

**SISTEM APLIKASI ELECTRONIC MAIL BERBASIS WEB MENGGUNAKAN
COLLABORATION DATA OBJECTS FOR WINDOWS NT**

ANDRI DARMADI



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2002**

@Hak cipta milik IPB University

IPB University

- Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



RINGKASAN

ANDRI DARMADI. Sistem Aplikasi *Electronic Mail* Berbasis *Web* Menggunakan *Collaboration Data Objects For Windows NT (Electronic Mail Application System Using Collaboration Data Objects For Windows NT)*. Dibimbing oleh AGUS BUONO dan HERU SUKOCO.

Electronic Mail merupakan fasilitas *internet* yang dapat mengatasi hambatan yang dihadapi manusia dalam berkomunikasi secara berjauhan. Berdasarkan mekanisme pengiriman dan penerimaan, *email* terdiri atas 3 jenis yaitu *Post Office Protocol (POP) email*, *email* berbasis *web* dan *email forwarder*. Mekanisme yang digunakan pada sistem ini adalah *email* berbasis *web*.

Sistem dibangun menggunakan sebuah komputer dengan sistem operasi Windows 2000 *Professional*. Penelitian dilakukan dengan mengamati situs-situs *email* berbasis *web*, dan mencoba menerapkannya pada aplikasi. Dalam pengembangan sistem digunakan bahasa pemrograman *Active Server Pages* menggunakan *Collaboration Data Objects for Windows NT (CDONTS)* sebagai *object email*. Pengujian aplikasi dilakukan dalam jaringan *intranet* dengan sistem operasi Windows 2000 *Server* sebagai *server* dan Windows 9x sebagai *client*.

Sistem dapat digunakan untuk mengirim dan membaca *email* dalam jaringan *intranet*. Untuk dapat menggunakan sistem ini pada *internet*, dibutuhkan alamat IP legal yang dapat digunakan oleh *server*. CDONTS memberikan kemudahan kepada pengembang dalam membangun sistem *messaging (email)* sederhana dan hanya membutuhkan PC dengan sistem operasi Windows NT sebagai *server*.

@ Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



**SISTEM APLIKASI *ELECTRONIC MAIL* BERBASIS *WEB* MENGGUNAKAN
*COLLABORATION DATA OBJECTS FOR WINDOWS NT***

@Hak cipta milik IPB University

ANDRI DARMADI

Skripsi
sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer
pada
Program Studi Ilmu Komputer

**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2002**



@Hak cipta milik IPB University

Judul : Sistem Aplikasi *Electronic Mail* Berbasis *Web* Menggunakan
Collaboration Data Objects for Windows NT
Nama : Andri Darmadi
NRP : G06496022

Menyetujui,

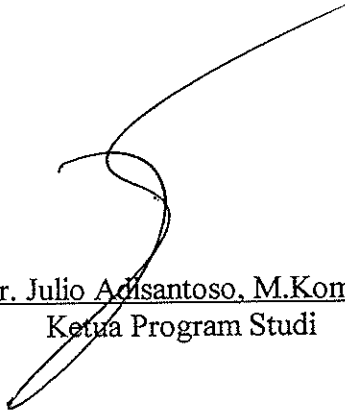


Ir. Agus Buono, M.Si., M.Komp.
Pembimbing I



Heru Sukoco, S.Si.
Pembimbing II

Mengetahui,



Ir. Julio Adisantoso, M.Komp.
Ketua Program Studi



Ir. Agus Buono, M.Si., M.Komp.
Ketua Jurusan

Tanggal Lulus : 25 APR 2002



RIWAYAT HIDUP

Penulis dilahirkan di Jakarta pada tanggal 30 Desember 1977 dari pasangan Mohammad Darmadi dan Ice Ayisah sebagai anak kedua dari lima bersaudara.

Tahun 1996 penulis lulus dari SMA Negeri 3 Jakarta dan pada tahun yang sama penulis memulai masa studi di Institut Pertanian Bogor melalui jalur Undangan Seleksi Masuk IPB sebagai civitas akademika Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam yang saat itu masih berstatus Program Studi.

Pada waktu masa belajar di SMA penulis merupakan pendiri Sic Seni Rupa SMA 3 Jakarta yang berada di bawah naungan Biro Kesenian dan pernah menjabat sebagai Wakil Ketua Sic Seni Rupa SMA 3.

@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



@Hak cipta milik IPB University

PRAKATA

Puji dan syukur penulis panjatkan kehadiran Allah SWT atas berkat, rahmat dan hidayah-Nya. Tema penelitian ini adalah aplikasi *email* berbasis *web*, dengan judul Sistem Aplikasi *Electronic Mail* Berbasis *Web* Menggunakan *Collaboration Data Objects for Windows NT*.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan, bimbingan, dan dorongan selama penyelesaian karya ilmiah ini, antara lain kepada :

1. Bapak Ir. Agus Buono, M.Si., M.Komp. dan Bapak Heru Sukoco, S.Si., selaku pembimbing yang telah memberikan bimbingan, arahan, saran dan bantuan dalam penyusunan karya ilmiah ini.
2. Mama, Papa serta saudara-saudariku yang menemani di rumah.
3. Mega yang telah memberikan dukungan dan bantuan dalam menyelesaikan skripsi.
4. Teman-teman seangkatan Bowo, Yusuf, Jarot, Indra, Basir, Panji-Risna, Erwin, Bhana, Habibie, OC, Riyadi, Tri, Mira, Uyuy, Wayan, Ake, Agung, Tuti, Andini, Sinta, Fitri, Maya, Daos, Yessi, Wildan dan Chandra.
5. Teman-teman SMA, Rico, Dinie, Yudish, Zuhail, Alhdy, Putra, Alif, dan teman-teman lainnya.
6. Teman-teman mahasiswa Jurusan Ilmu Komputer angkatan 32, 34 dan 35.
7. Pak Tisna, Pak Yana, Pak Fendy, Mas Farhan, Mas Firman, Bu Yayuk, Bu Eka, Bu Nining, Bu Tiktik.

Semoga karya ilmiah ini bermanfaat dan dapat digunakan sebaik-baiknya.

Bogor, Juni 2002

Andri Darmadi

Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

DAFTAR ISI

Halaman

DAFTAR TABEL	vi
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	vi
I. PENDAHULUAN	
I.1. Latar Belakang	1
I.2. Tujuan	1
II. TINJAUAN PUSTAKA	
II.1. Internet	1
II.2. Basis Data	1
II.3. <i>Electronic Mail</i>	1
II.4. Aplikasi <i>Email</i>	2
II.5. <i>Simple Mail Transport Protocol</i>	2
II.6. <i>Active Server Pages</i>	3
II.7. <i>Collaboration Data Objects</i>	3
II.8. <i>Collaboration Data Objects for Windows NT</i>	3
II.9 Metode Penulisan Program <i>Email</i> Lainnya	4
III. METODE PENELITIAN	
III.1. Metodologi	4
III.2. Fase Analisis	4
III.3. Fase Desain	
a. Desain <i>Output</i>	4
b. Desain <i>Input</i>	4
c. Desain Proses	4
d. Desain Basis Data	5
III.4. Fase Implementasi	5
III.5. Uji Coba Sistem	5
IV. HASIL DAN PEMBAHASAN	5
IV.1. Proses <i>Login SMTP</i>	6
IV.2. <i>Inbox</i>	6
IV.3. Proses Pembentukan <i>Email</i>	6
IV.4. Proses Pembentukan <i>Attachments</i>	6
IV.5. Proses Pembacaan <i>Email</i>	7
IV.5. Proses Transaksi <i>Email</i>	7
IV.6. <i>Address Book</i>	8
IV.7. <i>Block Address</i>	8
IV.8. Manipulasi Basis Data	8
IV.9. Kelebihan Sistem	8
IV.10. Kekurangan Sistem	9
V. KESIMPULAN DAN SARAN	
V.1. Kesimpulan	9
V.2. Saran	9
VI. DAFTAR PUSTAKA	9

@Hak cipta milik IPB University

IPB University



DAFTAR TABEL

	Halaman
1. Nama <i>field</i> dan tipe data tabel anggota	5
2. Nama <i>field</i> dan tipe data tabel blok	5
3. Nama <i>field</i> dan tipe data tabel address	5

