

```

4.3.2. Fungsi Pendeteksi Aktifitas Keyboard

Rutin perintah di bawah ini adalah untuk melakukan pendeteksian penekanan tombol *keyboard*, yang selanjutnya akan dibaca kode yang muncul dengan kondisi-kondisi yang telah ditetapkan dalam suatu mekanisme perulangan secara terus menerus dalam proses penulisan ke *file* log. Menggunakan *object textbox* bantuan untuk meletakkan data pencatatan sementara, kemudian dalam

range interval waktu tertentu akan disimpan dalam *file* log yang telah ditentukan. Kode karakter yang terbaca berdasarkan inisialisasi pada pemrogramannya, seperti `vbKeyA` untuk huruf A, dan seterusnya, hingga mencakup semua karakter aktif yang dapat diakses oleh pengguna, untuk masing-masing karakter telah ditetapkan data *string* sebagai informasi yang diperlukan dari suatu kegiatan pencatatan penekanan *keyboard*. Untuk pemrosesan karakter yang memerlukan penekanan tombol shift juga telah disertakan pengecekan menggunakan mekanisme **IIF**, dengan contoh sintaks sebagai berikut ***Text = Shf(Shift = 1, "!", "I")***. Penjelasan dari contoh perintah di atas adalah, jika tombol shift ditekan, maka karakter pertama yang akan dicetak, sedangkan jika tombol shift tidak mengalami penekanan, maka karakter kedua yang akan dicetak. Fungsi API yang digunakan adalah **GetAsyncKeyState**. Berikut cuplikan singkat kode untuk mengenali penekanan keyboard oleh pengguna.

```

Private Sub CheckKey()
  Dim Keycode As Integer
  Dim X As Integer
  Dim Shift As Integer

  Do
  DoEvents
    If currentwindow <> GetCaption(GetForegroundWindow) Then
      currentwindow = GetCaption(GetForegroundWindow)
    TxtLog.Text = TxtLog.Text & vbCrLf & "[" & Time & " - Current
    Window: " & currentwindow & "]:" & vbCrLf
    End If

    For Keycode = 8 To 255
      X = GetAsyncKeyState(Keycode) 'get the state of the key
      If X = -32767 Then
        Select Case Keycode
          Case 8 'backspace
            TxtLog.Text = TxtLog.Text + " [BACKSPACE] "
            .....

```

Rutin program di bawah ini akan menuliskan data-data yang terdapat pada *textbox* yang telah berisi ketikan *keyboard*, ke dalam *file* log yang telah ditentukan. Sebagai nilai standar untuk melakukan penulisan, sehingga menggunakan status dari *object* timerLog yang akan aktif menjalankan perekaman ke *file* text setiap 6 detik. Agar buffer tidak melakukan penulisan berkali-kali, maka perlu untuk menjalankan perintah pengosongan *object* *textbox* yang digunakan sebagai penampung untuk perekaman setelah selesai melakukan penulisan ke dalam *file* log.

```

Private Sub WriteToLog()
  Open savefile For Append As #1
  Print #1, TxtLog.Text
  Close #1
  TxtLog.Text = ""
End Sub

```

4.3.3. Fungsi Kendali Mouse & Keyboard

Identifikasi pengiriman koordinat dari komputer pengendali berupa data-data koordinat dari lokasi mouse serta aktifitas yang dilakukan sesuai dengan instruksi dari administrator akan dibaca oleh fungsi PARSEMOUSEEVENT pada aplikasi agen_kendali. Data yang dikirim tersebut adalah masing-masing lokasi dari kursor mouse, dengan membaca pergerakannya dari atas, bawah, kiri, dan kanan. Untuk mengenali tindakan klik yang dilakukan oleh administrator, maka semua aktifitas klik kiri maupun klik kanan, akan diidentifikasi berupa posisi x dan y. sedangkan untuk klik double, ada pengecualian terhadap pengguna yang melakukan perubahan konfigurasi untuk mengeksekusi aplikasi di bagian kanan

dari mouse, karena pada aplikasi ini hanya mengkonfigurasi penggunaan mouse secara normal yaitu penekanan dua kali terhadap klik mouse di sebelah kiri. Untuk kode sumbernya dapat dilihat pada bagian berikut ini:

```

Sub ParseMouseEvent(ByVal Arguments As String)
  Dim Temp() As String, x As Long, y As Long

  If InStr(1, Arguments, Chr(0)) = 0 Then Exit Sub

  Temp() = Split(Arguments, Chr(0))

  If UBound(Temp) < 2 Then Exit Sub
  .....
  .....
  .....
  Case Chr(6)
    modMain.LeftDown x, y
    modMain.LeftUp x, y
    modMain.LeftDown x, y

```

Aktifitas penggunaan keyboard dari sisi administrator pengendali dikenali oleh aplikasi pada sisi agen_kendali pada fungsi PARSEKEYBOARDEVENT. Setiap penekanan tombol di keyboard pengendali akan dibaca sebagai satu karakter yang kemudian akan dikirim langsung ke pihak penerima untuk selanjutnya dikenali karakter tersebut. Selanjutnya akan ditampilkan pada layar komputer dari agen_kendali mengikuti instruksi asal dari pihak pengendali. Hal ini dimungkinkan dengan adanya identifikasi byte dari variabel keycode yang diaktifkan dari data yang diterima.


```

Sub ParseKeyboardEvent(ByVal Arguments As String)
  Dim Temp() As String, Keycode As Byte
  If InStr(1, Arguments, Chr(0)) = 0 Then Exit Sub
  Temp() = Split(Arguments, Chr(0))
  If UBound(Temp) < 1 Then Exit Sub
  Keycode = CByte(Val(Temp(1)))
  Select Case Temp(0)
    Case Chr(1)
      modMain.KeyDown Keycode
    Case Chr(2)
      modMain.KeyUp Keycode
  End Select

```

4.3.4. Fungsi Administrasi Sistem

Pengenalan perintah untuk melakukan reboot, logoff, maupun shutdown secara remote oleh komputer pengendali ke komputer agen_kendali. Berikut akan diberikan terlebih dahulu deklarasi serta opsi yang dikenal oleh berkas pustaka **USER32.DLL** yang terdapat di direktori sistem dari Microsoft windows.

```

Declare Function ExitWindowsEx Lib "user32" (ByVal uFlags As Long, ByVal
dwReserved As Long) As Long

Public Const EW_RESTARTWINDOWS = 4
Public Const EW_REBOOTSYSTEM = 2
Public Const EW_SHUTDOWNWINDOWS = 1
Public Const EW_LOGOFFWINDOWS = 0

```

Implementasinya dalam mengeksekusi perintah untuk melakukan reboot system sesuai dengan kode yang dikirimkan dari sisi pengendali adalah sebagai berikut:

