

## **BAB III**

### **LANDASAN TEORI**

Landasan teori digunakan untuk menyelesaikan masalah secara sistematis. Bab ini akan membahas landasan teori yang meliputi hal-hal terkait dengan permasalahan yang ada dan landasan teori yang membahas tentang ilmu yang terkait dalam permasalahan tersebut.

#### **3.1 Konsep Dasar Sistem Informasi**

##### **3.1.1 Sistem**

Menurut Mulyadi (2008: 5), sistem adalah jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan. Sedangkan pengertian prosedur adalah suatu urutan kegiatan klerikal, biasanya melibatkan beberapa orang dalam satu departemen atau lebih, yang dibuat untuk menjamin penanganan secara transaksi perusahaan yang terjadi berulang-ulang.

##### **3.1.2 Informasi**

Menurut Kusri (2008:4), informasi merupakan data yang sudah diolah sedemikian rupa sehingga sesuai dengan yang dibutuhkan oleh penggunanya. Untuk memperoleh informasi yang berguna, pertama kali yang harus dilakukan adalah pengumpulan data, lalu diolah sehingga menjadi informasi. Ketika data telah menjadi suatu informasi, maka informasi tersebut akan menjadi terarah dan penting. Hal ini dikarenakan telah dilaluinya berbagai tahapan dalam

pengolahannya, yaitu pengumpulan data, data apa saja yang terkumpul, dan penemuan informasi yang diperlukan pengguna.

### 3.2 Internet

Sibero (2011a:10), Internet (Interconnected Network) adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet dapat juga dapat disebut jaringan alam suatu jaringan yang luas. Seperti halnya jaringan komputer lokal maupun jaringan komputer area, internet juga menggunakan protokol komunikasi yang sama yaitu TCP/IP (Transmission Control Protocol / Internet Protocol).

### 3.3 Aplikasi Web

Menurut Simarmata (2010), Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Fitur-fitur aplikasi web biasanya berupa data *persistence*, mendukung transaksi dan komposisi halaman web dinamis yang dapat dipertimbangkan sebagai hibridisasi antara hipermedia dan sistem informasi.

Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh *browser* web. *Client-side* mempunyai tanggung jawab untuk pengekseskuan proses bisnis.

Interaksi Web dibagi ke dalam tiga langkah yaitu:

1. Permintaan

Pengguna mengirimkan permintaan ke server web, via halaman web yang ditampilkan pada browser web.

2. Pemrosesan

Server web menerima permintaan yang dikirimkan oleh pengguna kemudian memproses permintaan tersebut.

3. Jawaban

*Browser* menampilkan hasil dari permintaan pada jendela *browser*. Halaman web bisa terdiri dari beberapa jenis informasi grafis (tekstual dari multimedia). Kebanyakan komponen grafis dihasilkan dengan tool khusus, menggunakan manipulasi langsung dan editor *WYSIWYG*.

### 3.4. Aplikasi Dekstop

Menurut Konixbam (2009) *Desktop Based Application* adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi *Internet* di suatu komputer otonom dengan *operating system* atau *platform* tertentu. Aplikasi *Desktop* difokuskan kepada aplikasi yang lebih independen. Hal tersebut bertujuan untuk mempermudah para pengguna aplikasi *desktop* dalam hal memodifikasi pengaturan aplikasi sehingga efektifitas, efesinsi waktu, dana, dan tenaga dapat lebih ditekankan semaksimal mungkin.

Secara garis besar pada pemrograman terutama pada aplikasi yang berbasis *desktop* dapat dibedakan menjadi dua macam yaitu pemrograman konvensional dan pemrograman visual.

1. Pemrograman konvensional merupakan metode mendesain suatu aplikasi, pemrograman dituntut untuk bisa menerapkan baris demi baris kode program agar bisa menghasilkan sebuah bentuk tampilan aplikasi yang dibuat dan akan memakan waktu lama.
2. Pemrograman visual merupakan metode pembuatan program dimana seorang programmer membuat koneksi antar objek-objek dengan cara

menggambar, menunjuk, dan mengklik pada diagram dan ikon dengan berinteraksi dengan diagram jalur.

