SISTEM MANAJER PADA SISTEM KOLABORASI BERBASIS WEB

Jasman Pardede
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Nasional
Jl. PKH.Hasan Mustapa No.23 Bandung 40124
Telp. (022) 7272215 ext. 18, Faks. (022) 7202892
E-mail: jasman@itenas.ac.id

ABSTRAK

Berkolaborasi dalam menyelesaikan suatu tugas satu kelompok merupakan suatu bagian penting dalam proses bisnis, karena setiap orang perlu untuk mendiskusikan gagasan-gagasannya, membagikan pemikirannya, mengkoordinasikan rencananya, menetapkan rencana kerja selanjutnya dan membuat suatu keputusan bersama. Perkembangan zaman saat ini, dimana tingkat mobilitas manusia yang semakin tinggi sehingga anggota kelompok sering tidak dapat berada pada tempat dan waktu yang sama. Lokasi yang tersebar dan sulitnya menyamakan waktu di antara sesama anggota dalam membahas suatu topik tertentu sangat menghambat kolaborasi. Untuk itu dibutuhkan suatu wadah yang dapat memfasilitasi kolaborasi tanpa memperhatikan lokasi dan waktu. Pesatnya perkembangan internet dan multimedia akhir-akhir ini telah ikut memicu pengembangan sistem kolaborasi, terutama sistem kolaborasi berbasis web. Untuk mengatur agar kolaborasi berjalan dengan baik pemakalah membangun suatu sistem yang disebut Sistem manajer. Sistem manajer merupakan aplikasi yang bertanggung jawab untuk mengatur dan memfasilitasi setiap client yang berkolaborasi dengan pengontrolan aktivitas pengguna, pengaturan sesi dan workspace dengan menyediakan fungsionalitas pengiriman teks dan gambar pada satu kelompok. Studi kasus yang digunakan dalam makalah ini adalah ruang kelas pertemuan maya.

Kata Kunci: berkolaborasi, kelompok, web, sistem manajer, client, sesi.

1. PENDAHULUAN

1.1 Latar Belakang

Kolaborasi merupakan suatu kegiatan yang memungkinkan sekelompok orang bekerja dan berkomunikasi secara bersama-sama dan mempunyai sekumpulan aturan atau kebijakan tertentu yang berlaku pada setiap orang dalam kelompok tersebut (*Bocig*, P., Chaffey, D., et al., 1999).

Perkembangan zaman saat ini, dimana tingkat mobilitas manusia yang semakin tinggi, sangat membutuhkan adanya dukungan dalam berkolaborasi pada lokasi yang berbeda-beda. Pengembangan aplikasi kolaborasi awalnya didominasi oleh aplikasi yang mendukung aktivitas secara tidak sinkron, seperti penggunaan dokumen bersama dalam forum diskusi. Kolaborasi interaktif secara sinkron dengan menggunakan teknologi jaringan lebih memberi kenyamanan bagi pengguna (Geon-Tae Ahn, et al., 2003).

Kolaborasi akan menjadi lebih interaktif jika dapat disajikan dalam bentuk gambar dengan menerapkan konsep *Computer Supported Cooperative Work* (*CSCW*). Gambar dapat dianalisa dan dibahas antar pengguna dengan menggunakan *chat system* pada terminal setiap pengguna melalui suatu *workspace*. Kolaborasi elektronik melalui *chat system* akan membawa setiap pengguna menyelesaikan suatu pekerjaan bersama tanpa harus bertemu muka di suatu tempat tertentu, sehingga akan menghemat biaya dan waktu.

Dengan komputasi jaringan, berbagai tipe perangkat lunak kelompok *collaborative* dapat digunakan. Pada umumnya, tipe-tipe dari perangkat lunak ini disebut *groupware*. *Groupware* digunakan oleh anggota *workgroup* untuk berkolaborasi yang kadang-kadang merujuk ke kelompok, *collaborative*, atau *computing.* Untuk mengontrol dan mengatur setiap pengguna di dalam menggunakan *groupware* diperlukan suatu sistem yang disebut *sistem manajer*.

1.2 Tujuan penelitian

Penelitian ini bertujuan untuk melakukan analisis, perancangan, dan pengimplementasian Sistem Manajer pada Sistem Kolaborasi Berbasis *web* yang memenuhi kebutuhan dalam mendukung kolaborasi pada suatu *website*.