```

Function RebootSystem()
  Dim RetVal As Integer
  RetVal = ExitWindowsEx(EW_REBOOTSYSTEM, 0)
  RebootComputer = RetVal
End Function

```

V. ANALISIS HASIL

5.1. Analisis Hasil

Mengetahui hasil kerja dari perangkat lunak ini secara rinci serta untuk mengetahui tingkat keberhasilan penggunaannya, telah diujicobakan penerapannya secara langsung pada suatu area jaringan lokal dan juga server virtualisasi di rumah tinggal dari penulis. Untuk hasil pengujian dapat dilihat pada tabel 5.1 untuk implementasi aplikasi pada sisi pengendali dan tabel 5.2 untuk implementasi aplikasi pada sisi agen_kendali, dengan menggunakan 3 komputer sebagai sistem uji coba.

Penulis melakukan pemantauan dari 1 mesin yang diposisikan sebagai pengendali pada komputer dengan alamat IP, misalkan menggunakan alamat sebagai berikut: 192.168.1.10, dan melakukan inisialisasi dua (2) komputer agen_kendali dalam satu area yaitu bunda dan anya. Basis dari sistem operasi yang telah ter-*install* adalah menggunakan Microsoft Windows XP baik pada sisi pengendali ataupun agen_kendali. Apabila pengguna target menggunakan versi sistem operasi Windows yang lain mulai dari Windows 98 SE, Windows ME, Windows Server 2000/2003, hingga Windows Vista 32 bit dan 64 bit, dapat disimpulkan bahwa aplikasi dapat berjalan dengan baik, karena dalam proses pembangunannya, aplikasi menggunakan fungsi-fungsi pustaka dari Windows yang secara umum akan terdapat di versi keluaran terakhir dari Microsoft.

Ujicoba pada fitur pengaksesan media penyimpanan menggunakan perangkat keras tambahan seperti flashdisk maupun harddisk external, dapat dikenali dengan

baik oleh aplikasi. Pengenalan terhadap perangkat keras tambahan tersebut belum dapat secara otomatis, tetapi pengguna harus mengklik tombol “Daftar drive server” agar informasi partisi atau drive dapat diperbaharui. Tidak menjadi masalah apakah perangkat tambahan tersebut diaktifkan ataupun dinonaktifkan setelah sistem pada agen_kendali terkoneksi dengan sitem pengendali..

Tabel 5.1. Hasil ujicoba implementasi aplikasi Pengendali

Aktifitas	192.168.1.10
Inisialisasi <i>Server</i>	Berhasil
Fitur Info Koneksi	Berhasil
Fitur Info Sistem	Berhasil
Fitur Pemantauan	Berhasil
Fitur Akses Harddisk	Berhasil
Fitur Akses Keyboard	Berhasil
Fitur Akses TaskManager	Berhasil
Fitur Kendali Layar	Berhasil

Indikator dari hasil ujicoba implementasi seperti yang terlihat pada informasi pada tabel 5.1 menunjukkan bahwa pada sisi pengendali semua fitur tersebut dapat diakses oleh pengguna, dengan melihat bahwa ketika fungsi-fungsi tersebut diklik sesuai dengan menu yang tersedia, maka akan tampil di layer pengguna, dan kemudian pengguna dapat mengeksekusi satu persatu fitur yang ada tersebut tanpa ada kegagalan atau pesan kesalahan dari aplikasi.

