

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian sebelumnya yang dilakukan oleh penulis lain yang dijadikan sebagai bahan pertimbangan. Penelitian ini dilakukan oleh Mellinda Ayu Mahanani dengan judul “Sistem Informasi Pembayaran SPP Berbasis *Web* pada SMK Negeri 1 Bancak”. Sistem Informasi ini masih menggunakan sistem konvensional sehingga kesalahan maupun ketidakamanan data bisa saja terjadi. Sistem ini berhasil mempermudah staf tata usaha dalam mengelola data administrasi pembayaran SPP siswa. Semua data dapat saling berintegrasi sehingga sangat efektif dan efisien. Sistem juga dapat menampilkan informasi pembayaran selain di lingkungan internal sekolah juga dapat digunakan di lingkungan eksternal yaitu wali murid dapat mengetahui informasi pembayaran SPP siswa yang bersangkutan. Dalam pengembangan sistemnya, penulis menggunakan metode *prototype* dan pengujian sistemnya menggunakan metode *blackbox testing*. Bahasa pemrograman menggunakan PHP dan *database MySQL*. Penulis ini menggunakan SMS sebagai notifikasi. Kekurangan notifikasi melalui SMS yaitu yaitu apabila terdapat gangguan pada jaringan telekomunikasi, maka sistem tidak dapat secara otomatis mengirim ulang pesan yang telah dikirimkan. Selain itu, penggunaan SMS sebagai notifikasi tidak bersifat gratis. Ada bebanan kepada *user* untuk membeli pulsa pada tiap kali mengirimkan SMS.[1]

Penelitian serupa dilakukan oleh Nur Khalimatus Sya'diyah dengan judul “Sistem Informasi Pembayaran SPP Berbasis *Web* pada SMP Muhammadiyah 2 Semarang dengan SMS *Gateway*”. Dalam pengujian sistemnya, penulis menggunakan metode *waterfall* dan *Unified Modeling Language* (UML) perangkat lunak yang dibangun menggunakan bahasa pemrograman PHP dan *MySQL*. Sistem ini mulai mengarah ke pihak orang tua/wali siswa, sehingga orang tua/wali siswa yang berada diluar kota dapat mengetahui berapa uang yang harus dibayarkan. Selain itu, sistem ini juga berhasil mempermudah petugas administrasi keuangan dalam menginput data siswa. Orang tua pada sistem ini hanya mendapatkan notifikasi melalui SMS terkait jadwal pembayaran SPP. [2]

Penelitian sebelumnya dilakukan oleh Yusran dengan judul “Perancangan Sistem Informasi Administrasi Pembayaran SPP Siswa Berbasis *Web*”. Permasalahan yang terjadi adalah proses penyampaian informasi kepada orang tua/wali siswa dilakukan dengan cara siswa memberitahukan orang tua/wali siswa sehingga tidak efektif. Penelitian ini menghasilkan sistem yang dapat mempermudah pihak sekolah khususnya bagian tata usaha dalam mengelola proses pembayaran SPP. Selain itu, dengan adanya sistem ini mempercepat proses pengolahan data. Orang tua murid juga merasakan kemudahan dalam mendapatkan informasi yang akurat dari pihak sekolah sehingga dapat mengurangi keterlambatan. Metode pengembangan sistem yang digunakan pada penelitian ini adalah *waterfall*. Bahasa pemrograman yang digunakan untuk membangun sistem ini, menggunakan bahasa pemrograman PHP dan MySQL sebagai *database*. [3]

Penelitian lainnya dilakukan oleh Erna Astriyani dengan judul “Perancangan Sistem Informasi Pembayaran SPP Berbasis Web Menggunakan Notifikasi SMS *Gateway*”. Permasalahan yang terjadi adalah informasi yang dikirimkan berupa data pembayaran SPP saja, belum terdapat notifikasi jatuh tempo atau peringatan pembayaran SPP. Penelitian ini berhasil mempermudah serta mempercepat pekerjaan pengelolaan data pembayaran SPP. Selain itu, sistem ini dapat membantu orang tua/wali mendapatkan informasi mengenai pembayaran SPP melalui SMS *Gateway*. Bahasa pemrograman yang digunakan untuk membangun sistem ini, menggunakan bahasa pemrograman PHP dan MySQL sebagai *database*. Penelitian ini menggunakan metode pengembangan sistem metode SDLC jenis *Waterfall*. [4]