2. STUDI KONSEP PEMBANGUNAN SISTEM KOLABORASI

Pengembangan aplikasi perangkat lunak yang dapat memfasilitasi kolaborasi perlu memperhatikan beberapa hal penting yaitu kebijakan dalam *collaborative system*, komunikasi kelompok, dukungan teknologi pemrograman jaringan dalam mendukung pengembangan aplikasi perangkat lunak *collaborative system*, serta bagaimana sistem manajer dapat mengatur sistem dalam berkolaborasi.

2.1 Kebijakan dalam Sistem Kolaborasi

Secara umum terdapat empat mode pengkoordinasian aktivitas-aktivitas yaitu; *parallel*, *pooled*, *sequential*, dan *reciprocal*.

a. *Parallel mode*, setiap pengguna mendekati/ menyelesaikan permasalahan sama sekali bebas dengan anggota kelompok lainnya.
b. *Pooled mode*, suatu kelompok menentukan struktur atau standar yang dipakai atau diterapkan dalam menyelesaikan suatu permasalahan dan setiap pengguna yang termasuk dalam kelompok berkontribusi terhadap gambaran kelompok kolektif.
c. *Sequential mode*, suatu kelompok menentukan tahapan-tahapan dalam proses penyelesaian masalah yang harus dikerjakan dengan suatu cara yang terurut bagi setiap anggota kelompok.
d. *Reciprocal mode*, di mana masing-masing pengguna dapat secara bebas bekerja pada suatu permasalahan (seperti pada *parallel mode*), tetapi terdapat beberapa aturan ketetapan yang ditentukan dalam pembuatan konstribusi oleh anggota individu.

Untuk *process control*, suatu *collaborative system* harus mendukung satu atau lebih mode pengkoordinasian aktivitas (Turoff, M. , 1999).

2.2 Komunikasi Kelompok

Komunikasi kelompok adalah komunikasi dengan pertukaran pesan di mana pesan dikirimkan oleh seorang anggota kelompok dan diterima oleh seluruh anggota kelompok. Untuk mengirimkan pesan ke setiap anggota perlu adanya suatu aturan yang disebut protokol komunikasi. Pesan yang dikirim kepada anggota-anggota kelompok tersebut disebut *multicast message*. Properti utama yang harus diperhatikan dalam membangun protokol komunikasi kelompok adalah *atomicity*, *ordering*, konkurensi, dan konsistensi pada data tereplikasi (Coulouris, G., Dollimore, J., Kindberg, T., 2001).

Terdapat beberapa cara/teknik untuk mengimplementasikan komunikasi kelompok dalam mendukung *collaborative system*, yaitu Socket dengan TCP/IP atau UDP/IP, *Remote Procedure Call* (*RPC*) dan Java RMI (Coulouris, G., Dollimore, J., Kindberg, T., 2001).

2.3 Pemrograman Jaringan dan Sinkronisasi Proses

Pengembangan aplikasi *collaborative system* tidak terlepas dari dukungan teknologi pemrograman jaringan, salah satunya adalah dengan **Pemrograman Java *Socket*.** Bahasa pemrograman Java sejak pertama kali diluncurkan telah menyatakan diri sebagai bahasa yang *general purpose*, berorientasi objek, dan konkuren (Gosling, James, et al. , 1996).

Konkurensi pada Java didasarkan atas mesin *virtual* Java yang mendukung banyak *thread* pada suatu saat. Setiap *thread* secara otonom mengeksekusi kode-kode Java baik di dalam mesin pemroses tunggal maupun banyak mesin pemroses. Saat mesin *virtual* Java mengeksekusi metode main dari kelas utama, mesin *virtual* Java telah menciptakan satu *thread* untuk mengeksekusi metode tersebut. *Thread* tersebut selanjutnya dapat menciptakan sejumlah *thread* lain (Mahmoud, Q.H., 2000). Setiap *thread* yang diciptakan pada Java memiliki memori kerja (*working memory*) yang digunakan untuk menyimpan *copy* dari variabel-variabel yang digunakan. Ketika suatu *thread* mengeksekusi program Java, *thread* melakukan operasi pada memori kerja, sementara memori utama digunakan sebagai *master copy*.

Java menyediakan mekanisme untuk sinkronisasi aktivitas konkuren banyak *thread* melalui *monitor*. *Monitor* merupakan konstruksi bahasa tingkat tinggi yang memungkinkan hanya satu *thread* pada satu saat yang dapat mengeksekusi satu blok kode yang diproteksi oleh *monitor*.

