

BAB II

LANDASAN TEORI

2.1 Gambaran Umum Perum Perhutani

2.1.1 Sejarah Perhutani

Sesuai dengan yang tercantum dalam situs resmi perhutani (<http://www.Perhutani-Unit2.co.id>) dapat diketahui sejarah pengelolaan hutan di Pulau Jawa dimulai sejak tahun 1200-an dengan pembalakan hutan alam oleh VOC. Kemudian mulai awal tahun 1800-an mulai dibangun hutan tanaman (khususnya Jati) oleh Willem Daendels sebagai Gubernur Jendral Hindia Belanda. Pada tahun 1865 dikeluarkan Undang-undang Kehutanan untuk Jawa dan Madura yang mengatur pengelolaan hutan di Pulau Jawa dan Madura.

Pada tahun 1775 mulai diperkenalkan penanaman jati sistem tumpang sari oleh Baurman. Kemudian pada tahun 1892 dibentuk organisasi teritorial kehutanan yang dinamakan Houtvesterij dan Djabatbedrijfs (Perusahaan Jati) oleh Hindia Belanda. Tahun 1898 disusun Rencana Perusahaan pertama oleh Bruisma.

Tahun 1942-1945 pada masa pemerintahan kolonial Jepang dan awal kemerdekaan pengelolaan hutan yang sudah baik mengalami kemunduran. Pada tahun 1950 dibentuk Djawatan Kehutanan oleh Pemerintah RI yang meliputi 5 (lima) Inspektorat. Wilayah Jawa Timur masuk Inspektorat IV (12 Daerah Hutan Jati : 373.635 Ha) dan Inspektorat V (10 Daerah Hutan dengan luas 926.371 Ha). Tahun 1963 Inspektorat di Jawa berubah bentuk menjadi PN Perhutani (Perusahaan Hutan Indonesia). Tahun 1972 berubah bentuk menjadi Perum

Perhutani berdasarkan PP No. 15/1972 yang disempurnakan dengan PP No. 36/1986 dan PP No. 53/1999.

Tujuan perusahaan adalah mengelola hutan sebagai ekosistem sesuai dengan karakteristik wilayah untuk mendapatkan manfaat yang optimal bagi perusahaan dan masyarakat sejalan dengan tujuan pembangunan nasional. Sifat usahanya adalah menyediakan pelayanan bagi kemanfaatan umum dan sekaligus memupuk keuntungan berdasarkan prinsip pengelolaan perusahaan dan kelestarian sumberdaya hutan.

Tahun 2001 Perum Perhutani berubah bentuk menjadi PT. Perhutani (Persero) berdasarkan PP No. 14 tahun 2001 dengan tugas dan wewenang yang sama

2.1.2 Surat perintah perjalanan dinas

Menurut Management Perhutani dalam buku Petunjuk Surat Menyurat (1999:45) Surat Perintah Perjalanan Dinas (SPPD) merupakan salah satu jenis dari surat intern dilingkungan perhutani yang mana surat ini hanya akan beredar didalam lingkungan Perhutani uni II. Surat ini memang sangat diperlukan oleh pihak Kepala Biro untuk memberi tugas kepada staff atau pejabat dibawahnya untuk melakukan dinas diluar kota. SPPD ini juga sebagai acuan oleh kasir perhutani untuk memberika dana sebagai biaya perjalanan dinas

2.2 Teori Aplikasi Otomasi Perkantoran

2.2.1 Otomasi perkantoran

Sesuai uraian oleh Barcomb(1999:30) dapat ditarik garis besar bahwa otomasi perkatoran merupakan suatu rangkaian kata yang terdiri dari otomasi

yang berarti pelaksanaan suatu operasi oleh mesin tanpa intervensi manusia kecuali sebagai aktivator, atau dapat juga didefinisikan sebagai pengalihan pekerjaan manusia kepada mesin, maupun ada pula yang menyebutnya suatu mekanisme yang mempermudah pekerjaan manusia

Dijelaskan pula oleh Barcomb (1999;70) bahwa otomasi perkantoran didefinisikan sebagai penggunaan teknologi untuk membantu manusia dalam melakukan *data processing* menjadi informasi atau dapat juga dikatakan sebagai suatu konsep, pendekatan ke pola pikir baru tentang bagaimana menangani informasi. Sehingga dalam mengembangkan suatu sistem otomasi perkantoran perlu melihat 11 asas perkantoran yang baik yaitu :

