

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 Kusriani (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 Biaya Operasional Sekolah

Biaya operasional sekolah adalah biaya yang diperlukan sekolah untuk menunjang proses pendidikan. Biaya operasional terdiri dari biaya personalia dan biaya nonpersonalia. Biaya personalia mencakup: gaji dan tunjangan yang melekat pada gaji yaitu tunjangan struktural, tunjangan fungsional, tunjangan profesi, dan tunjangan-tunjangan lain yang melekat dalam jabatannya. Biaya non personalia, antara lain biaya untuk: alat tulis sekolah, bahan dan alat habis pakai, yang habis dipakai dalam waktu satu tahun atau kurang, pemeliharaan dan perbaikan ringan, daya dan jasa transportasi/perjalanan dinas, konsumsi, asuransi, pembinaan siswa/ekstrakurikuler.

3.2.1 Pengelolaan BOP Pada SD Praja Mukti Surabaya

Pada SD Praja Mukti pengelolaan biaya operasional pendidikan dilakukan oleh Bagian Tata Usaha yaitu merekap siapa saja siswa yang sudah membayar dan belum membayar biaya operasional pendidikan pada SD Praja Mukti Surabaya.

3.2.2 Tahapan Pembayaran BOP Pada SD Praja Mukti Surabaya

Pada tahapan pembayaran BOP ini pertama Bagian Bendahara mencetak kartu investasi kemudian diberikan kepada wali kelas masing-masing yang nantinya akan dibagikan kepada siswa. Setelah siswa menerima kartu investasi tersebut kemudian siswa membayar BOP kepada Bagian Tata Usaha, yang kemudian oleh Bagian Tata Usaha akan direkapitulasi pembayaran BOP siswa tersebut.

3.3 *Internet*

Menurut Simarmata (2010), *Internet (Interconneted Network)* adalah jaringan komputer yang menghubungkan antar jaringan secara global, *internet* bisa juga disebut jaringan alam atau suatu jaringan yang luas. Seperti halnya jaringan komputer lokal maupun jaringan komputer area, *internet* juga menggunakan protokol komunikasi yang sama yaitu *Tranmission Control Protocol/Internet Protocol (TCP/IP)*.

3.4 *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 pengeksekusian 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*.

3.5 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 3.1 berikut (Jogiyanto, 2005).

Tabel 3.1 *Document Flowchart Symbols*

| No | Nama Simbol | Simbol | Fungsi |
|----|------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Terminator |  | Simbol ini digunakan untuk menunjukkan awal dan akhir suatu proses dokumen. |
| 2. | Document |  | Simbol ini digunakan sebagai <i>input</i> dan <i>output</i> baik secara manual ataupun dengan menggunakan komputer. |
| 3. | Manual Input |  | Simbol ini berfungsi untuk memasukkan data dengan menggunakan <i>online keyboard</i> . |
| 4. | Manual Process |  | Simbol ini menunjukkan kegiatan manual. |
| 5. | Offline Storage |  | Simbol ini merupakan dokumen yang diarsip dan diurutkan berdasarkan N (<i>numeric</i>), A (<i>alphabet</i>), C (<i>chronological</i>) |
| 6. | Flow |  | Simbol ini digunakan sebagai arah aliran dokumen. |

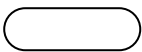


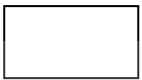
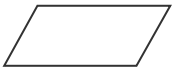




3.7 Diagram Alir Sistem (*System Flowchart*)

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. Diagram alir sistem digambar dengan menggunakan simbol-simbol yang ada pada Tabel 3.2 berikut.


