BAB II

LANDASAN TEORI

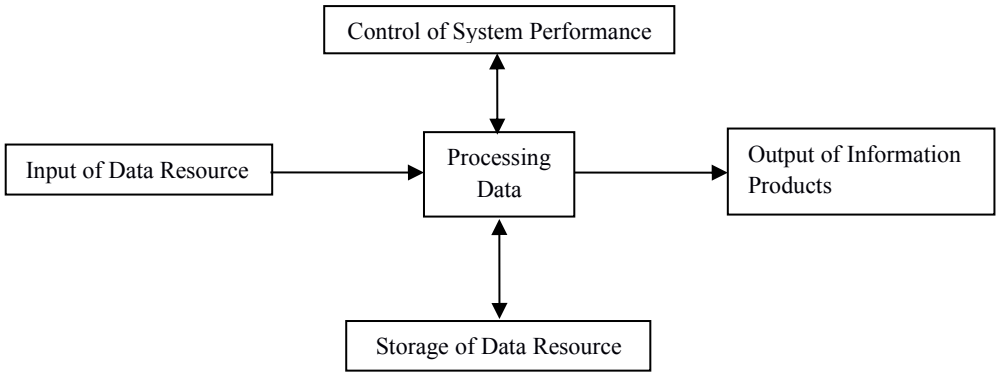
2.1 Informasi dan Data

Menurut Stephen A. Moscove dan Mark G. Simkin (1994) yang dikutip oleh Jogiyanto (1997:1), sistem adalah suatu kesatuan yang terdiri dari interaksi subsistem yang berusaha untuk mencapai tujuan (*goal*) yang sama. Suatu sistem mempunyai tujuan atau sasaran. Tujuan biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Sasaran menentukan masukan dan keluaran yang dihasilkan. Sistem dikatakan berhasil jika mencapai sasaran dan tujuan.

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam suatu organisasi. Informasi merupakan data yang telah diolah untuk menjadi bentuk yang lebih berguna bagi pihak penerima dan didalamnya mnenggambarkan suatu kejadian-kejadian (*event*) dam kesatuan nyata (*fact* dan *entity*) (Jogiyanto, 1997:25).

Sistem informasi terdiri dari *input*, proses, dan *output*. Pada proses terdapat hubungan timbal balik dengan dua elemen, yaitu kontrol kinerja sistem dan sumber-sumber penyimpanan data, baik berupa karakter-karakter huruf maupun berupa numerik. Saat ini data bisa berupa suara atau *audio* maupun gambar atau *video*. Data ini diproses dengan metode-metode tertentu dan akan menghasilkan *output* yang berupa informasi. Informasi yang dihasilkan dapat berupa laporan atau *report* maupun solusi dari proses yang telah dijalankan.

Penggambaran proses yang terjadi dalam suatu siklus sistem informasi seperti dijelaskan dalam Gambar 2.1.



Gambar 2.1 Proses Sistem Informasi
(Sumber: Herlambang dan Tanuwijaya, 2005)

Untuk menjadi bernilai bagi manager dan pembuat keputusan, informasi seharusnya memiliki karakteristik seperti tabel di bawah ini :

Tabel 2.1 Karakteristik keputusan

Akurat	Informasi yang akurat adalah informasi yang bebas dari *error*. Dalam beberapa kasus, informasi yg tidak akurat dihasilkan karena data yang digunakan pada pemrosesan tidak akurat (biasanya disebut *garbage in, garbage out* [GIGO])
Lengkap	Informasi yang lengkap berisi semua kebenaran(data) yang lengkap. Contoh, sebuah laporan investasi tidak akan lengkap tanpa adanya semua biaya penting.
Ekonomis	Informasi seharusnya ekonomis dalam pembuatannya. Para pembuat keputusan selalu akan membandingkan nilai guna informasi dan biaya yang dikeluarkan utk membuatnya.
Fleksibel	Informasi yang fleksibel dapat digunakan untuk berbagai tujuan. Misalnya, informasi jumlah inventori pada bag tertentu dapat digunakan oleh bagian penjualan untuk penutupan pada penjualan, oleh manajer produksi utk menentukan apakah inventori tsb

	perlu ditambah, dan oleh bag keuangan untuk menentukan nilai total investasi perusahaan utk bag inventori.
Handal	Informasi yang handal dapat diandalkan. Dalam banyak kasus, kehandalan sebuah informasi beragantung pada metode pengumpulan data tsb. Dalam contoh lain, kehandalan ini bergantung pada sumber dari informasi tsb.
Relevan	Informasi yang relevan penting bagi pembuat keputusan. Istilahnya, informasi bahwa harga kayu turun, tidak relevan bagi pabrik chip computer.
Simpel	Informasi seharusnya juga simpel, tidak terlalu rumit. Informasi yang mutakhir dan detil mungkin tidak dibutuhkan. Kenyataannya, informasi yang berlebih dapat menyebabkan overload informasi, dimana para pembuat keputusan mempunyai informasi berlebih dan tidak bisa menentukan yang mana yang penting.
Tepat waktu	Informasi yang tepat waktu adalah informasi yang ada pada saat yang dibutuhkan. Istilahnya, mengetahui cuaca minggu lalu, tidak akan membantu kita menentukan pakaian apa yang harus kita pakai pada hari ini.
Dapat dibuktikan	Informasi seharusnya dapat dibuktikan. Ini berarti anda dapat memeriksa untuk memastikan bahwa informasi tsb benar, mungkin dgn memeriksa sumber lain utk informasi yang sama.
Dapat diakses	Informasi seharusnya mudah diakses oleh pengguna utk mendapatkan bentuk informasi yang tepat dan disaat yang tepat utk mendapatkan yang mereka butuhkan.
Aman	Informasi seharusnya aman dari jamahan pengguna yang tidak berhak mengakses.