- a. Arus kerja (workflow) lancar
- b. Tidak terjadi duplikasi pekerjaan
- c. Minimalisasi *employee travelling*
- d. Menghindari pekerjaan tulis-menulis yang tidak perlu
- e. Maksimalisasi spesialisasi
- f. Penerapan *Management by exception*
- g. Minimalisasi penyimpangan/penyalahgunaan wewenang/sumber daya
- h. Menghindari pemeriksaan yang tidak perlu
- i. Optimalisasi alat-alat yang ada
- j. Minimalisasi penggunaan kertas
- k. Berasaskan kesederhanaan

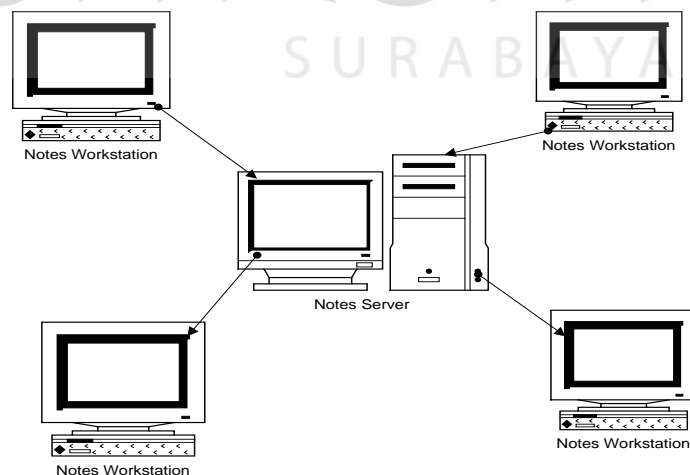
Sehingga dapat ditarik suatu pengertian umum bahwa OA tersebut merupakan konsep pengintegrasian berbagai komponen untuk melakukan data processing dengan bantuan teknologi secara maksimum dan intervensi minimum

dari manusia. Sehingga otomatisasi perkantoran ini akan sangat membantu dalam penyelesaian pekerjaan kantor.

2.2.2 Lotus notes

Menurut Kirkland (2000;45) Lotus Notes adalah suatu program groupware yang mengelola informasi untuk banyak pemakai ke sebuah jaringan komputer. Notes menggunakan aplikasi yang terdiri dari database untuk memungkinkan kita dan kolega kita berkomunikasi secara elektronik dan mengumpulkan serta membagi informasi dengan menggunakan komputer pribadi kita.

Notes pada dasarnya terdiri dari dua program: perangkat lunak terminal kerja yang beroperasi pada komputer kita (dan pada komputer kolega kita); dan perangkat lunak server yang beroperasi di lokasi sentral di jaringan. Terminal kerja Notes dapat pula mengoperasikan tipe perangkat lunak lain (misalnya seperti spreadsheet, pengolah kata dan sebagainya) kendati begitu, sebuah server Notes biasanya ditujukan semata-mata untuk menyajikan layanan Notes.



Gambar 2.1. Sistem Notes

Suatu sistem Notes terdiri dari terminal kerja dan server Notes yang beroperasi di *Local Area Network* atau LAN. Suatu sistem Notes dapat berisi lebih dari satu server Notes dan dapat beroperasi di lebih dari satu LAN. Server Notes tambahan menyediakan lebih banyak ruang untuk database dan mendistribusikan beban lalu lintas komunikasi. Dengan menggunakan dua LAN atau lebih, suatu sistem Notes dapat memberikan daerah geografis yang lebih luas. Pemakaian seperti ini memungkinkan beberapa cabang dari suatu perusahaan menggunakan bersama-sama suatu sistem Notes.

Ada tiga blok bangunan dari sebuah database Notes:

a. Dokumen

Meskipun masing-masing database Notes didesain untuk memenuhi kebutuhan kelompok kerja tertentu, semua database menggunakan bersama-sama struktur dasar yang mirip. Pada database apa saja, para pemakai memasukkan dan menggunakan bersama-sama informasinya di *dokumen*. Menurut Kirkland (2000:75) sebuah dokumen sama dengan sebuah record di sebuah database yang berkaitan atau dengan sebuah baris di suatu tabel. Meski begitu, tak seperti sebuah record di sebuah database yang berkaitan, sebuah dokumen Notes tidak mempunyai sembarang atas panjang yang sudah diatur sebelumnya. Sebuah dokumen Notes dapat terdiri dari satu surat atau beberapa halaman. Gambar 2.2 memperlihatkan sebuah dokumen Notes :

Gambar 2.2. Sebuah Dokumen Notes

b. Formulir dan Field

Menurut Tamura (1999:35) formulir mengontrol bagaimana para user memasukkan, mengedit, dan membaca dokumen disebuah database. Formulir pola untuk sebuah dokumen, dapat mengubahnya menjadi sebuah dokumen yang aktual dengan mengisi formulir kosong lalu menyimpannya disebuah database.