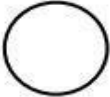

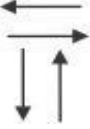
Tabel 3.2 *System Flowchart Symbols*

| No | Nama Simbol | Simbol | Fungsi |
|----|----------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| 1 | Terminator |  | Permulaan atau akhir program |
| 2 | <i>Flowline</i> |  | Arah alir program |
| 3 | <i>Preparation</i> |  | Proses inisialisasi/pemberian harga awal |
| 4 | Proses |  | Proses perhitungan atau proses pengolahan data |
| 5 | <i>Input / output data</i> |  | Proses <i>input</i> atau <i>output</i> data, parameter informasi |
| 6 | <i>Predifined process</i> |  | Permulaan sub program atau proses menjalankan sub program |
| 7 | <i>Decision</i> |  | Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya. |
| 8 | <i>On page connector</i> |  | Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman. |
| 9 | <i>Off page connector</i> |  | Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda. |

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 3.3 berikut.

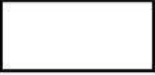

Tabel 3.3 *Flow Direction Symbols*

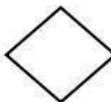
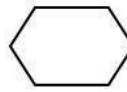
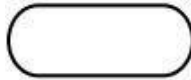

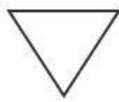

| No | Nama Simbol | Simbol | Fungsi |
|----|---------------------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 1. | <i>Offline Connector</i> |  | 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. |

2. *Processing Symbols*

Processing symbols merupakan simbol yang menunjukkan jenis operasi pengolahan data dalam suatu proses (Ladjamudin, 2005). Simbol-simbol tersebut dijelaskan pada Tabel 3.4 berikut.

Tabel 3.4 *Processing Symbols*

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

3.8 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:

3.8.1 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 3.5 berikut.

Tabel 3.5 *Context Diagram Symbols*

| 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 |


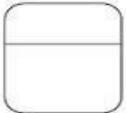


3.8.2 Diagram Rinci

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

3.8.3 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, 2007).

Tabel 3.6 DFD Symbols

| No | Nama Simbol | Simbol | Keterangan |
|----|------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <i>External Entity</i> |  | <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> . |
| 2 | <i>Process</i> |  | Proses adalah kegiatan yang dilakukan oleh orang atau komputer dari arus data yang masuk untuk menghasilkan arus data yang keluar. |
| 3 | <i>Data Store</i> |  | <i>Data store</i> merupakan tempat penyimpanan data yang berupa <i>file</i> maupun <i>database</i> di dalam sistem komputer. |
| 4 | <i>Data Flow</i> |  | <i>Data flow</i> atau aliran data yang mengalir diantara proses. Aliran data dapat digambarkan dari bawah ke atas, kiri ke kanan, maupun sebaliknya. |

Menurut Gane dan Sarson (1979), DFD terdiri atas empat simbol. Seperti pada Tabel 3.6 di atas. Setiap simbol memiliki aturan tersendiri dalam penggunaannya. Aturan-aturan tersebut antara lain:

A. *External Entity* (Entitas Luar)

External Entity atau Entitas Luar adalah entitas yang berada di luar sistem yang memberikan data kepada sistem (*source*) atau yang menerima informasi dari

sistem (*sink*). Entitas Luar ini bukanlah bagian dari sistem, bila suatu sistem informasi dirancang untuk satu bagian/departemen maka bagian lain yang masih terkait menjadi entitas luar. Aturan penggunaan untuk *external entity* antara lain:

1. 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.
2. Entitas luar diberi label dengan sebuah frase kata benda.

B. Process

Process adalah urutan pelaksanaan atau kejadian yang terjadi secara alami atau didesain, mungkin menggunakan waktu, ruang, keahlian atau sumber daya lainnya, yang menghasilkan suatu hasil. Suatu proses mungkin dikenali oleh perubahan yang diciptakan terhadap sifat-sifat dari satu atau lebih objek di bawah pengaruhnya. Aturan penggunaan sebuah proses antara lain:

1. Sebuah proses tidak hanya memiliki *output*. Jika sebuah objek hanya memiliki *output*, maka objek tersebut adalah *source*.
2. Sebuah proses tidak hanya memiliki *input*. Jika sebuah objek hanya memiliki sebuah *input*, maka objek tersebut adalah entitas luar.
3. Sebuah proses diberi label dengan sebuah frase kata kerja.

