

## BAB III

### LANDASAN TEORI

#### 3.1 Manajemen Jaringan

Manajemen jaringan adalah kemampuan untuk memonitor, mengontrol jaringan komputer dan komponen sistem. Manajemen jaringan mencoba menggunakan kekuatan komputer dan jaringan untuk mengatur dan mengelola sistem serta jaringan itu sendiri. Dalam melakukan hal itu, para manajer jaringan mengandalkan berbagai macam peralatan. Semakin kita memasuki era komputer pada setiap *desktop*, kita pun semakin menyandarkan diri pada manajemen jaringan sebagai wahana untuk menjamin bahwa segala sesuatunya dapat beroperasi dan memberikan pelayanan yang handal (Taufan, 2001).

*The International Organization for Standardization (ISO)* mendefinisikan sebuah model konseptual untuk menjelaskan fungsi manajemen jaringan. Fungsi manajemen jaringan terdiri sebagai berikut:

a. Manajemen Kesalahan (*Fault Management*)

Menyediakan fasilitas yang memungkinkan *administrator* jaringan untuk mengetahui kesalahan (*Fault*) pada perangkat yang dikelola, jaringan, dan operasi jaringan, agar dapat segera menentukan apa penyebabnya dan dapat segera mengambil tindakan (perbaikan).

b. Manajemen Konfigurasi (*Configuration Management*)

Memonitor informasi konfigurasi jaringan sehingga dampak dari perangkat keras atau pun lunak tertentu dapat dikelola dengan baik.

c. Pelaporan (*Accounting*)

Mengukur utilisasi jaringan dari pengguna atau *grup* tertentu untuk membantu dalam menjaga performa jaringan pada *level* tertentu yang dapat diterima.

d. Manajemen Performa (*Performance Management*)

Mengukur berbagai aspek dari performa jaringan termasuk pengumpulan dan analisis dari *data* statistik sistem sehingga dapat dikelola dan dipertahankan pada *level* tertentu yang dapat diterima.

e. Manajemen Keamanan (*Security Management*)

mengatur akses ke sumber daya jaringan sehingga informasi tidak dapat diperoleh tanpa izin.

Arsitektur manajemen jaringan terdiri dari elemen-elemen sebagai berikut:

a. *Network Management Station* (NMS)

Menjalankan aplikasi manajemen jaringan yang mampu mengumpulkan informasi mengenai perangkat yang dikelola dari agen manajemen yang terletak dalam perangkat.

b. Perangkat yang dikelola

Berupa semua jenis perangkat yang berada dalam jaringan, seperti *computer*, *printer*, atau pun *router*.

c. Agen Manajemen

Memberikan informasi mengenai perangkat yang dikelola kepada NMS dan dapat juga menerima informasi kendali/kontrol.

d. Protokol Manajemen Jaringan

Digunakan oleh NMS dan agen manajemen untuk bertukar informasi.

e. Informasi Manajemen

Merupakan informasi yang dipertukarkan antara NMS dan agen manajemen yang memungkinkan proses monitor dan kontrol dari perangkat.

### 3.2 *Simple Network Management Protocol (SNMP)*

*Simple Network Management Protocol* SNMP adalah sebuah protokol yang didesain untuk memberikan kemampuan kepada pemakai untuk memonitor dan mengatur jaringan komputernya secara sistematis dari jarak jauh atau dalam satu pusat kontrol saja.

Pengolahan ini dijalankan dengan mengumpulkan *data* dan melakukan penetapan terhadap *variabel-variabel* dalam elemen jaringan yang dikelola. SNMP merupakan protokol standar industri yang digunakan untuk memonitor dan mengelola berbagai perangkat di jaringan *Internet* meliputi *hub*, *router*, *switch*, *email Server*, *file Server*, *workstation* dan sistem manajemen jaringan secara jarak jauh (*remote*) (Onno, 2001).

