BAB II
LANDASAN TEORI

2.1 Pengertian Sistem Informasi

Menurut John F. Nash (1995:8) yang diterjemahkan oleh La Midjan dan Azhar Susanto, menyatakan bahwa Sistem Informasi adalah : Sistem Informasi adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksudmenata jaringan komunikasi yang penting, proses atas transaksi-transaksi tertentu dan rutin, membantu manajemen dan pemakaiintern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat.

2.2 Rapor

Rapor merupakan dokumen yang menjadi penghubung komunikasi baik antara pihak sekolah dengan orang tua peserta didik maupun dengan pihak-pihak lain yang ingin mengetahui tentang hasil belajar peserta didik pada kurun waktu tertentu (Direktorat Pembinaan SMP, 2006).

2.3 *Online*

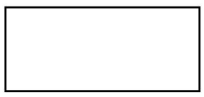
Online adalah suatu keadaan dimana sebuah *device* (peralatan) komputer terhubung dengan *device* lain, biasanya dengan menggunakan modem, *hotspot,* ataupun w*ireless* (Erfan, 2009).

2.4 E-R Diagram

Entity Relationship Diagram (E-R Diagram) adalah gambaran sistematis dari komponen-komponen ERD yaitu : model E-R yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang merepresentasikan seluruh fakta dari dunia nyata yang ditinjau (Fathansyah, 2007).

2.4.1 Entitas

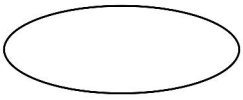
Entitas (*Entity*) merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Himpunan entitas (*Entity Set*) merupakan sekelompok entitas yang sejenis dan berada dalam lingkup yang sama. Himpunan entitas disimbolkan dengan persegi panjang (Fathansyah, 2007).



Gambar 2.1 Himpunan Entitas

2.4.2 Atribut

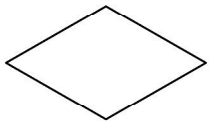
Atribut (*attributes*) adalah karakteristik yang mendeskripsikan karakteristik (*property*) dari entitas tersebut. Atribut disimbolkan dengan lingkaran/elips. Atribut yang berfungsi sebagai *key* primer digarisbawahi (Fathansyah, 2007).



Gambar 2.2 Atribut

2.4.3 Relasi

Relasi (*Relationship*) menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas, himpunan entitas tesebut membentuk himpunan relasi (r*elationship sets*). Himpunan relasi disimbolkan dengan belah ketupat (Fathansyah, 2007).



Gambar 2.3 Himpunan Relasi

2.4.4 Tingkatan Relasi

Tingkatan dari relasi menyatakan jumlah entitas yang berpartisipasi dalam suatu relasi. Tingkatan dari relasi terdiri dari 3 tingkatan yaitu :

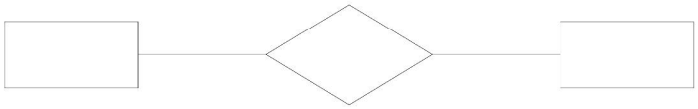
2.4.4.1 *Unary Degree*

Unary Degree (Tingkat Satu) merupakan relasi yang terjadi dalam sebuah himpunan entitas yang sama (Fathansyah, 2007).

Gambar 2.4 *Unary Degree*

2.4.4.2 *Binary Degree*

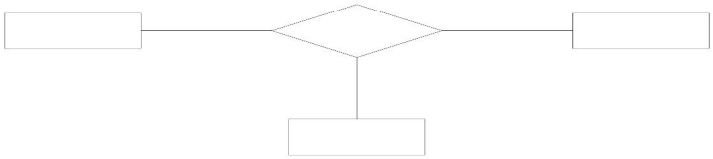
Binary Degree (Tingkat Dua) merupakan relasi yang terjadi di antara dua himpunan entitas yang berbeda (Fathansyah, 2007).



Gambar 2.5 *Binary Degree*

2.4.4.3 Ternary Degree

Ternary Degree (Tingkat Tiga) merupakan relasi yang terjadi di antara 3 himpunan entitas yang berbeda (Fathansyah, 2007).