C. Data Store

Data Store adalah sebuah *database* untuk transaksi sistem pengolahan yang menggunakan konsep-konsep gudang data untuk menyediakan data yang bersih untuk TPS (*Transaction Processing System*). Itu membawa konsep-konsep

dan manfaat dari gudang data ke bagian operasional bisnis. Aturan-aturan dalam menggunakan *data store* adalah sebagai berikut:

1. Data harus bergerak melalui proses di mana data diterima melalui suatu *source* untuk disimpan di *data store*.
2. Data tidak dapat bergerak langsung dari *data source* menuju *external entity*.
3. *Data store* diberi label dengan frase kata benda.

D. Data Flow (Aliran Data)

Data Flow adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data pada suatu sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. Aturan-aturan dalam menggunakan *data flow* antara lain:

1. Sebuah aliran data hanya menggunakan satu arah antar simbol.
2. 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.
3. Sebuah aliran data tidak dapat bergerak ke proses asalnya sehingga membutuhkan proses lain untuk menangani, menghasilkan, dan mengembalikan aliran data ke proses asal.
4. Aliran data atau *data flow* diberi label dengan frase kata benda.

3.9 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 beserta atribut relasinya serta mendokumentasikan kebutuhan sistem untuk pemrosesan data. ERD memiliki empat jenis objek, antara lain:

3.9.1 *Entity*

Entity 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).

3.9.2 Attribute

Menurut Connolly dan Begg (2002) atribut adalah deskripsi data yang mengidentifikasi 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 lima jenis, yaitu:

A. Simple Attribute

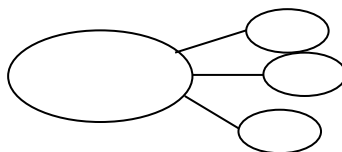
Simple Attribute adalah atribut yang terdiri dari komponen tunggal. *Simple Attribute* tidak dapat dibagi menjadi komponen yang lebih kecil.



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

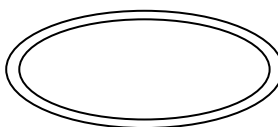
Single-value Attribute adalah atribut yang memegang nilai tunggal dari suatu entitas.



Gambar 3.3 *Single-value Attribute*

D. *Multi-value Attribute*

Multi-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*

3.9.3 *Keys*

Key adalah satu gabungan dari beberapa atribut yang dapat membedakan semua basis data (row) dalam tabel secara unik. *Key* di dalam *database* berfungsi sebagai suatu cara untuk mengidentifikasi dan menghubungkan satu tabel data dengan tabel yang lain. 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.

3.9.4 *Relationship*

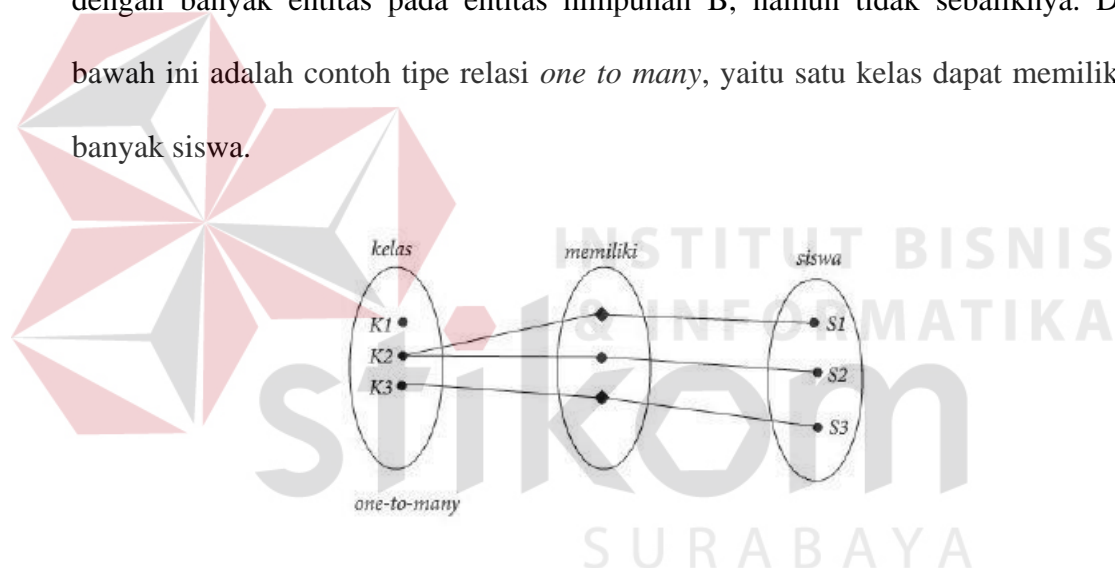
Menurut Whitten (2004) *relationship* adalah asosiasi bisnis alami antara satu entitas atau lebih. Dalam suatu relasi, entitas yang saling berelasi memiliki kata kerja aktif yang menunjukkan bahwa keduanya saling berelasi satu sama lain.