Untuk dapat berkomunikasi antar stasiun manajemen dan agen, maka diperlukan suatu protokol SNMP. Cara yang biasa dipakai SNMP adalah stasiun manajemen mengirim permintaan (*request*) ke agen tentang informasi atau memerintahnya untuk melakukan pembaharuan keadaannya dengan cara-cara tertentu. Idealnya, agen cukup menjawab pertanyaan diminta atau dikonfirmasi bahwa agen telah melakukan pembaharuan keadaan sesuai dengan permintaan manajer.

Ide dasar dari setiap manajemen jaringan adalah bahwa terdapat dua tipe sistem pada setiap jaringan yang terkonfigurasi yaitu: agen dan manajer. Yang

ditempatkan pada setiap titik pada jaringan yang akan diatur, termasuk PC, *Workstation*, *Server*, *bridge*, *router* dan lainnya termasuk modul agen.

Semua aplikasi jaringan pada umumnya berbagi protokol manajemen jaringan yang umum. Protokol ini menyediakan fungsi-fungsi *fundamental* untuk mengambil informasi manajemen dari agen dan mengirimkan perintah kepada agen. Protokol ini kemudian menggunakan fasilitas komunikasi seperti TCP/IP atau OSI (*Organisasi Standar Internasional*).

Akhirnya setiap agen memelihara basis informasi manajemen yang berisi informasi terbaru dan yang sebelumnya tentang konfigurasi dan lalu lintas lokalnya. Manajemen stasiun akan memelihara basis informasi manajemen *global* dengan informasi berisi rangkuman dari semua agen. *Simple Network Management Protocol* (SNMP) terdiri dari tiga elemen utama, yaitu:

1. Manajer
2. MIB (*Management Informasi Base*)
3. Agen

### 3.3 Apache

Menurut Apache dalam Teguh (2010), 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*.

*Server HTTP Apache* atau *Server Web/WWW Apache* adalah *Server web* yang dapat dijalankan di banyak sistem operasi (*Unix*, *BSD*, *Linux*, *Microsoft*

*Windows* dan *Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas *web/www* ini menggunakan HTTP.

*Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis *basis data* antar muka pengguna berbasis grafik (GUI) yang memungkinkan penanganan *Server* menjadi mudah.

*Apache* merupakan perangkat lunak sumber terbuka dikembangkan oleh komunikasi terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

### **3.4 Hypertext Preprocessor (PHP)**

*Hypertext Preprocessor* (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 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 hamper 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. *Programmer* asal Greenland ini membuat PHP pada tahun 1994 untuk menggantikan sebagian kecil kumpulan *script* dengan *Perl* yang digunakan untuk *maintenance* halaman *web* miliknya. Lerdorf mengawali menciptakan PHP untuk

menampilkan *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 menampilkan 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*. Sintak yang digunakan berasal dari bahasa C, *java* maupun *Perl*.

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

1. *Software* ini disebar dan dilisensikan sebagai perangkat lunak yang *open source*, maksudnya pendistribusian kode 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 menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.
3. Penulisan *script* PHP dapat menyatu dengan dokumen HTML, sehingga memudahkan pembuatannya. Untuk membedakan dengan sintak 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 digunakan *data* dari *database* dapat sangat mudah untuk dilakukan. *Database* yang didukung oleh PHP antara lain: *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.*

### 3.5 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 digunakan 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, table, 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* : memberikan 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. *Auxiliary*

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

### 3.6 Oracle 10g

*Oracle Database 10g* merupakan *database* yang pertama kali didesain untuk aktivitas *Enterprise Grind Computing*. *Oracle database 10g* mampu memotong biaya operasional sembari memberikan *services* yang berkualitas tinggi. Juga mampu secara merespon terhadap kebutuhan bisnis dengan resiko yang



rendah. Selain itu produk ini juga sangat mudah untuk dideploy dan juga dipelihara. Ada beberapa edisi yang tersedia dalam *Oracle database 10g*, yaitu :

1. *Standard Edition One*

Sangat mudah digunakan, mempunyai kekuatan yang bagus, performa yang baik dalam *workgroup* dan juga sudah mendukung aplikasi *internet* maupun *intranet*.

