

# Monitoring Beberapa Komputer di Laboratorium UPH Surabaya

James Wijaya, Kuswara Setiawan,  
Program Studi Sistem Informasi  
Universitas Pelita Harapan Surabaya  
Surabaya, Indonesia  
james.wijaya@uph.edu  
kuswara.setiawan@uph.edu

**Abstrak** - Pengawasan terhadap kegiatan praktikum di Laboratorium Komputer UPH Surabaya perlu dilakukan, karena terdapat penggunaan komputer yang tidak semestinya, di mana mahasiswa membuka dan menggunakan program selain yang diperlukan pada waktu praktikum berlangsung. Kegiatan ini dapat menyebabkan terganggunya konsentrasi kegiatan praktikum. Namun terdapat kesulitan untuk melakukan pengawasan karena terbatas jumlah asisten dan banyaknya komputer yang digunakan.

Salah satu solusi untuk mengatasi permasalahan tersebut, maka dirancanglah sebuah program aplikasi yang dapat mengawasi komputer melalui Local Area Network (LAN). Aplikasi ini dibuat dengan memanfaatkan teknologi client-server, di mana client adalah komputer yang digunakan oleh mahasiswa, sedangkan server adalah komputer yang digunakan khusus untuk mengawasi client tersebut.

Berdasarkan hasil uji coba yang dilakukan, dapat disimpulkan bahwa aplikasi dapat memenuhi kebutuhan untuk mengawasi penggunaan komputer client di Laboratorium Komputer. Adapun fungsi-fungsi pengawasan yang dapat dilakukan, antara lain : melihat tampilan layar monitor client (Capture Screen Client), Pengiriman pesan ke client berupa informasi maupun peringatan (Send Message), melakukan remote restart dan shutdown, melakukan penguncian dan pembebasan kembali terhadap keyboard dan mouse (Remote Lock-Unlock Keyboard and Mouse), mengunduh file dari server, dan mengunggah file ke server (File Transfer).

**Kata kunci:** Pemrograman Client-Server, Pemrograman Soket, Sistem Pemantauan

## I. PENDAHULUAN

Sistem pemantauan yang dapat dijumpai dalam kehidupan sehari-hari adalah penggunaan kamera *Closed-Circuit Television* (CCTV) yang terpasang pada area-area yang dianggap penting. CCTV digunakan untuk mengamati pergerakan objek atau suatu kejadian yang terekam secara *real-time* pada area tertentu, yang telah dipasang kamera CCTV. Penggunaan dari kamera CCTV bertujuan untuk meminimalkan tindakan kriminal yang terjadi. Berangkat dari pemikiran tersebut dicoba untuk menerapkan sistem pemantauan tersebut pada Laboratorium Komputer Universitas Pelita Harapan Surabaya, guna mempermudah para Asisten Laboratorium dalam mengawasi praktikum.

Pemantauan atau *monitoring* yang ada saat ini di Laboratorium Komputer UPH Surabaya adalah pengawasan kegiatan praktikum yang dikoordinasi oleh asisten dan Kepala Laboratorium. Selama para praktikan mengerjakan tugas praktikum, maka Asisten Laboratorium akan mengamati pekerjaan praktikan, sehingga dapat meminimalkan kecurangan oleh praktikan atau pun hal-hal yang tidak diperkenankan selama praktikum berlangsung.

Terdapat beberapa masalah mengenai penggunaan komputer yang tidak semestinya pada Laboratorium Komputer, antara lain para praktikan membuka dan menggunakan program selain yang

diperlukan pada waktu praktikum berlangsung. Kegiatan ini dapat menyebabkan terganggunya konsentrasi pada kegiatan praktikum. Namun terdapat kesulitan untuk melakukan pengawasan karena terbatasnya jumlah asisten dan banyaknya komputer yang digunakan.

Sebagai salah satu solusi untuk mengatasi permasalahan tersebut, maka dirancanglah sebuah program aplikasi yang dapat mengawasi komputer dari jarak jauh dengan memanfaatkan jaringan komputer *Local Area Network* (LAN), berteknologi *client server*. Program aplikasi ini dirancang untuk menyediakan informasi keadaan di dalam laboratorium sehingga dapat membantu para asisten dan Kepala Laboratorium dalam melakukan pengawasan terhadap penggunaan komputer. Informasi ini meliputi data-data komputer yang sedang digunakan serta status komputer tersebut. Selain itu, terdapat fasilitas untuk mengirimkan pesan kepada praktikan, mengunci komputer praktikan, melakukan *shut down* dan *reboot* atau *restart* komputer. Dengan adanya aplikasi ini, diharapkan pengawasan atau *monitoring* menjadi lebih efektif dan efisien, yang semula secara langsung (berkeliling) kini hanya dilakukan dengan menggunakan komputer yang berperan sebagai *server* untuk dapat memantau beberapa komputer,

memberikan informasi mengenai keadaan layar komputer yang diawasi secara *real time*.