3.10 Relation

Relation adalah hubungan antara tabel yang mempresentasikan hubungan antar objek di dunia nyata. Relasi merupakan hubungan yang terjadi pada suatu tabel dengan lainnya yang mempresentasikan hubungan antar objek di dunia nyata dan berfungsi untuk mengatur mengatur operasi suatu *database*. Relasi terdiri atas enam tipe, yaitu:

A. Relasi *one to many*

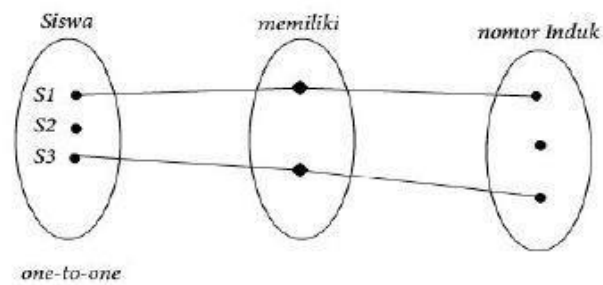
Relasi *one to many* berarti suatu entitas himpunan A dapat berhubungan dengan banyak entitas pada entitas himpunan B, namun tidak sebaliknya. Di bawah ini adalah contoh tipe relasi *one to many*, yaitu satu kelas dapat memiliki banyak siswa.



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*

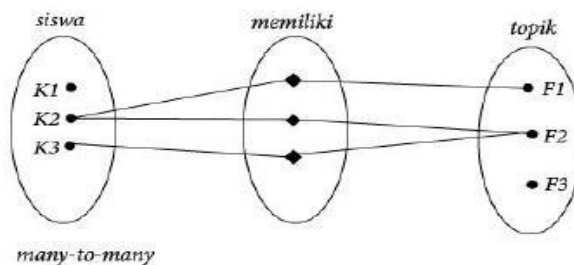
Relasi rekursif *one to one* adalah sebuah tipe relasi yang di mana entitasnya berpartisipasi lebih dari satu peran.

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 di mana satu entitas berhubungan dengan entitas lainnya yang dapat membentuk sirkulasi dalam relasi tersebut.

3.11 HTML

Menurut Prasetio (2010: 52), HTML atau *Hyper Text Markup Language* 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 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

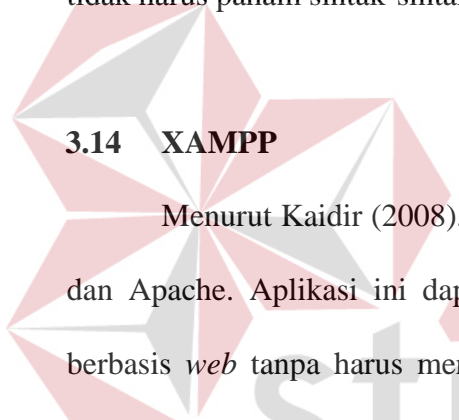
bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

3.13 PhpMyAdmin

Menurut Kaidir (2008), 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 Kaidir (2008), 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 atau sendiri-sendiri.



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA