

BAB III

LANDASAN TEORI

3.1 Pengertian Sekolah

Sekolah adalah tempat didikan bagi anak-anak. Tujuan dari sekolah adalah mengajarkan anak untuk menjadi anak yang mampu memajukan bangsa dan dalam membangun sebuah karakteristik kepada para generasi muda. Sekolah adalah sebuah lembaga yang dirancang untuk pengajaran siswa di bawah pengawasan guru yang dimana guru tersebut harus memiliki strategi pembelajaran.

Trianto (2010) Pembelajaran merupakan aspek kegiatan manusia yang kompleks, yang tidak sepenuhnya dapat dijelaskan. Pembelajaran secara simpel dapat diartikan sebagai produk interaksi berkelanjutan antara pengembangan dan pengalaman hidup. Pembelajaran dalam makna kompleks adalah usaha sadar dari seorang guru untuk membelajarkan siswanya (mengarahkan interaksi siswa dengan sumber belajar lainnya) dalam rangka mencapai tujuan yang diharapkan.

UU No. 20 Tahun 2003 tentang Sisdiknas Pasal 1 Ayat 20 “Pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar”.

Dari kedua pengertian tersebut dapat disimpulkan bahwa strategi pembelajaran adalah suatu kegiatan perencanaan yang dilaksanakan secara antara guru dan siswa untuk mencapai tujuan pembelajaran secara

optimal yang memungkinkan siswa dapat menangkap materi yang diberikan.

3.2 Presensi

Kehadiran siswa di sekolah biasa disebut dengan istilah presensi siswa. Presensi adalah pencatatan dan pengolahan data presensi yang dilakukan secara terus-menerus. Pencatatan dilakukan setiap hari sekolah dan dilakukan pelaporan kepada pihak Kesiswaan (Departemen Pendidikan dan Kebudayaan, 2006). Kehadiran dan ketidakhadiran siswa di sekolah dianggap merupakan masalah penting dalam pengelolaan siswa di sekolah, karena hal ini sangat erat hubungannya dengan prestasi belajar siswa.

Di samping itu, kehadiran dan ketidakhadiran siswa di sekolah merupakan gambaran tentang ketertiban suatu sekolah. Pengertian kehadiran di sekolah bukan hanya berarti siswa secara fisik ada di sekolah, melainkan yang lebih penting ialah keterlibatan siswa dalam kegiatan-kegiatan di sekolah.

Daftar presensi atau daftar hadir dimaksudkan untuk mengetahui frekuensi kehadiran siswa di sekolah sekaligus untuk mengontrol kerajinan belajar mereka. Tugas guru atau petugas yang ditunjuk adalah memeriksa dan memberikan tanda tentang hadir atau tidaknya seorang siswa satu kali dalam sehari. Pada proses belajar mengajar daftar kehadiran ini sangat penting karena daftar kehadiran tersebut sering kali

dijadikan sebagai bukti apakah siswa atau pengajar yang mengikuti proses belajar mengajar tersebut menghadiri kegiatan tersebut atau tidak.

3.3 *System Development Life Cycle (SDLC)*

Menurut Roger S. Pressman (2001), model ini biasa disebut juga dengan model waterfall atau disebut juga classic life cycle. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah Requirements (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian) dan *Maintenance* (perawatan).

a. *Requirement* (Analisis Kebutuhan Sistem)

Pada tahap awal ini dilakukan analisa guna menggali secara mendalam kebutuhan yang akan dibutuhkan. Kebutuhan ada bermacam-macam seperti halnya kebutuhan informasi bisnis, kebutuhan data dan kebutuhan *user* itu sendiri. Kebutuhan itu sendiri sebenarnya dibedakan menjadi tiga jenis kebutuhan. Pertama tentang kebutuhan teknologi. Dari hal ini dilakukan analisa mengenai kebutuhan teknologi yang diperlukan dalam pengembangan suatu sistem, seperti halnya data penyimpanan informasi. Kedua kebutuhan informasi, contohnya seperti informasi mengenai visi dan misi perusahaan, sejarah perusahaan, latar belakang perusahaan. Ketiga, Kebutuhan *user*. Dalam hal ini dilakukan analisa terkait kebutuhan *user* dan kategori *user*.

b. *Design* (Perancangan)

Selanjutnya, hasil analisa kebutuhan sistem tersebut akan dibuat sebuah design database, DFD, ERD, antarmuka pengguna/*Graphical User Interface* (GUI) dan jaringan yang dibutuhkan untuk sistem. Selain itu juga perlu dirancang struktur datanya, arsitektur perangkat lunak, detil prosedur dan karakteristik tampilan yang akan disajikan. Proses ini menterjemahkan kebutuhan sistem ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum memulai tahap implementasi.