2.4 Sistem Manajer pada Sistem Kolaborasi Berbasis Web

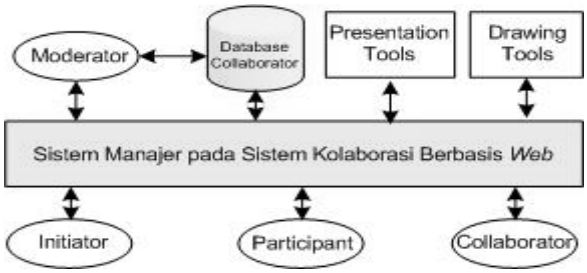
Sistem manajer merupakan suatu komponen aplikasi yang bertanggung jawab untuk mengatur beberapa *client* yang saling bekerja sama untuk melaksanakan suatu fungsi tertentu, dalam hal ini adalah pengontrolan aktivitas *client*, *session manager* dan *workspace*. *Session manager* adalah suatu *client* tertentu yang akan mengatur sekumpulan *client*, biasanya digunakan untuk mengawali dan mengakhiri sekumpulan *clients* yang ditentukan oleh pengguna. *Workspace* adalah suatu ruang lojik di mana aplikasi *groupware* dapat mengelola dan memanipulasi data internalnya. Sistem manajer akan mengatur setiap *workspace* pengguna sesuai dengan peran pengguna saat memasuki suatu ruangan.

3. HASIL DAN PEMBAHASAN

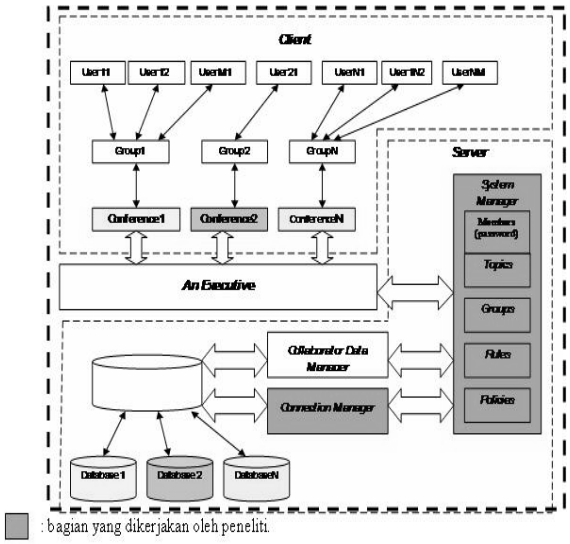
Studi kasus penelitian ini adalah sebuah ruang kelas pertemuan maya. Pada sebuah ruang kelas terdapat seorang pengajar dan beberapa siswa. Pengajar yang mengontrol satu ruangan disebut *initiator*. Siswa yang berperan aktif dan dikontrol oleh *initiator* disebut *participant*. Kelangsungan proses kolaborasi dikontrol oleh *moderator*. *Initiator* atau *participant* yang berada pada suatu ruang kelas disebut *collaborator*. Media kolaborasi setiap ruangan menggunakan sebuah *whiteboard* sebagai *drawing tools* dan *chat system* sebagai *presentation tools*. Gambaran umum dari perangkat lunak yang akan dibangun ditunjukkan pada Gambar 1**.**

Setiap *collaborator* yang mengikuti pembahasan topik yang sama akan dimasukkan dalam satu kelompok. Setiap *collaborator* akan memperoleh informasi yang sama dengan setiap pengguna dengan cara pengelompokkan *thread* pengguna menggunakan *ThreadGroup* sehingga *thread* lebih mudah dikontrol. Untuk mendukung suatu kelompok berkolaborasi lebih efektif digunakan

bantuan sistem jaringan komputer (*networked computer system*) dan menerapkan kebijakan-kebijakan yang ditranslasikan kepada aturan-aturan (*rules*), yang disebut *coordination policies*. Dari gambaran ini dapat dibuat arsitektur perangkat lunak Sistem Manajer pada Sistem Kolaborasi Berbasis *web*, seperti pada Gambar 2**.**