DAFTAR GAMBAR

	Halaman
1. Komponen standar konseptual <i>email</i> dalam satu <i>relay agent</i>	2
2. Komponen konseptual dalam <i>email</i> berbasis web	2
3. Properti CDONTS	3
4. Siklus hidup sistem	4
5. Proses <i>attachments</i>	7
6. Proses penerimaan dan pengiriman <i>Email</i>	7

DAFTAR LAMPIRAN

	Halaman
1. Sistem <i>Email</i> dengan 2 Sistem <i>Relay</i>	12
2. Relasi Antar Tabel	12
3. Model <i>Object</i> CDO	13
3. Flowchart Program	14
4. Tampilan <i>Login</i>	15
5. Tampilan <i>Inbox</i>	15
6. Tampilan <i>Address Book</i>	16

© Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
 2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
- b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

PENDAHULUAN

TINJAUAN PUSTAKA

Latar Belakang

Jarak dan biaya merupakan masalah yang dihadapi manusia untuk berkomunikasi apabila dilakukan dari tempat yang berjauhan. Sejalan berkembangnya teknologi, masalah dalam berkomunikasi tersebut dapat diatasi. Berbagai cara dan media telah ditemukan dan masih digunakan hingga saat ini untuk mengatasi masalah jarak dan biaya, seperti surat, telegram, fax dan telepon. Meskipun demikian, surat memiliki kelemahan dalam hal pengiriman. Fax, telepon dan telegram dapat mengatasi masalah waktu, tetapi memakan biaya yang tidak sedikit.

Pada tahun 1971, Ray Tomlinson menemukan cikal bakal teknologi *internet* (Setyomurti, 2001). Pada mulanya teknologi *internet* hanya berupa sebuah program pesan elektronik yang memungkinkan para peneliti dan pemrogram saling berbagi pesan pada suatu jaringan. Kemudian Tomlinson melakukan suatu percobaan yang diberi nama CYPNET. Dengan CYPNET, seseorang dimungkinkan memindahkan *file* antara komputer yang satu dengan lainnya meskipun letak komputer tersebut berjauhan.

Dengan ditemukannya teknologi *internet*, dunia komunikasi pun berkembang pesat. Salah satu fasilitas yang disediakan oleh *internet* adalah *electronic mail (email)*, yang merupakan solusi untuk mengatasi masalah yang selama ini dihadapi oleh manusia dalam berkomunikasi. Dengan *email* seseorang dapat mengirim pesan kepada kerabat, teman maupun rekan bisnis dengan waktu lebih cepat.

Email sebagai media komunikasi menarik minat banyak orang, sehingga jumlah pemakai teknologi *internet* semakin meningkat. *Email* dapat ditunjukkan kepada beberapa *user* maupun *grup*, tidak tergantung kepada jumlah komputer. Dengan besarnya minat atas *email* tersebut, maka bermunculanlah perusahaan-perusahaan jasa yang menawarkan fasilitas *email* ini, baik berbentuk *email* berbasis *Post Office Protocol (POP)* maupun *email* berbasis *web*.

Tujuan

Adapun tujuan penelitian ini adalah mempelajari dan mengembangkan sistem aplikasi *email* berbasis *web* menggunakan *Collaboration Data Objects for Windows NT*.

Internet

Internet adalah sekumpulan komputer yang tergabung bersama dalam suatu *gateway* yang menangani transfer data dan konversi pesan antara suatu jaringan ke jaringan lainnya, agar pesan tersebut dapat diterima. Protokol transfer utama yang digunakan oleh *world wide web (WWW)* adalah *Hypertext Transfer Protocol (HTTP)*. HTTP bersifat *request-response*, yaitu dalam protokol ini *client* menyampaikan pesan *request* ke *server* dan *server* kemudian memberikan *response* yang sesuai dengan *request* yang dikirimkan (Purbo et al. 1999).

Informasi yang ditampilkan di halaman *web* dapat berupa informasi yang statik maupun dinamik. Informasi yang dinamik biasanya berasal dari data yang terdapat dalam basis data.

Basis Data

Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya (Fathansyah, 1999). Kumpulan data yang saling berhubungan dan memiliki keterkaitan beserta tata cara pengaturan tertentu disebut basis data.

Untuk mengatur, menggunakan dan memanipulasi data diperlukan suatu perangkat lunak sistem pengelola basis data. Tujuan utama dari sistem pengelola basis data adalah kemudahan dan keefisienan dalam penggunaan, pengaturan dan manipulasi data.

Basis data relasional, sebagai salah satu model basis data memilah basis data ke dalam bentuk tabel. Setiap tabel terdiri atas baris (*record*) dan kolom (*atribut*). Setiap *record* terdiri atas atribut yang memiliki tipe data yang berbeda (*text*, *memo*, *number*, *dsb*).

Electronic Mail

Electronic mail (email) adalah transmisi suatu pesan pada jaringan. *Email* merupakan sinonim dari surat. Konsep *email* tidak dapat dipisahkan dengan alamat *email*. Alamat *email* menyediakan semua informasi yang dibutuhkan agar suatu pesan (*email*) dapat sampai kepada *user*.

Alamat *email* biasanya terdiri atas sebuah simbol yang memisahkan *id user* dengan *server* tujuan (*nama domain*). Simbol yang sering dilihat dan digunakan adalah simbol "@", yang dibaca "at" merupakan simbol format *internet*. Simbol

"I", yang juga dibaca "bang" tidak sering dijumpai merupakan format Unix to Unix Copy (UUCP).

Menurut Purbo (1999), *email* terdiri atas tiga komponen, yaitu *envelope*, *header* dan *body*. *Envelope* digunakan oleh *Message Transport Agent* (MTA) untuk pengiriman. *Header* terdiri dari *field-field* seperti *Received*, *Message-Id*, *From*, *Date*, *Reply-To*, *X-Phone*, *X-Mailer*, *To*, dan *Subject* digunakan oleh *user agent*. *Body* merupakan isi pesan yang ingin disampaikan oleh si pengirim kepada si penerima.

Aplikasi Email

Untuk dapat menggunakan fasilitas *email* diperlukan suatu aplikasi *email client*, sedangkan untuk mengatur dan menyampaikan sebuah diperlukan *email server*. *Email* yang dikirim akan disimpan di dalam *email server* yang kemudian akan diambil oleh penerima melalui *email client*.

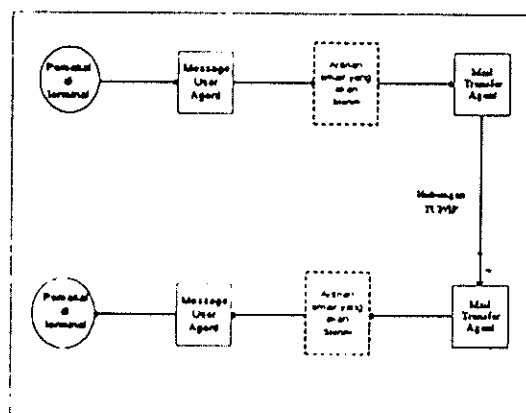
Menurut situs Media Informasi Komputer dan Teknologi, jenis *email* berdasarkan mekanisme pengiriman dan penerimaannya terdiri atas :

1. *Post Office Protocol (POP) email*, pemakaian protokol ini sama seperti pengambilan surat di kantor pos. Seperti pada surat, sebuah *email* akan melewati beberapa komputer yang berperan sebagai *router* atau *email server*, sebelum akhirnya tiba pada *email server* yang ada pada *Internet Service Provider* (ISP) di mana alamat *email* user terdaftar.
2. *Email berbasis web*, untuk penanganan *email* harus dilakukan dengan menggunakan *web-browser*. *Email client* ini memakai protokol *Internet Message Access Protocol (IMAP)* yang memungkinkan *user* untuk menggunakan fasilitas *email*.
3. *Email forwarder*, metode ini memberikan alamat *email* baru, tetapi tidak menyimpan *email* yang datang. Setiap *email* yang datang akan langsung diteruskan ke alamat *email* yang merupakan jenis pada nomor 2 dan 3.