2.2 Kriteria Pekerjaan

Setiap posisi/jabatan dalam suatu perusahaan akan memiliki setiap daftar pekerjaan (*job description*) yang harus diketahui dan dilaksanakan oleh setiap karyawan. Setiap kriteria pekerjaan haruslah memiliki penjelasan yang jelas mengenai apa yang harus dilakukan dan disusun sesuai dengan kemampuan setiap orang yang akan menempati posisi jabatan tersebut. Kriteria pekerjaan menurut Mathis dan Jackson (2002:78), menjelaskan apa-apa yang sudah dibayar oleh organisasi untuk dikerjakan oleh karyawannya. Setiap kriteria pekerjaan ini akan memberikan pengaruh satu sama lain untuk mendapatkan hasil pekerjaan yang sesuai dengan keinginan dari setiap perusahaan.

2.3 Kontrol Akses (*Access Control*)

2.3.1 Kebijakan Kontrol Akses

Persyaratan bisnis kontrol akses harus ditetapkan dan didokumentasikan. Peraturan dan hak kontrol akses untuk setiap pengguna atau kelompok pengguna harus dinyatakan dengan jelas dalam suatu pernyataan kebijakan tentang akses. Pengguna dan penyedia layanan harus diberi satu pernyataan persyaratan bisnis yang jelas yang harus dipenuhi untuk pengontrolan akses. Kebijakan harus mencakup hal berikut :

1. Persyaratan keamanan dari aplikasi bisnis perorangan.
2. Identifikasi dari seluruh informasi yang berhubungan dengan aplikasi bisnis.
3. Kebijakan diseminasi dan otorisasi informasi. Misalnya kebutuhan untuk mengetahui prinsip dan tingkat keamanan serta klarifikasi informasi.
4. Konsistensi antara kontrol akses dan kebijakan klarifikasi informasi dari sistem dan jaringan yang berbeda.

5. Peraturan yang relevan dan setiap kewajiban kontrak yang terkait dengan perlindungan akses ke data atau layanan.
6. Profil standar akses pengguna untuk kategori pekerjaan yang umum.
7. Manajemen hak akses di lingkungan yang terdistribusi dan terjaringan yang mengatur semua jenis koneksi yang tersedia.

2.3.2 Manajemen Akses User (*User Access Management*)

Tujuan manajemen akses user adalah untuk memastikan hanya pengguna yang memiliki hak mengakses Sistem Informasi dan mencegah akses illegal. Dalam menspesifikasikan aturan dalam manajemen kontrol akses, perlu untuk mempertimbangkan hal berikut :

1. Membedakan antara aturan yang selalu harus ditegakkan dan aturan yang bersifat pilihan atau kondisional.
2. Menetapkan aturan berdasarkan pemahaman "apa secara umum dilarang kecuali dinyatakan diperbolehkan", dari aturan yang lebih lemah misalnya "pada umumnya semua diperbolehkan kecuali dinyatakan dilarang".
3. Perubahan dalam penandaan informasi yang dilakukan secara otomatis oleh fasilitas pemrosesan informasi dan yang dilakukan atas pilihan pengguna.
4. Perubahan kewenangan pengguna yang dilakukan secara otomatis oleh sistem informasi dan yang dilakukan oleh administrator.
5. Aturan yang dipersyaratkan oleh administrator atau pejabat lain sebelum diberlakukan dan yang tidak diberlakukan.

Manajemen akses user meliputi hal berikut :

1. Registrasi pengguna (*User registration*)

Harus ada prosedur pendaftaran dan pengakhiran secara formal terhadap sebagai pengguna untuk memberikan akses menuju Sistem Informasi dan layanan seluruh kelompok pengguna. Akses dari pegguna layanan bagi Informasi harus dikontrol melalui proses pendaftaran pengguna secara formal, yang harus meliputi:

a. Penggunaan ID pengguna yang unik, agar pengguna dapat terhubung dan bertanggungjawab atas tindakannya. Penggunaan ID kelompok harus mendapatkan ijin apakah mereka diperbolehkan sesuai pekerjaan yang dilakukan.
b. Memeriksa apakah pengguna yang mempunyai otorisasi dari pemilik sistem, menggunakan untuk akses Sistem Informasi atau layanan. Persetujuan terpisah tentang hak akses dari manajemen juga diperlukan.
c. Memeriksa apakah tingkatan akses yang diberikan sesuai dengan tujuan bisnis dan konsisten dengan kebijakan organisasi tentang sistem keamanan, misalnya tidak menyalahgunakan pemisahan tugas.
d. Memberikan pengguna pernyataan secara tertulis tentang hak akses mereka.
e. Keharusan pengguna untuk menandatangani pernyataan yang menandakan bahwa mereka memahami tentang kondisi dari aksesnya.
f. Memastikan penyedia layanan tidak menyediakan akses hingga prosedur otorisasi dilengkapi.
g. Memelihara catatan resmi seluruh individu yang terdaftar untuk menggunakan layanan.

h. Mengakhiri hak akses pengguna yang telah pindah dari pekerjaannya atau meninggalkan organisasi.

i. Memeriksa secara periodic, dan mengakhiri, pengulangan pengguna ID dan catatan pengguna.

j. Menjamin bahwa ID pengguna yang sama tidak dikeluarkan kepada pengguna lain.

2. Manajemen hak istimewa atau khusus (*Privilege management*)

Alokasi dan penggunaan hak khusus (fitur atau fasilitas Sistem Informasi kelompok pengguna yang beragam memungkinkan adanya pengguna untuk menembus sistem atau kontrol aplikasi) harus dibatasi dan dikontrol. Penggunaan hak khusus tentang sistem yang tidak semestinya sering ditemukan sebagai faktor penyebab utama kegagalan sistem.