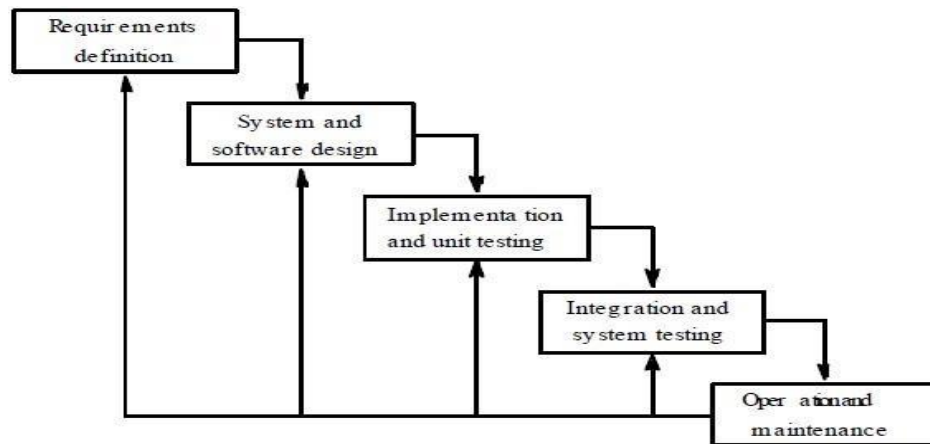
c. *Implementation* (Coding)

Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi hasil perancangan sebelumnya ke dalam sebuah bahasa pemrograman yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

d. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur ataukah tidak dan memastikan sistem terhindar dari error yang terjadi. Testing juga dapat digunakan untuk memastikan kevalidan dalam proses input, sehingga dapat menghasilkan output yang sesuai.

Komponen dari *System Development Life Cycle* (SDLC) ini seperti yang terlihat pada gambar 3.1 berikut ini :



Gambar 3.1 Tahapan Metode Waterfall menurut Roger S. Pressman

3.4 Aspek Dalam Administrasi Kelas

Aspek-aspek yang perlu diperhatikan dalam manajemen kelas yang baik adalah meliputi sifat kelas, pendorong kekuatan kelas, situasi kelas, tindakan efektif dan kreatif (Maman,1999).

Secara lebih terperinci kegiatan-kegiatan yang perlu dilaksanakan guru dalam memanajemen kelas sebagai aspek-aspek manajemen kelas yang tertuang dalam petunjuk pengelolaan kelas adalah.

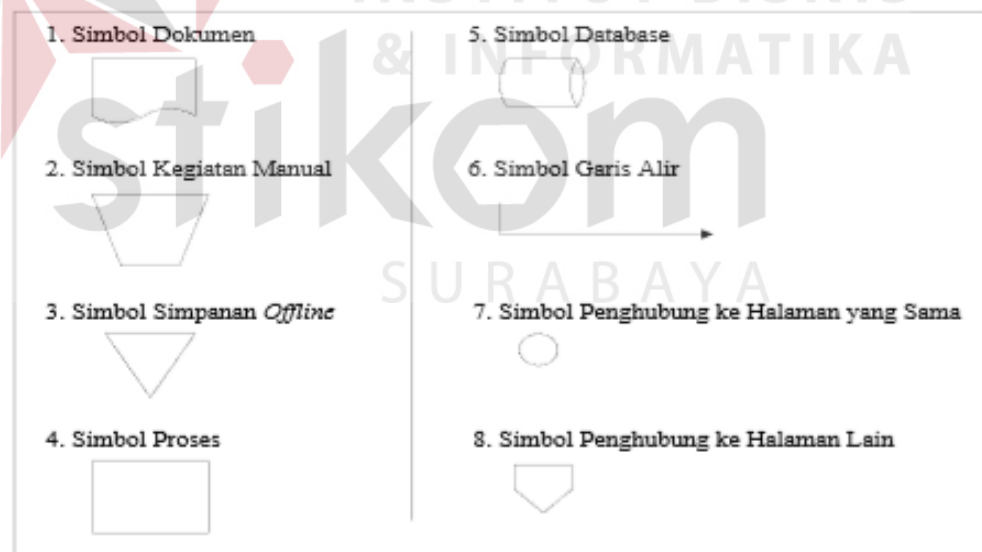
1. Mengecek kehadiran siswa.
2. Mengumpulkan hasil pekerjaan siswa, memeriksa dan menilai hasil pekerjaan tersebut.
3. Pendistribusian bahan dan alat secara adil dan proporsional kepada setiap siswa untuk melakukan praktik atau menggunakan alat dan bahan dalam proses belajarnya.
4. Mengumpulkan informasi dari siswa.
5. Mencatat data-data siswa yang menyangkut individu maupun maupun pekerjaan.

