

BAB III

LANDASAN TEORI

1.1 Aplikasi

Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data (Anisyah, 2000:30). Aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel (Dhanta, 2009:32). Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

1.2 Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan (Jogiyanto HM., 1999:692). Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang (Gordon B. Davis, 1991:28).

1.3 *Service Level Agreement*(SLG)

Service Level Agreement (SLG) singkatan dari atau jika diterjemahkan adalah, Perjanjian Tingkat Layanan. Pengertian *service level agreement* adalah bagian dari perjanjian layanan secara keseluruhan antara 2 (dua) entitas untuk peningkatan kinerja atau waktu pengiriman harus di perbaiki selama masa kontrak. Dua entitas tersebut biasanya dikenal sebagai penyedia layanan dan klien, dan dapat melibatkan perjanjian secara hukum karena melibatkan uang, atau kontrak lebih informal antara unit-unit bisnis internal.

SLG merupakan kesepakatan antara penyedia jasa dan pengguna jasa mengenai tingkat layanan (mutu). SLG merupakan komponen kunci dari keseluruhan strategi *Service Level Management* (SLM) suatu organisasi TI. Suatu SLG yang bagus sekaligus dapat berfungsi sebagai sarana komunikasi yang baik pula bagi Anda (perusahaan) dengan para pelanggan Anda dalam menangani harapan masing-masing pihak.

Dilihat dari definisinya, *service level agreement* lebih merupakan suatu kesepakatan, bukan suatu kontrak. Baik Anda bekerja pada bagian TI yang menjalankan pelayanan bagi pelanggan internal seperti untuk bagian personalia, maupun sebagai konsultan yang menawarkan jasa TI pada para pelanggan, SLG Anda haruslah merupakan keterangan yang jelas, baik bagi pihak yang menerima jasa maupun yang menyediakannya.

(<http://www.ebizasia.com/0218-2004/briefcase,0218,01.htm>, 2013)

1.4 Jaringan Telepon

Telepon berasal dari bahasa Yunani yaitu terdiri dari buah kata yaitu tele yang berarti jauh, dan phone berarti bunyi. Telepon merupakan pesawat atau alat penerima getaran bunyi dari jarak jauh. Pengertian teleon dalam kehidupan sehari-hari mengandung pengertian mikrofon dan telepon. Mikrofon berfungsi sebagai pengirim suara (yang menghadap mulut), sedangkan telepon berfungsi yang menerima suara (yang menempel pada telinga). (<http://id.shvoong.com/exact-sciences/physics/2112632-pengertian-telepon/#ixzz2iBUllkMk>, Oktober 2013).

1.5 Entity Relational Diagram (ERD)

Entity Relationship Diagram adalah suatu bentuk perencanaan *database* secara konsep fisik yang nantinya dipakai sebagai kerangka kerja dan pedoman dari struktur penyimpanan data. ERD digunakan untuk menggambarkan model hubungan data dalam sistem, dimana didalamnya terdapat hubungan entitas beserta atribut relasinya dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan data. ERD memiliki beberapa jenis model yaitu:

Tabel 3.1 Jenis-jenis ERD (CDM dan PDM)

No.	Jenis ERD	Keterangan
1.	<i>Conceptual Data Model (CDM)</i>	Merupakan model <i>universal</i> dan dapat menggambarkan semua struktur logic <i>database</i> (DBMS), dan tidak bergantung dari <i>software</i> atau

No.	Jenis ERD	Keterangan
		pertimbangan struktur <i>data storage</i> . Sebuah CDM dapat diubah langsung menjadi PDM.
2.	<i>Physical Data Model (PDM)</i>	Merupakan model ERD yang mengacu pada pemilihan <i>software</i> DBMS yang spesifik. Hal ini seringkali berbeda secara signifikan dikarenakan oleh struktur tipe <i>database</i> yang bervariasi, dari model <i>schema</i> , tipe data penyimpanan dsb.

ERD memiliki 4 jenis obyek, yaitu:

1. *Entity*

Sesuatu yang ada dan terdefiniskan bisa berupa nyata maupun abstrak yang dapat dibedakan satu dengan yang lainnya dan adanya hubungan saling ketergantungan. Ada 2 macam tipe *entity*, yaitu:

a. *Strong Entity*

Strong Entity merupakan tipe *entity* yang mempunyai *key attribute* untuk setiap individu yang ada di dalamnya.

b. *Weak Entity*

Weak Entity merupakan *entity* yang tidak memiliki *key attribute*, oleh karena itu *weak entity* harus dihubungkan dengan *strong entity* untuk menggunakan *attribute* kunci secara bersama-sama. (Abdul Kadir, 2008)

2. *Attribute*

Setiap *entity* memiliki beberapa *attribute*, yang merupakan ciri atau karakteristik dari *entity* tersebut. *Attribute* seting disebut juga data elemen atau *data field*. (Abdul Kadir, 2008)

3. *Key*

Beberapa elemen data memiliki sifat, dengan mengetahui nilai yang telah diberikan oleh sebagian elemen data dari *entity* tertentu, dapat diidentifikasi nilai-nilai yang terkandung dalam elemen-elemen data lain ada *entity* yang sama. Elemen penentu tersebut adalah sebagai elemen data kunci (*key*). (Abdul Kadir, 2008).

