

BAB III

LANDASAN TEORI

3.1 Konsep Dasar Sistem Informasi

3.1.1 Sistem

Menurut Herlambang (2005:116), definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

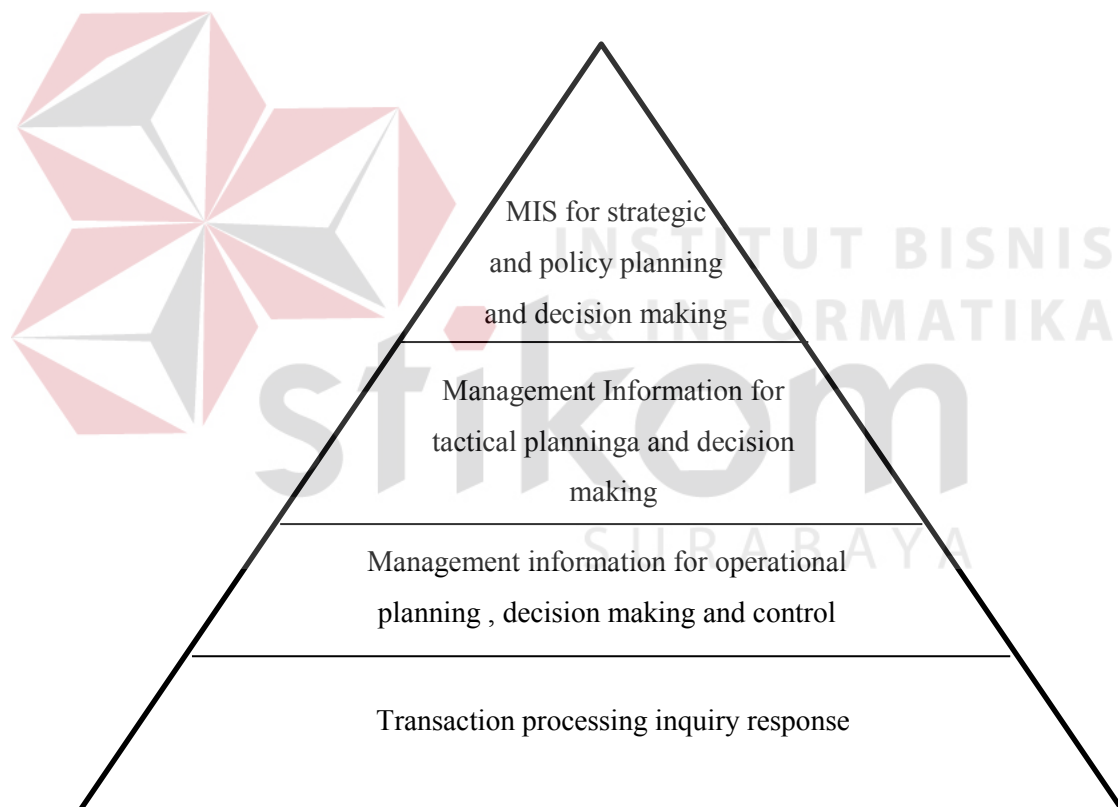
Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya.

3.1.2 Sistem Informasi

Menurut Herlambang (2005:121), data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, Informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya.

3.1.3 Sistem Informasi Manajemen

Menurut Bagus Kurniawan (2002:1) menyatakan bahwa sistem informasi manajemen (SIM) merupakan sebuah sistem terstruktur yang digunakan untuk mengelola data secara terkomputerisasi. Informasi yang dihasilkan dari suatu sistem informasi manajemen menjelaskan mengenai berbagai hal yang telah terjadi di masa lalu, yang sedang terjadi sekarang dan yang mungkin terjadi di masa depan. Didalam sistem informasi manajemen terdapat level atau tingkatan manajemen yang dapat dilihat pada gambar dibawah ini.



Gambar 3.1 Management Information System

3.1.4 Sistem Informasi Eksekutif

Menurut Irfan Subakti (2002:80), sistem informasi eksekutif adalah sistem berbasis komputer yang melayani informasi yang dibutuhkan oleh para eksekutif puncak. Sistem informasi eksekutif mendukung penyajian laporan

berupa grafik dan kemampuan drill-down, yaitu kemampuan untuk menguraikan data hingga detail. Karakteristik sistem informasi eksekutif :

Kualitas Informasi

1. Fleksibel.
2. Menghasilkan informasi yang benar.
3. Menghasilkan informasi yang sedia setiap saat.
4. Menghasilkan informasi yang relevan.
5. Menghasilkan informasi yang lengkap.
6. Menghasilkan informasi yang valid.