Formulir adalah bagian dari desain database dan dibuat oleh designer database

Setiap daerah yang kosong disebuah formulir dinamakan field dan dapat berisi jenis data tertentu. Sebuah field dapat berupa salah satu dari dua tipe data utama :

1. *Field yang dapat diedit*, seorang pemakai dapat memasukkan atau mengganti informasi.
2. *Field yang terkomputerisasi* memasok informasi secara otomatis. Dalam banyak hal, user tidak dapat mengganti informasi disebuah field yang terkomputerisasi. Contoh informasi yang dapat dimuat oleh tipe field ini adalah

nama penulis suatu dokumen dan tanggal suatu dokumen yang disusun. Sebuah field yang terkomputerisasi juga dapat menggabungkan informasi dari dua field lain atau lain.

Tanggal	09/01/2004
User Aktif	Kasi Anggaran

PERUM PERHUTANI
(PERUSAHAAN UMUM KEHUTANAN NEGARA)
UNIT II JAWA TIMUR
GRAHA PERHUTANI Jl. Gentengkali No.49 Surabaya
Telp. 5343851-5343852-5343853-5343854

SURAT PERINTAH PERJALANAN DINAS
No: / SPPD / UNIT II / 2004

1. a. Pejabat yang berwenang memberi perintah	a. <input type="text"/>
b. Jabatan	b. Kasi Anggaran
2. Nama pejabat yang diperintah	<input type="text"/>
3. a. Pangkat	a. <input type="text"/>
b. Jabatan	b. <input type="text"/>
c. Gaji pokok	c. <input type="text"/>
d. Tingkat menurut peraturan perjalanan dinas	d. <input type="text"/>
4. Maksud perjalanan dinas	<input type="text"/>
5. Alat angkut yang dipergunakan	<input type="text"/>
6. a. Tempat berangkat	a. <input type="text"/>
b. Tempat tujuan	b. <input type="text"/>
7. a. Lamanya perjalanan dinas	a. <input type="text"/>
b. Tanggal berangkat	b. <input type="text"/>
c. Tanggal harus kembali	c. <input type="text"/>

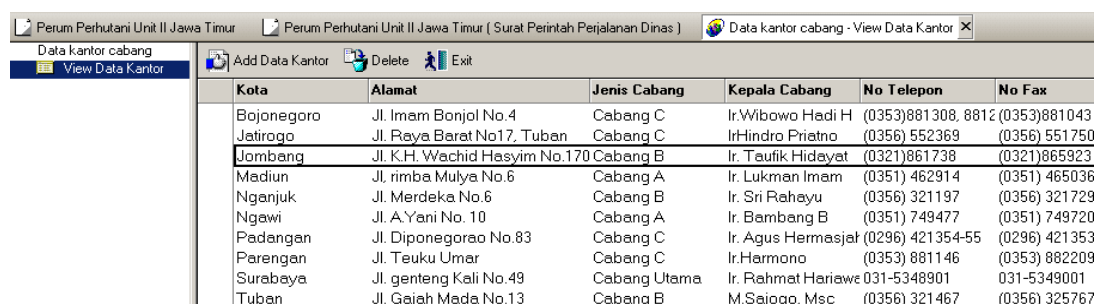
Gambar 2.3. Formulir dan Field Notes.

c. View

Masih dalam buku yang sama Tamura mengatakan (1999:50) dokumen database dikelola didalam *View* untuk membuatnya lebih mudah mencari dokumen dan memeriksa hubungan diantaranya. *View* adalah daftar informasi dari beberapa atau semua dokumen disebuah database dan masing-masing kolom menyajikan beberapa kombinasi data dari dokumen.

Setiap database memiliki paling tidak satu *view* dan dapat mempunyai banyak *view*. *View* yang muncul pertama kali saat membuka sebuah database

(disebut Main View) adalah default dan biasanya menampilkan semua dokumen database dalam suatu cara yang cocok dengan jenis informasi didalam database. Dibagian bawah View menu terdapat semua view didatabase.