Tabel 5.2. Hasil implementasi aplikasi Agen_Kendali

	Bunda 192.168.1.4	Anya 192.168.1.5
Inisialisasi Agen_Kendali	Berhasil	Berhasil
Terlihat di TaskList dengan nama aplikasi yang tersamarkan	Berhasil	Berhasil
Koneksi Dengan Pengendali	Berhasil	Berhasil
Akses Informasi Sistem	Berhasil	Berhasil
Akses Media Penyimpan	Berhasil	Berhasil
Upload File	Berhasil	Berhasil
Download File	Berhasil	Berhasil
Pemantauan Layar	Berhasil	Berhasil
Perekaman tampilan layar	Berhasil	Berhasil
Askes keyboard	Berhasil	Berhasil
Kendali Task Manager	Berhasil	Berhasil
Kendali Mouse & Keyboard	Berhasil	Berhasil
Remote Shutdown	Berhasil	Gagal
Remote Restart	Berhasil	Gagal
Reset Agen_Kendali	Gagal	Berhasil
Kelambatan Traffic	Tidak	Tidak

Indikator pengujian pada tabel 5.2 dimaksudkan untuk menjelaskan apakah kegiatan atau aktifitas yang dilakukan pada masing-masing fitur perintah yang diberikan dari pengendali, dapat dijalankan atau dieksekusi pada komputer yang menjalankan aplikasi agen_kendali. Parameter yang diberikan pada masing-masing pengujian adalah sama, sehingga diharapkan akan diperoleh hasil akhir atau kesimpulan dari tahap ujicoba dan implementasi dari aplikasi.

Hasil proses pengujian, secara garis besar aplikasi dapat berjalan dengan baik. Walaupun terdapat beberapa kegagalan akses dari fitur-fitur yang ada, seperti yang terlihat pada tabel hasil implementasi aplikasi agen_kendali di tabel 5.2. Pada komputer dengan nomor IP tertentu, sebagai contoh menggunakan nomor 192.168.1.4, terjadi kegagalan pada bagian reset agen_kendali. Hal ini dikarenakan aplikasi server yang sifatnya inisialisasi koneksi dan mekanisme kerja dari aplikasi dideteksi sebagai Trojan dari antivirus yang aktif. Hal ini terjadi karena pengguna tersebut melakukan instalasi lebih dari 2 aplikasi antivirus, sehingga sistem akan lebih baik dalam pengenalan aplikasi-aplikasi berbahaya.

Komputer dengan IP tertentu, sebagai contoh menggunakan alamat 192.168.1.5, mengalami kegagalan yang terjadi sewaktu melakukan eksekusi remote shutdown dan remote restart. Hal ini dikarenakan sistem yang terpasang pada komputer tujuan mengalami kerusakan. Setelah ditelaah lebih jauh, maka beberapa fitur dasar dari Microsoft windows tidak dapat berjalan dengan baik, seperti halnya shutdown, restart, maupun logoff, sehingga saran yang diberikan penulis adalah sistem target harus diinstall ulang.