2. *Standard Edition*

Fitur dasarnya sama seperti edisi *Standard Edition One*, tetapi dengan dukungan mesin yang lebih besar dan juga mendukung sistem *clustering* dan juga layanan *Real Application Clusters*.

3. *Enterprise Edition*

Memberikan efisiensi, reabilitas dan juga keamanan manajemen *data* untuk *mission-critical application*, seperti dalam lingkungan OLTP, *data warehouse* yang intensif melakukan *query* dan juga aplikasi *internet*.

4. *Oracle Real Application Clusters (RAC)*

*Oracle RAC* merupakan suatu lingkungan komputasi yang memanfaatkan interkoneksi antar komputer dengan menggunakan teknologi *cluster*.

5. *Oracle Partitioning*

*Oracle Partitioning* mampu untuk meningkatkan manajemen *data* bagi proses OLTP (*Online Transaction Processing*), *data mart* dan juga aplikasi *data warehouse* dengan menambahkan kemampuan pengaturan, ketersediaan dan juga performa pada struktur tabel dan index yang kompleks.

## 6. *Oracle OLAP*

*Oracle OLAP* merupakan mesin kalkulasi dengan skalabilitas dan performa yang tinggi untuk pemrosesan *data*, juga dilengkapi dengan administrasi dan manajemen yang terintegrasi sebagai solusi aplikasi yang membutuhkan analisa.

## 7. *Oracle Data Mining*

*Oracle Data Mining* mengijinkan perusahaan untuk membangun suatu sistem intelligence yang canggih sehingga dapat digunakan untuk menggali informasi lebih dalam dari *database* yang ada, menemukan fakta-fakta baru dan juga mengintegrasikan informasi tersebut ke dalam aplikasi bisnis.

## 8. *Oracle Spatial*

*Oracle Spatial* mengijinkan pengguna dan pembuat aplikasi untuk mengintegrasikan *data spasial* mereka ke dalam aplikasi *enterprise*.

Menurut Rio dalam Teguh (2010), *Oracle 10g* merupakan salah satu produk *Oracle* yang berupa *database Oracle*, dimana *database* versi ini lebih memberikan tekanan untuk kebutuhan *marketing*. Hal ini terlihat dari adanya huruf ‘g’ yang tercantum pada nama *database*, yang berarti *grid computing ready*. *Oracle 10g* terdiri dari 4 buah teknologi, yaitu :

### 1. *Automatic Storage Management (ASM)*

Berbentuk *console*, digunakan untuk manajemen ruang penyimpanan secara otomatis. Dengan ASM, kita dapat mengganti dan menambah *harddisk* pada saat *Server* sedang *online*. ASM juga sangat membantu dalam menstabilkan atau menyeimbangkan isi *data* dalam *harddisk*.

## 2. *Real Application Clusters (RAC)*

Keunggulan RAC :

1. Digunakan untuk meng-*cluster* (mengelompokkan) *node (Server)*.
2. Dapat menambahkan *node*, tanpa men-*shutdown node* yang aktif.
3. Jika *node* yang satu *down*, maka *node* yang lain akan tetap aktif.

## 3. *Oracle Streams*

Sebuah kesatuan *framework sharing* information, yang digunakan untuk membuat sebuah replikasi (*data replication*), sebagai jembatan komunikasi antar *Server*, dapat digunakan untuk mereplikasi dua *Server* atau lebih, sering digunakan untuk menyaring *data warehouse* dari *Server* yang sedang *online*. Biasanya digunakan untuk menggandakan *Server* dengan struktur skema *database* yang sama.

## 4. *Enterprise Management Grid Control*

Berbentuk *web browser*, dapat dijalan secara *online* maupun *offline (localhost)*, berfungsi untuk manajemen seluruh keperluan *database* secara visual seperti : *monitoring database, flashback data, control* memori yang digunakan, *control data, control user, control waktu*, dan masih banyak lagi.