Kota	Alamat	Jenis Cabang	Kepala Cabang	No Telepon	No Fax
Bojonegoro	Jl. Imam Bonjol No.4	Cabang C	Ir.Wibowo Hadi H	(0353)881308, 8812	(0353)881043
Jember	Jl. Reya Barat No17, Tuban	Cabang C	IrHindro Priatno	(0356) 552369	(0356) 551750
Jombang	Jl. K.H. Wachid Hasyim No.170	Cabang B	Ir. Taufik Hidayat	(0321)861738	(0321)865923
Madiun	Jl. rimba Mulya No.6	Cabang A	Ir. Lukman Imam	(0351) 462914	(0351) 465036
Nganjuk	Jl. Merdeka No.6	Cabang B	Ir. Sri Rahayu	(0356) 321197	(0356) 321729
Ngawi	Jl. A.Yani No. 10	Cabang A	Ir. Bambang B	(0351) 749477	(0351) 749720
Padangan	Jl. Diponegorao No.83	Cabang C	Ir. Agus Hermasjal	(0296) 421354-55	(0296) 421353
Parengan	Jl. Teuku Umar	Cabang C	Ir.Hermono	(0353) 881146	(0353) 882209
Surabaya	Jl. genteng Kali No.49	Cabang Utama	Ir. Rahmat Hariawe	031-5348901	031-5349001
Tuban	Jl. Gajah Mada No.13	Cabang B	M.Sajogo, Msc	(0356) 321467	(0356) 325767

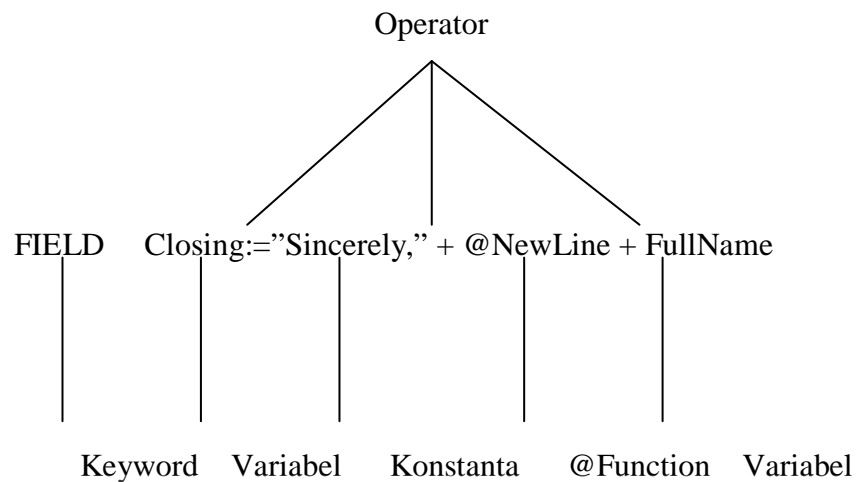
Gambar 2.4. View dari Suatu Notes.

2.2.3 Formula Language

Untuk menjelaskan maksud dari formula language dapat diambil beberapa point dari buku Tamura (1999:75) bahwa sebuah *formula* mengambil informasi dari notes dan mengerjakan berbagai operasi dengan data itu. Ada beberapa alasan menggunakan formula di sebuah formulir antara lain: untuk menjelaskan sebuah field, untuk mendefinisikan sebuah tombol, untuk mendefinisikan sebuah pop-up atau untuk menetapkan sebuah judul jendela.

Sebuah formula memerlukan informasi agar dapat bekerja dan instruksi apa yang dikerjakan dengan berbagai potongan data yang tersedia untuknya. Formula adalah kombinasi dari lima blok bangunan : variabel konstanta, operator, @Function, dan keyword. Dengan menggunakan formula, designer dapat menggabungkan lima komponen tersebut dalam berbagai cara yang luar biasa

banyaknya, untuk membuat suatu range formula dari yang paling dasar hingga paling rumit. Gambar 2.5 menunjukkan komponen-komponen dari suatu formula :



Gambar 2.5. Komponen-Komponen dari Suatu Formula

A. Variabel

Sebuah *variabel* adalah suatu nilai yang tidak pasti. variabel dapat (dan barangkali akan) mempunyai nilai yang berbeda pada waktu yang berbeda. Seringkali variabel disuatu formula akan menjadi nama field disebuah formula, user dapat menggunakan nama field lain untuk menghasilkan suatu nilai, atau sebagai argumen untuk @function. Berikut ini beberapa contoh pemakaian nama field tersebut di formula :

Tabel 2.1 Contoh Pemakaian Field

Bagaimana Dipakai	Contoh	Apa yang dikerjakan
▪ Sendiri	CustAddress	Menampilkan nilai di CustAddress