Antar muka user

1. Memiliki antarmuka user grafis yang canggih.
2. Memiliki antarmuka user yang *user-friendly*.
3. Akses informasi yang aman dan terjamin kerahasiaannya.
4. Waktu tanggapan atas respon cepat.
5. Dapat diakses dari berbagai tempat.
6. Memiliki prosedur akses yang dapat diandalkan.
7. Meminimalkan penggunaan keyboard.
8. Mendapatkan kembali informasi yang diinginkan secara cepat.
9. Didesain sesuai kebutuhan manajemen dari para eksekutif.
10. Memiliki *self-help* menu.

Kemampuan teknis

1. Akses ke kumpulan informasi (global).
2. Akses ke e-mail.
3. Penggunaan yang ekstensif dari data eksternal.

4. Interpretasi tertulis.
5. Indicator-indikator masalah yang dapat disorot (highlight).
6. Hypertext dan hypermedia.
7. Analisis ad hoc.
8. Presentasi dan analisis multidimensional.
9. Penyajian informasi dalam bentuk hirarki.
10. Jalinan terpadu grafis dan teks dalam layar yang sama.
11. Penyediaan manajemen berdasarkan laporan pengecualian.
12. Menyajikan tren / kecenderungan, rasio dan penyimpangan.
13. Penyediaan akses ke data historis dan data terkini.
14. Pengorganisasian di seputar CSF.
15. Penyediaan informasi pada berbagai level detail.
16. Menyaring, mengompres / memadatkan, melacak data kritis.
17. Mendukung penjelasan terhadap permasalahan yang bersifat terbuka.

3.1.5 Analisa dan Perancangan Sistem

Analisis sistem dilakukan dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

Menurut Kendall (2003:7), Analisa dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

3.2 Konsep Dasar Basis Data

3.2.1 Sistem Basis Data

Menurut Marlinda (2004:1), sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu Perangkat Keras (*Hardware*), Sistem Operasi (*Operating System*), Basis Data (*Database*), Sistem (Aplikasi atau Perangkat Lunak) Pengelola Basis Data (DBMS), Pemakai (*User*), dan Aplikasi (Perangkat Lunak) lain (bersifat opsional).

Tabel 3.1 Keuntungan dan Kerugian Sistem Basis Data

Keuntungan Sistem Basis Data	Kerugian Sistem Basis Data
Mengurangi Kerangkapan data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga update dilakukan berulang-ulang	Diperlukan tempat penyimpanan yang besar
Mencegah ketidakkonsistenan	Diperlukan tenaga yang terampil dalam

	mengolah data
Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang	Perangkat lunaknya mahal
Integritas dapat dipertahankan	Kerusakan sistem basis data dapat mempengaruhi department yang terkait

3.2.2 Database

Menurut Yuswanto (2005:2), *database* merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara database Relasional dan Non Relasional. Pada database Non Relasional, sebuah database hanya merupakan sebuah file.

Menurut Marlinda (2004:1), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai), masalah keamanan (*security*), masalah integrasi (kesatuan), dan masalah data *independence* (kebebasan data).

3.2.3 Database Management System

Menurut Marlinda (2004:6), *Database Management System* (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis Data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

3.3 Interaksi Manusia dan Komputer

Menurut Rizky (2006:4), Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-faktor utama dalam lingkungan interaksinya.

Deskripsi lain dari IMK adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif. Dikatakan juga bahwa sebuah desain antar muka yang ideal adalah yang mampu memberikan kepuasan terhadap manusia sebagai pengguna dengan faktor kapabilitas serta keterbatasan yang terdapat dalam sistem.

Pada implementasinya, IMK dipengaruhi berbagai macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan, antar muka, kendala dan produktifitas.

3.4 Grafik (*Highcharts*)

Highchart adalah *charting library* yang ditulis dengan bahasa javascript murni. Highchart menawarkan intuitif, grafik interaktif ke situs web atau aplikasi

web. Saat ini highchart mendukung grafik tipe *line*, *spline*, *area*, *areaspline*, *column*, *bar*, *pie*, dan *scatter*. Highchart sudah dapat berjalan di semua browser modern termasuk pada iPhone/iPad. Browser standar menggunakan SVG untuk proses render grafik, sedangkan internet explorer menggunakan VML untuk penggambaran grafik.