3. Manajemen *password* user (*User password management*)

Password adalah alat umum untuk memvalidasi identitas pengguna untuk mengakses Sistem Informasi atau layanan. Alokasi dari *password* harus dikontrol melalui proses manajemen yang formal, pendekatannya harus :

a. Pengguna harus menandatangani pernyataan untuk menjaga *Password* pribadi secara rahasia dan *password* kelompok hanya untuk anggota kelompok tersebut (ini dapat tercakup dalam batasan dan prasyarat perjanjian kerja).

b. Menjamin, bahwa pengguna dipersyaratkan untuk memelihara *Password*-nya, dimana *password* sementara yang aman disediakan dan mengharuskan mereka mengganti *password* sesegera mungkin. *Password* sementara yang

diberikan ketika pengguna lupa *password*-nya hanya boleh disediakan jika terdapat identifikasi pengguna yang jelas.

c. Mensyaratkan *password* sementara diberikan ke pengguna secara aman. Penggunaan pihak ketiga atau pesan surat elektronik secara terbuka (teks secara jelas) harus dihindarkan. Pengguna harus memberitahukan bahwa mereka telah menerima *password*. *Password* tidak boleh disimpan dalam sistem komputer bila tidak terlindungi (dengan melihat teknologi lain untuk identifikasi dan otentikasi pengguna, misalnya biometric, yaitu pengesahan sidik jari, tandatangan dan penggunaan piranti keras token, sebagai cotoh kartu chip, tersedia dan harus dipertimbangkan jika perlu).

4. Tinjauan terhadap hak akses user (*Review of user access rights*)

Untuk memelihara kontrol yang efektif terhadap akses ke data dan layanan Informasi, manajemen satu proses formal secara berkala untuk mengkaji ulang hak pengguna terhadap akses agar :

a. Hak akses pegguna dapat dikaji ulang dalam rentang waktu secara berkala (dianjurkan setiap 6 bulan) dan setelah ada perubahan.

b. Otorisasi untuk hak khusus harus dikaji ulang dalam rentang waktu yang lebih sering (dianjurkan setiap waktu 3 bulan).

c. Alokasi hak khusus diperiksa dalam rentang waktu secara berkala untuk memastikan bahwa tidak ada hak khusus diminta tanpa ijin.

5. Tanggung jawab pengguna (*user respon sibilities)*

Untuk mencegah akses pengguna tanpa ijin. Kerjasama pengguna terhadap hak akses sangat penting bagi sistem keamanan yang efektif. Pengguna harus diingatkan tentang tanggung jawab untuk menjaga control akses yang efektif,

khususnya yang terkait dengan penggunaan *password* dan keamanan peralatan pengguna.

2.3.3 Prosedur Perubahan Kontrol (*change control procedures*)

Untuk meminimalisir kerusakan system informasi, harus ada control yang ketat terhadap implementasi perubahan. Prosedur formal control perubahan harus dilaksanakan dengan tegas.

Manajer harus memastikan bahwa prosedur keamanan dan kontrol tidak dilanggar, dan pemograman pendukung hanya diberi akses pada bagian sistem yang diperlukan untuk pekerjaannya, serta adanya perjanjian dan persetujuan formal untuk setiap perubahan diminta. Perubahan piranti lunak aplikasi dapat mempengaruhi lingkungan operasional. Bila dimungkinkan, prosedur kontrol perubahan aplikasi dan operasi harus terintegrasi.

Proses ini mencakup :

1. Memelihara catatan tingkat otoritas yang disetujui;
2. Memastikan perubahan diajukan oleh pengguna yang berhak;
3. Mengkaji ulang prosedur kontrol dan integritas untuk memastikan bahwa mereka tidak menjadi rawan karena perubahan;
4. Mengidentifikasi semua piranti lunak komputer, informasi, database organisasi dan piranti keras yang membutuhkan penyesuaian;
5. Mendapat persetujuan formal untuk proposal yang lengkap sebelum pekerjaan dimulai;
6. Memastikan bahwa pengguna dengan otorisasi menerima perubahan sebelum implementasi;

7. Memastikan bahwa implementasi dilakukan untuk meminimalisir gangguan bisnis;
8. Memastikan bahwa system dokumentasi telah diperbarui saat selesainya setiap perubahan, serta pengarsipan dan pembuangan dokumentasi lama;
9. Memelihara satu versi kontrol untuk semua pemutakhiran piranti lunak;
10. Memelihara bukti pemeriksaan (audit trail) dari semua permintaan perubahan;
11. Memastikan bahwa dokumentasi operasional dan prosedur pengguna dirubah sesuai kebutuhan;
12. Memastikan bahwa implemetasi perubahan berlangsung pada waktu yang tepat dan tidak mengganggu proses bisnis terkait.

2.3.4 Kontrol Akses Informasi dan Aplikasi (*Application and information access control*)

Kontrol akses aplikasi dan informasi bertujuan untuk mencegah akses illegal yang terdapat dalam sistem – sistem aplikasi. Kontrol akses aplikasi dan informasi ini meliputi hal – hal berikut.

1. Pembatasan akses informasi (*information access restriction*)

Pengguna sistem aplikasi, termasuk staff pendukung, harus disediakan akses ke informasi dan fungsi sistem aplikasi sesuai dengan kebijakan kontrol akses yang ditentukan, berdasarkan kebutuhan aplikasi usaha individu dan tidak berubah terhadap kebijakan akses informasi organisasi.