Tabel 2.1.(Lanjutan)

<ul style="list-style-type: none"> ▪ Dengan satu field lain atau lebih untuk menghasilkan suatu nilai 	<code>Firstname+" "+lastName</code>	Menampilkan data difield FirstName dengan data diField lastname dipisahkan oleh sebuah spasi
<ul style="list-style-type: none"> ▪ Sebagai argumen untuk function 	<code>@Uppercase(LastName)</code>	Menampilkan data dari Field LastName dalam huruf besar

B. Konstanta

Formula juga membutuhkan potongan informasi statis yang disebut konstanta. User menggunakan konstanta untuk memasok karakter yang tepat yang sesuai dengan keinginan, agar kalkulasi oleh formula. Berikut ini deskripsi tentang tiga tipe konstanta:

a. Untaian teks

Sembarang karakter, termasuk spasi, diapit dalam tanda kutip.

Contoh formula `"Profile for" +CompanyName`

Akan menampilkan untaian teks Profile for dengan data difield CompanyName.

b..Angka

Tidak ada persyaratan format kusus (misalnya user tidak harus mengapit angka dalam tanda kutip) ntuk sebuah angka disuatu formula.

Contoh formula `Cost * 2` Akan menampilkan nilai Cost yang dikalikan dengan 2.

c. Nilai waktu dan tanggal

Angka-angka diapit dalam tanda kurung siku([]). Default pemisah adalah slash(/). Sebagai contoh user dapat menampilkan tanggal [11/07/99]

C. Operator

Operator menyajikan instruksi matematis bagi suatu formula (Misal operator * menjelaskan Notes agar menampilkan nilai). Notes mengevaluasi operator dalam serangkaian urutan yang dikenal sebagai urutan preseden. Jika ada dua operator disuatu formula mempunyai urutan yang sama, (misal dua penjumlahan) Notes mengajarkan operasi dari kiri ke kanan.

Urutan urutan di formula Notes adalah:

- a. Tanda kurung
- b. Pangkat dalam aljabar
- c. kalian
- d. bagian
- e. jumlahan
- f. Pengurangan

Contoh dalam persamaan $7 + 4 * 6 = 31$, Notes mengerjakan operator perkalian sebelum penjumlahan. Walau begitu dapat digunakan tanda kurung untuk menindih urutan operasi natural. Kalau user melakukan tanda kurung disekitar $7 + 4$ dalam contoh ini, hasilnya akan berubah menjadi $(7 + 4) * 6 = 66$

D. Keyword Formula

Ada lima *keyword formula* yang dapat dipakai untuk mengerjakan fungsi-fungsi kusus disuatu formula Notes. User harus meletakkan keyword formula diawal suatu baris formula dan mengakhiri dengan titik koma, kecuali

untuk keyword `SELECT` dan `ENVIRONMENT`. Berikut ini lima keyword formula yang tersedia untuk dipakai Notes :

- a. `DEFAULT` dipakai untuk menetapkan default nilai sebuah field temporer, yang memastikan bahwa field tidak kosong.]
- b. `FIELD` dipakai untuk menetapkan sebuah nilai ke field yang disimpan didokumen. Dapat dipakai untuk mengganti informasi di field yang sudah ada atau untuk membuat sebuah field baru. Contoh,

```
FIELD FullName:= first + " " + Lst;
```

membuat sebuah field baru yang dinamakan `FullName` dengan mendaftarkan data di field `First` dan field `Last` menggunakan spasi diantaranya.

- c. `REM` dipakai untuk menambahkan ucapan, penjelasan, atau komentar ke suatu formula. Ucapan dapat menjadi piranti dokumentasi yang bermanfaat bagi formula yang rumit, terutama jika para pemakai lain perlu menyalin atau mengedit formula. user harus menyertakan teks `REM` dalam tanda kutip. Pada contoh berikut ini, pernyataan `REM` (yang merupakan tiga baris pertama) dipakai untuk menjelaskan formula yang mengikutinya:

```
REM "This Formula display the company's city,";
REM "State, and zip code on the same line";
REM "With a space between each";
City + " " + State + " " + ZipCode
```

E. `SELECT`

Dipakai untuk mengidentifikasi dokumen tertentu atau dokumen yang berisi informasi spesifik di sebuah field tertentu. Contoh,

```
SELECT Company ="SLR Communications"
```