### 3.7 *System Flow*

*System Flow* adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap *sub-sistem* (Hartono, 1998:10).

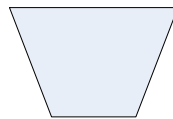
Terdapat berbagai bentuk simbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *off-line storage*, *on-page reference*, dan *off-page reference*.

*Terminator* merupakan bentuk symbol yang digunakan sebagai tanda dimulainya jalan proses atau pun tanda akhir dari sebuah pengerjaan suatu sistem. Simbol dari *terminator* dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Terminator*.

*Manual operation* digunakan untuk menggambarkan sebuah proses kerja yang digunakan tanpa menggunakan komputer sebagai medianya (menggunakan proses manual). Simbol dari *manual operation* dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Manual Operation*.

*Document* merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik. Simbol dari *document* dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Document*.

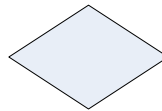
*Process* adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. Simbol dari *process* dapat dilihat pada Gambar 3.4.

Gambar 3.4 *Process*.

*Database* digunakan sebagai media penyimpanan *data* yang bersifat terkomputerisasi. Simbol dari *database* dapat dilihat pada Gambar 3.5.

Gambar 3.5 *Database*.

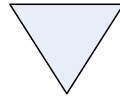
*Decision* merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar atau salah. Simbol dari *decision* dapat dilihat pada Gambar 3.6.

Gambar 3.6 *Decision*.

*Manual input* digunakan untuk melakukan proses *input* kedalam *database* melalui *keyboard*. Simbol dari *manual input* dapat dilihat pada Gambar 3.7.

Gambar 3.7 *Manual Input*.

*Off-line storage* merupakan bentuk media penyimpanan yang berbeda dengan *database*, dimana media penyimpanan ini menyimpan dokumen secara manual atau lebih dikenal dengan nama arsip. Simbol dari *off-line storage* dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Off-Line Storage*.

*On-page reference* digunakan sebagai simbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus *data* yang ada terlalu jauh dalam permasalahan letaknya. Simbol dari *on-page reference* dapat dilihat pada Gambar 3.9.



Gambar 3.9 *On-Page Reference*.

*Off-page reference* memiliki sifat yang sedikit berbeda dengan *on-page reference*, karena simbol ini hanya digunakan apabila arus *data* ada dilanjutkan ke halaman yang berbeda. Simbol dari *off-page reference* dapat dilihat pada Gambar 3.10.



Gambar 3.10 *Off-Page Reference*.

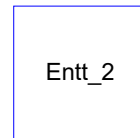
### 3.8 Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) sering digunakan untuk menggambarkan suatu sistem yang telah ada atau baru yang telah dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (Hartono, 1999:700). Diagram ini digunakan untuk menggambarkan arus data di dalam sistem secara terstruktur dan jelas. Selain itu DFD juga merupakan

gambaran dari sistem yang baik. Adapun 4 (empat) simbol yang sering dipakai dalam DFD, yaitu:

### 1. *External Entity*

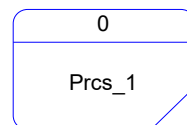
Simbol ini merupakan simbol *external entity*, digunakan sebagai sumber dari inputan sistem atau tujuan dari *output* sistem. Seperti pada Gambar 3.11.



Gambar 3.11 *Simbol External Entity*.

### 2. *Process*

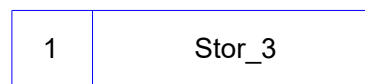
Simbol proses dimana sering digunakan untuk melakukan perubahan terhadap input yang masuk sehingga menghasilkan data dari perubahan input yang diolah tadi. Seperti pada Gambar 3.12.



Gambar 3.12 *Simbol Process*.

### 3. *Data Store*

Simbol dari penyimpanan data, sering digunakan sebagai simpanan dari data yang dapat berupa suatu *file* atau basis *data*. Seperti pada Gambar 3.13.

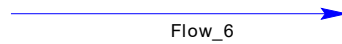


Gambar 3.13 *Simbol Data Store*.