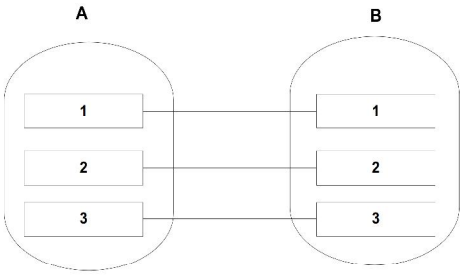
Gambar 2.6 *Ternary Degree*

2.4.5 Kardinalitas/Derajat Relasi

Kardinalitas relasi adalah batasan yang menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain (Fathansyah, 2007). Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misalnya A dan B) dapat berupa :

2.4.5.1 Satu ke Satu (*One to One*)

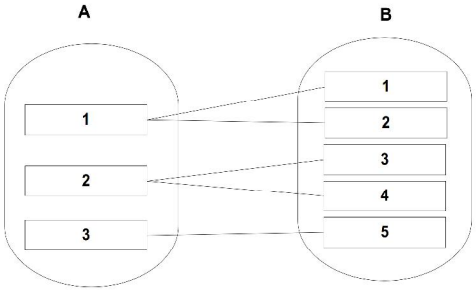
Relasi satu ke satu berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, begitu juga sebaliknya.



Gambar 2.7 Relasi Satu ke Satu

2.4.5.2 Satu ke Banyak (*One to Many*)

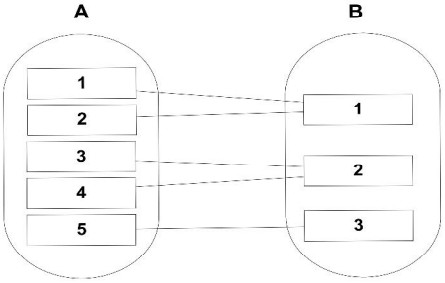
Relasi satu ke banyak berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.



Gambar 2.8 Relasi Satu ke Banyak

2.4.5.3 Banyak ke Satu (*Many to One*)

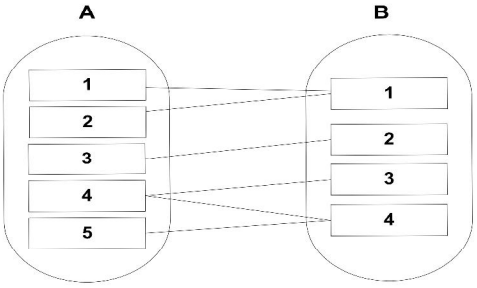
Relasi banyak ke satu berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya.



Gambar 2.9 Relasi Banyak ke Satu

2.4.5.4 Relasi Banyak ke Banyak (*Many to Many*)

Relasi banyak ke banyak berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.



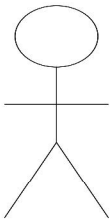
Gambar 2.10 Relasi Banyak ke Banyak

2.5 *Use Case* Diagram

Menurut Raharjo dan Mahastama (2009), *use case* diagram adalah sekumpulan skenario yang dihubungkan satu sama lain dengan satu tujuan yang sama dari pengguna. Sedangkan menurut Khairul (1994), *use case* diagram merupakan sekumpulan skenario sukses dan gagal yang mendeskripsikan seorang aktor yang menggunakan suatu sistem untuk mencapai suatu tujuan. Dengan kata lain, *use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, bukan "bagaimana".

Sebuah *use case* merepresentasikan suatu interaksi antara aktor dengan sistem (Khairul,1994).

Use case diagram dapat didefinisikan sebagai deskripsi sistem, lingkungan dan relasi antara sistem dengan lingkungannya. Deskripsi fungsionalitas sistem dari sudut pandang user (external view). Deskripsi kumpulan aksi yang dilakukan sistem untuk menghasilkan luaran pada actor. Adapun notasi-notasi yang digunakan dalam *use case* adalah sebagai berikut :

Simbol	**Nama**	**Keterangan**
	*Actor*	*Actor* adalah pengguna sistem. *Actor* tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output.
	Use Case	*Use case* digambarkan sebagai lingkaran elips dengan nama *use case* dituliskan di dalam elips tersebut.

	Association	*Association* digunakan untuk menghubungkan *actor* dengan *use case*. *Association* digambarkan dengan sebuah garis yang menghubungkan antara *actor* dengan *Use Case.*
<<*extend*>>	*Extend*	Relasi yang digunakan jika *use case* yang satu mirip dengan *use case* yang lain.
<<*include*>>	*Include*	Relasi jika terdapat perilaku yang mirip dengan beberapa *use case*.