Aplikasi yang harus dipertimbangkan dalam mendukung kebutuhan pembatasan akses:

a. Menyediakan menu untuk mengontrol akses terhadap fungsi sistem aplikasi
b. Membatasi pengetahuan pengguna atas informasi atau fungsi sistem aplikasi yang mereka tidak dapat otorisasi mengakses, dengan mengedit dokumentasi pengguna.
c. Mengontrol hak akses pengguna, misalnya membaca, menulis, menghapus dan mengeksekusi.
d. Memastikan bahwa output dari sistem aplikasi yang menanganiinformasi penting, yang relevan untuk penggunaan output, terkirim hanya kepada terminal dan lokasi yang berijin, termasuk mengkaji ulang secara berkala. Output semacam itu untuk memastikan redundansi informasi hilang.

Fasilitas sistem keamanan harus digunakan untuk melarang akses ke sistem aplikasi. *Logical Access* terhadap *software* dan informasi harus dilarang untuk pengguna, sistem aplikasi seharusnya terkait dengan hal – hal berikut :

a. Mengontrol akses pengguna terhadap informasi dan fungsi sistem aplikasi, dalam hubungannya dengan kebijakan kontrol akses bisnis yang ditetapkan;
b. Menyediakan perlindungan dari akses tidak berwenang untuk semua penggunaan dan sistem operasi *software* yang mampu menjalankan sistem dan kontrol aplikasi;
c. Tidak menyalahgunakan sistem keamanan sistem lain yang sumber informasinya terbagi;

d. Mampu untuk menyediakan akses informasi hanya untuk pemilik, individu lain yang diijinkan, dan kelompok pengguna yang ditetapkan.

2. Isolasi sistem yang sensitive (*sensitive system isolation*)

Sistem sensitif membutuhkan lingkungan komputasi khusus (terisolasi). Sebagian sistem aplikasi sangat sensitif terhadap potensi kehilangan sehingga membutuhkan penanganan khusus. Sensitifitas dapat mengindikasikan bahwa sistem aplikasi harus dijalankan pada komputer khusus, seharusnya hanya membagi *resources* dengan sistem aplikasi yang dipercaya, atau tidak mempunyai batasan. Yang harus diperhatikan dalam isolasi sistem sensitif :

a. Sensitifitas sistem aplikasi harus diidentifikasi secar eksplisit dan didokumentasikan oleh pemilik aplikasi.
b. Ketika aplikasi sensitif dijalankan pada lingkungan bersama, sistem aplikasi yang akan dibagi *resources* nya harus teridentifikasi dan disetujui oleh pemilik aplikasi sensitf.

2.3.5 Identifikasi Kelemahan (*Vulnerability Identification*)

Vulnerability adalah kekurangan atau kelemahan di dalam prosedur keamanan informasi, perencaaan, implementasi atau kontrol internal di dalam organisasi terhadap penjagaan informasi yang dimiliki, dimana kelemahan ini dapat menimbulkan atau memicu ancaman (*threat*). Tujuan utama dari tahap ini adalah organisasi memahami kelemahan yang dimiliki dalam Sistem Manajemen Informasinya. Tabel 2.2 berikut memberikan contoh yang dimiliki oleh organisasi.

Tabel 2.2 Contoh Kelemahan (*Vulnerability*)

Kelemahan (*Vulnerability)*	Sumber Ancaman	Aksi
Akses ID karyawan yang telah berhenti tidak dihapus dari sistem	Karyawan yang telah berhenti	Akses tanpa hak,illegal
Administrasi firewall membolehkan guest ID	User tanpa hak (*unautborized*)	Akses illegal
Ruang server menggunakan penyemprot air untuk menghindari kebakaran, tidak ada pelindung atau anti air yang digunakan untuk server	Api, orang yang iseng	Penyemprotan air dapat menyala secara terduga

Metode yang digunakan untuk menentukan kelemahan pada sistem di organisasi dapat berupa pencarian sumber – sumber kelemahan (*Vulnerability source*), pengujian sistem keamanan (*system security testing*) maupun membuat daftar kebutuhan keamanan (*security requirement check list*).

a. Mencari sumber – sumber kelemahan

Untuk mendapatkan kelemahan – kelemahan yang dimiliki oleh sistem manajemen keamanan informasi organisasi dapat menggunakan metode mencari sumber – sumber kelemahan. Sumber – sumber kelemahan dapat ditemukan dengan menganalisa data – data berikut :

- Data teknik sistem teknologi informasi yang digunakan

- Data kelemahan sistem software yang digunakan (dapat dilihat dalam buku panduan yang disertakan oleh vendor)

- Dokumentasi analisa resiko

b. Uji sistem keamanan

Metode kedua untuk mencari kelemahan sistem adalah dengan melakukan uji sistem keamanan. Uji sistem keamanan dapat dilakukan dengan cara metode proaktif artinya sistem keamanan di uji secara langsung untuk menemukan kelemahan. Cara lain adalah dengan melakukan testing atau uji kepada seluruh pengguna sistem, apakah pengguna dapat mengakses sistem melebihi dari yang dibolehkan dari hasil tersebut akan ditemukan kelemahan sistem.

c. Membuat daftar kebutuhan keamanan

Metode ketiga untuk mencari kelemahan sistem adalah dengan membuat daftar kebutuhan keamanan (bias berupa *check list*). Berdasarkan daftar kebutuhan keamanan yang telah dibuat kemudian dibandingkan dengan sistem keamanan yang telah ada, perbedaan anatara kebutuhan keamanan dan sistem keamanan yang ada dapat ditentukan kelemahan sistem keamanan yang ada. Daftar kebutuhan keamanan dapat dikelompokkan kedalam beberapa area, antara lain : area manajemen, operasional dan teknik. Contoh mengenai daftar tersebut dapat dilihat pada tabel 2.3.