Gambar 1. Gambaran umum aplikasi

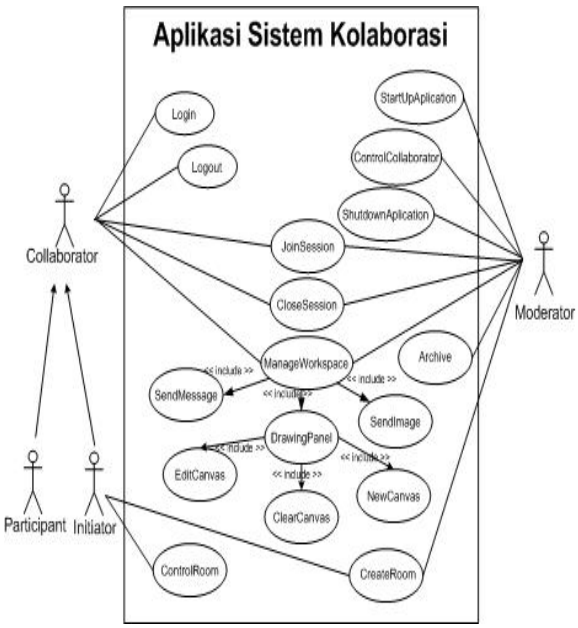


Gambar 2. Arsitektur perangkat lunak

Dari fungsional perangkat lunak yang hendak dibangun, diperoleh use case diagram seperti yang ditunjukkan pada Gambar 3.

Aplikasi yang dikembangkan pada penelitian ini menggunakan mode pengkoordinasian aktivitas secara *parallel mode*, di mana setiap pengguna dapat menggunakan *workspace*-nya tanpa adanya interupsi dari pengguna lain.

Dalam menerapkan *concurrency control* menggunakan konkurensi yang dimiliki Java atas mesin *virtual* Java yang mendukung banyak *thread* pada suatu saat, yang sering disebut dengan istilah *multithreading*. Untuk menerapkan *access control*, sistem akan mengidentifikasi setiap pengguna berdasarkan *users id* yang dimiliki oleh setiap pengguna. Pengguna yang berperan sebagai *initiator* akan mengatur hak akses setiap pengguna lainnya, apakah pengguna dapat menggambar pada *whiteboard* atau hanya dapat menulis pada *workspace*, hanya memperhatikan pengguna lain atau dapat menulis pada *workspace* dan menggambar pada *whiteboard*. Pengaturan hak akses ini dilakukan dengan pengiriman *token* ke pengguna oleh *initiator*.



Gambar 3. Use case diagram

Untuk menerapkan *session management* pada sistem akan selalu memperhatikan keberadaan setiap pengguna yang mau bergabung atau meninggalkan sesi tanpa mengganggu pengguna lain dan hanya memberikan informasi teks ke pengguna lain yang berada dalam satu kelompok mengenai keberadaan pengguna. Keberadaan pengguna diperoleh dari pengiriman *token* saat terjadi event pada *workspace*.

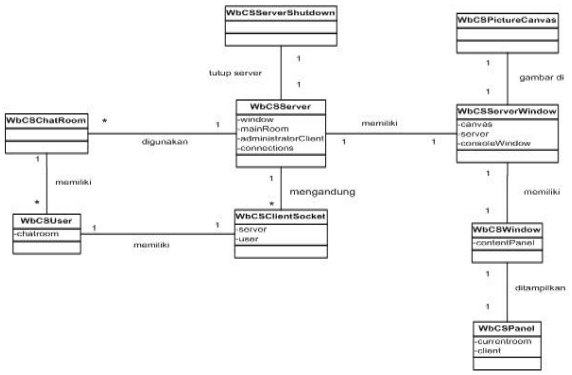
Aplikasi dalam mengacu kelompok *awareness* akan mengikuti hal berikut :

a. *Presence awareness***,** yaitu setiap pengguna baru yang memasuki sistem akan dikonfirmasi ke setiap pengguna lainnya berupa teks ‘pengguna masuk ke ruangan *topik’*.
b. *Engagement awareness***,** yaitu menginformasikan setiap aktivitas pengguna baik aktivitas menggambar atau sedang menuliskan pesan berupa teks ‘*aktivitas : pengguna’*.
c. *Structural awareness***,** yaitu mengidentifikasi tipe pengguna, baik sebagai *initiator*, *participants*, atau *moderator*.
d. *Workspace awareness***,** yaitu memperhatikan setiap perubahan yang terjadi pada *workspace*.