### 4. *Data Flow*

Simbol yang menggambarkan aliran *data*, yang sering digunakan untuk menghubungkan antara proses dengan proses, proses dengan sumber proses dan

proses dengan tujuan. Sedangkan anak panahnya menunjukkan arah aliran *data*-nya. Seperti pada Gambar 3.14.



Gambar 3.14 *Simbol Data Flow*.

### 3.9 Alamat IP

Alamat IP (*Internet Protocol Address* atau sering disingkat IP) adalah deretan angka binar antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer *host* dalam jaringan *internet*. Panjang dari angka ini adalah 32-bit (untuk IPv4 atau IP versi 4), dan 128-bit (untuk IPv6 atau IP versi 6) yang menunjukkan alamat dari komputer tersebut pada jaringan *internet* berbasis TCP/IP. Sistem pengalamatan IP ini terbagi menjadi dua, yakni:

1. IP Versi 4 (IPv4)
2. IP Versi 6 (IPv6)

Perbandingan Alamat IPv4 dan IPv6 dapat dilihat di Tabel 3.1 dibawah ini.

Tabel 3.1 Perbandingan Alamat IPv4 dan IPv6

Kriteria	Alamat IP versi 4	Alamat IP versi 6
Panjang alamat	32 bit	128 bit
Menggunakan kelas alamat	Belakang tidak digunakan lagi, mengingat telah tidak relevan dengan perkembangan jaringan <i>internet</i> yang pesat	Tidak
Alamat <i>multicast</i>	Kelas D, yaitu 224.0.0.0/4	FF00:/8



Kriteria	Alamat IP versi 4	Alamat IP versi 6
Alamat <i>broadcast</i>	Ada	Tidak ada
Alamat yang belum ditentukan	0.0.0.0	::
Alamat <i>loopback</i>	127.0.0.1	::1
Alamat IP publik	Alamat IP publik IPv4, yang ditetapkan oleh otoritas <i>internet</i> (IANA)	Alamat IPv6 <i>unicast Global</i>
Alamat IP Pribadi	Alamat IP pribadi IPv4, yang ditetapkan oleh otoritas <i>internet</i>	Alamat IPv6 <i>unicast site-local</i> (FEC0::/48)
Konfigurasi alamat otomatis	Ya (APIPA)	Alamat IPv6 <i>unicast link-local</i> (FE80::/64)
Representasi Teksual	<i>Dotted decimal format notation</i>	Calon <i>hexdecimal format notation</i>
Fungsi <i>Prefiks</i>	<i>Subnet mask</i> atau panjang <i>prefiks</i>	Panjang <i>prefiks</i>
Resolusi alamat DNS	<i>A Resource Record (Single A)</i>	<i>AAAA Resource Record (Quad A)</i>

### 3.10 Analisis dan Perancangan Sistem

Analisis sistem merupakan tahap yang paling penting dari suatu pengembangan sistem karena merupakan tahap awal untuk melakukan evaluasi

permasalahan yang terjadi serta kendala-kendala yang dihadapi dari sebuah sistem yang telah berjalan.

Analisis sistem itu sendiri dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluai permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Hartono, 1999).

Untuk itu, diperlukan ketelitian dalam mengerjakan, sehingga tidak dapat kesalahan dalam tahap selanjutnya, yaitu tahap perancangan sistem. Langkah-langkah yang diperlukan di dalam menganalisa sistem adalah:

1. Tahap perencanaan sistem
2. Tahap analisis sistem
3. Tahap perancangan sistem
4. Tahap penerapan sistem
5. Membuat laporan dari hasil analisa

Pada tahap perancangan, dilakukan identifikasi masalah serta diperlukan adanya analisa yang digunakan untuk menentukan faktor-faktor yang menjadi permasalahan dalam sistem yang telah ada atau digunakan.