Beberapa keunggulan yang dimiliki oleh aplikasi *desktop* yaitu:

1. Dapat berjalan dengan independen, tidak perlu menggunakan sebuah *web browser*.
2. Tidak memerlukan koneksi internet
3. Prosesnya lebih cepat dibanding aplikasi *web*

Kekurangannya yaitu :

1. Harus menginstal aplikasinya terlebih dahulu jika ingin menjalankannya
2. Bermasalah pada lisensi karena membutuhkan banyak lisensi pada setiap komputer yang berbeda-beda
3. Biasanya memerlukan *hardware* yang mempunyai kualitas yang baik atau cukup tinggi.




### 3.5 Website

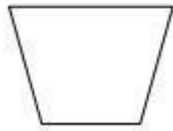
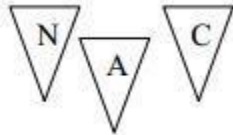

Menurut Yuhefizar dkk (2009;2), website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan. Selain itu, website dapat juga digunakan sebagai alat promosi, tetapi bukan sebagai alat promosi pertama.

Kelebihan website dibandingkan dengan media cetak maupun elektronik adalah kelengkapan informasi yang disajikan dengan biaya yang relatif murah. Kekurangannya adalah produk yang ditampilkan serta pasar yang dituju lebih *segmented* (terpusat pada kalangan/kelompok konsumen tertentu). Oleh karena itu, harus memanfaatkan kekurangannya menjadi *Strong Point* dalam pemasaran.

### 3.6 Diagram Alir Dokumen (*Document Flowchart*)

Menurut Jogiyanto (2005), diagram alir dokumen atau *paperwork flowchart* merupakan diagram alir yang menunjukkan arus laporan dan formulir beserta tembusannya. Berdasarkan pengertian di atas dapat disimpulkan bahwa diagram alir dokumen adalah diagram yang menggambarkan aliran seluruh dokumen. Diagram alir dokumen ini menggunakan simbol-simbol yang sama dengan diagram alir sistem. Diagram alir dokumen digambar dengan menggunakan simbol-simbol yang ada pada tabel berikut (Jogiyanto, 2005):

No	Nama Simbol	Simbol	Fungsi
1.	<b><i>Terminator</i></b>		Simbol ini digunakan untuk menunjukkan awal dan akhir suatu proses dokumen.
2.	<b><i>Document</i></b>		Simbol ini digunakan sebagai <i>input</i> dan <i>output</i> baik secara manual ataupun dengan menggunakan computer.
3.	<b><i>Manual Input</i></b>		Simbol ini berfungsi untuk memasukkan data dengan menggunakan <i>online keyboard</i> .

4.	<b>Manual Process</b>		Simbol ini menunjukkan kegiatan manual.
5.	<b>Offline Storage</b>		Simbol ini merupakan dokumen yang diarsip dan diurutkan berdasarkan N ( <i>numeric</i> ), A ( <i>alphabet</i> ), C ( <i>chronological</i> )
6.	<b>Flow</b>		Simbol ini digunakan sebagai arah aliran dokumen.

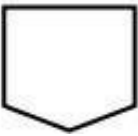
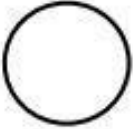

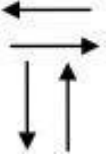
**Tabel 3.1** Simbol-simbol *Document Flowchart*

### 3.7 Diagram Alir Sistem (*System Flowchart*)

Diagram alir sistem merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan untuk mengolah data dan menghubungkan antar peralatan tersebut (Oetomo, 2002). Diagram alir sistem ini tidak digunakan untuk menggambarkan langkah-langkah dalam memecahkan masalah tetapi hanya menggambarkan prosedur pada sistem yang dibentuk.