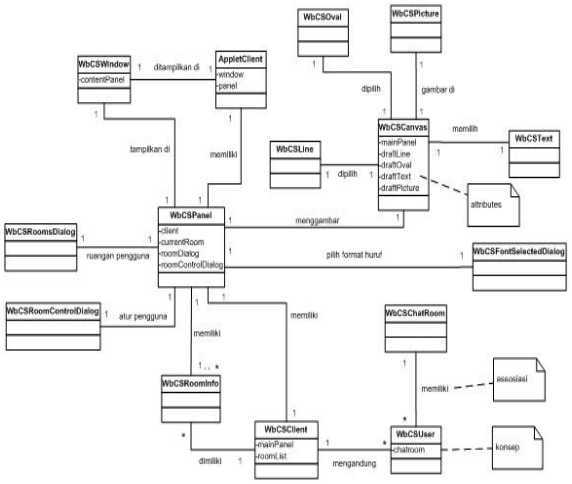
Karakteristik bentuk komunikasi yang digunakan dalam aplikasi ini adalah *document comunication* yang berupa teks dan gambar.

3.1 Perancangan Sistem

Model konseptual dari aplikasi dari sisi client digambarkan pada Gambar 4. Sedangkan model konseptual aplikasi dari sisi server digambarkan pada Gambar 5.



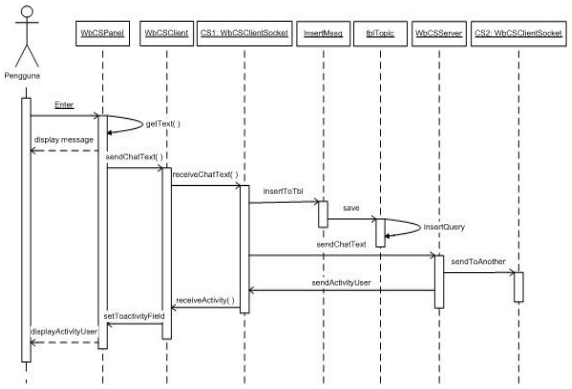
Gambar 4. Model konseptual pada sisi server



Gambar 5. Model konseptual pada sisi client

3.2 Pemodelan Dinamis

Untuk memodelkan aspek dinamis dari suatu sistem yang hendak dibangun maka dibuat skenario *sequence diagram*, dalam hal ini diambil salah satu *use case SendMessage* ditunjukkan pada Gambar 6.



Gambar 6. Sequence diagram SendMessage

3.3 Implementasi Sistem

Perangkat lunak Sistem Manajer pada Sistem Kolaborasi Berbasis *web*, memerlukan dukungan perangkat lunak lain dalam implementasinya seperti, sistem operasi Windows dan Linux, bahasa pemrograman Java serta *servlet engine* Apache Tomcat 5.5 sebagai *server side web application* berbasis Java (JSP/servlet) dan MySQL Server 5.0 pada suatu server**.**

3.4 Komunikasi antar Komponen Perangkat Lunak

Dalam memanfaatkan teknologi Java, sebuah objek A dapat mengirimkan message kepada objek lain (B) yang tidak dikenal pada saat desain kelas A melalui mekanisme **event** dan **event listener.** Proses di *client* dalam berkomunikasi dengan *server* menggunakan protokol TCP yang bersifat *connection-oriented* dan *reliable*.

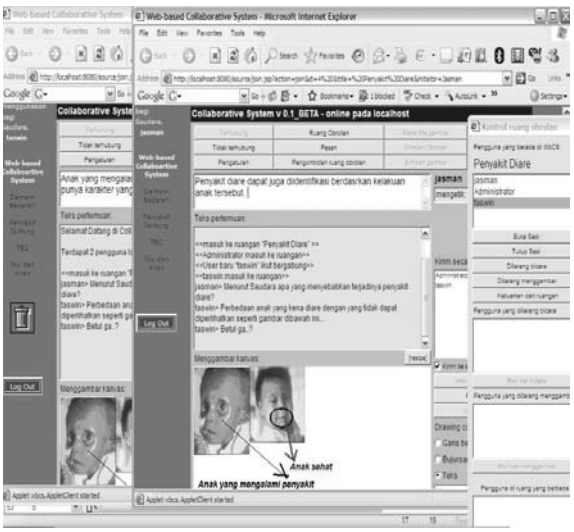
4. TEKNIK PENGUJIAN

Teknik pengujian yang digunakan adalah teknik pengujian *black box testing*. *Black box testing* atau disebut juga *behavioral testing*, memiliki fokus pada kebutuhan fungsional dari perangkat lunak. *Black box testing* memungkinkan pemrogram untuk memperoleh sekumpulan kondisi masukan (*input*) yang akan secara penuh menjalankan semua kebutuhan fungsional untuk sebuah program. Dalam hal ini diambil salah satu butir uji, yaitu pengiriman pesan teks seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Pengujian pengiriman pesan teks