*Data-data* yang baik yang berasal dari sumber-sumber internal seperti misalnya laporan-laporan, dokumen *observasi*, maupun sumber-sumber di luar lingkungan sistem seperti pemakai sistem, dikumpulkan sebagai bahan pertimbangan analisa. Jika semua permasalahan sudah di-identifikasi, dilanjutkan dengan mempelajari dan memahami alur kerja dari sistem yang digunakan.

Kemudian diteruskan dengan menganalisa dan membandingkan sistem yang terbentuk dengan sistem yang sebelumnya di gunakan. Dengan adanya perubahan tersebut, maka langkah selanjutnya adalah membuat laporan-laporan hasil analisa sebelumnya dan sistem yang akan diterapkan. Perancangan sistem adalah proses menyusun atau mengembangkan sistem informasi yang baru. Dalam tahap ini, harus dipastikan bahwa semua persyaratan untuk menghasilkan informasi dapat terpenuhi.

Hasil sistem yang dirancang harus sesuai dengan kebutuhan pemakai, karena rancangan tersebut meliputi perancangan mulai dari sistem yang umum hingga diperoleh sistem yang lebih spesifik. Dari hasil rancangan tersebut, dibentuk pula rancangan *database* disertai dengan struktur *file* antara sistem yang satu dengan sistem yang lainnya. Selain itu, dibentuk pula rancangan *input* dan *output system*, misalnya menentukan berbagai bentuk *input data* dan isi laporan.

Apabila di dalam perancangan sistem terdapat kesalahan maka kita perlu melihat kembali dari analisa sistem yang telah dibuat. Sehingga dapat di ambil kesimpulan bahwa analisa sistem mempunyai hubungan erat dengan perancangan sebuah sistem.

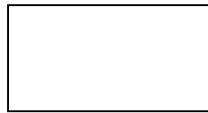
### **3.11 Entity Relationship Diagram (ERD)**

Proses *reverse engineering* terhadap suatu basis *data* menjadi suatu kebutuhan bagi perancang basis *data* untuk mengetahui struktur dari sebuah basis *data*. Struktur tersebut biasanya dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD).

ERD dibagi menjadi dua macam yaitu: *Conceptual Data Model* (CDM), dan *Physical Data Model* (PDM). Simbol-simbol yang sering digunakan adalah:

### 1. Entity

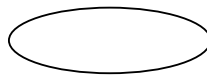
*Entity* merupakan sesuatu yang mudah diidentifikasi. Sebuah *entity* bisa berupa *obyek*, tempat, orang, konsep, atau aktivitas. *Entity* dinyatakan dalam simbol persegi panjang. Simbol *entity* pada gambar 3.15.



Gambar 3.15 *Entity*.

### 2. Atribut

Atribut merupakan penjelasan-penjelasan dari *entity* yang membedakan *entity* satu dengan yang lain. Sebuah atribut juga merupakan sifat-sifat dari sebuah *entity*. Atribut dinyatakan dalam simbol *ellips*. Simbol atribut pada gambar 3.16.



Gambar 3.16 *Atribut*.

### 3. Relationship

*Relationship* adalah penghubung antara suatu *entity* dengan *entity* yang lain dan merupakan bagian yang sangat penting di dalam mendesain *database*. Ada tiga tipe *relationship* yang dikenal yaitu :

#### a. One-to-One Relationship

Jenis hubungan antar tabel yang menggunakan secara bersama sebuah kolom *primary key*. Jenis hubungan ini tergolong jarang digunakan, kecuali untuk alasan keamanan atau kecepatan akses *data*. Seperti departemen hanya

mengerjakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.

*b. One-to-Many Relationship*

Jenis hubungan antar tabel dimana satu *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Jenis hubungan ini yang paling sering digunakan. Misalnya satu pekerjaan hanya dikerjakan oleh satu departemen saja, namun satu departemen dapat mengerjakan beberapa macam pekerjaan sekaligus.

*c. Many-to-Many Relationship*

Jenis hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.

*d. Many-to-One Relationship*

Jenis hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan satu *record* pada tabel lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, namun satu pekerjaan hanya dikerjakan oleh satu departemen saja.