Penulis melakukan proses ujicoba implementasi aplikasi agen_kendali pada beberapa varian sistem operasi Microsoft Windows, yang dapat dilihat keberhasilannya pada tabel 5.3, sehingga dapat diperoleh gambaran sampai sejauh mana kompatibilitas aplikasi dalam penyesuaian terhadap berbagai varian sistem operasi dari Microsoft. Ujicoba dilakukan menggunakan aplikasi virtualisasi sistem operasi VMWARE versi 6.5, dengan kelebihan adalah memudahkan penulis untuk melakukan proses instalasi berbagai sistem operasi tanpa harus

melakukan interaksi secara fisik terhadap sumber daya harddisk. Sebagai tambahan, pengguna dapat pula menggunakan berbagai alternatif virtualisasi lain, sebagai contoh Virtual PC dan Virtual Box.

Tabel 5.3. Implementasi aplikasi *control agent* pada varian SO Windows

Varian SO Windows	Status	Keterangan
Windows 98 SE	Berhasil	Memerlukan <i>file</i> MSVBVM60.DLL
Windows XP SP2	Berhasil	
Windows Server 2003 Enterprise	Berhasil	
Windows Vista 32 Bit	Berhasil	Memerlukan <i>file</i> MSWINSCK.OCX versi 6.00.81694.
Windows Vista 64 Bit	Berhasil	Penempatan <i>file</i> MSWINSCK.OCX berada di lokasi direktori \Windows\System

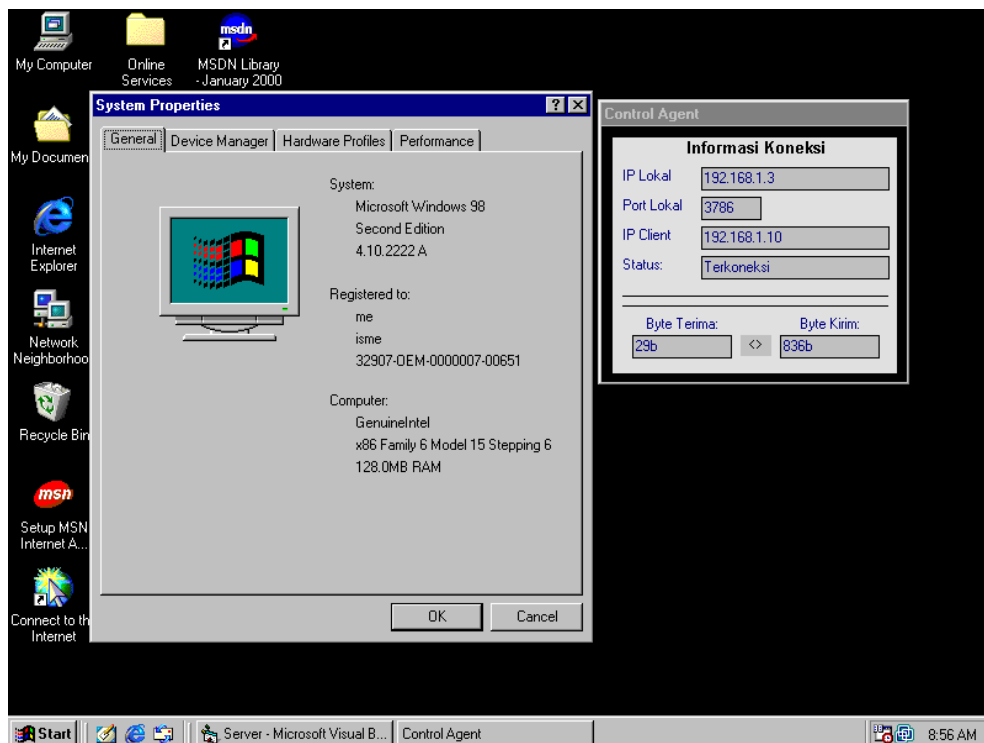
Tabel 5.3 memperlihatkan bahwa aplikasi *control agent* yang diimplementasikan pada komputer yang akan dipantau dapat bekerja pada banyak varian sistem operasi keluaran Microsoft. Meski pada waktu ujicoba terdapat perlakuan khusus di Windows Vista 32 bit yaitu harus menggunakan *file library* **MSWINSCK.OCX** versi 6.00.81694. Informasi tersebut penulis peroleh dari hasil penelusuran di forum-forum IT di Internet. Jika versi *file library* tersebut menggunakan versi yang lain yang lebih baru, maka akan timbul error pada waktu aplikasi *agen_kendali* dijalankan karena salah inisialisasi fungsi dan juga membutuhkan ketergantungan terhadap *file library* lain yang lebih baru sehingga

pengguna perlu melakukan proses *update library* secara simultan. Gambar 5.1 sampai 5.5 memvisualisasikan keberhasilan implementasi aplikasi *agen_kendali* pada masing-masing sistem operasi.

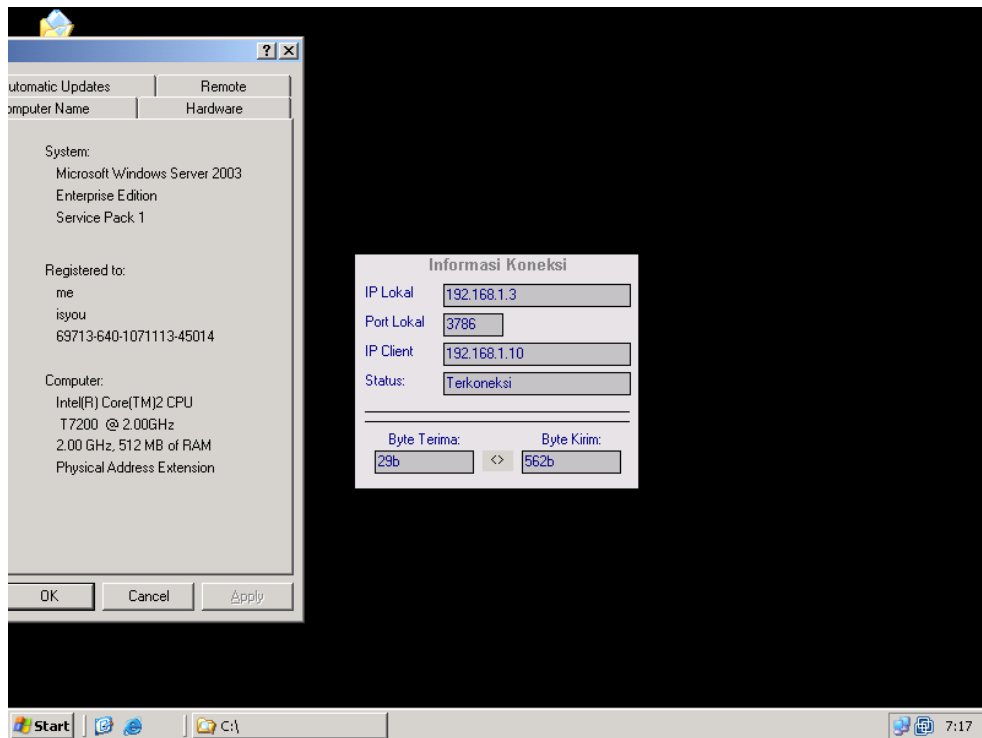
Sistem yang telah dikonfigurasi pada registry untuk menjalankan secara otomatis aplikasi setelah proses *booting*, menjadi solusi untuk tindakan administrasi dari pihak pengguna yang suatu waktu melakukan penambahan *user* pada sistem. Pemilihan lokasi dari aplikasi *agen_kendali* pada direktori sistem yang tidak disadari oleh pengguna umum, seperti misalnya ditempatkan pada direktori standar windows pada komputer penulis yaitu C:\Windows\System32, menjadi pilihan yang tepat. Kemudian untuk mencegah upaya penghilangan aplikasi secara paksa oleh pengguna, maka sistem akan secara otomatis memiliki duplikasi dari *file* *agen_kendali* yang akan ditempatkan pada lokasi direktori yang tersembunyi.