#### 1. *Flow Direction Symbols*



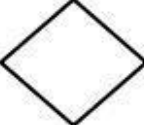
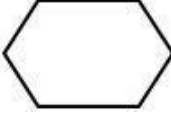


*Flow direction symbols* digunakan untuk menghubungkan antara satu simbol dengan simbol lainnya (Ladjamudin, 2005). Simbol ini disebut *connecting line*. Simbol-simbol tersebut dijelaskan pada tabel di bawah ini :

No	Nama Simbol	Simbol	Fungsi
1.	<i>Offline Connector</i>		Fungsi dari simbol ini adalah menyambungkan antara suatu proses dengan proses lainnya di halaman yang berbeda.
2.	<i>Connector</i>		Fungsi dari simbol ini adalah menyambungkan antara, suatu proses dengan proses lainnya di halaman yang sama.
3.	<i>Communication Link</i>		Fungsi dari simbol ini adalah mentransisi suatu data atau informasi dari setiap lokasi.
4.	<i>Flow</i>		Fungsi dari simbol ini adalah menyatakan jalannya arus suatu proses.

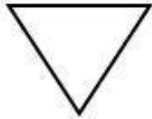

**Tabel 3.2** *Flow Direction Symbols*

## 2. *Processing Symbols*

*Processing symbols* merupakan simbol yang menunjukkan jenis operasi pengolahan data dalam suatu proses (Ladjamudin, 2005). Simbol-simbol tersebut dijelaskan pada tabel di bawah ini:

No.	Nama Simbol	Simbol	Fungsi
1.	<i>Offline Conector</i>		Simbol ini berfungsi untuk menyambungkan satu proses dengan proses lainnya di halaman yang berbeda.
2.	<i>Manual Process</i>		Simbol ini berfungsi untuk melakukan prosedur atau proses tanpa menggunakan komputer.
3.	<i>Decision</i>		Simbol ini berfungsi untuk melakukan pengecekan. Biasanya menghasilkan jawaban ya atau tidak.
4.	<i>Predefined Process</i>		Simbol ini berfungsi sebagai tempat penyimpanan nilai awal.
5.	<i>Terminal</i>		Simbol ini berfungsi untuk menyatakan permulaan atau penghentian suatu program.
6.	<i>Key Operation</i>		Simbol ini berfungsi untuk menyatakan suatu jenis operasi yang diproses dengan



			menggunakan mesin yang memiliki <i>keyboard</i> .
7.	<b><i>Offline Storage</i></b>		Simbol ini digunakan untuk menyimpan data ke suatu media tertentu.
8.	<b><i>Manual Input</i></b>		Simbol ini berfungsi untuk memasukkan data dengan menggunakan <i>online keyboard</i> .


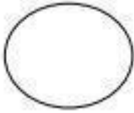

**Tabel 3.3** *Processing Symbols*

### 3.8 Diagram Konteks (*Context Diagram*)

Diagram konteks merupakan sebuah model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem (Whitten, 2004). Menurut Oetomo (2002), terdapat beberapa hal yang perlu diperhatikan dalam membuat diagram konteks, diantaranya:

1. Kelompok pemakai, baik internal maupun eksternal perusahaan.
2. Identifikasi kejadian-kejadian yang mungkin terjadi dalam penggunaan sistem.
3. Arah anak panah yang menunjukkan aliran data.
4. Setiap kejadian digambarkan dalam bentuk yang sederhana dan mudah dipahami oleh pembuat sistem.

Suatu diagram konteks hanya mengandung satu proses saja, biasanya diberi nomor proses 0. Proses ini mewakili proses dari seluruh sistem dengan dunia luarnya. Simbol-simbol yang digunakan dalam membuat diagram konteks digambarkan pada tabel di bawah ini:

No	Nama Simbol	Simbol	Fungsi
1.	<i>Terminator</i>		Simbol ini digunakan untuk berkomunikasi dengan sistem aliran data.
2.	<i>Process</i>		Simbol ini berfungsi untuk mewakili suatu aktifitas yang ada pada sistem.
3.	<i>Flow (Aliran data)</i>		Simbol ini digunakan untuk menunjukkan arah dari aliran