Berbeda dengan penelitian-penelitian terkait yang dilakukan sebelumnya. Adapun penelitian yang dikembangkan berjudul “Sistem Informasi Pembayaran SPP pada SMK Makmur 1 Cilacap Berbasis *Web*” mempunyai perbedaan dengan penelitian yang dilakukan sebelumnya. Perbedaan dengan sistem yang sebelumnya yaitu penambahan notifikasi yang akan dikirimkan oleh sistem kepada orang tua siswa melalui *email*, diharapkan dapat lebih membantu mengingatkan jadwal pembayaran SPP sehingga tidak terjadi keterlambatan pembayaran, adanya fitur untuk orang tua agar dapat melihat pembayaran telah dilakukan atau belum, proses pembayaran dapat dilakukan secara *online* sehingga dapat lebih mempermudah apabila orang tua ingin membayarkan secara langsung tanpa perlu mendatangi sekolah. Metode pengembangan yang digunakan adalah metode *Waterfall*.

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan melaksanakan suatu fungsi untuk mencapai tujuan. Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem dan sasaran sistem. [5]

2.2.2 Pengertian Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata. [5]

2.2.3 Sistem Informasi

Sistem informasi merupakan suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan. Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.[6]

2.2.4 Sumbangan Pembangunan Pendidikan (SPP)

Hayati (2013) Menurut kamus besar bahasa Indonesia definisi pembayaran adalah dari suku kata “pem.ba.yar.an” adalah proses, cara, perbuatan membayar, maka pengertian pembayaran SPP adalah proses membayar SPP yang dilakukan berulang-ulang, sekali dalam satu bulan, dasar hukum pengusutan SPP adalah keputusan bersama tiga menteri, yaitu: (1) Menteri Pendidikan dan Kebudayaan (No. 0257/K/1974), (2) Menteri dalam Negeri (No. 221 tahun 1947), (3) Menteri Keuangan (No.

Kep. 1606/MK/11/1974). SPP dimaksudkan untuk membantu pembinaan pendidikan seperti yang ditunjukkan pada pasal 12 keputusan tersebut yakni untuk membantu penyelenggaraan sekolah, kesejahteraan personal, perbaikan sarana dan prasarana dan kegiatan supervisi. Selanjutnya pada pasal 18 dinyatakan bahwa kedudukan Kepala Sekolah dalam pengelolaan SPP adalah bendaharawan khusus yang bertanggungjawab dalam penerimaan, penyetoran, dan penggunaan dana yang telah ditentukan terutama Dana Penunjang Pendidikan (DPP).

2.2.5 Rekayasa Perangkat Lunak

Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model sistem, dan cara penggunaannya. RPL merupakan pembangunan sebuah perangkat lunak dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomis yang dipercaya dan bekerja secara efisien menggunakan mesin. Rekayasa Perangkat Lunak (*Software Engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.[7]