Aplikasi *agen_kendali* juga dapat dieksekusi pada arsitektur komputer yang berbasis 64 bit. Hal ini dibuktikan pada ujicoba untuk diimplementasikan di sistem operasi Windows Vista 64 bit. Tetapi terdapat sedikit perlakuan khusus pada penanganan *file* sistem yang digunakan oleh aplikasi, yaitu **MSWINSCK.OCX** dan **ZLIB.DLL** yang harus ditempatkan pada lokasi direktori \Windows\System, yang berbeda dengan implementasi di versi Windows yang lain yang ditempatkan pada direktori \Windows\System32. Kemudian pengguna perlu terlebih dahulu melakukan proses register file **MSWINSCK.OCX** sehingga dapat diinisialisasi fungsinya oleh sistem. Tahapan register tersebut administrator dapat melalui utility “Command Prompt” dengan memilih bekerja sebagai

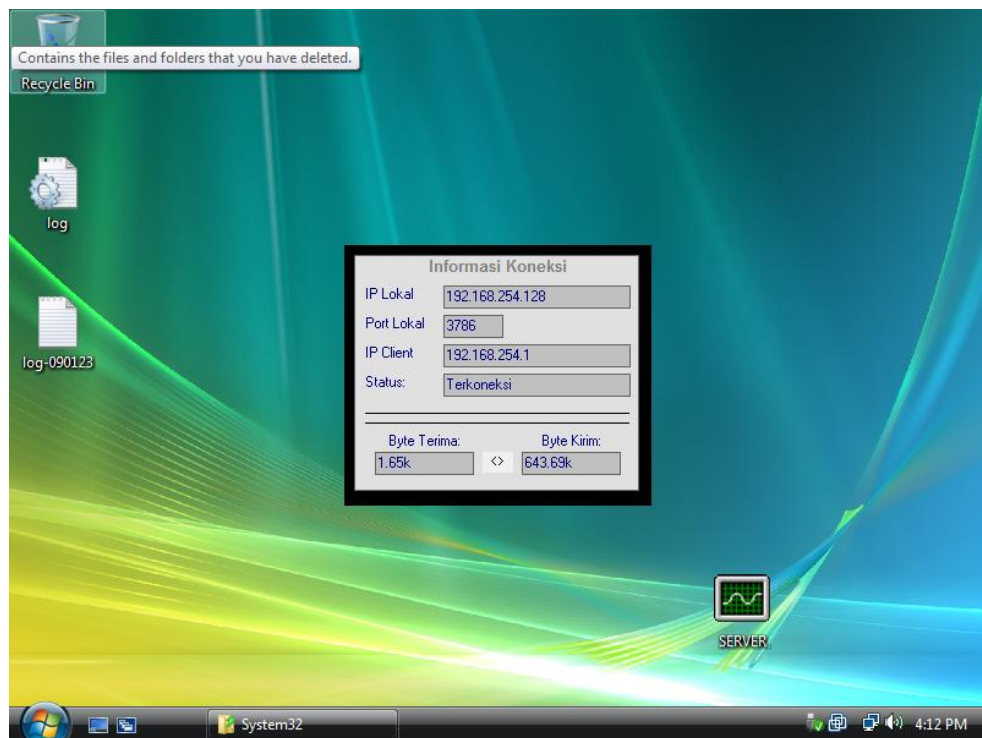
administrator (caranya adalah dengan melakukan klik kanan pada *shortcut* “command prompt” dan memilih menu “Run as Administrator”). Kemudian administrator akan mendapat tampilan mode DOS untuk selanjutnya mengetikkan perintah **REGSVR32 MSWINSCK.OCX** untuk melakukan register *file* ke dalam sistem.



Gambar 5.1. Implementasi pada sistem operasi Windows 98 SE.



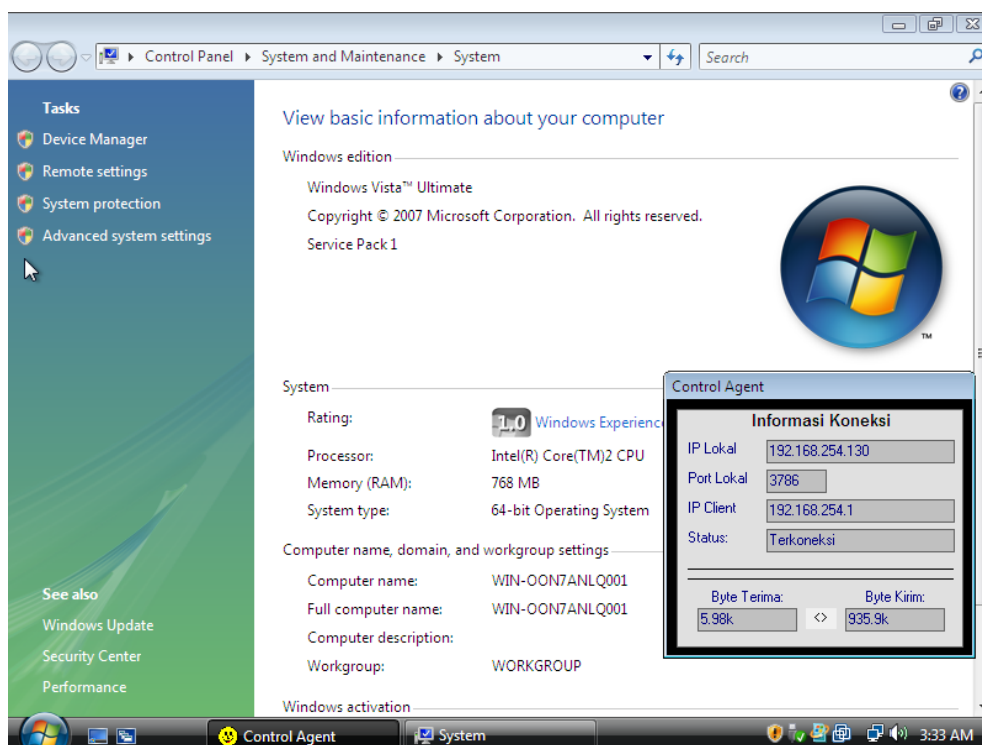
Gambar 5.2. Implementasi pada sistem operasi Windows Server 2003.



Gambar 5.3. Implementasi pada sistem operasi Windows Vista 32 bit.

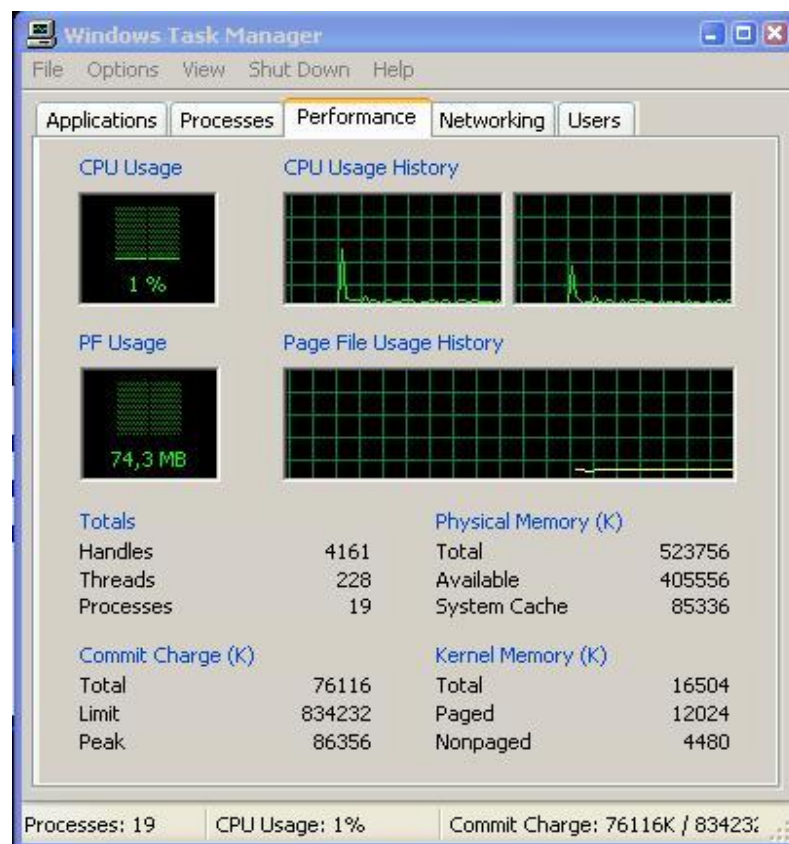


Gambar 5.4. Implementasi pada sistem operasi Windows XP SP2.



Gambar 5.5. Implementasi pada sistem operasi Windows Vista 64 bit.

Visualisasi kinerja dari *resources* pada mesin agen_kendali selama fungsi perekaman dijalankan, dapat menggunakan aplikasi standar dari Microsoft Windows, yaitu Task Manager. Untuk mengaktifkannya, lakukan penekanan tombol pada keyboard secara bersamaan, yaitu CTRL+ALT+DEL, dan akan tampil visualisasi seperti yang terlihat pada gambar 5.6.



Gambar 5.6. Kondisi sistem setelah setelah aplikasi diaktifkan.

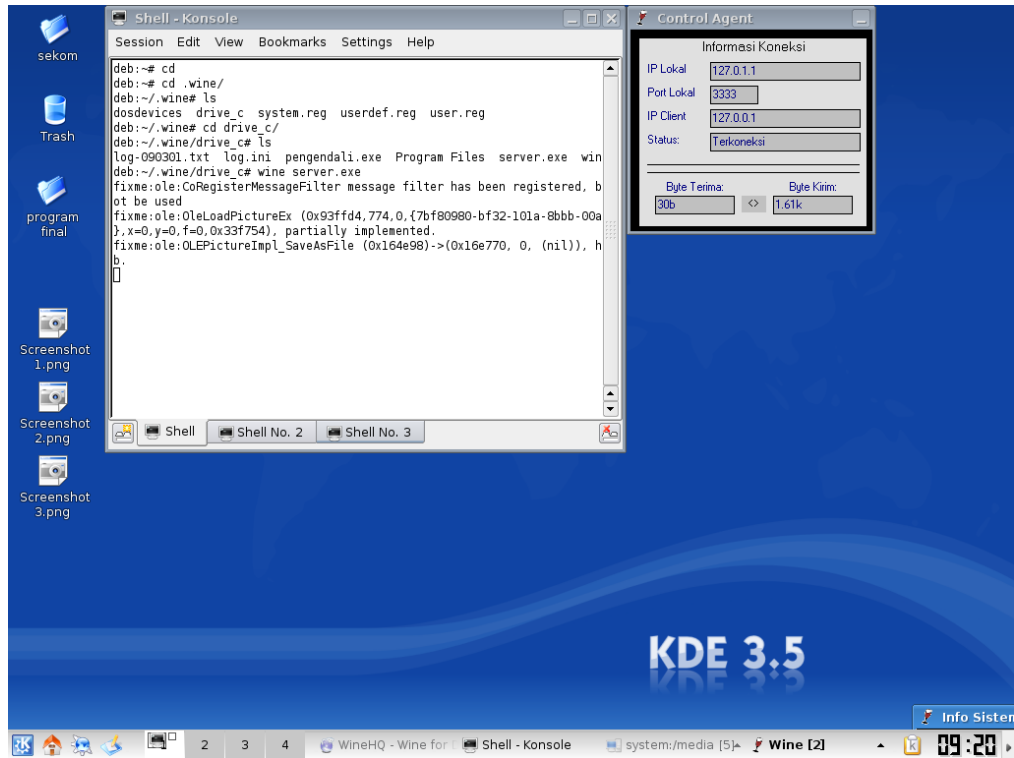
Kinerja sistem pada waktu ujicoba tidak akan terpengaruh secara signifikan, karena selain ujicoba ini dilakukan pada komputer yang spesifikasi perangkat kerasnya termasuk baik, juga karena aplikasi hanya menggunakan sumberdaya dari sistem secara minimal. Hal ini didukung oleh sifat eksekusi

aplikasi yang menunggu instruksi dari sisi pengendali, untuk selanjutnya menjalankan fitur yang diminta.

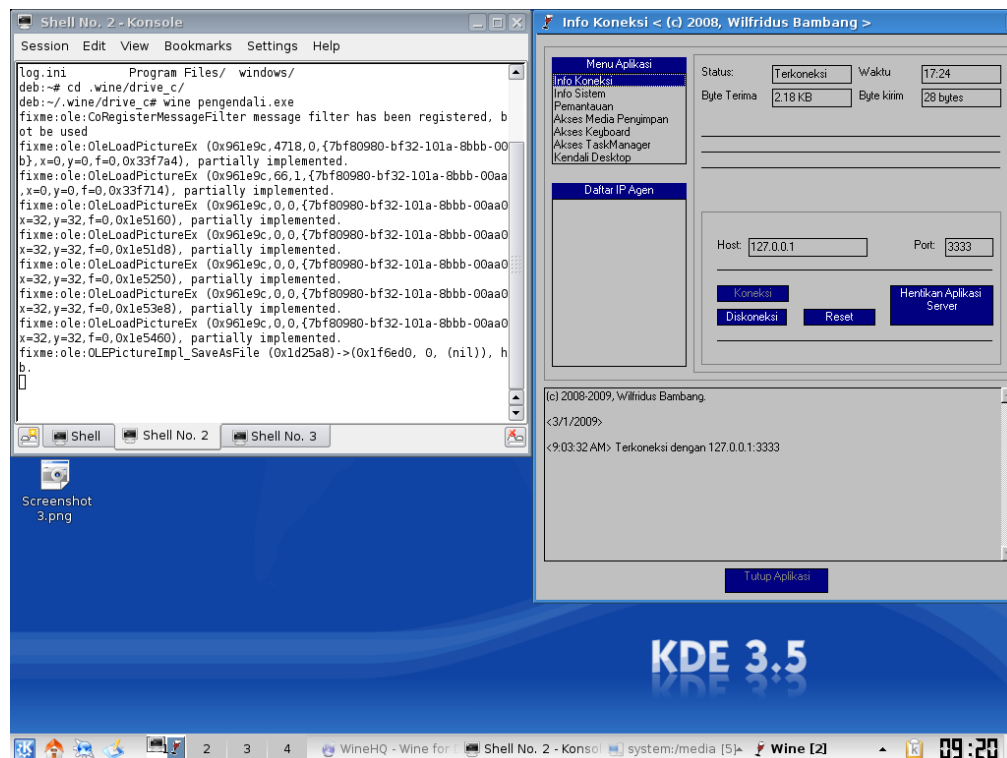
Penerapan identifikasi nomor IP komputer pada jaringan menggunakan metode DHCP (*Dynamic Host Configuration Protocol*) dimana komputer yang terhubung pada jaringan akan diberikan alamat IP yang acak tidak akan mengganggu kinerja dari aplikasi. Hal ini disebabkan karena pada waktu implementasi pertama kali ketika aplikasi *agen_kendali* dijalankan pada komputer, maka administrator akan melakukan konfigurasi terhadap aplikasi dengan memberikan nomor IP dari komputer pengendali serta nomor Port yang akan digunakan. Hal tersebut dilakukan dengan asumsi bahwa komputer yang menjalankan aplikasi pengendali adalah diberikan alamat IP secara static, yang berada dalam subnet yang sama dengan komputer yang menjalankan aplikasi *agen_kendali*.