Simple Mail Transport Protocol

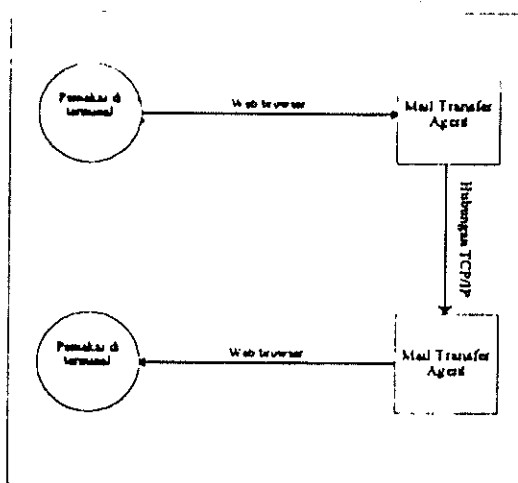
Simple Mail Transport Protocol (SMTP) merupakan protokol TCP/IP yang digunakan untuk mengirimkan pesan dari satu komputer ke komputer yang lain. Dalam internet protokol ini digunakan untuk mengirim pesan (*email*). Secara konseptual, pada SMTP pemakai berhubungan dengan *Message User Agent (MUA)* atau aplikasi *email* di terminal. MUA bertugas untuk menangani pertukaran *email* yang akan dikirim

atau diterima dengan menggunakan protokol TCP/IP. MUA kemudian akan melakukan hubungan TCP/IP untuk berkomunikasi dengan *Message Transfer Agent (MTA)*. Setelah terhubung, MUA akan mengirim dan mengambil *email* milik *user* yang terdapat *server MTA*. *Email* yang telah dikirim ke MTA selanjutnya akan diteruskan oleh MTA kepada *server MTA* tujuan. (Gambar 1). Bila menggunakan dua sistem relay antar organisasi dapat dilihat pada Lampiran 1.



Gambar 1. Komponen standar konseptual *email* dalam satu relay agent

Dalam sistem aplikasi *email* berbasis *web*, tidak diperlukan MUA. Aplikasi yang diperlukan oleh *user* agar dapat menggunakan fasilitas *email* adalah sebuah *web browser* seperti Internet Explorer untuk mengakses fasilitas *email* yang terdapat pada *server* (Gambar 2).



Gambar 2. Komponen konseptual *email* berbasis *web*

Active Server Pages

Dokumen yang ditampilkan di dalam *web browser* dapat berisi informasi statik dan dinamik. Dokument statik adalah dokumen yang ditempatkan pada media penyimpanan *server* dan tidak mengalami banyak perubahan secara umum. Dokumen dinamik dibentuk setelah data yang ada pada *server* diproses oleh *scripts* dan menghasilkan tampilan berisi informasi yang merupakan hasil dari kueri *client* yang dikirimkan ke *server*. *Active Server Pages* (ASP) merupakan bahasa *scripts* yang dikembangkan oleh Microsoft untuk mengembangkan suatu halaman *web* yang dinamik dan dapat berinteraksi dengan pengguna pada suatu *aplikasi web*.

ASP dapat melakukan interaksi dengan basis data. Interaksi dengan basis data ini dilakukan dengan menggunakan salah satu komponen ASP, yaitu *ActiveX Data Object* (ADO) (Saleh, 2000). Dengan ADO ASP dapat mengambil, mengolah dan menghapus data.

Collaboration Data Objects

Collaboration Data Objects (CDO) teknologi *object library* Microsoft yang bertujuan memberikan pengembang suatu cara mudah dalam memberikan pelayanan *messaging* (Mitchell et al. 1999). Sebelumnya CDO bernama *OLE Messaging* dan kemudian berganti menjadi *Active Messaging* sebelum akhirnya dikenal dengan nama CDO.

CDO dapat digunakan dalam pengembangan sistem *message board* berbasis *web*, otomatisasi sistem penjawab *email*, otomatisasi informasi *meeting rooms*, status *server monitor*, dan aplikasi lainnya. CDO memiliki banyak *objects* didalam arsitekturnya (lampiran 3.)

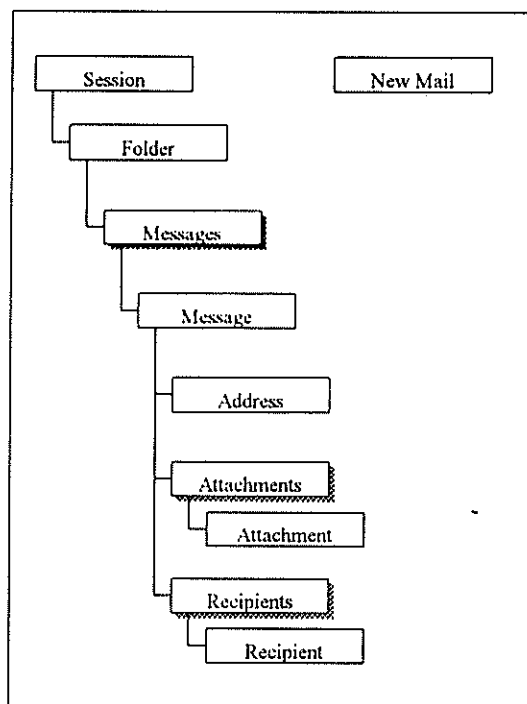
CDO dapat dikembangkan di atas program Visual Basic (versi 3 dan yang lebih baru), Microsoft C/C++, ASP dengan menggunakan VBScript dan JScript.

Collaboration Data Objects for Windows NT

CDO dikembangkan di atas sistem operasi Microsoft Windows, baik versi 9x dan versi NT. CDO yang dikembangkan di atas sistem operasi berbasis Windows NT dinamakan *Collaboration Data Objects for Windows NT* (CDONTS).

Dalam pengembangan sistem *email* berbasis *web* ini *library* yang digunakan adalah CDONTS yang merupakan subbagian dari CDO (Gambar 3). CDONTS terdiri atas beberapa *object* yang berfungsi untuk membentuk dan membaca *email*. *Objects* tersebut adalah *object Session, Folder,*

Message, Address Entry, Attachments, dan *recipients* (Gambar 3). CDONTS memiliki 2 *object* utama, yaitu *Session* dan *New Mail*.



Gambar 3. Properti CDONTS

Session Object menyimpan informasi yang digunakan untuk mengakses *mailbox user*, yaitu *name user* dan *alamat email user*. *Session object* terdiri atas anak *object Folder*, dimana *object Folder* merupakan *parent object* dari *Messages*. *Messages objects* merupakan koleksi berbagai *message (email)* yang terdapat dalam *mailbox user*.

Folder object menampilkan sebuah *folder* yang menampung *email/messages* untuk setiap *Session*. Pada SMTP Terdapat 2 *Folder object*, yaitu *Inbox* dan *Outbox*. Metode yang dapat dilakukan pada kedua *folder* ini adalah menghapus semua pesan.

Messages collection merupakan kumpulan *Message object* yang terdapat pada *Folder object*. Melalui *Messages collection* inilah tiap *Message object* dapat diolah oleh *user* baik itu membaca maupun menghapus.

New Mail object digunakan untuk membentuk *email* baru yang akan dikirim. Adapun properti yang dimiliki oleh *New Mail object* ini yaitu *to, cc, bcc, subject, importance, from, body* dan *attachments*. Metode yang dilakukan pada *New*

Mail ini adalah mengirim *email* dan menghapus/batal mengirim.