Memilih dokumen dengan teks "SLR Communications" di field `Company`

F. ENVIRONMENT

Dipakai untuk menetapkan suatu variabel untuk menetapkan suatu variabel yang unik dimiliki user, maka formulanya:

```
CurrentCustNumb      :=      @TextToNumber      (@Environment
CustomerNumber));
NextCustNumber := CustomerNumber + 1;
Environment CustNumber := @Text (NextCustNumber)
```

Memanggil nomor pelanggan yang disimpan di NOTES.INI milik pemakai, mengkonversikannya menjadi sebuah nomor, dan menyimpannya sebagai nomor pelanggan yang aktual. Lalu environment menambahkan 1 ke nomor pelanggan, mengkonversikannya kembali menjadi teks, dan menyimpan nomor yang baru di NOTES.INI sebagai nomor pelanggan selanjutnya. Formula ini berguna untuk mengotomasi tugas menetapkan nomor pelanggan.

G. Formula @Function

Notes memperlengkapi user dengan “mini formula” yang telah terpasang didalamnya atau *@Function* yang mengerjakan berbagai kalkulasi yang sudah didefinisikan secara spesifik pada tipe data yang berbeda.

Berikut ini beberapa garis pedoman jika user menggunakan *@Function*:

- a. *@Function* selalu mengembalikan suatu nilai yang tipe datanya merupakan teks (termasuk field data keyword), angka, waktu, logika.
- b. Beberapa *@Function* memerlukan *argumen* (nilai yang digunakan untuk mengerjakan operasi) untuk menghasilkan suatu nilai sewaktu *@Function* yang lain memproduksi suatu nilai. Contoh dari *@Function* yang memerlukan sebuah argumen adalah:

```
@ProperCase(FirstName)
```

c. @Function ini mengkonversi data di field FirstName menjadi kapitalisasi nama yang pantas; huruf pertama adalah huruf besar, yang lainnya adalah huruf kecil.

@Now

memasukkan tanggal dan waktu yang aktual serta tidak memerlukan lain.

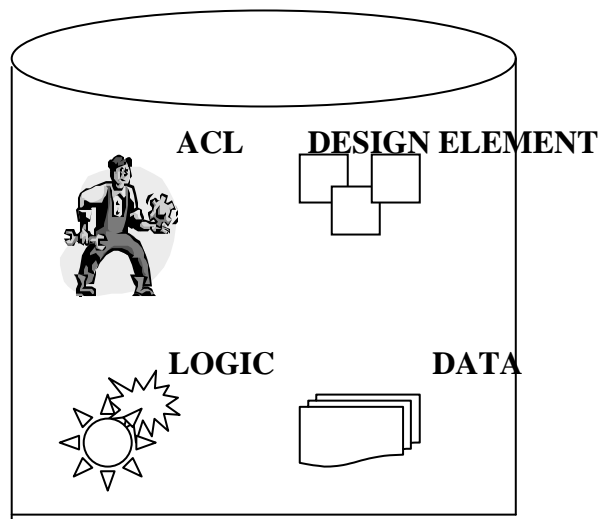
2.2.4 Lotus Domino Server

Dalam bukunya Kirkland (2000:79) menjelaskan lotus domino server adalah adalah suatu program server pendukung Lotus Notes, yang dimasukkan pada masing-masing komputer yang terpisah dan diakses melalui LAN, WAN atau modem untuk dapat di digunakan oleh semua user dalam suatu group.

Domino server adalah rancangan dari database-database Notes. Izin atau penolakan akses oleh user pada database tergantung pada daftar kontrol akses yang diolah pada setiap database. Misal, user1 diberi akses untuk menulis dan mengedit isi dokumen dalam database, user2 hanya diberi akses untuk membaca dokumen yang ada pada database, selain untuk akses diluar itu maka akan ditolak oleh database.

Database Domino berisi aplikasi-aplikasi dari Notes yang telah dibuat. Desain aplikasi dan data dimasukkan secara bersama-sama didalam sebuah file yang disebut Notes Storage Facility (NSF) yang merupakan ekstensi standar untuk file-file database. Apabila user membuat database baru, Notes secara otomatis menambahkan ekstensi ini setelah nama file tersebut.

Gambar 2.6 adalah ilustrasi grafik dari Database domino dan komponen yang ada didalamnya :



Gambar 2.6. Notes Storage Facility (NSF)

Berikut ini adalah komponen dari database Domino beserta fungsi dan contohnya:

a. Komponen ACL

Komponen ini berfungsi membatasi aplikasi user dan apa yang dapat dilakukan user dalam aplikasi tersebut.