Beberapa istilah yang digunakan dalam *use case* adalah sebagai berikut:

a. Aktor (*Actors*)

Aktor adalah sesuatu yang memiliki perilaku seperti orang (diidentifikasi dengan peran), sistem komputer, atau organisasi (Khairul, 1994).

Terdapat beberapa jenis aktor dalam *use case* yaitu sebagai berikut :

1. Aktor Utama (*Primary Actor*)

Aktor utama memiliki tujuan pengguna yang dicapai melalui penggunaan layanan sistem. Diidentifikasi untuk menemukan tujuan pengguna (*user goals*) yang mengarahkan *use case*.

2. Aktor Pendukung (*Supporting Actor*)

Aktor pendukung menyediakan suatu layanan (misalnya informasi) pada sistem. Diidentifikasi untuk mengklarifikasi *interface* dan protokol eksternal.

3. Aktor di Luar Panggung (*Offstage Actor*)

Aktor luar panggung memiliki kepentingan perilaku dalam *use case,* tetapi bukan utama ataupun pendukung. Melainkan diidentifikasi untuk menjamin agar semua kepentingan terpenuhi. Kepentingan aktor luar panggung (o*ffstage actor*) terkadang tidak kentara atau mudah dilupakan jika tidak diberi nama secara eksplisit.

b. Skenario (*Scenarios*)

Skenario adalah rangkaian aksi dan interaksi tertentu antara aktor dan sistem. Skenario disebut juga *use case instance* (Khairul, 1994).

Use Case memiliki tiga format yaitu :

a. Ringkas (*Brief*)

Ringkasan suatu paragraf pendek biasanya berupa skenario sukses utama, contoh proses penjualan. Ringkasan dibuat saat analisis kebutuhan awal untuk mendapatkan gambaran umum tentang subjek dan lingkup (Anonim a, 2012).

b. Sederhana (*Casual*)

Beberapa paragraf pendek, biasanya berupa skenario sukses utama dan skenario sukses alternatif contoh menangani barang yang dikembalikan

(*Handle Return*). Kedua skenario tersebut dibuat pada waktu yang sama dengan format ringkas (*brief)* (Anonim a, 2012).

c. Detail (*Fully Dressed*)

Semua tahapan dan variasi ditulis secara detail dan ada bagian pendukung, seperti prakondisi dan jaminan sukses. Setelah banyak *use case* diidentifikasi dan ditulis dalam format ringkas, selanjutnya selama *workshop requirement* pertama beberapa (sekitar 10%) *use case* yang bernilai tinggi dan signifikan secara arsitektur ditulis secara detail (Anonim a, 2012).

2.6 MySQL

MySQL adalah suatu perangkat lunak database relasi (*Relational Database Management System* / RDBMS), seperti halnya Oraqle Postgresql MS SQL, dan sebagainya. MySQL AB menyebut produknya sebagai database open source terpopuler didunia. Berdasarkan riset dinyatakan bahwa di *platform* web, dan baik untuk kategori open source maupun umum. MySQL adalah database yang paling banyak dipakai. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya (Kasmono, 2009).

Menurut Ridwan Sanjaya dan Onno W. Purbo (2006), MySQL mempunyai tiga sub bahasa, yaitu :

1. *Data Definition Language* (DDL)

DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut basis data, tabel atribut (kolom), batasan-batasan terhadap suatu atribut, serta hubungan antara tabel. Yang termasuk dalam kelompok DDL ini adalah *Create, Alter*, dan *Drop.*

2. *Data Manipulation Language* (DML)

   DML adalah kelompok perintah yang berfungsi untuk memanipulasi data dalam basis data. Misalnya untuk pengambilan, penyisipan, pengubahan dan penghapusan data. Perintah yang temasuk dalam kategori DML adalah:

   a. *select* : memilih data;
   b. *insert* : menambah data;
   c. *delete* : menghapus data;
   d. *update* : mengubah data.

3. *Data Control Language* (DCL)

   DCL berisi perintah untuk mengendalikan pengaksesan data. Pengendalian data dapat dilakukan berdasarkan per pengguna, per tabel, per kolom, maupun per operasi yang boleh dilakukan. Perintah-perintah yang termasuk DCL adalah:

   1. *grant* : memberikan kendali pengaksesan data;
   2. *revoke* : mencabut kemampuan pengaksesan data;
   3. *block table* : mengunci tabel.