Metode Penulisan Program *Email* Lainnya

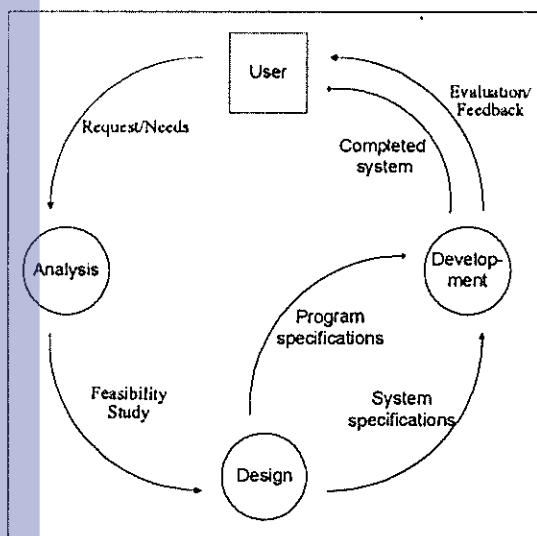
Penulisan pemrograman *email* dapat dilakukan dengan bahasa lain seperti PHP: *Hypertext Preprocessor* (PHP) dan *Common Gateway Interface* (CGI). Kedua bahasa pemrograman ini tidak menggunakan objek untuk mengirim *email*, melainkan langsung berhubungan dengan SMTP pada *server* yang telah memiliki MTA. Program dengan menggunakan instruksi yang terdapat pada MTA memerintahkan MTA untuk membuat *email* baru dan mengirim sesuai tujuannya. Metode ini mengharuskan pengembang untuk mempelajari instruksi-instruksi yang terdapat pada MTA untuk dapat menyediakan fasilitas *messaging*.

METODOLOGI PENELITIAN

Metodologi

Metodologi yang digunakan dalam mengembangkan sistem aplikasi *email* berbasis *web* ini adalah mengikuti tahapan-tahapan pada siklus hidup sistem.

Siklus hidup sistem adalah proses evolusi yang diikuti dengan pengimplementasian sistem atau subsistem informasi berbasis komputer (Edwards, 1985). Menurut Edwards (1985), sistem hidup sistem terdiri dari tiga fase : analisis, desain, implementasi. Gambar 4 menunjukkan fase dalam siklus hidup sistem



Gambar 4. Siklus Hidup Sistem

Fase Analisis

Aktivitas yang dilakukan pada tahap ini adalah meneliti beberapa sistem aplikasi *email* berbasis *web* yang sudah ada. Tahap ini dilakukan untuk meneliti fasilitas dasar yang diberikan sistem aplikasi *email* berbasis *web*.

Sistem yang dikembangkan harus dapat menyediakan fasilitas *email*, seperti membaca, menulis, menerima dan mengirim *email* secara *online*.

Untuk dapat menggunakan fasilitas *email*, pengguna diharuskan mendaftarkan diri terlebih dahulu dengan memberikan informasi yang diperlukan. Informasi ini akan disimpan di dalam basis data untuk digunakan kemudian.

Fase Desain

Tahap desain ini mencakup beberapa hal :

1. Desain *Output*

Output utama dari sistem ini adalah penanganan *email*. Desain *Output* sistem mencakup :

- Penyajian informasi mengenai *email* beserta propertinya. Informasi ini harus dapat memberikan gambaran mengenai *email* yang ada. Baik itu berupa alamat pengirim, subyek dan waktu penerimaan *email*.
- Sistem dapat membuat *email* yang kemudian dikirimkan untuk berkomunikasi dengan *user* lainnya.

2. Desain *Input*

Sebelum menggunakan fasilitas *email*, pengguna harus mendaftarkan diri terlebih dahulu agar dapat memakai fasilitas *email* dengan memasukkan *id* dan *password* untuk kemudian diperiksa oleh *server*, guna menjaga kerahasiaan. Setelah memasukkan *id* dan *password*, keterangan mengenai *id* tersebut disimpan dalam *session*, sehingga pengguna tidak perlu memasukkan data kembali sampai melakukan *logout*.

Pengiriman *email* dapat dilakukan apabila properti *email* yang diperlukan sudah terpenuhi. *Input* yang diperlukan properti *email* adalah From, To, Subject, Importance dan Body. Properti tersebut dapat diinput oleh *user* melalui *keyboard*.

Pada penggunaan fasilitas lain yang seperti *address book*, diperlukan *input* berupa data rekan yang akan dimasukkan ke dalam basis data.

3. Desain Proses

Sistem ini bertujuan untuk memberikan pelayanan *email* secara *online*. Alur proses kerja dari program ini dapat dilihat pada Lampiran 4.

4. Desain Basis Data

Untuk menggunakan fasilitas *email*, *user* harus memasukkan beberapa informasi yang akan disimpan ke dalam basis data untuk digunakan kemudian. Adapun informasi yang diperlukan adalah identitas *user*, *email* serta *id user* dan kata *password* menjaga kerahasiaan dan keamanan pengguna. Basis data juga menyimpan data lain yang dapat diakses pada fasilitas yang disediakan.

Basis data pada sistem ini terdiri atas 3 tabel, yaitu :

1. Tabel anggota.
2. Tabel blok.
3. Tabel address.

Tiap tabel terdiri dari *field-field* yang memiliki tipe data. Tipe data dari *field* tersebut dapat dilihat pada tabel 1, 2 dan 3 di bawah.

Tabel anggota menyimpan data diri anggota yang dimasukkan pada saat pengguna mendaftarkan diri. Selain itu data ini juga dapat diubah dengan menggunakan fasilitas yang telah disediakan. Tabel Anggota berisi data seperti nama, alamat, nomor telepon dan tanggal lahir (Tabel 1).

Tabel 1. Nama *field* dan tipe data tabel anggota

Nama <i>field</i>	Tipe data
nama	Text
keluarga	Text
telepon	Text
alamat	Text
<i>user</i>	Text
password	Text
pertanyaan	Text
jawaban	Text

Tabel blok menyimpan *id user*, alamat *email* yang ingin ditolak (Tabel 2). Tabel ini berfungsi menyimpan alamat *email* yang tidak diinginkan oleh *user*.

Tabel 2. Nama *field* dan tipe data tabel blok

Nama <i>field</i>	Tipe data
<i>User</i>	Text
blokemail	Text
ID blok	Text

Tabel address menampung data-data rekan *user*. *Field* yang terdapat dalam tabel address adalah *id user*, nama rekan, alamat *email* rekan, alamat, dan nomor telepon rekan (Tabel 3).

Tabel 3. Nama *field* dan tipe data tabel address

Nama <i>field</i>	Tipe data
<i>user</i>	Text
Nama	Text
Keluarga	Text
<i>email</i>	Text
alamat	Text
telepon	Text
ID	AutoNumber

Fase Implementasi

Pengembangan sistem ini dilakukan di dalam sistem operasi *Windows 2000 Profesional* yang juga berfungsi sebagai *server*, baik sebagai *web server*, basis data *server* dan *email server*. Untuk pengaturan *email* digunakan *Windows 2000 SMTP Virtual Server*.

Bahasa pemrograman yang digunakan adalah *HTML*, *Visual Basic Script* dalam penulisan *ASP*. Objek-objek yang digunakan dalam sistem ini adalah *CDONTS* untuk pengembangan *email*.

Untuk penulisan program dan perancangan aplikasi digunakan *Ms FrontPage 2000*. Basis data dibangun dengan menggunakan *Microsoft Access 2000*.

Uji coba Sistem

Uji coba sistem dilaksanakan dalam lingkungan *intranet* yang berjalan di atas *Windows 2000 Profesional* dengan fasilitas *Internet Information Services 5.0* serta *SMTP Virtual Server*. Aplikasi *email client* yang digunakan adalah *Internet Explorer 5.0* yang juga merupakan aplikasi *web browser*.

HASIL DAN PEMBAHASAN

Sistem aplikasi *email* berbasis *web* merupakan aplikasi yang berjalan masih di dalam lingkungan jaringan *intranet*. Uji coba dilakukan dengan menggunakan fasilitas yang sudah tersedia di dalam laboratorium komputer milik Jurusan Ilmu Komputer. Dalam uji coba ini digunakan 10 komputer yang digunakan untuk mengakses dan menggunakan aplikasi *email* secara bersamaan.

Sistem ini menyediakan fasilitas dasar *email* berbasis *web* kepada *user* yang telah memiliki alamat pada *server*. Proses yang terdapat pada sistem ini adalah *login*, *inbox*, *compose*, *address book*, dan *block email*.

Login merupakan halaman pertama yang ditemui oleh *user* ketika akan memeriksa *email*, selain itu *form login* merupakan cara untuk menciptakan privasi bagi *user*. Daftar *email* yang berada dalam *mailbox* dapat dilihat pada *inbox*. Pengiriman *email* dilakukan dengan mengirim informasi yang diperlukan dalam pembentukan suatu *email* melalui *form compose*. Setelah informasi yang diperlukan sudah terpenuhi, informasi tersebut akan dikirim ke *server* untuk diproses menjadi sebuah *email* dan dikirim. Pada *address book user* dapat menyimpan data mengenai informasi rekan seperti *address book* pada umumnya. Sedangkan untuk menolak *email* yang tidak diinginkan, *user* dapat melakukannya dengan *form block email*.