Contoh: seorang user dapat membaca data, tetapi tidak bisa membuat dokumen.

b. Desain Elemen

Desain elemen berfungsi membangun blok dari aplikasi. Pengembang menggunakan desain elemen untuk mengontrol bagaimana user berinteraksi dengan data dalam aplikasi tersebut.

Contoh : Pada halaman depan suatu aplikasi terdapat text, grafik dan link.

c. Logika (Logic)

Komponen ini membantu pengembang untuk menambahkan logika kedalam desain elemen yang berfungsi untuk:

- Mengkalkulasi, modifikasi, atau mengakses nilai suatu field.
- Mengotomasikan tugas.

Contoh: menyimpan data ketika user mengklik tombol.

d. Data

Data adalah informasi dari suatu aplikasi, dimana user membuat, modifikasi, dan membaca data melalui desain elemen

Contoh : nama, alamat, nomer telepon dari customer

Domino server merupakan rancangan dari database-database Notes. Ijin atau penolakan akses oleh pemakai pada database tergantung pada daftar kontrol akses yang diolah pada setiap database. Sebagai contoh, pemakai1 diberi akses untuk menulis dan mengoreksi isi dokumen dalam database, pemakai2 hanya diberi akses untuk membaca dokumen yang ada pada database, untuk akses lain diluar hak yang telah diberikan akan ditolak oleh database. Berikut ini tingkatan akses pemakai Notes:

a. No Access

Para user sama sekali tidak mempunyai akses ke database.

b. Depositor

Para user dapat menyusun dokumen, tetapi tidak dapat membaca sembarang dokumen, bahkan dokumen yang telah disusun oleh para user

c. Reader

Para user dapat membaca dokumen, tetapi tidak dapat menyusunnya.

d. Author

Para user dapat menyusun dan membaca dokumen serta dapat mengedit dokumen yang disusun oleh para user.

e. Editor

Para user dapat menyusun, membaca, dan mengedit semua dokumen.

f. Designer

Para user memiliki semua hak akses Editor dan juga dapat memodifikasi icon database serta dokumen-dokumen About Database dan Using Database, macro, formulir dan tampilan.

g. Manager

Para user memiliki semua hak akses designer dan dapat pula memodifikasi ACL serta menghapus database dari server.

2.2.5 Lotus Domino Designer

Dalam Pengembangan otomasi perkantoran ini menggunakan tool dalam lotus yang disebut lotus Domino Designer. Lotus Domino Designer ini merupakan tipe kedua dari *Notes client*. Aplikasi ini memiliki perbedaan dalam *user interface* dan ijin dari Notes Client yang biasa.

Dengan Domino Designer kita dapat membuat aplikasi yang dapat digunakan baik dalam bentuk *web* ataupun aplikasi dalam Notes Client. Domino Designer ini sudah dilengkapi dengan (IDE) Integrated Development Environment, yang menyediakan kemampuan untuk mengembangkan aplikasi

dengan bahasa pemrograman yang sedang berkembang saat ini yaitu Java dan Java Script yang mana berguna untuk membangun aplikasi berbasis web.

2.2.6. Local Area Network (LAN)

Menurut Tanenbaum(1997:25) local area network (LAN) merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan workstation dalam kantor perusahaan atau pabrik-pabrik untuk memakai bersama *resource* dan saling bertukar informasi. LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteristik: (1) ukuran, (2) teknologi transmisi, dan (3) topologinya.

Dalam pembuatan aplikasi surat menyurat dalam lingkungan kantor dengan teknologi Lotus Notes sangat diperlukan guna mendukung office automation, sehingga karyawan dapat saling bertukar informasi dengan cepat secara paperless.

LAN mempunyai ukuran yang terbatas, yang berarti bahwa waktu transmisi pada keadaan terburuknya terbatas dan dapat diketahui sebelumnya. Hal ini juga memudahkan manajemen jaringan.

2.2.7 E-Mail (Electronic Mail) / Surat Elektronik

Dijelaskan dalam buku Reber(1996:40) aplikasi email ini merupakan satu kesatuan dengan Lotus Notes dan Domino sejak pertama kali dikenalkan. Dengan sudah memasyarakatnya email saat ini diperlukan sebuah sistem yang dapat dengan stabil dan dapat diandalkan untuk mengelola email tersebut.

Dalam buku Burke (1999:60) sistem dalam Domino email server dapat mengelola beberapa database mail, selain itu dapat juga mengatasi jumlah user dengan jumlah banyak karena aplikasi ini merupakan aplikasi client / server, dimana server dapat berkomunikasi langsung dengan dengan server lain, mengirim dan menerima mail serta berkomunikasi dengan client juga.