Arsitektur Aplikasi Sistem Pemantauan Komputer Client Pada Laboratorium Komputer yang akan dibuat adalah client-server desktop application. Aplikasi client ditaruh pada komputer yang digunakan oleh praktikan, sedangkan aplikasi server ditaruh pada komputer yang melakukan pemantauan.

## II. OSI LAYER

OSI (*Open System Interconnection*) adalah suatu standar komunikasi antarmesin yang terdiri atas tujuh lapisan. Ketujuh lapisan tersebut mempunyai peran dan fungsi yang berbeda satu sama lainnya. Setiap *layer* memiliki peran khusus pada proses komunikasi data. Model OSI *Layer* terbagi ke dalam dua kelompok, yaitu : *upper layer* yang berfokus pada *user application* dan bagaimana *file* ditampilkan pada komputer, dan *lower layer* yang menjadi bagian terpenting dalam komunikasi data melalui jaringan aktual.[1]

Tujuh lapisan dari Model OSI yaitu:

### 1) Physical Layer

**Layer ini** bertanggung jawab atas proses data menjadi bit dan mentransfernya melalui media, seperti kabel, dan menjaga koneksi fisik antar sistem.

### 2) Data Link Layer

**Layer ini** menyediakan *link* untuk data, memaketkannya menjadi *frame* yang berhubungan dengan *hardware* kemudian diangkut melalui media komunikasinya dengan kartu jaringan, mengatur komunikasi *layer physical* antara sistem koneksi dan penanganan *error*.

### 3) Network Layer

**Layer ini** bertanggung jawab menentukan alamat jaringan, menentukan rute yang harus diambil selama perjalanan, dan menjaga urutan antrian trafik di jaringan. Data pada *layer* ini berbentuk paket.

### 4) Transport Layer

**Layer ini** bertanggung jawab membagi data menjadi segmen, menjaga koneksi logika *end-to-end* antar terminal, dan menyediakan penanganan *error (error handling)*.

### 5) Session Layer

**Layer ini** menyiapkan saluran komunikasi dan terminal dalam hubungan antarterminal, mengkoordinasikan proses pengiriman dan penerimaan serta mengatur pertukaran data.

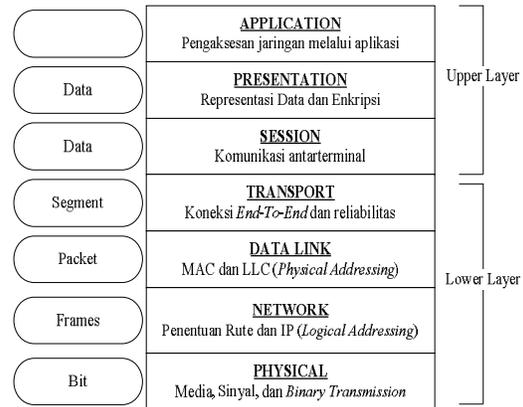
### 6) Presentation Layer

**Layer ini** bertanggung jawab bagaimana data dikonversi dan diformat untuk transfer data. Pada *layer* ini terjadi penerjemahan,

kompresi, dan enkripsi data.

### 7) Application Layer

*Layer* ini mengatur interaksi *user* dengan aplikasi (*software*) yang dipakai, mengatur bagaimana aplikasi tersebut dapat mengakses jaringan, dan membuat pesan-pesan tersebut dianggap *error*.



Gambar 1 : Tujuh Lapisan Model OSI

Terkait dengan pembagian layer tersebut, maka aplikasi yang dikembangkan terletak pada *Application Layer*.

## III. PEMROGRAMAN SOKET

Aplikasi yang dikembangkan di sini berjalan pada sistem operasi Windows terutama Windows XP untuk sisi *client* dan dikembangkan dengan menggunakan bahasa pemrograman Visual Basic (VB) .NET. Pemrograman socket pada .NET dapat digunakan dengan menggunakan *library* System.Net.Sockets. Model pemrograman socket yang digunakan di sini adalah *asynchronous*.