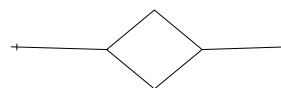
Menurut Sutanta (2004), relasi antar entitas dapat digambarkan melalui salah satu gambar 3.17 dan gambar 3.18 dari pilihan berikut ini:

1. Pilihan 1

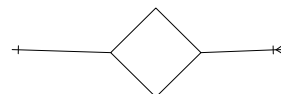
Jenis relasi

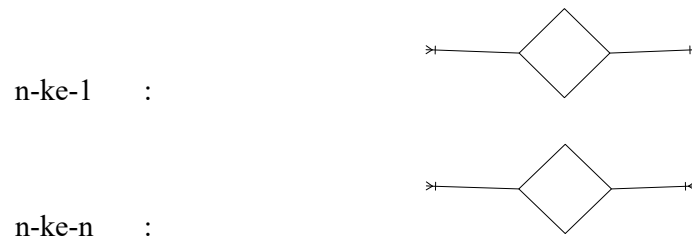
Simbol yang digunakan

1-ke-1 :



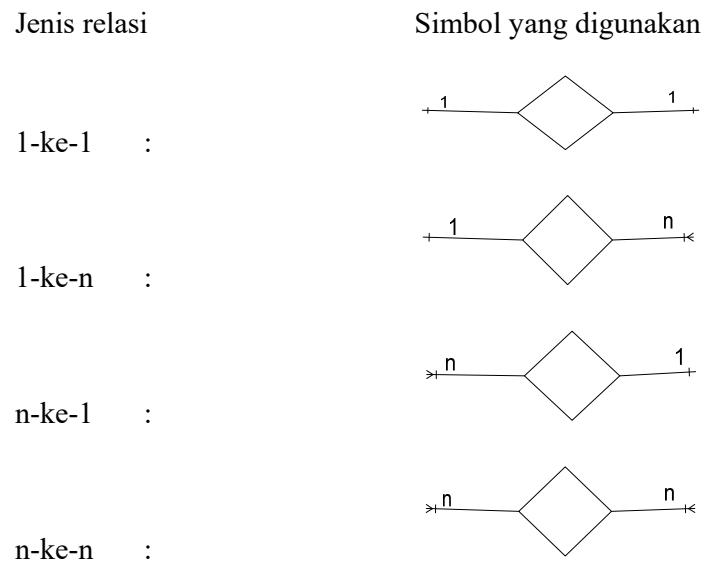
1-ke-n :





Gambar 3.17 Simbol relasi antar entitas (pilihan 1).

## 2. Pilihan 2



Gambar 3.18 Simbol relasi antar entitas (pilihan 2).

ERD dapat digambarkan menggunakan salah satu dari pilihan di atas, namun penggunaannya harus konsisten. Jika menggunakan simbol pilihan 1, maka untuk seluruh bagian ERD harus menggunakan simbol kelompok pilihan 1.

## 3. Kunci relasi

Kunci relasi atau *key* adalah suatu properti yang menentukan apakah suatu kolom pada tabel sangat penting atau tidak. Berdasarkan macamnya, kunci relasi terdiri dari:

a. Kunci kandidat

Yaitu satu atau gabungan minimal atribut yang bersifat unik yang dapat digunakan untuk mengidentifikasi setiap *record* dalam relasi.

b. Kunci *Primer*

Yaitu bagian atau salah satu dari kunci kandidat yang digunakan sebagai kunci utama untuk membedakan setiap *record* dalam relasi. Kunci *Primer* biasa disebut sebagai *primary key*.

c. Kunci Alternatif

Yaitu bagian dari kunci kandidat yang tidak digunakan sebagai kunci utama.

d. Kunci Penghubung

Kunci penghubung atau *foreign key* yaitu satu atau gabungan sembarang atribut yang menjadi kunci utama dalam relasi lain yang mempunyai hubungan secara logik. Kunci penghubung dan kunci utama harus memiliki tipe dan ukuran *data* yang sama.