6. Pemeliharaan arsip tentang kegiatan dalam kelas sebagai tanggungjawab bersama sehingga dapat memberikan informasi baik bagi guru maupun bagi siswa.
7. Memberikan tugas/PR.

3.5 System Flow

Menurut J.E. Kendall (2003) *System flow* atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan urutan-urutan dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dikerjakan sistem. Simbol-simbol yang digunakan dalam system flow ditunjukkan pada

Gambar 3.2



Gambar 3.2. Simbol-simbol pada System Flow

1	Simbol dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual atau komputer.
2	Simbol kegiatan manual	Menunjukkan pekerjaan manual.
3	Simbol simpanan <i>offline</i>	Menunjukkan file non-komputer yang diarsip.
4	Simbol proses	Menunjukkan kegiatan proses dari operasi program komputer.
5	Simbol <i>database</i>	Menunjukkan tempat untuk menyimpan data hasil operasi komputer.
6	Simbol garis alir	Menunjukkan arus dari proses.
7	Simbol penghubung	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.

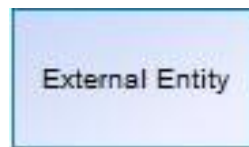
3.6 Data Flow Diagram (DFD)

Menurut J.E. Kendall (2003), DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik agar data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas.

3.6.1 Simbol-simbol yang digunakan dalam DFD

1. *External Entity*

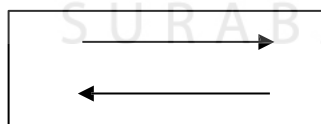
Sebuah elemen sistem atau sistem yang lain yang menghasilkan informasi bagi transformasi oleh perangkat lunak, atau menerima informasi yang dihasilkan oleh perangkat lunak. Simbol *External Entity* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Simbol *External Entity*

2. *Data Flow*

Data Flow atau aliran data disimbolkan dengan tanda panah. Data Flow menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Symbol Data Flow dapat dilihat pada Gambar 3.4.



Gambar 3.4 Simbol *Data Flow*

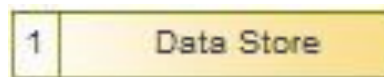
3. *Process*

Mempresentasikan sebuah proses atau transformasi yang diaplikasikan ke data untuk mengubahnya dengan berbagai macam cara. Simbol *Proces* dapat dilihat pada Gambar 3.5.

Gambar 3.5 Simbol *Proses*

4. *Data Store*

Data Store adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Simbol *Data Store* dapat dilihat pada Gambar 3.6.

Gambar 3.6 Simbol *Data Store*

3.6.2 *Context Diagram*

Context Diagram merupakan langkah pertama dalam pembuatan DFD. Pada context diagram dijelaskan sistem yang dibuat dan *eksternal entity* yang terlibat. Dalam context diagram harus ada arus data yang masuk dan arus data yang keluar.

3.6.3 *Data Flow Diagram Level 0*

DFD level 0 adalah langkah selanjutnya setelah context diagram. Pada langkah ini, digambarkan proses-proses yang terjadi dalam Aplikasi.

3.6.4 Data Flow Diagram Level 1

DFD Level 1 merupakan penjelasan dari DFD level 0. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD level 0.

3.7 Entity Relational Diagram

Menurut J.E. Kendall (2003), Entity Relational Diagram (ERD) merupakan penggambaran hubungan antara beberapa entity yang digunakan untuk merancang database yang akan diperlukan.

3.8 Adobe Dreamweaver

Menurut Sibero, (2011), *Dreamweaver* merupakan sebuah produk *web developer* yang dikembangkan oleh *Adobe Systems Inc.*, sebelumnya produk *Dreamweaver* dikembangkan oleh *Macromedia Inc.*, yang kemudian sampai saat ini perkembangannya diteruskan oleh *Adobe Systems Inc.*, *Dreamweaver* dikembangkan dan dirilis dengan kode nama *Creative Suit (CS)*.

Menurut Sigit (2010), “*Dreamweaver* adalah sebuah *HTML editor profesional* untuk mendesain web secara visual dan mengelola situs atau halaman web”.

Menurut Wahana Komputer (2010), ”Adobe Dreamweaver merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah website, baik secara grafis maupun dengan menuliskan kode sumber secara langsung”.

Berdasarkan definisi di atas, maka dapat disimpulkan dreamweaver adalah suatu aplikasi yang digunakan dalam membangun atau membuat sebuah *web*.