Proses Login SMTP

Input yang diperlukan untuk melakukan *login SMTP* adalah nama *user* dan alamat *email user* yang terdapat pada *server*. Tidak diperlukannya *password* untuk melakukan *login*, menyebabkan kelemahan keamanan privasi pada sistem. Untuk menjaga keamanan privasi maka dibuatlah basis data yang dapat menampung *password* untuk tiap *user*. Agar dapat menggunakan fasilitas *email*, *user* harus memasukkan *id user* dan *password user*, yang kemudian akan dicocokkan dan diproses dengan basis data untuk menghasilkan nama *user* dan *id user* yang diperlukan *login SMTP*. Nama dan *id user* disimpan dalam *session objects* dan akan digunakan untuk menggunakan fasilitas *email*. Apabila terdapat kesalahan dalam mengetik *user id* dan *password* maka *user* akan diarahkan pada halaman *login* (Lampiran 5).

Proses *login SMTP* dilakukan pada *form Verification.asp*, bila berhasil *user* akan diteruskan pada halaman *home.asp*. Adapun kode program untuk melakukan *login SMTP* adalah sebagai berikut :

```
Set objsesi = Server.Createobject("CDONTS.Session")
objsesi.LogonSMTP Session("user"), Session("mailbox")
Set Session("session") = objsesi
```

Inbox

Inbox menampilkan daftar *email* yang terdapat dalam *mailbox user*. Daftar *email* diperoleh dengan membaca *messages collection*

yang terdapat pada *folder inbox* yang sesuai dengan *id user*.

Dalam daftar tersebut ditampilkan pula alamat pengirim, tingkat keperluan, subyek, dan waktu penerimaan *email*. *Email* yang ditampilkan pada daftar ini tidak terurut secara teratur, sehingga menyulitkan *user* pada waktu melihat daftar tersebut. Untuk memudahkan *user* dalam memeriksa *email*, daftar *email* tersebut dapat disusun berdasarkan waktu diterimanya *email*, subject, maupun alamat pengirim sesuai keinginan *user* (Lampiran 6).

Pengaturan daftar *email* awal akan diatur berdasarkan waktu diterimanya *email*, yang selanjutnya dapat diubah agar sesuai dengan pengurutan yang diinginkan. Adapun algoritma yang digunakan untuk mengurutkan daftar *email* adalah algoritma pengurutan *bubble sort*. Berikut adalah penggalan kode program pengurutan :

```
for i = 1 to colpesan.Count - 1
for j = i+1 to colpesan.Count
if colpesan(sort(i)).Sender >
colpesan(sort(j)).Sender Then
temp=sort(i)
sort(i)=sort(j)
sort(j)=temp
end if
Next
Next
```

Proses Pembentukan Email

Data yang diperlukan dalam pembuatan sebuah *email* diperoleh melalui *form compose*, di mana pada *form* ini *user* memasukkan komponen *email* yang diperlukan. Setelah itu, informasi yang telah diinput akan dikirimkan kepada *form Send.asp* untuk diolah untuk membentuk sebuah *email* dengan menggunakan CDONTS. Adapun cara penulisan CDONTS dalam pembentukan *email* adalah sebagai berikut :

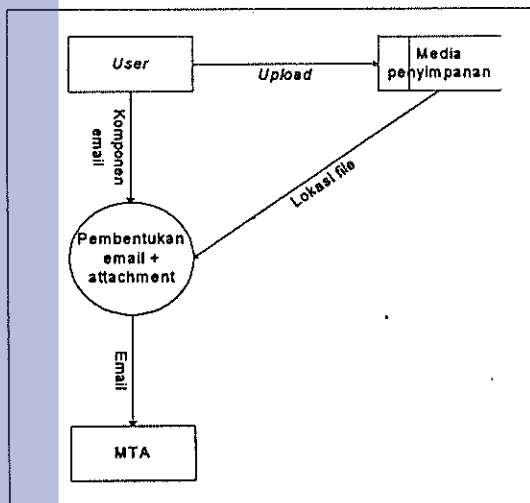
```
Dim surat, FileName, FilePath Set surat =
Server.CreateObject("CDONTS.NewMail") surat.From =
Session("mailbox") surat.To = Request.QueryString("to")
surat.Cc = Request.QueryString("cc") surat.Bcc =
Request.QueryString("bcc") surat.Importance =
Request.QueryString("importance") surat.Subject =
Request.QueryString("subject") surat.Body =
Request.QueryString("body") if request("file")<>"" Then
FileName = "D:\myweb3\" + Session("mailbox") + "\" +
Request("file") FilePath = "D:\myweb3\" +
Session("mailbox") surat.Attachfile FileName end if
surat.Send Set surat = Nothing if request("file")<>"" Then
iso.DeleteFolder(FilePath)
```

Proses Pembentukan Attachments

Tidak seperti pada aplikasi email berbasis POP, untuk menggunakan attachments pada aplikasi email berbasis web diperlukan langkah-langkah yang harus dilakukan (Gambar 5). Adapun langkah-langkah yang dilakukan adalah sebagai berikut :

1. Melakukan upload terhadap file yang ingin dikirimkan.
2. Setelah user berhasil melakukan upload, server akan menaruh file tersebut ke dalam suatu folder.
3. File yang telah disimpan oleh server akan dimasukkan ke dalam envelope email setelah properti-properti email yang lain telah terpenuhi.
4. Setelah email dikirim, maka file tersebut akan dihapus dari server.

Proses attachments ini dilakukan dengan menggunakan mengakses properti dari email yang akan dikirim dan memberikan lokasi attachments yang telah diupload (lihat kode program proses pembentukan email).



Gambar 5. Proses Attachments

Proses Pembacaan Email

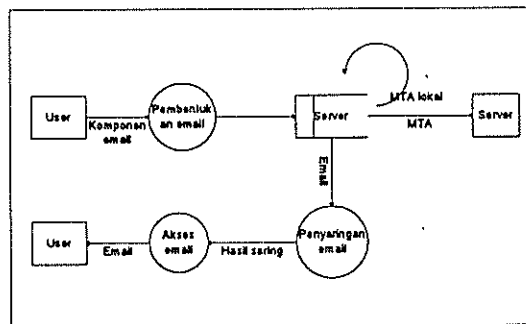
Membaca email dapat dilakukan dengan membaca properti-properti email, adapun properti email yang ditampilkan adalah pengirim, waktu penerimaan, subyek dan isi email beserta attachment bila ada. Informasi di atas diekstrak oleh CDONTS dengan kode program sebagai berikut :

```
From : <%=colpesan(index).Sender%>
Subject : <%=colpesan(index).Sender%>
Isi : <%=colpesan(index).Text%>
```

Pada form ini dapat dilakukan penghapusan email, membalas (reply) dan melanjutkan (forward).

Proses Transaksi Email

Tidak seperti pada sistem aplikasi email lainnya, sistem aplikasi email berbasis web merupakan aplikasi bersifat request-response, sehingga segala proses yang terjadi dilaksanakan di dalam server (Gambar 6). Setelah semua properti yang dibutuhkan untuk membentuk envelope email, maka email akan segera dikirimkan setelah instruksi 'send' dikerjakan. Email yang akan dikirim diterima oleh MTA, dan MTA akan memeriksa tujuan email, bila tujuan email terdapat pada remote host, maka akan disampaikan kepada server yang mengatur remote host tersebut, bila ditujukan kepada local host, email akan disimpan oleh server di dalam hardisk. Secara fisik email akan disimpan di dalam suatu direktori yang telah ditetapkan oleh administrator, dalam sistem ini email disimpan pada direktori "C:\inetpub\mailroot\Drop" dengan extension eml untuk setiap email.