Tabel 2.3 Contoh daftar kebutuhan keamanan

Area Keamanan	Kebutuhan Keamanan
Manajemen	Kontrol akses, manajemen asset, penilaian resiko, keamanan organisasi dan lainnya
Operasional	Manajemen operasional, komunikasi, manajemen kelangsungan bisnis dan lainnya
Teknik	Keamanan fisik, pembangunan sistem informasi, keamanan SDM dan lainnya

2.4 Perancangan Sistem

2.4.1 Data Flow Diagram (DFD)

Menurut Jogiyanto (1989: 699) DFD atau yang sering disebut *Bubble Chart* atau diagram, model proses, diagram alur kerja atau model fungsi adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alir data baik secara manual maupun komputerisasi. *DFD* merupakan alat pembuat model yang sering digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluaran.

Untuk memudahkan proses pembacaan *DFD*, maka penggambaran *DFD* disusun berdasarkan tingkatan atau *level* dari atas ke bawah, yaitu:

1. *Context Diagram*

Merupakan diagram paling atas yang terdiri dari suatu proses dan menggambarkan ruang lingkup proses. Hal yang digambarkan dalam diagram konteks adalah hubungan terminator dengan sistem dan juga sistem dalam suatu

proses. Sedangkan hal yang tidak digambarkan dalam *Context Diagram* adalah hubungan antar *terminator* dan *data source*.

2. *Diagram Zero* (Level 0)

Merupakan diagram yang berada diantara diagram konteks dan diagram detail serta menggambarkan proses utama dari *DFD*. Hal yang digambarkan dalam *Diagram Zero* adalah proses utama dari sistem serta hubungan *entity*, proses, alur data dan *data source*.

3. *Diagram Detail* (Primitif)

Merupakan penguraian dalam proses yang ada dalam *Diagram Zero*. Diagram yang paling rendah dan tidak dapat diuraikan lagi.

2.4.2 Apache

Untuk menjalankan aplikasi web membutuhkan *web-server*. Apache adalah *web-server* yang mendukung bahasa PHP sehingga dapat dipakai untuk implementasi aplikasi berbasis PHP. *Web-server* akan menerjemahkan bahasa PHP yang dipakai pada aplikasi *score online* untuk ditampilkan secara visual pada *browser* (Apache, 2005).

2.4.3 *Hypertext Preprocessor (PHP)*

PHP adalah *server side scripting envirotment* yang dapat digunakan untuk membuat dan menjalankan aplikasi-aplikasi di *web-server* agar lebih interaktif dan *programmable.* dengan PHP aplikasi-aplikasi yang ada di *web-server* benar-benar dijalankan di *web-server* tanpa mengharuskan adanya tambahan atau syarat tertentu untuk sisi client (*web browser*). PHP biasanya dijadikan sebagai *module* dalam suatu web agar bisa mengeksekusi file-file PHP

yang tersedia di *web-server*. PHP dapat berjalan di hampir seluruh platform, *open source* dan berlicensi GNU *Public License* (GPL). (Welling, 2001).

PHP pada mulanya di tulis sebagai sebuah kumpulan dari CGI dengan menggunakan bahasa pemrograman C oleh *programmer* bernama Rasmus Lerdorf. *Programer* asal Greenland ini membuat PHP pada tahun 1994 untuk mengantikan sebagian kecil kumpulan *script* dengan Perl yang digunakan untuk *maintenance* halaman web miliknya. Lerdorf mengawali menciptakan PHP untuk menanpilkan *resume* miliknya dan mengumpulkan beberapa data, seperti berapa banyak lalu lintas data yang diterima dalam halaman web miliknya. (Welling, 2001).

Setelah mengalami perkembangan oleh suatu kelompok open source (termasuk Rasmus) maka mulai versi 3 php menanpakan keunggulan sebagai salah satu bahasa server yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini . maka jadilah PHP disebut sebagai *Hypertext Preprocessor*.

Aplikasi yang dibangun dengan PHP memiliki kelebihan tersendiri. Beberapa kelebihan yang dimiliki PHP antara lain :

1. Software ini disebarkan dan dilicensikan sebagai perangkat lunak yang *open source*, maksudnya pendistribusian oaker programnya disertakan juga kode programnya dan biasanya secara gratis.
2. Dengan menggunakan PHP *script* maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunkan aplikasi yang dibuat dengan menggunakan *script* PHP.

3. Penulisan *script* PHP dapat menyatu dengan dokumen HTML, sehingga memudahkan pembuatannya. Untuk membedakan dengan sintaks HTML dan PHP maka dibuatlah kesepakatan tag yang digunakan oleh PHP.
4. Kemampuan PHP yang paling diandalkan dan signifikan adalah dukungan kepada banyak database. Membuat halaman web yang menggunakan data dari database dapat sangat mudah untuk dilakukan. Database yang didukung oleh PHP antara lain: adabas D, dBase, Empress, IBM DB2, Infomix, Ingers, Interbase, Frontbase, File Pro(read only), SQL Server, MySQL, Oracle, ODBC, PostgresSQL, Solid, Sysbase, Velocis, dan unix DBM.

2.4.4 Structured Query Language (SQL)

Pada umumnya semua *engine* database mengadopsi bahasa standar SQL yaitu bahasa yang digunakan untuk memanipulasi dan memperoleh data dari sebuah database relasional. SQL membuat seorang developer atau administrator database melakukan hal-hal berikut :

a. Mengubah struktur sebuah database.
b. Mengubah pengaturan keamanan sistem.
c. Memberikan hak akses kepada pengguna untuk mengakses database atau tabel
d. Memperoleh informasi dari database.