### 3.12 Dreamweaver CS 5

*Dreamweaver* merupakan *software* aplikasi yang digunakan sebagai HTML editor *profesional* untuk mendesain *web* secara visual. Aplikasi ini juga yang biasa dikenal dengan istilah WYSIWYG (*What You See Is What You Get*), yang intinya adalah bahwa anda tidak harus berurusan dengan *tag-tag* HTML untuk membuat sebuah situs. Selain itu *Dreamweaver* juga memberikan keleluasaan untuk menggunakan sebagai media penulisan bahasa pemrograman *web*.

*Adobe* kembali mengeluarkan varian terbaru dari *Dreamweaver* yaitu *Dreamweaver CS 5*. Ada beberapa fitur baru yang dapat ditemukan pada versi terbaru *Dreamweaver CS 5* ini, diantaranya integrasi dengan *Adobe Business Catalyst*, integrasi *Adobe Browser Lab*, integrasi dengan CMS, pengecekan CMS, dan lain-lain. Selain itu juga ada beberapa fitur dalam versi *Dreamweaver CS 4* yang tidak disertakan atau ditemukan kembali dalam *Dreamweaver CS 5*, diantaranya fasilitas pembuatan *web* album foto, penambahan *flash paper*, beberapa *behaviour javascript*, dan lain-lain (madcoms, 2011).

*Dreamweaver CS 5* mempunyai ruang kerja yang dapat digunakan untuk mendesain sebuah halaman *web*. Selain itu kita juga dapat mengubah tampilan umum dari ruang kerja *Dreamweaver CS 5*. Adapun elemen-elemen ruang kerja dari *Dreamweaver CS 5* adalah sebagai berikut:

a. *Application Bar*

Berada dibagian paling atas jendela aplikasi *Dreamweaver CS 5*. Baris ini berisi tombol *workspace (workspace switcher)*, *CS Live*, *Menu* dan aplikasi lainnya.

b. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *design* atau tampilan kode. Selain itu juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.

c. *Jendela Dokumen*

Lembar kerja tempat membuat dan mengedit desain halaman *web*.



d. *Workspace Switcher*

Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Dreamweaver CS 5*. Sebagai contoh mengubah tampilan menjadi tampilan *classic*, yaitu tampilan ruang kerja *Dreamweaver* versi sebelumnya.

e. *Panel Groups*

Kumpulan panel yang saling berkaitan, *panel-panel* ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. *Panel* ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara *default*, *panel groups* berisi *panel Adobe Browser Lab, Adobe Business Catalyst, Insert, CS 5 Styles, AP Element* dan *Files*.

f. *Tag Selector*

Terletak dibagian bawah jendela dokumen, satu baris dengan status *bar*. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. *Tag selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja desain.

g. *Property Inspector*

Digunakan untuk melihat dan mengubah berbagai *property* objek atau *teks* pada jendela desain. *Property* untuk satu objek dengan objek lainnya selalu berbeda-beda. Jendela ini tidak dapat diuraikan pada tampilan jendela kode.

h. *Toolbar Standard*

Baris *toolbar* ini berisi tombol-tombol yang mewakili perintah pada menu *File* dan *Edit*, diantaranya adalah perintah *New*, *Open*, *Save*, *Save All*, *Cut*, *Copy*, *Paste*, *Undo* dan *Redo*.

i. *Toolbar Style Rendering*

Secara *default* disembunyikan. *Toolbar* ini berisi tombol-tombol untuk menampilkan desain dalam media yang berbeda. Selain itu juga digunakan untuk mengaktifkan dan menonaktifkan *style* CSS.

j. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya tampil pada jendela kode.

k. *Toolbar Browser Navigation*

*Toolbar* ini merupakan *toolbar* baru yang ada didalam *Dreamweaver CS 5* dan letaknya tepat berada diatas jendela dokumen. *Toolbar* ini berisi tombol-tombol yang digunakan sebagai navigasi didalam *browser*.