Identifikasi	**TWbCS-01**		
Nama Butir Uji	Pengiriman Pesan Teks		
Tujuan	Media dalam mengirimkan pesan teks ke *workspace* pengguna dan menyimpan teks ke tabel		
Deskripsi	Pengguna menulis **string teks** ke teks field aplikasi kemudian tekan "ENTER"		
Kondisi Awal	Menu aplikasi applet menampilkan antar muka menu ruangan yang sedang dimasuki oleh pengguna		
Pengujian			
Skenario Uji			
a. Masukkan **string teks** yang akan dikirim b. Pilih pengguna yang akan menerima pesan c. Tekan "ENTER"			
Kriteria Evaluasi Hasil			
a. Pengguna mempunyai hak untuk mengirimkan teks b. Pengguna yang dalam satu sesi tidak mengaktifkan tombol "Ignore user(s)"			
Kasus dan Hasil Uji (Data Normal)			
Masukan	Harapan	Pengamatan	Kesimplan
String teks	Tampilkan **string teks** ke *Workspace*. Simpan **string teks** ke tabel.	**String teks** yang dikirim ditampilkan dan data disimpan di tabel	**[**X**]** Terima **[ ]** Tolak

Hasil pengujian pengiriman pesan teks dan gambar oleh pengguna direpresentasikan seperti pada Gambar 7.



Gambar 7. Menu teks area penulisan komentar dan pengiriman gambar

5. KESIMPULAN DAN HASIL PENGUJIAN

5.1 Hasil Pengujian

Hasil pengujian memperlihatkan bahwa perangkat lunak yang dibangun bekerja dengan baik sesuai dengan spesifikasi kebutuhan, analisis, dan perancangan. Hasil pengujian ini tidak menemukan kesalahan jika prosedur pelaksanaan operasi dijalankan dengan benar.

5.2 Kesimpulan

Berdasarkan penelitian yang dilakukan, beberapa kesimpulan yang dapat diambil sebagai berikut :

a. Pada penelitian ini telah dibangun sebuah aplikasi Sistem Manajer pada Sistem Kolaborasi Berbasis *web* yang berfungsi sebagai wadah dalam mendukung pekerjaan yang membutuhkan kerja sama kelompok dengan memperhatikan aspek-aspek kelompok *awareness*. Perangkat lunak ini memiliki kemampuan untuk pengiriman teks dan gambar.
b. Sistem Manajer pada Sistem Kolaborasi Berbasis *web* adalah suatu aplikasi manajemen sistem dalam pengaturan komputer-komputer *client* yang terhubung ke server, pengaturan hak akses dan pemantauan aktivitas.
c. Aplikasi Sistem Manajer pada Sistem Kolaborasi Berbasis *web* telah menerapkan properti-properti yang mendukung koordinasi komponen *Computer Supported Cooperative Work*.
d. Konkurensi pada Java didasarkan atas mesin *virtual* Java yang mendukung banyak *thread* pada suatu saat, yang sering disebut dengan istilah *multithreading*, mampu mendukung komunikasi kelompok di dalam pengembangan aplikasi *collaborative system* dan dalam pengelompokkan *thread* dilakukan dengan menggunakan *ThreadGroup* sehingga *thread* dapat lebih mudah dikontrol.

PUSTAKA

Bocig, P., Chaffey, D., et al. (1999). *Business Information Systems Technology, development and Management*, Prentice-Hall.

Coulouris, G., Dollimore, J., Kindberg, T. (2001). *Distributed Systems Concepts and Design*, 3th edition, Pearson Education.

Geon-Tae Ahn, Jin-Hong Kim, Myung-Joon Lee. (2003). *A Web-based Collaborative System and Its Application.* School of Computer Engineering and Information Technology.

Diakses pada 1 Maret 2007 dari http://cs.nstu.ru/IST2003/IT-I/IT-I-03.pdf

Gosling, James et al. (1996). *The Java Language Specification***,** Sun Microsystems.

Mahmoud, Q.H.. (2000). *Distributed Programming with Jav**a,*** Manning Publication Co.

Turoff, M. (1999). *An End to Student Segregation*: *No More Seperation Between Distance Learning and Regular Courses***:** A summary of the invited plenary for the Telelearning 99 meeting in Montreal, Canada.