Perintah-perintah SQL secara umum dapat dikelompokkan menjadi lima macam, yaitu :

1. *Data Definition Language* (DDL)

Adalah perintah SQL yang digunakna untuk menjelaskan objek dari database. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka database. Prinsipnya adalah:

a. *Create*: untuk membuat/menciptakan obyek database
b. *Alter*: untuk memodifikasi/mengubah obyek database
c. *Drop*: untuk menghapus obyek database
d. Obyek *database* yang dimaksud terdiri dari database, tabel, *index*, dan *view*

2. *Data Manipulating Language (DML)*

Adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi database. SQL menyediakan 4 perintah DML:

a. *Select:* digunakan untuk mengambil data dari database
b. *Delete*: digunakan untuk menghapus data pada database
c. *Insert:* menambahkan data ke database
d. *Update:* memodifikasi data ke database

3. *Security*

Adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Antara lain terdiri atas:

a. *Grant:* memberi akses kepada user tertentu untuk akses ke database
b. *Revoke:* mencabut hak akses dari user

4. *Integrity*

Adalah perintah-perintah yang digunakan untuk menjaga kesatuan data. Contoh: *recover table*: untuk memperbaiki tabel pada database

5. *Auxilliary*

Adalah perintah-perintah pelengkap atau tambahan seperti: *unload* dan *rename*.

2.4.5 MySQL

Menurut Didik Dwi Prasetyo (2004 :18) MySQL merupakan salah satu database server yang berkembang di lingkungan open source dan didistribusikan secara free (gratis) dibawah lisensi GPL.

MySQL merupakan RDBMS (*Relational Database Management System*) server. RDBMS adalah program yang memungkinkan pengguna database untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya.

1. Keunggulan MySQL

Beberapa keunggulan dari MySQL yaitu :

a. Cepat, handal dan Mudah dalam penggunaannya

   MySQL lebih cepat tiga sampai empat kali dari pada database server komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.

b. Didukung oleh berbagai bahasa

Database server MySQL dapat memberikan pesan error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.

c. Mampu membuat tabel berukuran sangat besar

Ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.

d. Lebih Murah

MySQL bersifat open source dan didistribusikan dengan gratis tanpa biaya untuk UNIX platform, OS/2 dan Windows platform.

e. Melekatnya integrasi PHP dengan MySQL

Keterikatan antara PHP dengan MySQL yang sama-sama software open source sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan database server lainnya. Modul MySQL di PHP telah dibuat built-in sehingga tidak memerlukan konfigurasi tambahan pada file konfigurasi php.ini.

2. Keistimewaan MySQL

1. **Portabilitas**. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. **Perangkat lunak sumber terbuka (open source)**. MySQL didistribusikan sebagai open source sehingga dapat digunakan secara gratis.
3. **Multi-user**. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. **Performance tuning**, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. **Ragam tipe data**. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. **Perintah dan Fungsi**. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. **Keamanan**. MySQL memiliki beberapa lapisan keamanan seperti password yang terenkripsi.
8. **Skalabilitas dan Pembatasan**. MySQL mampu menangani basis data dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. **Konektivitas**. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau named pipes (NT).
10. **Lokalisasi**. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar Muka**. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. **Klien dan Peralatan**. MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. **Struktur tabel**. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.5 Testing dan Implementasi

Menurut Standar ANSI/IEEE 1059, Testing adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects/error/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Menurut Romeo (2003:3), Testing *software* adalah proses mengoperasikan software dalam suatu kondisi yang dikendalikan untuk:

3. Verifikasi.

Melakukan pengecekan atau pengetesan entitas – entitas, apakah telah berlaku sebagaimana telah ditetapkan

4. Mendeteksi error.

Untuk menentukan apakah sesuatu hal terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut terjadi dimana seharusnya mereka ada.

5. Validasi.

Apakah spesifikasi yang ditetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya?

Menurut Romeo(2003:33), Dengan adanya perkembangan teknologi internet, berkembanglah kebutuhan aplikasi berbasis web, baik untuk keperluan internet organisasi. Terdapat beberapa hal yang berkaitan dengan kualitas aplikasi berbasis web, antara lain :

1. Komplesitas Aplikasi.

Web merupakan aplikasi yang paling berkembang saat ini, baik dari segi kompleksitas, manajemen query pada *database* yang sangat besar, atau metode *searching* yang ada. Web site lebih kompleks dari yang terlihat, karena web site menggunakan teknologi GUI, *Network Connectivity* dan *Database Acces.* Beberapa pengamat menyatakan bahwa teknologi *client/server* akan digantikan oleh *internet,* tapi kenyataan yang berkembang adalah teknologi gabungan dari keduannya. Inilah alasan mengapa *client/server testing* yang dibahas sebelumnya juga berkaitan dengan subbab ini.

2. Keterbatasan Alat Bantu.

Hal yang tidak dapat dibantah adalah alat bantu pengembangan aplikasi berbasis web saat ini masih memiliki keterbatasan yang sangat menggangu. Aplikasi web dibangun dengan alat bantu standar yang menghasilkan pages statis, sehingga pengguna tidak dapat dengan mudah men-*download* data ke *desktop analysis tool* seperti *excel spreadsheet.*

Produk web merupakan aplikasi yang paling cepat mengalami penambahan versi oleh karena itu manajemen tes yang diperlukan juga harus handal, karena hal ini berhubungan dengan kualitas dari aplikasi itu sendiri.

3. Kompatibilitas

Web pages akan terlihat berbeda jika dilihat dari *Web Browser* yang berbeda, karena perbedaan implementasi dari HTML *standart*.