Gambar 6. Pengiriman dan Penerimaan Email

Pengaturan mailbox untuk tiap user tidak diperlukan pada servis SMTP. Semua email yang diterima akan disimpan pada satu folder "C:\inetpub\mailroot\Drop". Pada servis SMTP tidak terdapat dan tidak diperlukan pengaturan mailbox untuk setiap user. Email yang ditampilkan kepada user akan disaring berdasarkan alamat tujuan (Gambar 6), bila alamat tujuan sesuai dengan alamat user maka user dapat melihat email yang ditujukan untuknya.

Proses penyaringan ini dilakukan pada saat melakukan login SMTP. Sewaktu melakukan login alamat email yang tepat, karena servis SMTP tidak melakukan validasi terhadap alamat email yang akan diakses. Apabila pada waktu

melakukan *login* SMTP terdapat kesalahan *input*, maka *user* akan diarahkan ke dalam *mailbox* sesuai dengan yang diketikannya. Dalam hal ini tentunya apabila salah penulisan satu karakter saja dalam melakukan *login* SMTP, *user* akan diarahkan ke dalam *mailbox* yang bukan miliknya (dalam hal ini *mailbox* kosong). Dalam keadaan ini, *user* tidak akan melihat pesan kesalahan, melainkan hanya akan mendapati suatu *mailbox* yang kosong.

Address book

Seperti buku alamat pada umumnya, fasilitas ini menyimpan informasi mengenai rekan yang telah *diinput* di dalam basis data. Tampilan utama *address book* adalah daftar nama, *email*, alamat, serta nomor telepon rekan (*addresslist.asp*). Penambahan data ke dalam basis data dapat dilakukan dengan memasukkan *input* sesuai urutannya. Sedangkan untuk mengedit informasi rekan yang terdapat pada basis data, dapat dilakukan dengan memilih nama rekan yang ingin diedit. Dengan mengklik nama rekan, dijalankan suatu proses pengalihan halaman disertai kode yang kemudian akan diolah oleh halaman berikutnya (*Addressbook.asp*). *Addressbook.asp* menampilkan data rekan, baik yang ingin dimasukkan maupun edit. Data berikutnya akan dikirimkan ke halaman *Addressproses.asp* untuk diproses dan dimasukkan ke dalam basis data. Penghapusan data dari basis data dilakukan dengan memilih "Delete" yang berada pada satu baris dengan data yang ingin dihapus (Lampiran 7). Seperti pada proses sebelumnya, dengan mengklik Delete berarti memanggil suatu fungsi yang mengalihkan *user* ke halaman *Addressproses* disertai dengan kode hapus untuk data yang ingin dihapus tersebut.

Address book melakukan identifikasi terhadap *user* pada waktu *user* melakukan *login*. Identitas *user* kemudian digunakan untuk mengambil data pada *server* untuk kemudian ditampilkan. Adapun bahasa kueri yang digunakan untuk mengambil identitas *user* adalah sebagai berikut :

```
strSQL = "SELECT * FROM address WHERE user = '" & Session("mailbox") & "' ORDER BY nama"
```

Address book melakukan validasi terhadap *input* yang telah *diinput* oleh *user*, dengan mengirim pesan kesalahan agar kesalahan tersebut dapat diperbaiki.

Block Address

Dengan menggunakan fasilitas *block address user* dapat mencegah *email* tertentu yang menuju *mailbox user*. Proses ini juga berhubungan dengan basis data.

Pencegahan *email* dilakukan dengan memasukkan alamat *email* yang ingin dicegah ke dalam basis data, sehingga dengan demikian *email* yang tidak diinginkan dapat segera dihapus ketika sampai ke alamat *user*. Apabila suatu saat *user* ingin menerima kembali *email* dari alamat tersebut, maka *user* dapat membatalkan *email* yang telah dicegah dari basis data.

Form Block Address mengambil data sesuai dengan identitas *user* yang *diinput* ketika melakukan *login*. Pengambilan identitas ini dilakukan dengan bahasa kueri sebagai berikut :

```
strSQL = "SELECT * FROM blok WHERE User = '" & Session("mailbox") & "' ORDER BY blokemai"
```

Manipulasi Database

Komunikasi sistem dengan basis data ditangani oleh ADO. ADO memungkinkan sistem untuk melakukan manipulasi terhadap basis data. Manipulasi basis data digunakan untuk meningkatkan keamanan dan privasi *user* dan menyajikan informasi yang diinginkan. Penulisan kueri guna memanipulasi basis data yang digunakan pada aplikasi ini sederhana, sehingga penulisannya dilakukan di dalam bahasa pemrograman. Contoh kueri yang digunakan pada *form verification.asp* dalam memeriksa *user* dan *password* yang diberikan adalah sebagai berikut :

```
strSQL = "SELECT * FROM anggota WHERE user = '" & Request("user") & "' & Request("password") & "'@ilkom.net";"
```

Penyediaan fasilitas *Address book* dan *Block Address* dilakukan dengan menggunakan teknologi ADO. ADO digunakan dalam fasilitas di atas karena fasilitas di atas berhubungan erat dengan basis data.

Kelebihan Sistem

Sistem aplikasi *email* berbasis *web* menggunakan CDONTS yang merupakan subset dari *object* CDO. CDONTS merupakan *object* yang kecil dan sederhana yang memungkinkan penyediaan sistem *messaging* yang sederhana dalam waktu singkat.

Penggunaan CDONTS dalam sistem ini memberikan kemudahan dalam penulisan program, dalam hal ini pengembang tidak perlu

memiliki pengetahuan yang mendalam terhadap *email*.

Kode program *email* yang dibangun dengan menggunakan CDONTS lebih mudah dibaca dibandingkan penulisan program *email* dengan menggunakan bahasa lainnya. Perbandingan penulisan kode program *email* menggunakan ASP (CDONTS) dengan PHP dapat dilihat pada kode berikut :

Kode ASP :

```
Dim surat,
Set surat = Server.CreateObject("CDONTS.NewMail")
surat.From = Session("mailbox")
surat.To = Request.QueryString("to")
surat.Cc = Request.QueryString("cc")
surat.Bcc = Request.QueryString("bcc")
surat.Importance = Request.QueryString("importance")
surat.Subject = Request.QueryString("subject")
surat.Body = Request.QueryString("body")
surat.Send
Set surat = Nothing
```

Kode PHP:

```
Ini_set("SMTP", "172.16.1.17");
Ini_set("sendmail_from", me@xyz.com);
Mail("bertha@hotmail.com", "Tes", "Message Here");
```

Kekurangan Sistem

Sistem dibangun dengan menggunakan SMTP *Virtual Server* yang tersedia pada Windows 2000. dengan demikian sistem ini tidak menyediakan fasilitas untuk mengakses *email* dengan menggunakan metode POP3. Untuk menyediakan fasilitas POP3, maka dibutuhkan POP *Virtual Server* yang tidak disediakan oleh Windows 2000.

Email yang datang disimpan kedalam satu *folder* merupakan kelemahan lain sistem ini. CDONTS secara langsung akan mengikuti pengaturan pada IIS/SMTP *Virtual Server*, sehingga pengembang tidak dapat membangun pengaturan *mailbox* yang lain agar pengaturan *email* menjadi lebih baik. menyimpan *email* di dalam satu *folder* yang kemudian akan disaring hasilnya.

KESIMPULAN DAN SARAN

Kesimpulan

Sistem aplikasi *email* berbasis *web* adalah suatu aplikasi *email* yang berjalan di dalam jaringan dan bersifat *request-response*. Keunggulan yang dimiliki sistem aplikasi *email* berbasis *web* adalah memungkinkan *user* untuk memeriksa, membaca dan mengirim *email* dari berbagai tempat yang terdapat komputer dan memiliki hubungan ke dalam jaringan *internet*.

Kelemahan yang terdapat dalam sistem ini adalah penggunaan sistem dilakukan secara *online*, sehingga menghabiskan waktu lebih banyak dalam penggunaannya dibandingkan sistem aplikasi *email* berbasis POP.

CDONTS merupakan *object library* sederhana yang memungkinkan pengembang untuk membentuk sistem *email* yang sederhana baik dalam kegunaan maupun pengembangannya. Saat ini keterbatasan CDONTS dalam melakukan pengiriman dan penerimaan *email* hanya dapat dilakukan dengan protokol SMTP.