Pemrograman Soket *Asynchronous* atau pemrograman socket *event-driven* merupakan salah satu solusi pemrograman jaringan guna mengatasi komunikasi socket dengan banyak koneksi jaringan yang berbeda sekaligus dalam satu proses.[2] Dalam hal skalabilitas, model pemrograman *asynchronous* merupakan model terbaik, karena bertambahnya koneksi tidak menambah proses baru atau *thread*. Selain itu, pemrograman dengan model ini mudah diimplementasikan atau diintegrasikan dengan model pemrograman berbasis *Graphics User Interface*. Namun dalam pembuatannya lebih kompleks.

## IV. ANALISA INPUT

Input dari aplikasi ini adalah alamat IP dan *port* yang digunakan komputer *server* untuk menerima koneksi dan melakukan komunikasi dengan komputer *client* yang akan dikontrol atau dipantau. Sedangkan untuk proses-proses setelah koneksi terbentuk, input yang dibutuhkan adalah alamat IP

dan *Computer* atau *Host Name* dari komputer *client* yang bergabung, guna mempermudah pemantauan dan pengontrolan.

V. ANALISIS PROSES

Proses yang dilakukan oleh sistem antara lain :

1. Pengidentifikasian komputer *client*.
2. Aplikasi *server* mengirimkan perintah pada aplikasi *client* yang menunggu perintah.
3. Aplikasi *client* menjalankan atau mengeksekusi proses yang diperintahkan.
4. Aplikasi *client* mengirimkan hasilnya pada aplikasi *server*.

VI. ANALISA OUTPUT

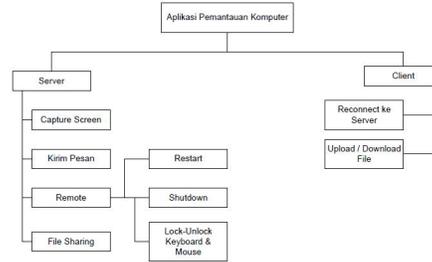
Output yang dapat diberikan oleh sistem antara lain :

1. Daftar semua komputer yang telah menjalankan aplikasi *client*.
2. Informasi sistem, yaitu berisi informasi *hostname* dan alamat IP, pencatatan waktu pertama kali melakukan koneksi ke *server*, dan status dari komputer *client* (*Connected, Locked, Unlocked*).
3. Pemantauan, dilakukan dengan memantau tampilan monitor komputer *client*. Tampilan monitor akan dikirimkan ke *server* tiap kali *server* melakukan *request* akan tampilan monitor *client*.
4. Kontrol, yaitu terdiri dari pengontrol *power* (*shutdown*, dan *restart*), mengunci dan mengaktifkan *hardware* (monitor, *keyboard*, dan *mouse*).
5. Sistem Pesan, berupa pengiriman pesan *pop-up* (*warning*, dan *information*).

VII. ARTISTEKTUR APLIKASI

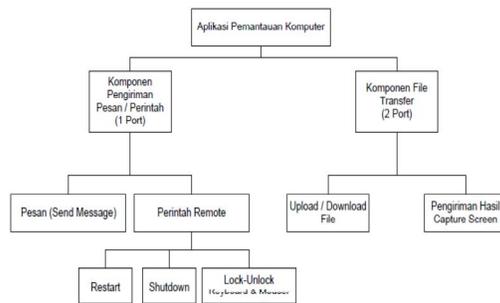
Aplikasi Pemantauan Komputer dibagi menjadi dua bagian, yaitu aplikasi *server* dan aplikasi *client*. Aplikasi *server* dapat melakukan *Capture Screen Client*, mengirimkan pesan ke *client*, pusat pertukaran *file* (*file sharing*), dan melakukan pengendalian jarak jauh, berupa : *restart*, *shutdown*, *lock-unlock keyboard* dan *mouse*. Sedangkan aplikasi *client* hanya dapat melakukan koneksi ulang ke *server*, *upload file* ke *server*, dan *download file* dari *server*.

Aplikasi Pemantauan Komputer menggunakan 2 komponen yang menangani komunikasi antara *client* dan *server*, yaitu komponen untuk menangani pengiriman pesan atau perintah (*Send Message, Remote Restart, Shutdown, Lock-Unlock Keyboard dan Mouse*), dan komponen untuk menangani pertukaran *file*.