**Tabel 3.4** *Simbol-simbol Context Diagram*

### 3.9 Data Flow Diagram (DFD)

Menurut Whitten (2004), *Data Flow Diagram (DFD)* merupakan alat yang menggambarkan aliran data melalui sistem. Dalam pembuatan DFD, terdapat beberapa tingkatan yang bertujuan untuk menghindari aliran data yang rumit. Tingkatan tersebut dimulai dari tingkatan tertinggi ke bentuk yang lebih rinci. Tingkatan DFD terdiri atas:

#### 1. Diagram Konteks (*Context Diagram*)


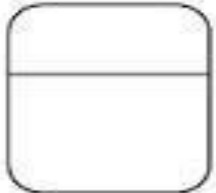

Diagram konteks merupakan sebuah model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem (Whitten, 2004).

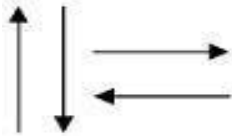
## 2. Diagram Rinci

Diagram rinci menggambarkan rincian dari proses yang ada pada tingkatan sebelumnya. Diagram ini merupakan diagram dengan tingkatan paling rendah dan tidak dapat diuraikan lagi.

## 3. Diagram Level 0

Diagram level 0 merupakan diagram aliran data yang menggambarkan sebuah *event* konteks. Diagram ini menunjukkan interaksi antara *input*, *output*, dan *data store* pada setiap proses yang ada (Nugroho, 2009).

Nama Simbol	Simbol	Keterangan
<i>External Entity</i>		<p><i>External entity</i> merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang akan memberikan <i>input</i> ataupun menerima <i>output</i>.</p>
<i>Process</i>		<p>Proses adalah kegiatan yang dilakukan oleh orang atau komputer dari arus data yang masuk untuk menghasilkan arus data yang keluar.</p>
<i>Data Store</i>		<p><i>Data store</i> merupakan tempat penyimpanan data yang berupa <i>file</i> maupun <i>database</i> di dalam sistem komputer.</p>

<b>Data Flow</b>		<p><i>Data flow</i> atau aliran data yang mengalir diantara proses. Aliran data dapat digambarkan dari bawah ke atas, kiri ke kanan, maupun sebaliknya.</p>
------------------	---	---

**Tabel 3.5** *Simbol-simbol DFD*

Menurut Gane dan Sarson (1979), DFD terdiri atas empat simbol.

Setiap simbol memiliki aturan tersendiri dalam penggunaannya. Aturan-aturan tersebut antara lain:

**1. *External Entity (Entitas Luar)***

Aturan penggunaan untuk *external entity* antara lain:

- a. Data harus bergerak melalui proses, selama data tersebut berhubungan dengan sistem. Jika data tidak berhubungan dengan proses, maka aliran data tidak perlu ditampilkan pada DFD.
- b. Entitas luar diberi label dengan sebuah frase kata benda.

**2. *Process***

Aturan penggunaan sebuah proses antara lain:

- a. Sebuah proses tidak hanya memiliki output. Jika sebuah objek hanya memiliki output, maka objek tersebut adalah source.
- b. Sebuah proses tidak hanya memiliki input. Jika sebuah objek hanya memiliki sebuah input, maka objek tersebut adalah entitas luar.
- c. Sebuah proses diberi label dengan sebuah frase kata kerja.

**3. *Data Store***

Aturan-aturan dalam menggunakan *data store* adalah sebagai berikut:

- a. Data harus bergerak melalui proses dimana data diterima melalui suatu *source* untuk disimpan di *data store*.
- b. Data tidak dapat bergerak langsung dari *data source* menuju *external entity*.
- c. *Data store* diberi label dengan frase kata benda.

#### 4. **Data Flow (Aliran Data)**

Aturan-aturan dalam menggunakan *data flow* antara lain:

- a. Sebuah aliran data hanya menggunakan satu arah antar simbol.
- b. Sebuah cabang pada aliran data memiliki arti data yang sama dari satu lokasi menuju ke satu atau lebih proses, tempat penyimpanan data, serta entitas luar.
- c. Sebuah aliran data tidak dapat bergerak ke proses asalnya sehingga membutuhkan proses lain untuk menangani, menghasilkan, dan mengembalikan aliran data ke proses asal.
- d. Aliran data atau *data flow* diberi label dengan frase kata benda.