2.2.6 Rekayasa Web

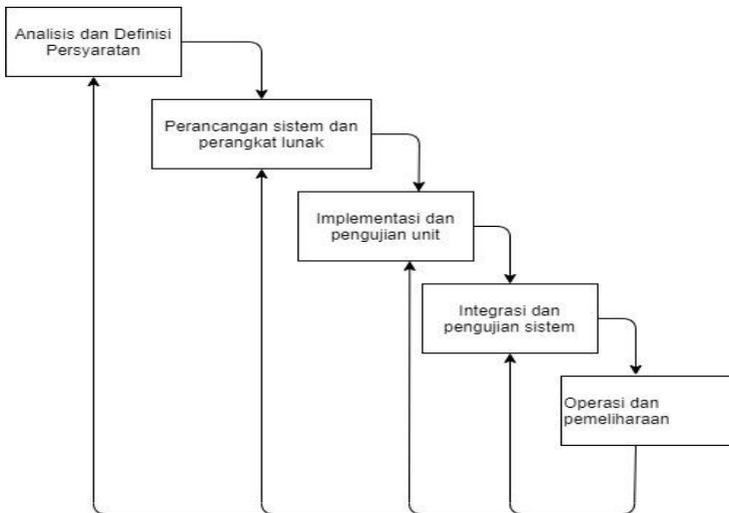
Rekayasa Web adalah proses yang digunakan untuk menciptakan aplikasi web yang berkualitas tinggi. Rekayasa web mengadaptasi rekayasa perangkat lunak dalam hal konsep dasar yang menekankan pada aktifitas teknis dan manajemen. Namun demikian adaptasi tidak secara utuh, tapi dengan perubahan dan penyesuaian. Rekayasa web gabungan antara web publishing (suatu konsep yang berasal dari printed publishing) dan aktifitas rekayasa perangkat lunak. Dikatakan demikian karena desain sebuah aplikasi web menekankan pada desain grafis, desain informasi, teori hypertext, desain sistem dan pemrograman

2.2.7 Waterfall

Model *Waterfall* adalah sebuah metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Dalam proses implementasi metode *Waterfall* ini, sebuah langkah akan diselesaikan terlebih dahulu dimulai dari tahapan yang pertama sebelum melanjutkan ke tahapan yang berikutnya.[8] Transisi antara tahapan yang sebelumnya sudah sepenuhnya selesai dilaksanakan sehingga tidak ada tahapan yang berjalan secara bersamaan.

Model *Waterfall* menurut Ian Sommerville dipilih karena pendekatannya yang sistematis dan terurut serta dapat melibatkan

pengulangan beberapa tahap atau seluruh tahap, setelah melalui tahap fase siklus hidup akhir. Tahapan-tahapannya meliputi analisis kebutuhan, desain sistem, penulisan kode program, pengujian program, dan penerapan program serta pemeliharaan.



Gambar 2. 1 Metode waterfall menurut Sommerville

Berikut adalah penjelasannya :

1. Analisis kebutuhan perangkat lunak
Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
2. Desain
Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean.
3. Pembuatan kode program
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminal.

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur struktur kontrol, maka perhatian berfokus pada domain informasi.

Pengujian *black-box* ini terdapat beberapa proses. Proses-proses yang ada dalam pengujian ini diantaranya :

- a. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
- b. Pemilihan jenis *input* yang memungkinkan menghasilkan *output* dengan benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
- c. Menentukan *output* untuk satu jenis *input*.
- d. Pengujian dilakukan dengan *input-input* yang telah benar-benar diseleksi.
- e. Melakukan pengujian
- f. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan.
- g. Menentukan fungsionalitas yang seharusnya ada pada perangkat lunak yang sedang diuji.

5. Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Pada tahap ini dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2.8 Struktur Data

Struktur data adalah cara menyimpan atau merepresentasikan data di dalam komputer agar bisa dipakai secara efisien. Berikut adalah bagian-bagian yang ada pada struktur data :

1. *Flowchart*

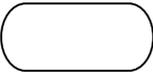
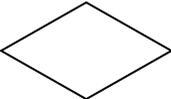
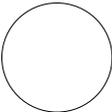
Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Ada beberapa jenis-jenis *flowchart* diantaranya :

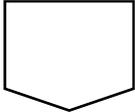
- A. Bagan alir sistem (*system flowchart*)
- B. Bagan alir dokumen (*document flowchart*)
- C. Bagan alir skematik (*schematic flowchart*)
- D. Bagan alir program (*program flowchart*)
- E. Bagan alir proses (*process flowchart*)

Simbol-simbol dalam *flowchart* dapat dilihat pada tabel 2.1

Tabel 2. 1 *Simbol Flowchart*

No	Simbol	Fungsi
1	 <p>Flow Direction Symbol</p>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>connecting line</i> .