Web pages dapat diakses dari beberapa *platform*yang berbeda, seperti Win NT, Win 95, OS/2, Mac dan lain-lain. Ini artinya testing perlu dilakukan pada berbagai *platform* dan konfigurasi yang berbeda.

4. Performansi

Hal yang paling sulit untuk dites adalah pengukuran kecepatan akses. *Response Time* dari Web, karena hal itu bukan hal yang mudah untuk dipecahkan dengan biaya yang murah.

Banyak faktor yang menjadi penyebab seperti *loads* yang tidak dapat diprediksi, Web yang menjadi favorit bisa menerima ribuan penggunjung per-hari bandingkan dengan Web biasa yang pengunjungnya hanya ratusan.

5. Kegunaan

Beberapa pengguna mungkin punya harapan sendiri-sendiri tentang bagaimana *web site* yang menarik. Seperti contohnya *Web Pages* harus dapat

dengan mudah untuk disimpan. Oleh karena itu *Web pages* harus terlihat atraktif agar menarik perhatian dari pengguna. Ada beberapa pengguna yang sangat sensitif dan terganggu jika keluar ata masuk dari suatu *Web pages* tanpa suatu *permission* atau *awaraness.*

6. Keamanan

Sistem keamanan merupakan hal yang sangat penting dalam aplikasi berbasis web, karena aplikasi ini dibangun untuk dapat diakses oleh pengguna atau aplikasi yang baik itu dalam suatu intranet ataupun extranet dengan sama baiknya. Hak akses eksternal memang dibatasi tapi tidak menutup kemungkinan terjadinya *hacking* terhadap aplikasi.

7. Organisasional

Telah dijelaskan diatas bahwa teknologi ini merupakan inovasi yang sangat fenomenal. Oleh karena itu mungkin dalam perkembangannya yang kurang diperhatikan adalah kendali kualitas dan standar testing yang baik. Yang terjadi pada pengembangan intranet yan mengambil alih semua proses pembangunan dari suatu aplikasi Web mulai dari desain hingga proses testing.

Dalam beberapa organisasi intranet mambuat kekacuan karena kurangnya koordinasi. Setiap orang mempunyai web internal pribadi. Setiap orang punya ide sendiri-sendiri bagaimana membuat harus membuat web-nya, apa isinya, dan bagaimana harus berjalan. Sehingga terjadi kekacauan pada kepemilikan dan hak akses informasi juga pertanyaan siapa yang bertanggung jawab atas kualitas dari informasi dan maintenance dari aplikasi itu sendiri.

Tipe-tipe testing pada aplikasi berbasis web, antara lain :

1. *Content* dan *Funcionality testing.* Testing terhadap isi dan fitur seperti yang terdapat pada *Web site* umumnya, pastikan sudah lengkap dan berjalan sesuai dengan yang diinginkan.
2. *Feature interaction testing.* Banyak pengguna yang secara simultan mengakses satu site yang sama dan tidak boleh terjadi interfrensi antara mereka.
3. *Usability testing.* Melakukan testing apakah Web site sudah *user friendly*.
4. *Database testing.* Memastikan *database* dapat diakses dari Web site yang mempunyai kendali integritas dan kecukupan data.
5. *Security* dan *control testing.* Memastikan *site* ini aman, termasuk *account setup, billing, dan dari unauthorized acces.*
6. *Connectivity testing.* Pastikan *Web site* dapat melakukan *connection* atau *disconnection.*
7. *Interoperability testing.* Pastikan semua *Web Browser* dari semua versi da jenis komputer yang berbeda dapat berjalan dengan baik pada aplikasi ini.
8. *Cross platform* dan *configuration testing.* Pastikan perilaku dari sistem kompatibel dalam *platform* dan konfigurasi yang berbeda.
9. *Performance* dan *Stress testing.* Ukur kemampuan, *response time* dan semua proses yang terjadi dalam keaadaan *workloads* di atas rata-rata, rata-rata atau dibawah rata-rata.
10. *Internazionalization testing.* Pastikan *site* tidak membingungkan atau menyerang pengguna.

11. *Beta testing.* Undang beberapa pengguna terpilih untuk melakukan eksperimen pada *site* anda dan mintalah *feedback* pada mereka sebelum *web site* itu diluncurkan.
12. *Standart Compilance testing.* Pastikan *Web site* itu kompetibel dengan *internet standart*, apakah terlihat sama meskipun menggunakan *browser* atau *search engines* yang berbeda.

2.6 Standarisasi Perusahaan

Pada Unit Sistem Informasi (ISDC), SISFO berawal dari proyek Mekanisasi Administrasi Telekomunikasi (MEKADATEL) pada tahun 1977 yang bertujuan untuk melakukan mekanisasi terhadap system billing.Pada tahun berikutnya, Bagian Pengelolaan Data (OLAHTA) didirikan dibawah tanggung jawab Direktorat Keuangan c.q. bagian Keuangan Wilayah Telekomunikasi.

Karena perkembangan bisnis telekomunikasi, dirasakan perlu untuk mengembangkan Unit Kerja OLAHTA menjadi SUBDITDATA (Sub Direktorat Pengolahan Data) dibawah kendali BAGOPTEK (Bagian Operasi Teknik). Pada tahun 1990-an, saat perubahan era komputerisasi dari mini computer menjadi miniframe, dibentuk PUSTEKSI (Pusat Teknologi Informasi dan Sistem Informasi). Hingga 1992 PUSTEKSI berada dibawah kendali DIREKTORAT OPTEK, selanjutnya PUSTEKSI berada dibawah DITPRANTEK. Perkembangan terus berlanjut, dimana sistem informasi menjadi salah satu layanan dukungan dari PT. TELKOM. Berdasarkan keputusan Direksi tanggal 22 Februari 1995, dibentuk Divisi sistem informasi (SISFO) sebagai salah satu divisi pendukung dilingkungan PT. TELKOM.