4. *Relationship*

Relationship menggambarkan hubungan yang terjadi antar *entity* yang mewujudkan pemetaan antar *entity*. Bentuk *relationship*, yaitu:

a. *One to One Relationship*

Hubungan satu *entity* dengan satu *entity* yang lain.

b. *One to Many Relationship*

Hubungan antar *entity* satu dengan *entity* yang lainnya adalah satu berbanding banyak.

Untuk normalisasi yang dipakai dalam pembahasan kali ini adalah bentuk normalisasi bentuk pertama. Yang disebut dengan bentuk normal pertama (1NF) adalah suatu keadaan yang membuat setiap perpotongan baris dan kolom dalam relasi hanya berisi satu nilai. Untuk membuat relasi agar berada dalam bentuk normal pertama, perlu langkah untuk untuk menghilangkan atribut-atribut bernilai ganda. Hal yang penting lainnya yang

perlu dilakukan setelah melakukan normalisasi ke bentuk pertama adalah menentukan kunci primernya. Kunci primer dapat dipilih melalui determinan–determinan yang muncul dalam relasi yang membuat setiap baris dapat diidentifikasi secara unik (tidak ada yang kembar). Kalau tidak ada yang determinan dengan satu atribut yang memenuhi, pilihlah gabungan atribut yang dapat digunakan untuk membedakan antara satu baris dengan baris lainnya. (Abdul Kadir , 2008:130).

1.6 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan alat yang digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir ataupun lingkungan fisik dimana data tersebut akan disimpan (Jogiyanto HM., 2005:700). Pengertian *Data Flow Diagram (DFD)* adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam obyek kemudian melewati suatu proses yang mentransformasikan ke tujuan yang lain, yang ada pada objek lain. (Wijaya, 2007).

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 dan pembahasannya berupa global 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

digambarkan dalam *Context Diagram* adalah hubungan antar *terminator* dan *data source*.

2. *Diagram level 0*

Merupakan diagram yang berada diantara diagram konteks dan diagram detail serta menggambarkan proses utama dari *DFD*. Hal yang digambarkan dalam diagram level 0 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. *Data Flow Diagram* (*DFD*) memiliki empat komponen, yaitu:

3.1 *Terminator*

Terminator atau *External Entity* atau kesatuan luar yang mewakili entitas *external* yang berkomunikasi dengan sistem yang dikembangkan. *Terminator* merupakan kesatuan di lingkungan sistem yang dapat berupa orang atau sistem yang berada di lingkungan luar sistem yang memberikan masukan maupun yang menerima keluaran dari sistem serta berupa bagian atau divisi diluar sistem yang berkomunikasi dengan sistem. *Terminator* ini sering juga disebut entitas *external*.

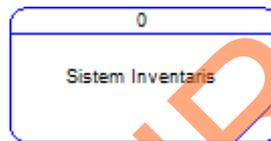
3.2 Proses

Proses sering dikenal dengan nama *Bubble*, fungsi, atau informasi. Komponen proses menggambarkan bagian dari sistem yang

mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi atau empat persegi panjang tegak dengan sudut tumpul. Proses diberi nama untuk menerangkan proses atau kegiatan apa yang sedang dilaksanakan dan setiap proses harus diberi penjelasan yang lengkap sebagai berikut:

a. Identifikasi Proses

Umumnya berupa angka yang menunjukkan nomor dari proses dan ditulis pada bagian atas simbol.



Gambar 3.2 Simbol Proses

b. Nama Proses

Menunjukkan apa yang sedang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya.

Nama proses diletakkan dibawah identifikasi proses.

4. *Data Store*

Data Store digunakan sebagai sarana untuk pengumpulan data. *Data store* disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada satu ujungnya atau dua garis horizontal. Suatu nama perlu diberikan pada *data store* menunjukkan nama filenya. *Data store* biasanya berkaitan dengan penyimpanan file atau database yang dilakukan secara terkomputerisasi. *Data store* dihubungkan dengan alur data hanya pada komponen proses pengertiannya sebagai berikut:

- a. Alur data dari *store* yang berarti sebagai pengaksesan data untuk suatu proses.
- b. Alur data ke proses berarti melakukan *update* data seperti menambah data, mengurangi data maupun mengubah data.