Ujicoba menjalankan aplikasi pengendali dan *agen_kendali* pada sistem operasi Linux dilakukan dengan menggunakan aplikasi **WINE** (<http://www.winehq.org>). Distro Linux yang digunakan untuk ujicoba ini adalah Debian Etch v4.0, dengan versi dari wine yang digunakan adalah v0.9.25. Kedua aplikasi dapat dieksekusi dengan baik seperti yang terlihat pada visualisasi pada gambar 5.7 dan 5.8, dengan langkah yang harus dilakukan terlebih dahulu pada waktu ujicoba adalah menyalin *file library* yang diperlukan oleh aplikasi ke dalam direktori sistem dari aplikasi wine yang terdapat pada lokasi **/root/.wine/drive_c/windows/system32/** yaitu *file-file* berikut ini: **ZLIB.DLL**,

MSWINSCK.OCX, COMDLG32.OCX, MSVBM60.DLL, dan MSCOMCTL.OCX.

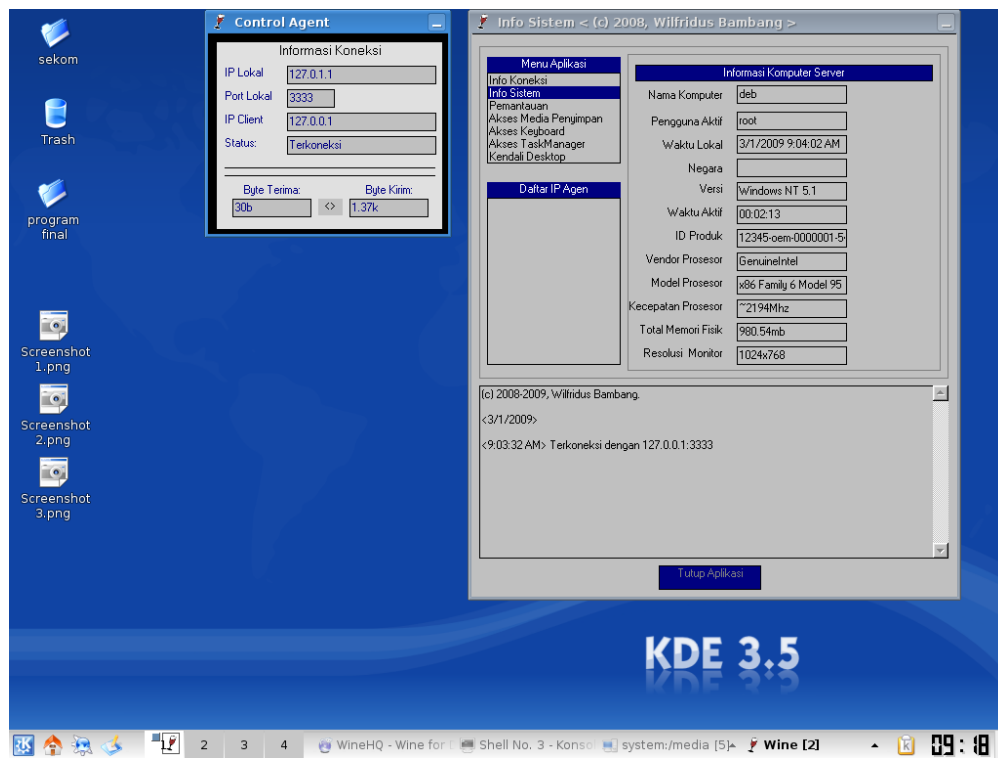


Gambar 5.7. Visualisasi eksekusi aplikasi agen_kendali dengan wine.



Gambar 5.8. Visualisasi eksekusi aplikasi pengendali dengan wine.

Fitur komunikasi dan identifikasi antara aplikasi pengendali dengan agen_kendali dapat berjalan dengan baik. Hal tersebut dapat dilihat dari dapat dilakukannya koneksi menggunakan alamat IP dan port seperti yang terlihat visualisasinya pada gambar 5.9. Pada bagian form informasi sistem juga dapat menampilkan informasi seperti ketika menjalankan aplikasi pada sistem yang berbasis Microsoft Windows. Untuk pendeteksian penekanan keyboard serta identifikasi nama aplikasi yang aktif juga berjalan dengan baik. Akan tetapi, tidak semua fitur dapat berfungsi dalam eksekusi dengan emulator ini, seperti perekaman tampilan layar, pengaksesan media penyimpanan, dan pengendalian sumberdaya mouse dan keyboard. Hal ini diakibatkan karena fungsi-fungsi tersebut secara spesifik hanya dapat berjalan pada lingkungan (*environment*) Microsoft Windows.



Gambar 5.9. Visualisasi komunikasi aplikasi pengendali dengan agen_kendali.

VI. KESIMPULAN DAN SARAN

6.1. Kesimpulan

1. Perangkat lunak ini dapat membantu administrator jaringan untuk melakukan pemantauan terhadap penggunaan komputer dalam suatu jaringan, dengan melakukan pemantauan terhadap layar monitor serta disertai pula fungsi untuk mengendalikan keyboard dan mouse aktif dari pengguna. Selanjutnya dilengkapi pula dengan pemantauan terhadap penekanan keyboard. Informasi yang dihasilkan dari aktifitas pemantauan ini, menjadi suatu bukti digital yang otentik dan diperlukan untuk melakukan suatu reaksi atas aksi dari kegiatan yang tidak seharusnya dilakukan pada komputer publik.