Dalam era globalisasi, sistem informasi memainkan peranan yang sangat penting pada setiap proses bisnis yang dilakukan oleh pelaku bisnis di Indonesia maupun diluar Indonesia. Sejalan dengan perkembangan teknologi yang semakin pesat dan semakin beragam keinginan customer, PT. TELKOM sebagi penyedia jasa telekomunikasi dituntut untuk memberikan layanan sesuai kebutuhan pengguna jasa, sebagai bekal dalam menghadapi persaingan dan tantangan dimasa depan.

Menyadari pentingnya peranan informasi dalam menghadapi persaingan ini, PT. TELKOM membentuk Divisi Sistem Informasi (SISFO) sebagai penyedia sistem informasi bagi perusahaan dan penanggung jawab pengelolaan infrastruktur sistem informasi PT. TELKOM diseluruh Indonesia, untuk menunjang operasi dan strategi PT. TELKOM. Dengan pengalaman lebih dari 30 tahun mengelola sistem informasi PT. TELKOM dan penguasaan akan proses bisnis industri telekomunikasi, SISFO mempunyai modal yang cukup besar untuk menjadi penyedia jasa sistem informasi yang handal.

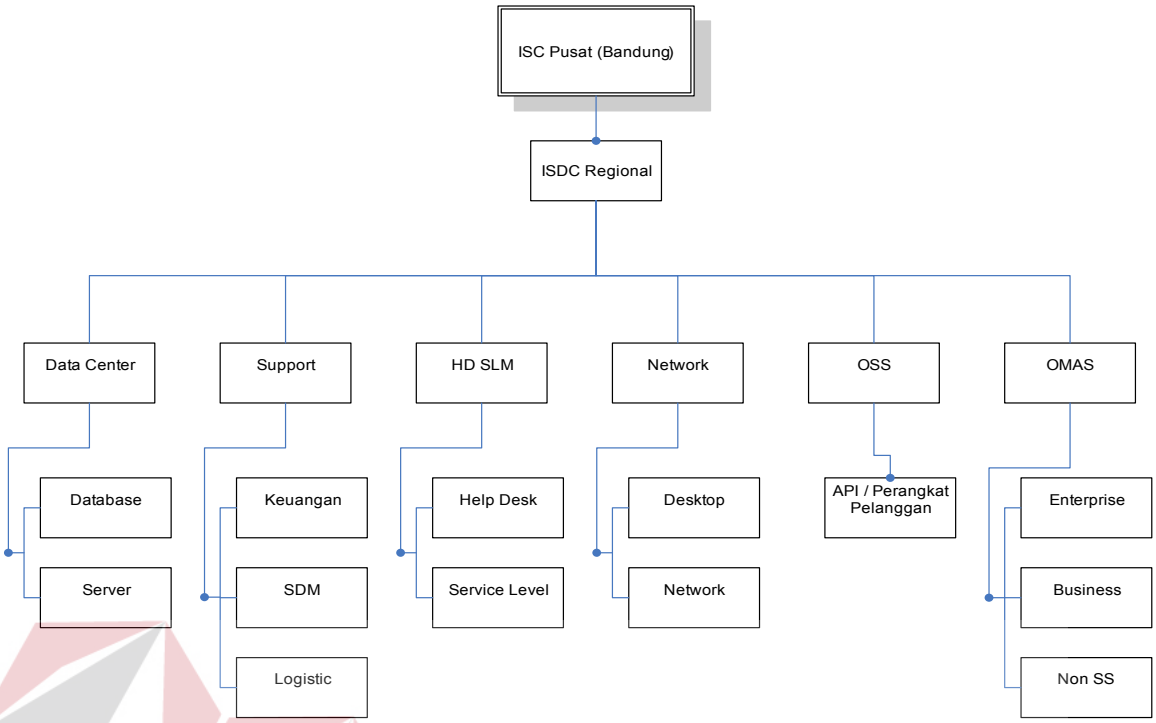
Maka setiap aplikasi yang dimiliki oleh PT. Telkom Indonesia dibutuhkan keamanan yang baik sehingga diperlukan sistem informasi yang melindungi hak akses aplikasi – aplikasi yang dimana penggunaan dan pengelola disusun dengan baik seperti :

- Admin ums (*user management system*) dimana pada bagian ini sangat berperan penting untuk mengelola hak akses *user* pada aplikasi yang akan digunakan, admin akan mengatur jabatan – jabatan *user* yang dapat menggunakan aplikasi yang ada di PT. Telkom sehingga aplikasi tidak bisa

disalah gunakan oleh *user* yang tidak mempunyai ijin memasuki aplikasi tersebut.

- Admin *host/device* (aplikasi) dimana pada bagian ini memiliki tugas untuk memberikan persetujuan dalam proses permintaan hak akses *user* , pada proses ini admin *host/device* (aplikasi) jika persyaratan yang diajukan oleh *user* terpenuhi maka persetujuan akan diberikan dan admin akan memberikan *username* dan *password* aplikasi yang akan digunakan dan apabila penunjang belum terpenuhi maka persetujuan tidak akan diterima (ditolak) dengan memberikan alasannya sehingga *user* dapat mengajukan kembali permintaan hak akses *user* tersebut.

Host/Device merupakan bagian dari divisi – divisi yang terdapat di PT. Telkom Indonesia. Setiap *Host/Device* memiliki aplikasi – aplikasi yang digunakan didalam setiap *Host/Device*. Penjelasan *Host/Device* dapat digambarkan pada schema berikut :



Gambar 2.2 Struktur PT. Telekom Indonesia divisi ISC