2.2.8 Analisa dan perancangan sistem.

Analisa sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.

Analisis sistem dilakukan setelah tahap perencanaan sistem. Adapun langkah-langkah dasar dalam melakukan analisis sistem :

1. Identify, yaitu mengidentifikasi masalah
2. Understand, yaitu memahami kerja dari sistem yang akan kita buat.
3. Analyze, yaitu menganalisa sistem yang kita
4. Report, yaitu membuat laporan hasil analisis permasalahan

Setelah analisis sistem dilakukan, tahap selanjutnya adalah perancangan sistem. Perancangan sistem dapat didefinisikan sebagai tahap setelah

1. Perancangan sistem secara umum
2. Perancangan sistem secara terinci
 - a. Workflow

Pada tahap awal kita harus mengetahui secara benar tentang workflow yang ada di Perum Perhutani Unit II Surabaya agar dalam proses pengembangan aplikasi dapat sesuai dengan workflow yang sudah ada. Pengertian dari workflow

itu sendiri yaitu suatu otomasi dari proses surat menyurat, baik secara keseluruhan maupun sebagian, dimana dokumen, informasi atau tugas diteruskan dari satu *partisipan* ke *partisipan* lain sesuai dengan prosedur, ketentuan yg berlaku

b. Data Flow Diagram (DFD)

Pada tahap analisis dan perancangan sistem, penggunaan notasi dapat membantu komunikasi dengan pemakai/user sistem untuk memahami sistem tersebut secara logika. Diagram yang menggunakan notasi –notasi untuk menggambarkan arus dari data sistem ini dikenal dengan nama Diagram Arus data (Data Flow Diagram).

Data Flow Diagram (DFD) digunakan untuk menggambarkan arus data didalam sistem secara terstruktur dan jelas, menggambarkan arus data dari suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika, tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan.. DFD juga dapat merupakan dokumentasi dari sistem yang baik. Dengan adanya data flow diagram akan mempermudah dalam melakukan analisa sistem. Sehingga pada akhirnya hasil dari pengembangan software otomasi perkantoran dapat dilihat dan apakah sudah sesuai dengan sistem yang ada di lapangan.

DFD berfungsi untuk menggambarkan proses aliran data yang terjadi didalam sistem dari tingkat yang tertinggi sampai tingkat terendah, yang memungkinkan kita untuk melakukan dekomposisi atau membagi sistem kedalam bagian-bagian yang lebih sederhana. DFD memiliki empat simbol yaitu :

1. *External entity*, merupakan kesatuan (entity) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang akan memberi input atau menerima output dari sistem
2. *Data Flow* (Arus Data), menunjukkan aliran data diantara proses, data store dan external entity, yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.
3. *Process* (proses), menunjukkan kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk menghasilkan arus data yang keluar dari proses.
4. *Data store* (Penyimpanan Data), menunjukkan penyimpanan data yang berupa file atau database dalam komputer

c. Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara entity beserta relasinya. Entity merupakan sesuatu yang ada dan terdefinisi di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap entity biasanya mempunyai atribut yang merupakan ciri entity tersebut. Sedangkan relasi adalah hubungan antar entity yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar entity.

2.2.9. Interaksi manusia dan komputer

Dapat diambil sedikit intisari dari buku Barcomb (1989:30) bahwa interaksi manusia dan komputer mengandung maksud bahwa manusia dan komputer dapat saling tukar menukar informasi layaknya percakapan antara orang dengan komputer. Faktor manusia atau ergonomi dalam sistem komputer memperhitungkan bagaimana komputer berinteraksi dengan manusia. Untuk

seperti hubungan pemakai dengan komputer secara baik adalah dengan membaginya kedalam suatu kategori minimal 8 kategori:

1. Pemakai komputer.
2. Alat input.
3. Bahasa input.
4. Rancangan dialog.
5. Pemandu user.
6. Pesan yang timbul.
7. Rancangan layar.
8. Waktu respon komputer.

Dokumentasi komputer merupakan salah satu tipe dari petunjuk pemakai yaitu merupakan salah satu penyimpanan detail yang biasanya menawarkan grafik dan kata-kata, untuk bermacam-macam contoh untuk menjelaskan masalah dan konsep. Dalam bidang otomasi perkantoran keakuratan dari input sangat menentukan keakuratan dari output otomasi perkantoran itu.