2. Aplikasi yang diaktifkan pada suatu sistem dengan kondisi firewall yang aktif, harus dikonfigurasi berbagai *role* atau peraturan lalu lintas data pada firewall sehingga fitur-fitur yang tersedia dalam aplikasi dapat berjalan dengan baik. Hal ini dikarenakan aplikasi memerlukan aktifasi (kondisi *permit* atau *enable*) *dua* (2) port aktif yang digunakan untuk melakukan pendeteksian nomor IP dari komputer agen serta komunikasi utama antara aplikasi pengendali dan aplikasi *agen_kendali*.
3. Sistem yang akan menjalankan aplikasi pengendali maupun *agen_kendali* memerlukan *dua* (2) *file library* yaitu **ZLIB.DLL** yang digunakan untuk fungsi pembuatan dan kompresi data gambar, serta **MSWINSCK.OCX**, untuk memanfaatkan fitur-fitur pemrograman windows socket dari Microsoft.

6.2. Saran

Pengembangan perangkat lunak ini masih memiliki keterbatasan yang dapat dijadikan acuan untuk pembangunan di masa yang akan datang sehingga akan menciptakan suatu sistem pemantauan yang lebih sempurna. Salah satu contoh keterbatasan tersebut adalah aplikasi yang dibangun ini tidak dapat melakukan pemantauan banyak komputer target pada satu waktu. Sehingga pada waktu implementasinya administrator atau pengguna yang berada pada sisi pengendali mengakses secara bergantian komputer target tersebut. Hal ini tentu akan mempersulit pihak administrator jika harus melakukan administrasi ke banyak pengguna.

Keterbatasan lain adalah aktifitas pemantauan dan pengendalian hanya dapat berjalan di atas platform sistem operasi berbasis Microsoft Windows atau bersifat homogen, dan juga harus langsung berinteraksi dalam bentuk *desktop application*. Akan lebih baik jika aplikasi yang dikembangkan dapat bekerja pada berbagai macam platform sistem operasi lain seperti misalnya linux atau macintosh, dengan membangun aplikasi memanfaatkan bahasa pemrograman yang mendukung fitur tersebut yang salah satu contohnya adalah JAVA. Kemudian untuk tahap implementasinya, administrator juga dapat memanfaatkan media lain seperti misalnya aplikasi browser yang mendukung penggunaan java applet sehingga memungkinkan aplikasi berbasis java yang dibangun dapat diakses dan menjadi antarmuka pengendalian komputer target.

DAFTAR PUSTAKA

Gary Wroblewski, The Herbert H, and Grace A Dow. "Building a Virtual Infrastructure", Central Michigan University, 2007.

Knut Holmqvist, Till Halbach, and Thor Kristoffersen. "Virtualization as a Strategy for Maintaining Future Access to Multimedia Content", 2009 First International Conference on Advances in Multimedia, 2009.