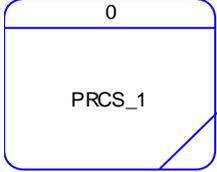
5. Alur Data

Alur data dapat digambarkan dengan anak panah yang menuju ke dalam proses maupun ke luar proses. Alur data digunakan untuk menerangkan perpindahan data atau informasi dari suatu bagian ke bagian lainnya. Syarat-syarat pembuatan sebuah DFD adalah:

- a. Pemberian nama untuk setiap komponen DFD.
- b. Pemberian nomor pada proses DFD.
- c. Penggambaran DFD serapi mungkin.
- d. Menghindari pembuatan DFD yang rumit.
- e. Memastikan DFD dibangun secara konsisten.

Table 1.1 Simbol *Data Flow Diagram*

No	Simbol	Nama Simbol	Keterangan
1		<i>External Entity</i> atau <i>Boundary</i>	Simbol ini menunjukkan kesatuan dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan pengaruh berupa input atau menerima output

2		Data Flow atau Aliran Data	Aliran data dapat digambarkan dengan tanda panah dan garis yang diberi nama dari aliran data tersebut
3		Proses	Dalam simbol tersebut dituliskan nama proses yang akan dikerjakan oleh sistem dari transformasi aliran data yang keluar. Suatu proses mempunyai satu atau lebih input data dan menghasilkan satu atau lebih output data.
4		Data Store	Data store merupakan simpanan dari data yang dapat berupa file atau catatan manual, dan suatu agenda atau buku. Data store digunakan untuk menyimpan data sebelum dan sesudah proses lebih lanjut

1.7 Hypertext Preprocessors (PHP)

PHP adalah bahasa yang sederhana namun kuat dirancang untuk membuat konten HTML. Sejak dimulai pada tahun 1994, PHP telah mampu menyaingi

bahasa *web* karena keunggulan popularitas bahasa dan kemudahan penggunaan. (Tatroe, 2002).

PHP pertama kali dibuat oleh Rasmus Lerdroft. PHP sering digunakan untuk membangun web dinamis dimana proses keseluruhan berjalan pada web server dan menampilkan hasilnya pada web browser.

PHP adalah bahasa pemrograman untuk membuat situs web dinamis dan interaktif. PHP berjalan di server web dan melayani pengunjung dengan halaman web sesuai permintaan. (Doyle, 2009 :p3)

1.8 *My Structure Query Language (MySQL)*

MySQL merupakan bahasa pemrograman *open-source* yang paling populer dan banyak digunakan di lingkungan Linux. Kepopuleran ini karena ditunjang oleh performansi *query* dari *database*-nya yang jarang bermasalah. (Allen dan Hornberger, 2002: p220). MySQL adalah sebuah program pembuatan database yang bersifat *open source*, artinya siapa saja dapat menggunakannya secara bebas. (Nugroho, 2004: p29)

MySQL dikembangkan oleh MySQL AB, sebuah perusahaan komersial yang membangun layanan bisnisnya melalui *database* MySQL. Awal mula pengembangan MySQL adalah pengguna *mSQL* untuk koneksi ke tabel mempergunakan rutin *level* rendah (*ISAM*). Setelah beberapa pengujian diperoleh kesimpulan *mSQL* tidak cukup cepat dan fleksibel untuk memenuhi kebutuhan. Sehingga dihasilkan suatu antarmuka *SQL* baru pada *database* tetapi dengan *API* yang mirip *mSQL*. *API* ini dipilih sedemikian sehingga memudahkan *porting* kode.

1.9 Oracle Database Management System

Oracle *Database* 10g dirancang untuk secara efektif digunakan pada segala sesuatu mulai dari *server blade* kecil ke *server SMP* dan *cluster* terbesar dari semua ukuran. (Oracle Edison Group, 2005). Oracle Database 10g merupakan peningkatan yang signifikan dari versi-versi Oracle sebelumnya. Fitur-fitur baru membuat pengembang dan administrator memiliki kontrol lebih mudah terhadap penampung, pemroses dan pengambilan data. (Loney dan Bryla, 2007 : p4)

Oracle adalah *relational database management system* (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi. Oracle Server menyediakan solusi yang efisien dan efektif karena kemampuannya dalam hal sebagai berikut (Larry Ellison, 1977):

- Dapat bekerja di lingkungan client/server (pemrosesan tersebar)
- Menangani manajemen space dan basis data yang besar
- Mendukung akses data secara simultan
- Performansi pemrosesan transaksi yang tinggi
- Menjamin ketersediaan yang terkontrol
- Lingkungan yang terreplikasi