3.5 PHP

Menurut Rasmus Lerdorf, Kevin Tatroe dan Peter MacIntyre (2006:5), PHP (akronim dari PHP Hypertext Preprocessor) merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server. Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server. Ketika menggunakan PHP sebagai *server-side embedded script language* maka server akan melakukan hal-hal sebagai berikut:

1. membaca permintaan dari *client/browser*
2. mencari halaman/page di server
3. melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page
4. mengirim kembali halaman tersebut kepada client melalui internet atau intranet.

3.5.1 Sejarah PHP

Menurut Rasmus Lerdorf, Kevin Tatroe dan Peter MacIntyre (2006:2), PHP diciptakan oleh Rasmus Lerdorf, seorang programmer Unix dan Perl. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung yang melihat resume Rasmus Lerdorf. Skrip ini selanjutnya dikemas menjadi tool yang disebut “*Personal Home Page*”. Paket inilah yang menjadi cikal bakal PHP.

Pada bulan September-Oktober 1995, Rasmus menciptakan PHP/FI Versi 2.0. Pada versi ini *programmer* dapat menempelkan kode terstruktur di dalam tag HTML dan kode ini juga bisa berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks sambil jalan. Pada tanggal 6 Juni 1998, PHP versi 3.0 berhasil dirilis. PHP 3 ini mempunyai kinerja yang lebih tinggi, berkemampuan *object-oriented*, *syntax highlighting*, *array* multidimensi dan dapat diperluas melalui mekanisme *extension*. Dukungan dan fiturnya pun jauh bertambah. PHP merupakan salah satu bahasa yang populer.

3.5.2 Fitur-Fitur PHP

Menurut Rasmus Lerdorf (2006:4), Fitur-fitur yang dimiliki PHP antara lain:

1. *Acces Logging*
2. *Access Restriction*
3. *mSQL Support*
4. *postgre support*
5. *DBM support*
6. *RFC-1867 file upload support*
7. *Variable, arrays, associative arrays*
8. *User defined function with static variable and recursion*

9. *Conditional while and loops*
10. *Extended regular expressions*
11. *Raw HTTP header control*
12. *Dynamic GIF image creation*

3.5.3 Cara Kerja PHP

Model kerja HTML diawali dengan permintaan suatu halaman web oleh browser. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, browser mendapatkan alamat dari web server, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh web server. Informasi yang disampaikan ke web server antara lain adalah nama browser, versinya dan sistem operasinya.

Selanjutnya, *web server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkannya ke layar pemakai. Jika yang diminta adalah sebuah halaman PHP, maka prinsipnya serupa dengan kode HTML hanya saja, ketika berkas PHP yang diminta didapatkan oleh web server, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke web server, selanjutnya *web server* menyampaikan ke klien.

3.6 Unified Modelling Language (UML)

3.6.1 Sejarah Unified Modelling Language (UML)

Menurut Martin Fowler (2004:1), *Unified Modelling Language (UML)* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk

menentukan, visualisasi, merancang dan mendokumentasikan artifact dari sistem software, untuk memodelkan bisnis dan sistem non software lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

3.6.2 Artifak UML

UML menyediakan beberapa notasi dan artifact standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Menurut Martin Fowler (2004:3), Artifak didalam UML didefinisikan

sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman. Yang harus diperhatikan untuk menjaga konsistensi antar artefak selama proses analisis dan desain adalah bahwa setiap perubahan yang terjadi pada satu artefak harus juga dilakukan pada artefak sebelumnya. Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut :

1. *use case diagram*
2. *class diagram*
3. *behavior diagram*
4. *statechart diagram*
5. *activity diagram*
6. *interaction diagram*
 - a. *sequence diagram*
 - b. *collaboration diagram*
13. *implementation diagram*
14. *component diagram*
15. *deployment diagram*

Diagram-diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa. Dibuatnya berbagai jenis diagram diatas karena setiap sistem yang kompleks selalu paling baik jika melalui pendekatan himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencukupi untuk melihat sistem yang besar dan kompleks. Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-

beda dalam proses rekayasa. Diagram-diagram tersebut dibuat agar model yang dibuat semakin mendekati realitas.