### 3.10 Entity Relationship Diagram (ERD)

Pengertian *Entity Relation Diagram* (ERD) menurut Jogiyanto (2001) adalah suatu komponen himpunan entitas dan relasi yang dilengkapi dengan atribut yang mempresentasikan seluruh fakta. ERD digunakan untuk menggambarkan model hubungan data dalam sistem yang di dalamnya terdapat hubungan entitas berserta atribut relasinya serta mendokumentasikan kebutuhan sistem untuk pemrosesan data. ERD memiliki 4 jenis objek, antara lain:

## 1. Entity

Entitas adalah kelompok orang, tempat, objek, kejadian atau konsep tentang apa yang diperlukan untuk menyimpan data (Whitten, 2004). Setiap entitas yang dibuat memiliki tipe untuk mengidentifikasi apakah entitas tersebut bergantung dengan entitas lainnya atau tidak. Tipe entitas merupakan kumpulan objek yang memiliki kesamaan properti yang teridentifikasi oleh perusahaan dan memiliki keberadaan yang independen (Connolly & Begg, 2002). Tipe entitas terdiri atas dua jenis, yaitu:

### a. *Strong Entity*

*Strong entity* adalah tipe entitas yang tidak bergantung pada keberadaan jenis entitas lainnya. Suatu entitas dikatakan kuat apabila tidak tergantung pada entitas lainnya.

### b. *Weak Entity*

*Weak Entity* adalah tipe entitas yang bergantung pada keberadaan jenis entitas lain yang saling berhubungan. Karakteristik *weak entity* terletak pada entitas *occurrence* yang tidak dapat teridentifikasi secara unik. Entitas *occurrence* adalah sebuah objek yang secara unik dapat teridentifikasi dengan tipe entitas (Connolly & Begg, 2002).

## 2. Attribute

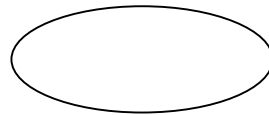
Menurut Connolly dan Begg (2002) atribut adalah deskripsi data yang mengidentifikasikan dan membedakan suatu entitas dengan entitas lainnya. Setiap atribut memiliki domain untuk mendefinisikan nilai-nilai potensial yang dapat menguatkan atribut. Atribut domain adalah kumpulan nilai-nilai

yang diperbolehkan untuk satu atau lebih atribut (Connolly & Begg, 2002).

Atribut dapat dibedakan menjadi 5 jenis, yaitu:

a. ***Simple Attribute***

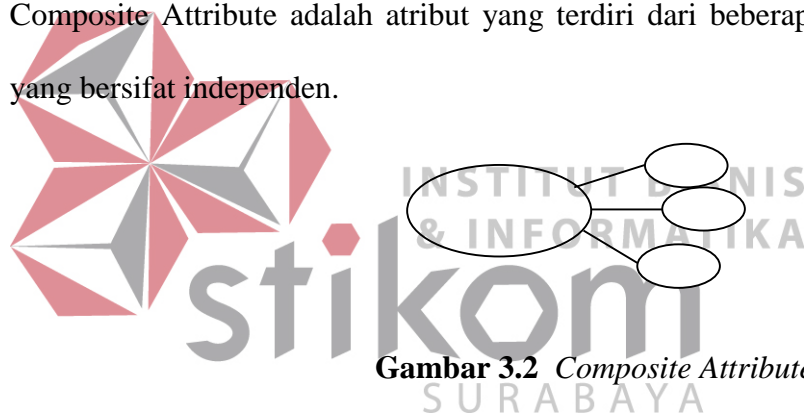
Simple Attribute adalah atribut yang terdiri dari komponen tunggal. Sempel atribut tidak dapat dibagi menjadi komponen yang lebih kecil.



**Gambar 3.1** *Simple Attribute*

b. ***Composite Attribute***

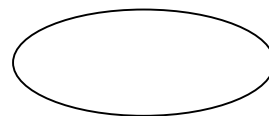
Composite Attribute adalah atribut yang terdiri dari beberapa komponen yang bersifat independen.