No	Simbol	Fungsi
2	 <p>Terminator Symbol</p>	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.
3	 <p>Processing Symbol</p>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4	 <p>Manual Operation Symbol</p>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
5	 <p>Decision Symbol</p>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
6	 <p>Connector (On-page)</p>	Menyederhanakan hubungan antar simbol yang letaknya berjauhan.

No	Simbol	Fungsi
7	 Connector (Off-page)	Menghubungkan simbol dalam halaman berbeda.
8	 Input-Output Symbol	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
9	 Document Symbol	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.
10	 Manual Input	Memasukkan data secara manual dengan menggunakan online keyboard.

2. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design (OOAD)*. Macam-macam dari *Unified Modeling Language (UML)* antara lain : *use case diagram*, *sequence diagram* dan *si diagram*.

a. Use Case

Use case diagram atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Secara kasar use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi–fungsi itu. [9][10][10][10][10][9][8][8][8][8][8]

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol *Use Case* dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol Use Case Diagram

No	Simbol	Nama Simbol dan Keterangan
1		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i> , Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri(independent).

No	Simbol	Nama Simbol dan Keterangan
3		<i>Generalization</i> , Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i> , Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i> , yang menghubungkan antara objek satu dengan objek lainnya
7		<i>System</i> , Menspesifikasikan paket yang menampilkan sistem secara terbatas
8		<i>Use Case</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan

No	Simbol	Nama Simbol dan Keterangan
		suatu hasil yang terukur bagi suatu actor.
9		<i>Collaboration</i> , Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i> , Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case*. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence Diagram* menambahkan dimensi waktu pada interaksi diantara obyek[7]. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2. 3 *Simbol Sequence Diagram*

No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Usecase</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
2.		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i> , Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i> , Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>System</i> , Menspesifikasikan paket yang menampilkan sistem secara terbatas.

2.2.9 Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk menginformasikan suatu perusahaan instansi, dalam bahasan tertentu [8]

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language (SQL)*.

MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. *MySQL* yang biasa kita gunakan adalah *MySQL Free Software* yang berada dibawah Lisensi *General Public License (GPL)*[9]. *MySQL* juga dapat didefinisikan sebagai sebuah database server, dapat juga berpersion sebagai *client* sehingga sering disebut *database client / server yang open source* dengan kemampuan dapat berjalan baik di Operasi Sistem maupun dengan *Platform Windows* maupun *Linux*.

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama *MySQL AB*, yang pada saat itu bernama *TcX Data Konsult AB* sekitar tahun 1994-1995. *MySQL* sudah ada sejak 1979. *MySQL* termasuk jenis *Relation Database Management System (RDBMS)* digunakan oleh banyak portal-portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran *MySQL* dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap digunakan dalam *MySQL*. Sebuah basis data yang terdapat pada *MySQL* mengandung satu atau beberapa tabel yang terdiri dari sejumlah baris dan kolom.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu *SQL (Structured Query Language)*. *SQL* adalah suatu bahasa (language) yang digunakan untuk mengakses data di dalam sebuah database relasional. *SQL* sering juga disebut dengan istilah *query*, dan bahasa *SQL* secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh server database atau software database mengenal dan mengerti bahasa *SQL*. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah *SQL* yaitu :

1.) ***Data Definition Language (DDL)***

DDL adalah sub perintah dari bahasa *SQL* yang digunakan untuk membangun kerangka sebuah database, dalam hal ini *database* dan *table*. Terdapat tiga perintah penting dalam DDL, yaitu :

- a.) *CREATE*: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru view baru, dan kolom baru. Contoh: *CREATE DATABASE nama_database*.
- b.) *ALTER*: perintah *ALTER* berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel,

menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh: *ALTER TABLE* nama_tabel *ADD* nama_kolom datatype.

- c.) *DROP*: perintah *DROP* berfungsi untuk menghapus database atau tabel.

Contoh: *DROP DATABASE* nama_database.