2.7 *Web Server*

Menurut Purbo (2001) *Web Server* adalah software yang menjadi tulang belakang dari *world wide web* (www). Web server menunggu permintaan dari client yang

menggunakan browser seperti netscape navigator, Internet Explorer, Mozilla, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permntaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Dan ini mempunyai format standar, disebut dengan format SGML (*Standar General Markup Language*). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser tersebut. Contohnya, jika data yang dikirim berupa gambar, browser yang hanya mampu menampilkan teks (misalnya lynx) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja.

2.7.1 Apache

Apache adalah perangkat lunak yang dapat digunakan sebagai *web* server. *Hypertext Transfrer Protocol Daemon* (HTTPD) *server* yang menerjemahkan perintah-perintah *tag* dari *Hypertext Transfrer Protocol* (HTTP) (Kadir, 2002).

2.7.2 Hypertext Transfer Protokol (HTTP)

Hypertext Transfer Protokol (HTTP) merupakan protokol dasar dari *World Wide Web* (WWW) yang dapat diterapkan dalam aplikasi *client-server* yang melibatkan *hypertext* (standar dokumen dalam pembuatan aplikasi *web*).

Penggunaan *http* yang khas adalah antara *web browser* dengan *web server*. Untuk menyediakan kehandalan dalam kinerjanya, *http* menggunakan koneksi TCP (*Transmission Control Protocol*), akan tetapi *http* merupakan suatu protokol yang tidak jelas, karena setiap transaksi yang ditangani diperlukan secara mandiri

(*independent),* dimana setiap transaksi yang dilakukan akan menciptakan suatu koneksi TCP baru antara *client* dengan *server,* kemudian memutuskan koneksi tersebut setelah transaksi dilengkapi.

Fungsi dari *http* bukanlah hanya sebagai protokol yang digunakan untuk mentransfer *hypertext,* melainkan protokol yang digunakan mentransmisikan informasi dengan tingkat efisiensi yang diperlukan untuk membuat lompatan *hypertext.* Data-data yang ditransmisikan oleh protokol ini dapat merupakan *text* asli, *hypertext,* audio, gambar atau informasi-informasi lainnya yang bisa diakses melalui *internet*. Kehandalan lain dari *http* adalah sifatnya yang sangat *flexibel* dalam format-format yang dapat ditangani, ketika seorang c*lient* mengajukan permintaan kepada *server*, maka *client* bisa memasukkan daftar prioritas format-format yang ditanganinya (Beletz, 2006).

2.7.3 Hypertex Markup Language (HTML)

HTML merupakan standar yang digunakan dalam *platform* tampilan dan isi dari sebuah dokumen pada halaman *web*. Dalam penggunaannya, html didefinisikan dalam *tag-tag* yang dipergunakan untuk mengatur tampilan dan isi dari dokumen *hypertext* tersebut seperti ukuran huruf, warna, *header* dokumen, pembuatan *link*, pengaturan paragraf, serta pembuatan tabel (Sunarprihartono, 2002).

Yang dapat dilakukan dengan HTML yaitu:

a) mengontrol tampilan dari *web page* dan kontennya,
b) mempublikasikan dokumen secara *online* sehingga bisa di akses dari seluruh dunia,

c) membuat *online form* yang bisa di gunakan untuk menangani pendaftaran, transaksi secara *online*,

d) menambahkan objek-objek seperti *image*, audio, video dan juga *java applet* dalam document HTML.

Contoh penulisan kode HTML :

```
<html>
<head>
<title>judul</title>
</head>
<body>
        kode-kode HTML
</body>
</html>
```

Pasangan *tag* <html> dan </html> menandakan bahwa kode yang terdapat didalamnya adalah kode yang diterjemahkan oleh *browser* sebagai dokumen HTML.

Bagian yang terdapat dalam <html> dan </html> terbagi atas :

1) kepala (*head*);

2) badan (*body*).

2.8 Personal Home Pages (PHP)

Menurut Sunarprihartono (2002), PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Maksud dari

server-side scripting adalah *syntax* dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada dokumen html. Pembuatan *web* ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman *web*.

PHP merupakan *script* untuk pemrograman. *Script web server side* adalah *script* untuk membuat dokumen *Hypertext Markup Language* (HTML). Dokumen yang dihasilkan melalui aplikasi bukan dokumen yang dibuat dengan editor teks. Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi mudah, proses *update* data dapat dilakukan menggunakan aplikasi yang disebut *script* PHP (Sidik, 2002).