**Gambar 3.2** *Composite Attribute*

c. ***Single-value Attribute***

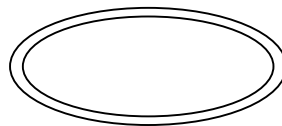
Single-value Attribute adalah atribut yang memegang nilai tunggal dari suatu entitas.



**Gambar 3.3** *Single-value Attribute*

d. ***Multi-value Attribute***

Muti-value Attribute adalah atribut yang dapat memegang nilai lebih dari suatu entitas.



**Gambar 3.4** *Multi-value Attribute*

e. ***Derived Attribute***

Derived Attribute adalah atribut yang mewakili turunan nilai sebuah atribut yang saling berkaitan dan belum tentu dalam tipe entitas yang sama.



**Gambar 3.5** *Derived Attribute*

3. **Keys**

Menurut Connolly dan Begg (2002) *keys* terdiri atas beberapa jenis, yaitu:

a. ***Candidate Key***

*Candidate key* merupakan *set* minimal dari suatu atribut yang secara unik mengidentifikasi setiap *occurrence* dari tipe entitas. *Candidate key* tidak boleh *null* (kosong).

b. ***Primary Key***

Sebuah *candidate key* yang dipilih untuk mengidentifikasi secara unik tiap kejadian pada suatu entitas. *Primary key* harus bernilai *unique* dan tidak boleh *null* (kosong).

c. ***Composite Key***



Sebuah *candidate key* yang mempunyai dua atribut atau lebih. Suatu atribut yang membentuk *composite key* bukanlah kunci sederhana karena *composite key* tidak membentuk kunci senyawa.

d. ***Alternate Key***

Sebuah *candidate key* yang tidak menjadi *primary key*. *Key* ini biasa disebut dengan *secondary key*.

e. ***Foreign Key***

Himpunan atribut dalam suatu relasi yang cocok dengan *candidate key* dari beberapa relasi lainnya. *Foreign key* mengacu pada *primary key* suatu tabel.

Nilai *foreign key* harus sesuai dengan nilai *primary key* yang diacunya.

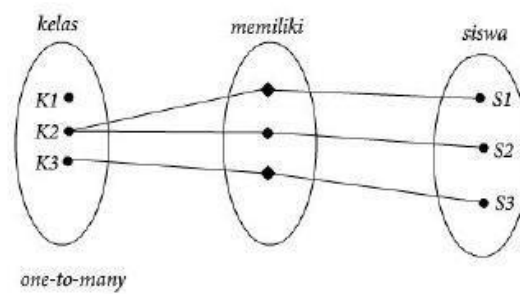
4. **Relationship**

Menurut Whitten (2004) *relationship* adalah asosiasi bisnis alami antara satu entitas atau lebih. Dalam suatu relasi, entitas yang saling berelasi memiliki kata kerja aktif yang menunjukkan bahwa keduanya saling berelasi satu sama lain.

Relasi terdiri atas enam tipe, yaitu:

a. **Relasi *one to many***

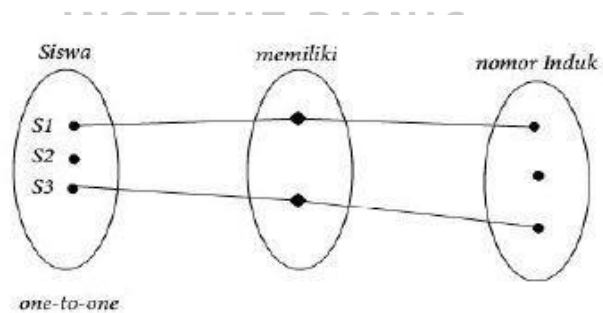
Relasi *one to many* berarti suatu entitas himpunan A dapat berhubungan dengan banyak entitas pada entitas himpunan B, namun tidak sebaliknya. Di bawah ini adalah contoh tipe relasi *one to many*, yaitu satu kelas dapat memiliki banyak siswa.