2.) **Data Manipulation Language (DML)**

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Terdapat 4 (Empat) perintah penting dalam DML, yaitu :

- a.) *INSERT*: perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika database dan tabel sudah dibuat. Contoh: *INSERT INTO* nama_tabel *VALUES* (data1, data2, dst...);
- b.) *SELECT*: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh: *SELECT* nama_kolom1, nama_kolom2 *FROM* nama_tabel;
- c.) *UPDATE*: perintah update digunakan untuk memperbaharui data pada sebuah tabel. Contoh: *UPDATE* nama_tabel *SET* kolom1=data1, kolom2=data2, ... *WHERE* kolom=data;
- d.) *DELETE*: perintah delete digunakan untuk menghapus data dari sebuah tabel.

Contoh: *DELETE FROM* nama_tabel *WHERE* kolom=data

3.) **Entity Relationship Diagram (ERD)**

ERD adalah suatu model untuk menjelaskan mengenai hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki hubungan antar relasi. *Entity Relationship Diagram* digunakan untuk memodelkan struktur data serta hubungan antar data, untuk dapat menggambarkannya digunakan beberapa notasi serta simbol.[13]Komponen dalam menyusun *Entity Relationship Diagram* (ERD) antara lain sebagai berikut :

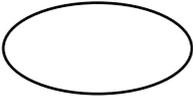
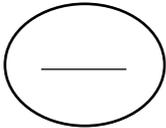
- a. Entitas merupakan suatu objek dalam dunia nyata yang bisa dibedakan dengan objek lain, ebagai contoh murid, guru, yayasan . Entitias tersebut terdiri dari beberapa atribut sebagai contoh : NIS,nama,alamat,umur dan lain sebagainya. Atribut NIS adalah atribut yang unik yang digunakan untuk mengidentifikasi atau membedakan siswa dengan siswa yang lain. Pada tiap-tiap

entitas harus memiliki 1 atribut unik yang dikenal dengan sebutan primary key.

- b. Atribut merupakan entitas pasti yang memiliki elemen yang berfungsi untuk dapat mendeskripsikan karakteristik dari suatu entitas tersebut seperti contoh di atas . Isi dari atribut tersebut memiliki sesuatu yang bias mengidentifikasi isi elemen yang satu dengan yang lainnya. Terdapat dua jenis Atribut antara lain sebagai berikut :
 1. Identifier (key) yang berfungsi sebagai penentu entity secara unik (primary key).
 2. Descriptor (nonkey attribute) digunakan untuk dapat menspesifikasikan karakteristik dari sebuah entity yang tidak unik.
- c. Relasi adalah suatu hubungan antara beberapa entitas..himpunan relasi antar entitas pemetaan kardinialit terdiri dari :
 1. *One-to-one*
Setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua. Hubungan antara file pertama dan file kedua adalah satu berbanding satu.
 2. *One-to-many*
setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua. Hubungan antara file pertama dan file kedua adalah satu berbanding banyak atau banyak berbanding satu
 3. *Many-to-one*
Kebalikan dari *relation One To Many* dimana setiap baris data dari tabel pertama dihubungkan lebih dari satu baris ke tabel kedua. Hubungan antara file pertama dan file kedua adalah banyak berbanding.

Simbol-simbol yang dipakai dalam pembuatan *Entity Relational Diagram* (ERD) dapat dilihat pada Tabel 2.4.

Tabel 2.4 *Simbol Entity Relational Diagram*

No	Simbol	Fungsi
1	 <p style="text-align: center;">Entity</p>	Memberikan identitas pada entitas yang memiliki label dan nama.
2	 <p style="text-align: center;">Relationship</p>	Mengetahui jenis hubungan yang ada antara 2 file.
3	 <p style="text-align: center;">Attribute</p>	Memperjelas atribut yang dimiliki oleh sebuah entitas.
4	 <p style="text-align: center;">Primary Key Attribute</p>	Atribut yang menjadi kunci utama dan tidak boleh sama dengan atribut yang lain.
5	 <p style="text-align: center;">Link</p>	Menghubungkan atribut dengan entitas dan entitas dengan relasi.

~Halaman Ini Sengaja Dikosongkan~