3.9 *Hypertext Preprocessor (PHP)*

Mengacu pada pendapat Nixon (2009), PHP adalah *server-side scripting language* yang awalnya dirancang untuk pengembangan web untuk menghasilkan halaman web yang dinamis. Untuk tujuan ini, kode PHP tertanam ke dokumen sumber HTML dan diterjemahkan oleh sebuah web server dengan PHP prosesor modul, yang menghasilkan dokumen halaman website. PHP dapat digunakan di sebagian besar web server dan sebagai juru mandiri, pada hampir setiap sistem operasi dan platform secara gratis.

PHP adalah bahasa scripting umum yang terutama cocok untuk pengembangan web sisi server di mana PHP umumnya berjalan pada web server. Kode PHP di *file* yang diminta dilaksanakan oleh PHP *runtime*, biasanya untuk membuat halaman konten website yang dinamis atau membuat gambar dinamis yang digunakan di situs website atau di tempat lain. PHP Dapat pula digunakan untuk baris perintah scripting dan digunakan di Aplikasi GUI pada sisi klien. PHP dapat digunakan hampir di semua *web server*, di hampir semua sistem operasi dan platform, dan dapat digunakan dengan banyak sistem manajemen database relasional (RDBMS). PHP dapat diunduh secara gratis dan PHP Group menyediakan kode sumber lengkap bagi pengguna untuk membangun, menyesuaikan dan mengembangkannya untuk mereka gunakan sendiri.

3.10 *My Structured Query Language*

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi *General Public License* (GPL). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.

MySQL merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian data yang dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dibanding PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

MySQL adalah satu dari sekian banyak sistem *database*, merupakan terobosan solusi yang tepat dalam aplikasi *database*. Didukung oleh ribuan bahkan jutaan komunitas pengguna di internet yang siap membantu. Selain itu juga tersedia *mailing list* dan *homepage* khusus yang memberikan tutorial serta dokumentasi lengkap (Didik, 2003). Tipe data pada MySQL dapat dilihat pada tabel 3.1.

No	Tipe Data	Deskripsi
1	BIGINT(lenght)	Integer 8 byte
2	TEXT	TEXT?BLOB dengan maksimum 64 Kb
3	CHAR(NUM)	Fixed string dengan panjang antara 1 sampai 255
4	DATE	YYYY-MM-DD
5	DATETIME	YYYY-MM-DD HH:MM:SS
6	DECIMAL(length, dec)	Decimal
7	DOUBLE	Bilangan double precision floating-point
8	DOUBLE PRECISION	Bilangan double precision floating-point
9	FLOAT	Bilangan floating-point
10	INTEGER	Integer 4 byte
11	ENUM	Enumerasi
12	LONGTEXT/LONGBLOB	Maksimum 2^{23}
13	MEDIUMTEXT/MEDIUMBLOB	Maksimum 16777216
14	MEDIUMINT	Integer 3 byte
15	NUMERIC	Sama dengan tipe data DECIMAL
16	REAL	Sama dengan DOUBLE
17	SET	Objek string dengan beberapa nilai
18	SMALLINT	Integer 2 byte

19	TINYTEXT/TINYBLOB	Teks/binary dengan maksimum 255
20	TINYINT	Integer 1 byte
21	VARCHAR (NUM)	Variable length string 1 <= NUM <=255
22	TIME	HH:MM:SS
23	TIMESTAMP	YYYYMMDDHHMMSS

Tabel 3.1 Tipe Data MySQL

Data Manipulation Language (DML) merupakan perintah yang digunakan untuk menampilkan, mengubah, menambah, dan menghapus baris dalam tabel (Sanjaya, 2005). Perintah-perintah untuk menampilkan, mengubah, menambah, dan menghapus terdiri dari.

1. *Select*

Perintah ini digunakan untuk menampilkan isi tabel yang ada pada *database*. Perintah *select* digunakan untuk menampilkan baik secara keseluruhan isi tabel maupun sebagian isi tabel. Perintah ini juga bisa menampilkan data dari tabel yang dihubungkan dengan tabel-tabel yang lain.

2. *Insert*

Perintah ini digunakan untuk menambahkan data ke dalam *database*. Pengisian yang dilakukan bisa untuk seluruh *field* atau hanya sebagian *field* saja.

3. *Update*

Perintah update digunakan untuk mengubah data dalam sebuah *database*.

Perubahan yang dilakukan bisa untuk seluruh *field* atau hanya sebagian *field* saja.

4. *Delete*

Perintah ini digunakan menghapus *record* dengan kriteria tertentu yang ada pada *database*.