**Gambar 3.6** Tipe relasi *one to many*

b. **Relasi *one to one***

Relasi *one to one* berarti setiap entitas himpunan A hanya berhubungan dengan satu entitas himpunan B, begitu juga sebaliknya. Di bawah ini adalah contoh tipe relasi *one to one*, yaitu satu siswa pasti hanya memiliki satu nomor induk.



**Gambar 3.7** Tipe relasi *one to one*

c. **Relasi rekursif *one to one***

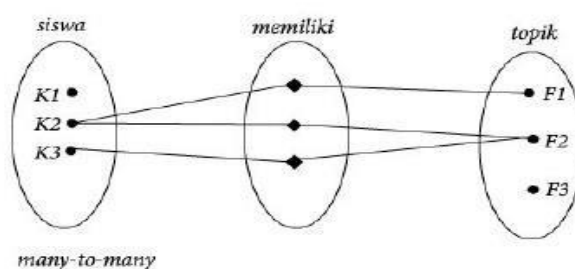
Relasi rekursif *one to one* adalah sebuah tipe relasi yang dimana entitasnya berpartisipasi lebih dari satu peran.

d. **Relasi *superclass/subclass***

Untuk setiap relasi *superclass* / *subclass*, entitas *superclass* diidentifikasi sebagai entitas induk dan entitas *subclass* sebagai anggotanya.

e. **Relasi *many to many***

Relasi *many to many* berarti setiap entitas himpunan A dapat berhubungan dengan entitas pada himpunan B, begitu juga sebaliknya. Sebagai contoh pada Gambar 3.8, yaitu satu siswa memiliki banyak topik dan setiap topik dapat dilihat atau dimiliki banyak siswa.



**Gambar 3.8** Tipe relasi *many to many*

f. **Relasi kompleks**

Relasi kompleks adalah tipe relasi yang dimana satu entitas berhubungan dengan entitas lainnya yang dapat membentuk sirkulasi dalam relasi tersebut.

### 3.11 HTML

Menurut Prasetio (2010;52), HTML adalah bahasa yang mempelopori hadirnya web dan internet. Bahasa ini merupakan bahasa pemrograman yang digunakan oleh sebagian besar situs web yang dikunjungi oleh setiap orang. HTML saat ini dikenal oleh hampir semua komputer yang ada di dunia dan merupakan cara paling *universal* untuk membuat sebuah dokumen. HTML tidak memiliki variasi format terbaik dan bahkan tidak menjamin bahwa halaman web yang dibuat akan

tampak sama persis di setiap browser, tetapi perlu diingat bahwa tanpa HTML, tidak akan ada internet.

### 3.12 PHP Hypertext Preprocessor

*PHP Hypertext Preprocessor* (PHP) adalah bahasa pemrograman scripting sisi server (*server-side*), bahasa pemrograman yang digunakan oleh server web untuk menghasilkan dokumen *Hypertext Markup Language* (HTML) *on-the-fly* (sidik, 2005: 323).

Menurut Kaidir (2008: 2), PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu penampilan berdasarkan permintaan terkini. Misalnya, bisa menampilkan *database* ke halaman web. Pada prinsip PHP mempunyai fungsi yang sama dengan skrip-skrip seperti *Active Server Page* (ASP), *Cold Fusion*, atau *perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, Skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

### 3.13 PhpMyAdmin

Menurut Firdaus (2007: 15) PhpMyAdmin adalah suatu program open source yang berbasis web yang dibuat menggunakan aplikasi PHP. Program ini digunakan untuk mengakses database MySQL. Program ini mempermudah dan mempersingkat kerja penggunanya. Dengan kelebihanannya, para pengguna awam tidak harus paham sintak-sintak SQL dalam pembuatan database dan tabel.

### 3.14 XAMPP

Menurut (Widijanuarto, 2010), Xampp adalah sebuah aplikasi yang berisi MySQL dan Apache. Aplikasi ini dapat membantu untuk membangun sebuah aplikasi berbasis web tanpa harus menginstal MySQL dan Apache secara terpisah/sendiri-sendiri.