Marcin Lubonski, Valérie Gay, and Andrew Simmonds, "A Conceptual Architecture for Adaptation in Remote Desktop Systems Driven by the User Perception of Multimedia", 2006.

Muhammad Wannous and Hiroshi Nakano, "NVLab, a Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network Computing Technologie", 2009.

"What is Virtualization", RedHat, 2006.

LAMPIRAN

Lampiran 1. Perancangan Antarmuka Perangkat Lunak *Remote desktop*

Perancangan ini akan membahas mengenai rancangan antarmuka perangkat lunak yang nantinya akan diimplementasikan. Antarmuka hanya disediakan untuk perangkat lunak *remote desktop* sisi pengendali, dimana pengguna pada sisi pengendali akan saling berinteraksi dengan perangkat lunak yang diimplementasikan pada sisi agen_kendali. Untuk visualisasi dari antarmuka, dapat dilihat pada gambar 1 sampai dengan gambar 8.

Info Koneksi < © 2008-2009, Wilfridus Bambang >

Menu Aplikasi

- Info Koneksi
- Info Sistem
- Perekaman Layar
- Akses Media Penyimpan
- Akses Keyboard
- Kendali TaskManager
- Kendali Desktop

Daftar IP Agen Kendali

Status Waktu

Byte Terima Byte Kirim

Host Port

Koneksi

Diskoneksi

Reset

Hentikan Aplikasi Server

© 2008-2009, Wilfridus Bambang

Tutup Aplikasi

Gambar 1. Perancangan Antarmuka Fungsi Info Koneksi

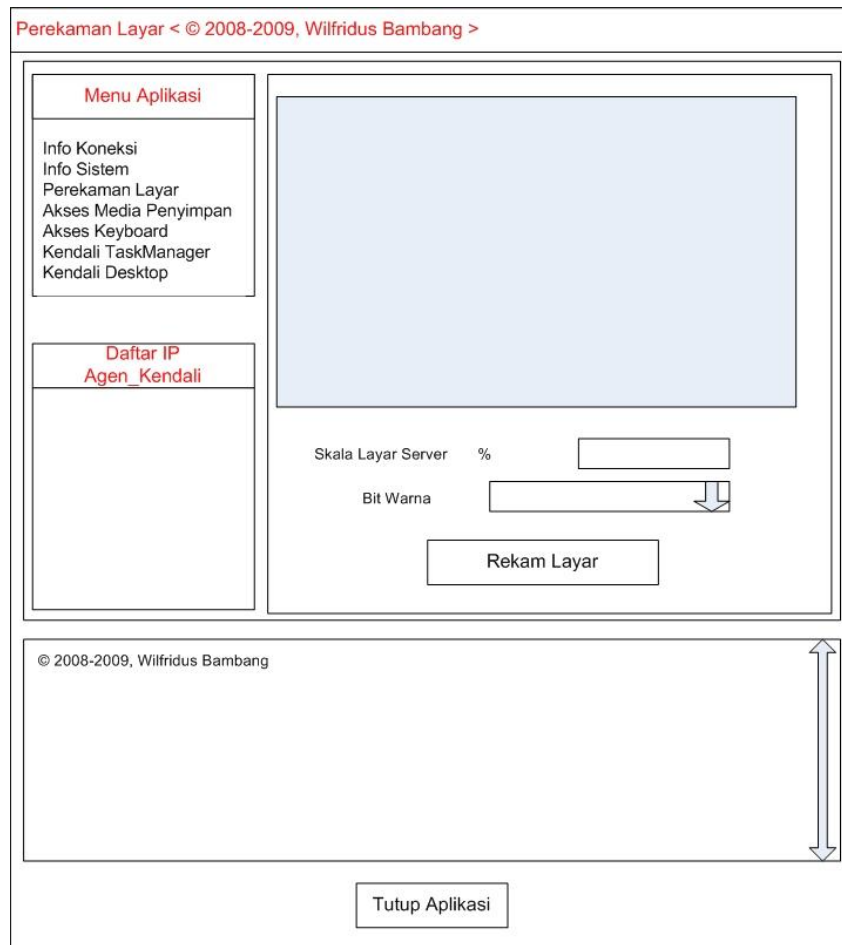
Info Sistem < © 2008-2009, Wilfridus Bambang >

<p>Menu Aplikasi</p> <p>Info Koneksi Info Sistem Perekaman Layar Akses Media Penyimpan Akses Keyboard Kendali TaskManager Kendali Desktop</p>	<p>Informasi Komputer Server</p> <p>Nama Komputer <input type="text"/></p> <p>Pengguna Aktif <input type="text"/></p> <p>Waktu Lokal <input type="text"/></p> <p>Negara <input type="text"/></p> <p>Versi <input type="text"/></p> <p>Waktu Aktif <input type="text"/></p> <p>ID Produk <input type="text"/></p> <p>Vendor Prosesor <input type="text"/></p> <p>Model Prosesor <input type="text"/></p> <p>Kecepatan Prosesor <input type="text"/></p> <p>Total Memori Fisik <input type="text"/></p> <p>Resolusi Monitor <input type="text"/></p>
<p>Daftar IP Agen_Kendali</p>	

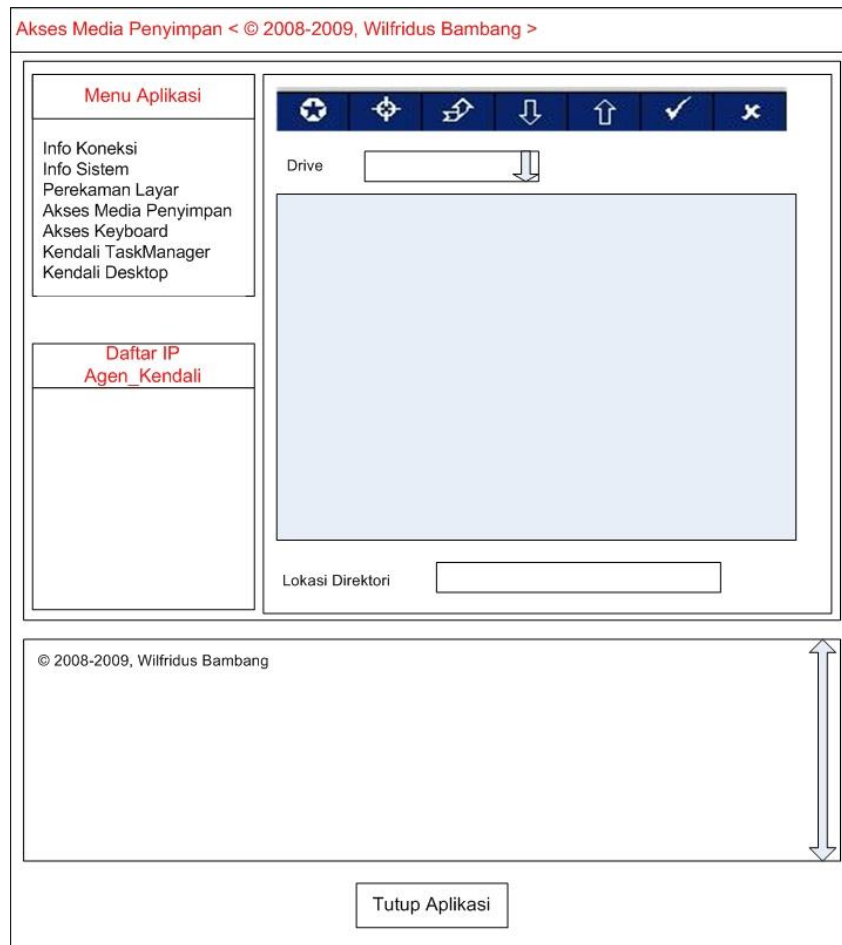
© 2008-2009, Wilfridus Bambang

Tutup Aplikasi

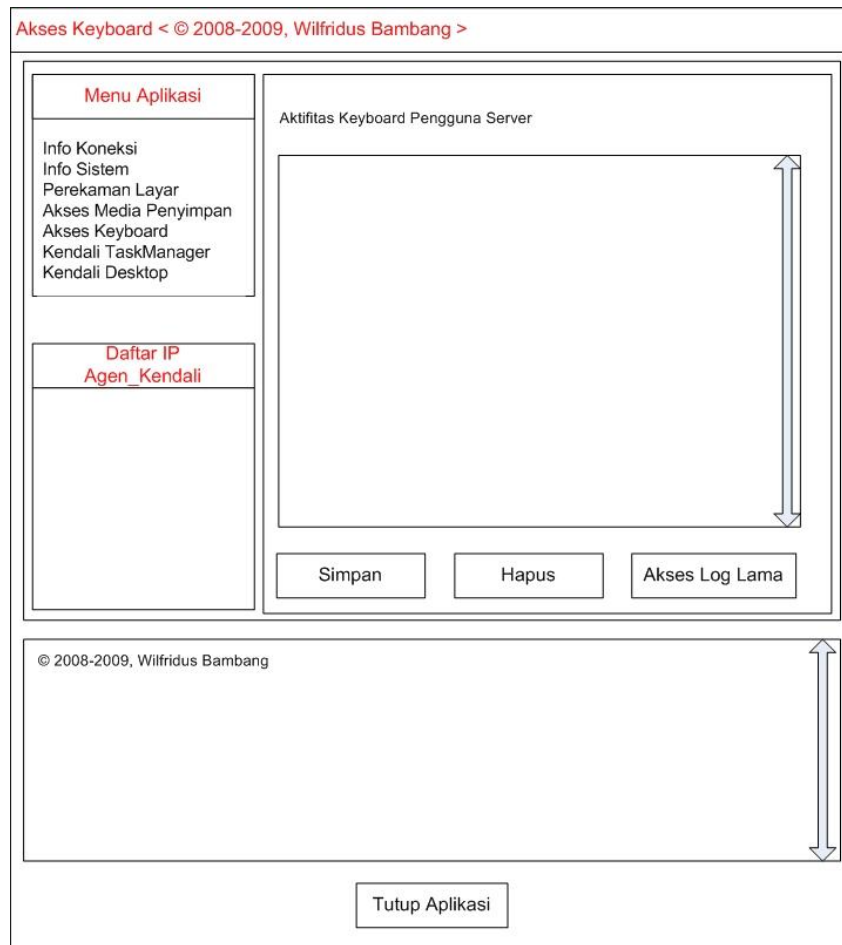
Gambar 2. Perancangan Antarmuka fungsi Info Sistem



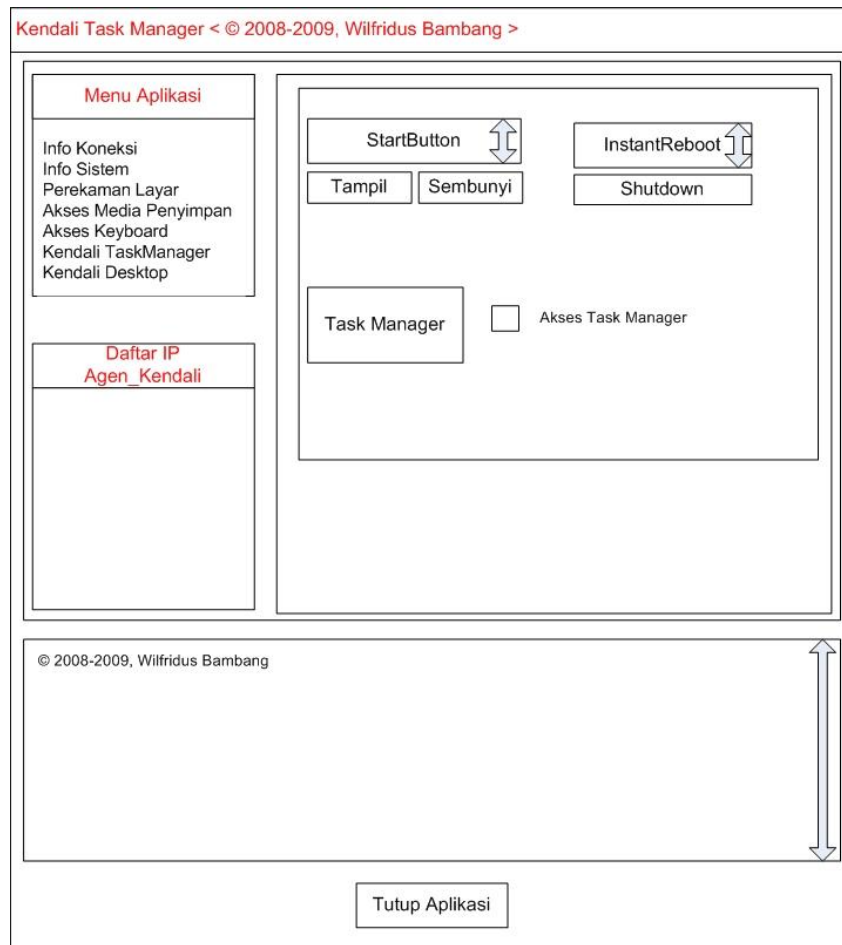
Gambar 3. Perancangan antarmuka fungsi Pemantauan



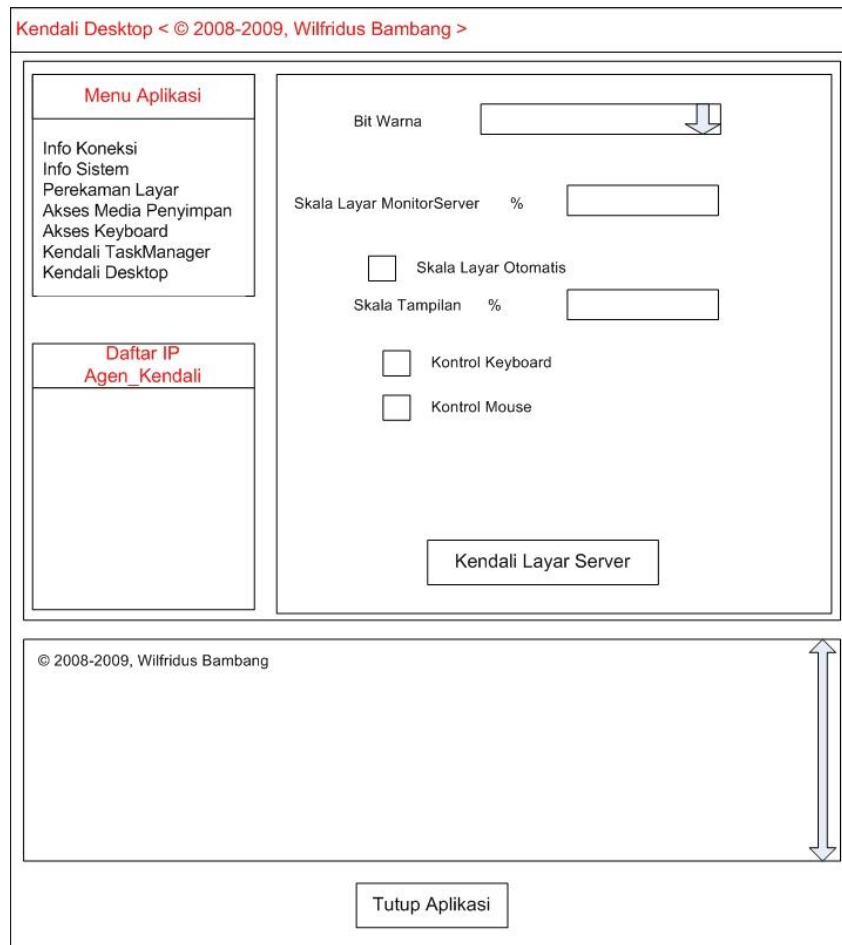
Gambar 4. Perancangan antarmuka fungsi Akses Harddisk



Gambar 5. Perancangan antarmuka Akses Keyboard



Gambar 6. Perancangan antarmuka fungsi Akses Task Manager



Gambar 7. Perancangan antarmuka fungsi Kendali Layar

Server

Informasi Koneksi

IP Lokal

Port Lokal

IP Client

Status

Kirim:

Terima:

Gambar 8. Perancangan antarmuka aplikasi Server