Saran

Sistem aplikasi *email* berbasis *web* dikembangkan dalam lingkungan Windows 2000 *Profesional* yang tidak menyediakan pelayanan untuk membangun suatu *Domain Name Server* (DNS) yang dapat diberikan tugas sebagai *Mail Exchanger* untuk *email* yang ditujukan kepada *server*. Dengan demikian apabila *email* yang dikirim tidak terdaftar pada *server*, tidak akan diberikan pesan kesalahan kepada *user* yang mengakibatkan *user* tidak dapat mengetahui hasil pengiriman *email* tersebut.

Sistem ini dibangun dengan menggunakan CDONTS yang disediakan untuk lingkungan Windows 2000 *Profesional*. CDONTS merupakan pengembangan dari CDO untuk Windows NT yang memiliki kelebihan dari pendahulunya. Akan tetapi untuk lebih dapat mengoptimalkan aplikasi *email* dapat dicoba dengan menggunakan CDO dengan menggunakan Microsoft Exchange *Server* yang lebih memiliki banyak kemampuan dalam pengaturan *email*.

DAFTAR PUSTAKA

- Edwards, P. 1985. *System Analysis, Design and Development with Structred Concept*. Holt, Reinhart & Windston, New York.
- Fathansyah, Ir. 1999. *Basis Data*. Informatika, Bandung.
- Jenis *Email*, Media Informasi Komputer dan Teknologi. (<http://pawang.web.id>)
- Mitchell et al. 1999. *Profresional CDO Programming*. Wrox Press Ltd, Birmingham.

Purbo, O. W., et al. 1999. *Buku Pintar Internet TCP/IP Standar, Desain dan Implementasi*. Elex Media Komputindo, Jakarta.

Salih, M. 2000. *Active Server Pages Fundamental (Objects and Component)*.
Internet
(<http://msdn.microsoft.com/library/tecchart/>)

Setyomurti. 2001. Maret 2001. Happy birthday to QWERTYUIOP. @ha Vol.7 (43 s/d 45)

@Hak cipta milik IPB University

IPB University





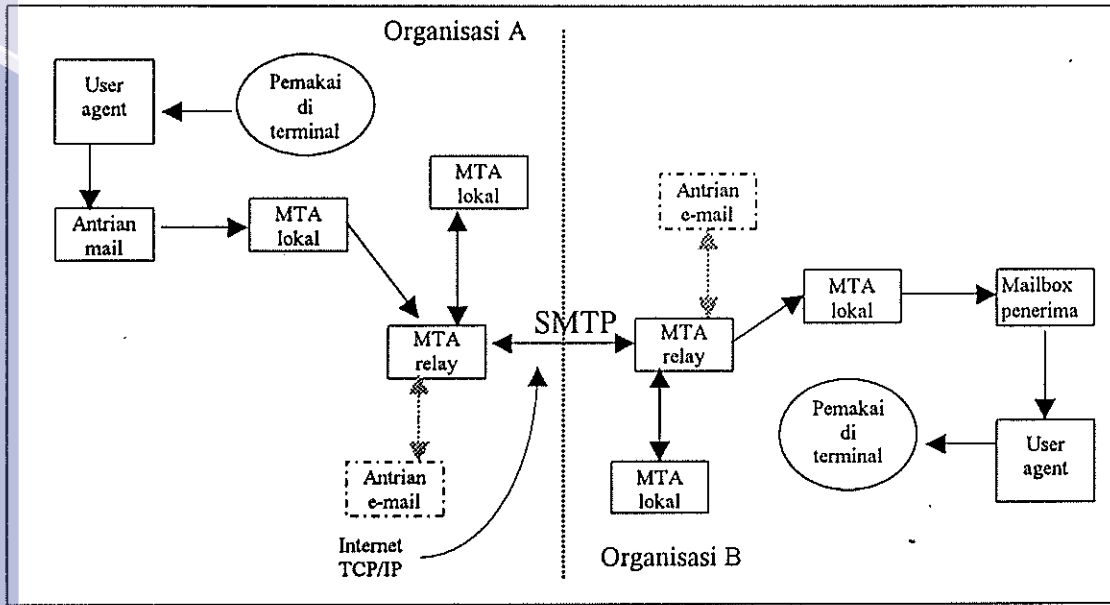
@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

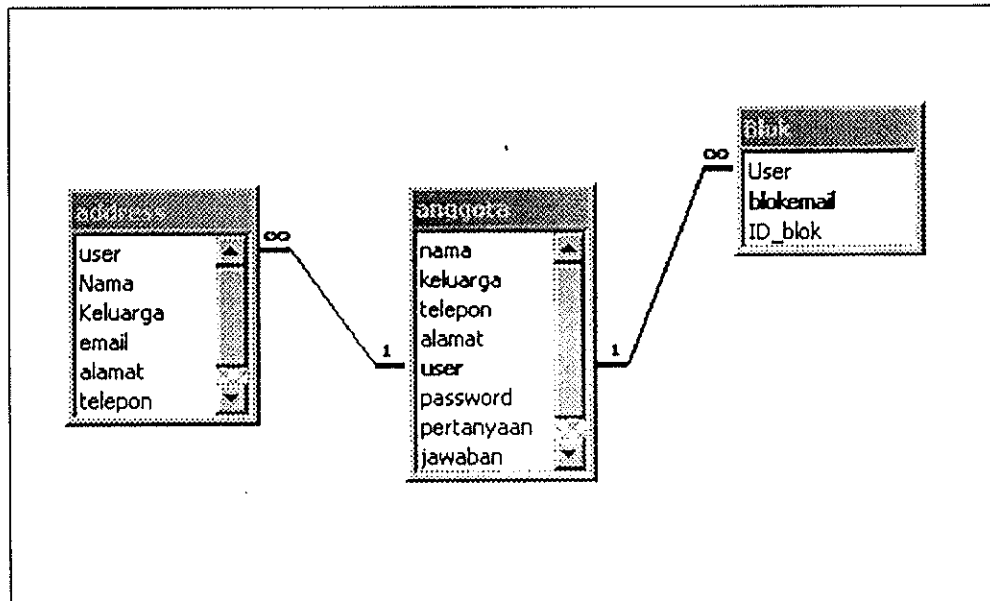
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