Gambar 2 : Arsitektur Aplikasi

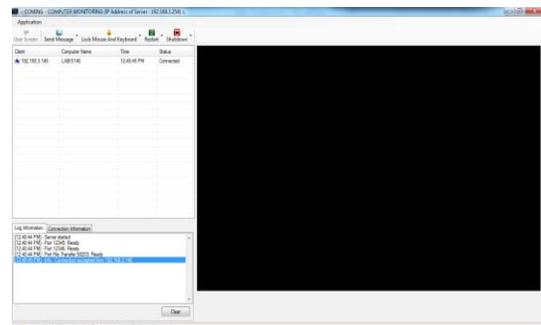
Komponen yang digunakan untuk menangani pengiriman pesan atau perintah hanya menggunakan 1 nomor *port* sedangkan komponen untuk pertukaran *file* menggunakan 2 nomor *port*, yaitu 1 nomor *port* digunakan untuk tukar-menukar file antara *client* dan *server*, dan 1 nomor *port* berikutnya digunakan untuk mengirim dan menerima file gambar hasil *capture screen*.



Gambar 3 : Penggunaan Komponen

VIII. HASIL APLIKASI

Bagian utama dari aplikasi ini ialah aplikasi *server*, di mana aplikasi *server* ini pada saat pertama kali dijalankan akan menunggu *client* yang masuk. Jika terdapat *client* yang masuk, maka nama komputer dan IP Address, waktu masuk, dan status akan ditampilkan. Tampilan aplikasi *server* sebagai berikut.

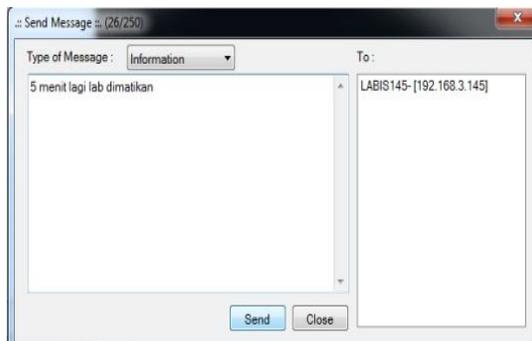


Gambar 4 : Tampilan Server

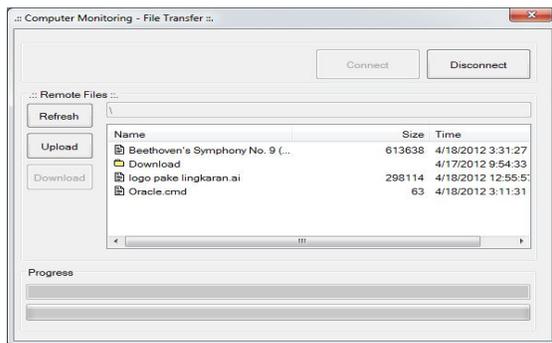
Pada bagian *server* terdapat tombol-tombol untuk melakukan perintah pada *client*. Perintah-perintah yang dapat dilakukan *server* adalah: pengiriman pesan *pop-up* ke *client*, mengunci /

membuka *mouse* dan *keyboard* komputer *client*, *restart*, *shutdown* komputer *client*, mengambil tampilan layar *client*.

Gambar 5 : Pengiriman Pesan ke *client*



Aplikasi *server* dipasang pada komputer yang melakukan pengawasan (komputer utama), sedangkan aplikasi *client* dipasang pada komputer praktikan.



Gambar 6 : Upload-Download File

## IX. KESIMPULAN

Dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa aplikasi pemantauan komputer yang telah dibuat dapat membantu mengatasi permasalahan yang terdapat pada Laboratorium Komputer UPH Surabaya. Karena penggunaan jaringan lokal oleh aplikasi maka tidak terdapat *delay* yang cukup lama untuk pengiriman data yang cukup besar seperti melihat tampilan layar dari komputer *client*.

## REFERENSI

- [1] Sopandi, Dede. *Instalasi dan Konfigurasi Jaringan Komputer*. Bandung: Informatika Bandung, 2008.
- [2] Rushing, Sam. *Asynchronous Socket Programming*.  
[http://www.nightmare.com/medusa/async\\_sockets.html](http://www.nightmare.com/medusa/async_sockets.html) (diakses 18 Januari 2012)
- [3] Ganyob. Building Client/Server applications with VB.NET for secure private file sharing. n.d.  
<http://www.codeproject.com/Articles/12318/B>

uilding-Client-Server-applications-with-VB-NET-fo (accessed February 28, 2012).

- [4] Gintoro, Adrian Tjahjana, Dedy Irawan, William. "SISTEM SMART CLASS ROOM BERBASIS SMART CARD DAN BAHASA PEMROGRAMAN C++." Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010), 2010