LAMPIRAN

Lampiran 1. Sistem Email Dengan 2 Sistem Relay

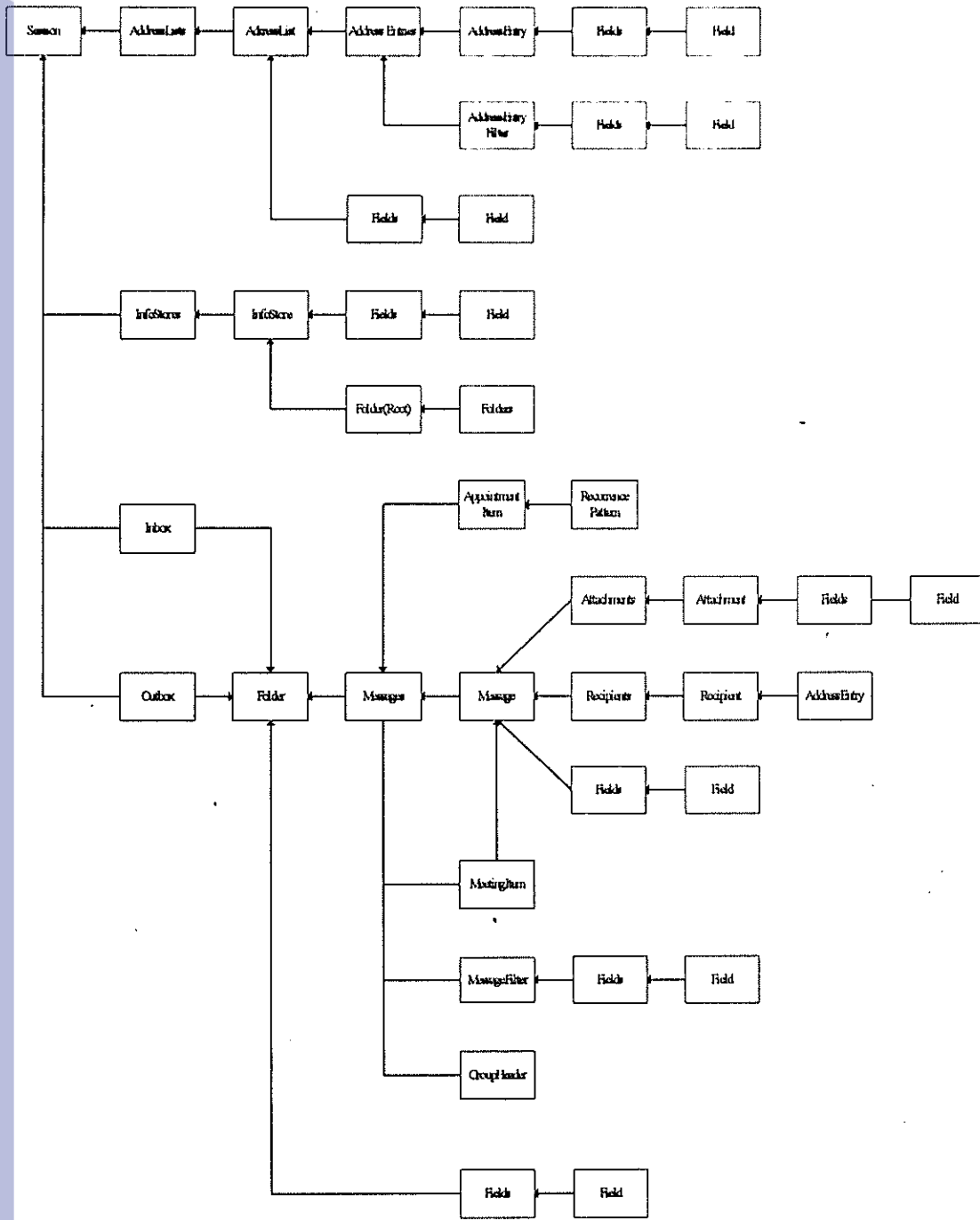


Lampiran 2. Relasi Antar Tabel

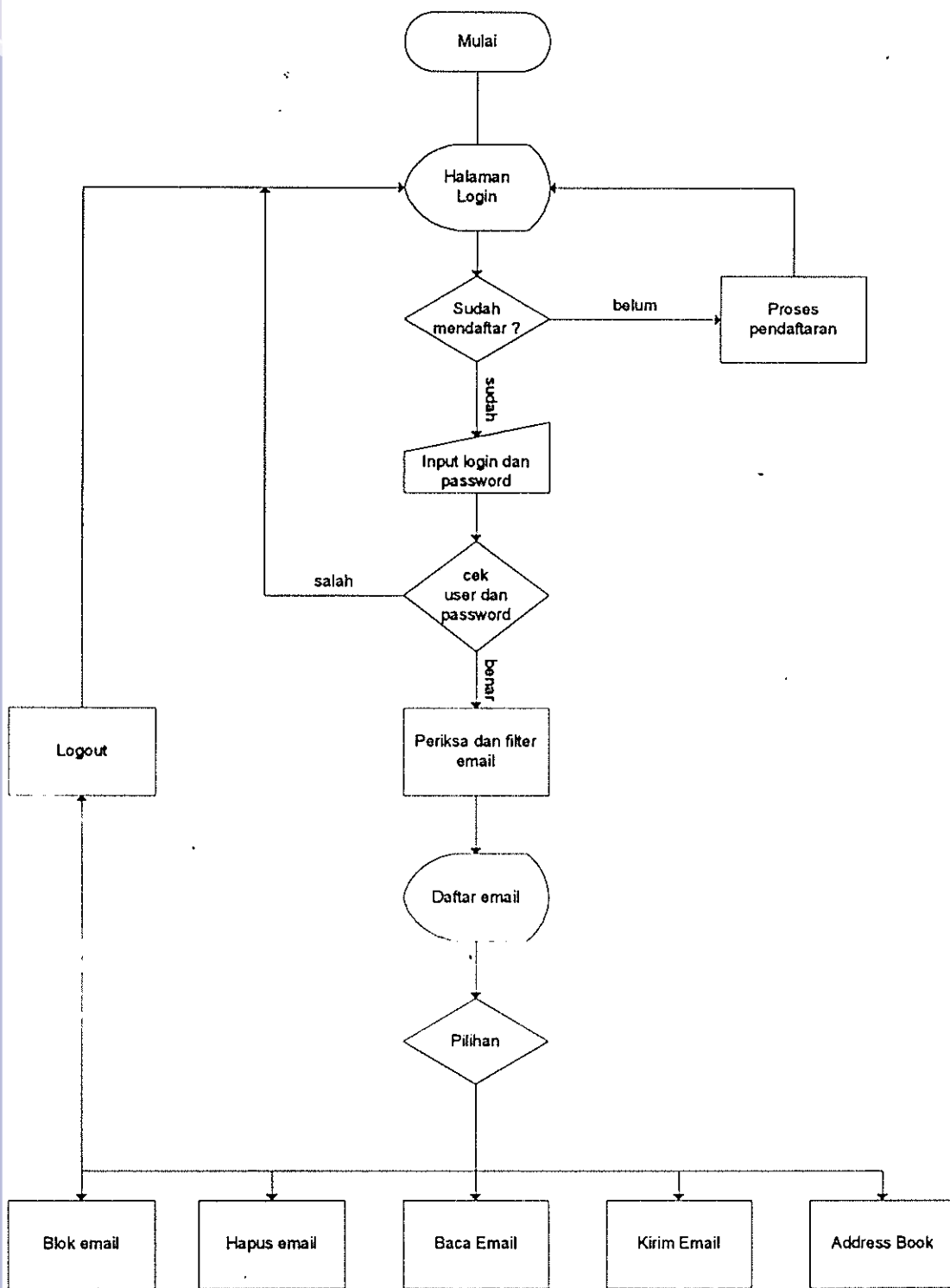


Lampiran 3. Model *Object* CDO

@Hak cipta milik IPB University



Lampiran 4. Alur Data Program



@Hak cipta milik IPB University

IPB University

Lampiran 5. Tampilan *Login*

Welcome to ILKOM.NET webmail
 Please login
 User Name : @ilkom.net
 Password :

[Daftar](#)

[Lupa password](#)

Lampiran 6. Tampilan *Inbox*

[home](#) [inbox](#) [address book](#) [change info](#) [logout](#)
[compose](#)

	<u>From</u>	<u>Subject</u>	<u>Received</u>
<input type="checkbox"/>	andri@ilkom.net	<u>fdgfsdg</u>	2/21/2002 8:56:23 AM
<input type="checkbox"/>	! andri@ilkom.net	<u>gadget</u>	2/21/2002 8:33:23 AM
<input type="checkbox"/>	andri@ilkom.net	<u>remote</u>	2/13/2002 2:37:01 PM
<input type="checkbox"/>	andri@ilkom.net	<u>coba aja</u>	2/10/2002 8:03:20 PM

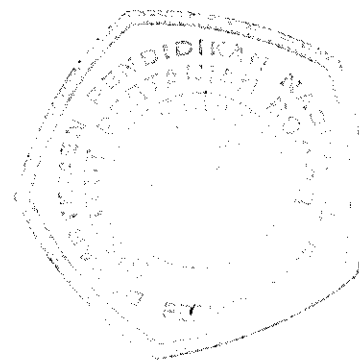
 << >>

© Hak cipta milik IPB University

Lampiran 7. Tampilan Address Book

@ Hak cipta milik IPB University

New Contact					Block Address					Home					Personal Data					Log Out				
No	Name	Email	Address	Telephone																				
1	<u>andri_d</u>	andri30@yahoo.com	USA	431456	Delete																			
2	<u>Bill</u>	bill@ilkom.net	anywhere	6745325	Delete																			
3	<u>indra</u>	indra@ilkom.net	Jakarta	43523	Delete																			
4	<u>Irawan</u>	qwonk@ilkom.net	Bandung	664323	Delete																			
5	<u>Mega</u>	mega@ilkom.net	Jakarta Utara	234557	Delete																			
6	<u>Yusuf</u>	yusuf@ilkom.net	Ahmad Yani	3254355	Delete																			



Hak Cipta Dilindungi Undang-undang
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
 2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.