

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian pertama yang telah dilakukan oleh Irma Yunita, Sunardi (2017) dengan judul “Sistem Informasi Bimbingan dan Konseling di MTs Miftahul Arifin dengan menggunakan PHP dan MYSQL”. Metode penelitian dan model pengembangan yang digunakan yaitu model pengembangan berbasis *System Development Life Cycle* (DLC). Dalam model ini ada beberapa tahapan yang harus dilalui sebelum sistem benar-benar dapat diimplementasikan pada sekolah. Tahapan-tahapan tersebut antara lain adalah adanya planning, analisis, desain, pengkodean, testing, maintenance. Penelitian ini bertujuan untuk mengelolah proses kegiatan bimbingan konseling di MTs. Miftahul Arifin dengan mengisi form bimbingan. Penelitian ini lebih berfokus pada mengelola data skor, ketentuan skor dan data pelanggaran^[3].

Penelitian kedua dilakukan oleh Nufan Balafif, Budiman, Zainal Muttaqin (2016) dengan judul “Peningkatan Pelayan Bimbingan Konseling di MAN Tambakberas Jombang dengan Sistem Informasi Bimbingan Konseling berbasis Web”. Sistem ini dibuat untuk memberikan informasi hasil Bimbingan Konseling dengan siswa kepada orang tua siswa secara cepat yang tidak terbatas dengan tempat maupun waktu. Pengembangan sistem menggunakan metode USDP (*Unified Software Development Process*). Penelitian ini bertujuan untuk membantu proses layanan serta pelaporan terhadap hasil bimbingan dan konseling. Penelitian ini lebih berfokus pada layanan bimbingan konseling secara online antara siswa dan guru BK, serta mencatat pelanggaran yang telah lakukan oleh siswa tersebut^[4].

Penelitian ketiga dilakukan oleh Cornelia Miyastuti Cahyadi, Gaguk Susanto (2020) dengan judul “Rancang Bangun Sistem Informasi Bimbingan Konseling di SMK Tamansiswa Mojokerto berbasis Web menggunakan Metode *Waterfall*”. Sistem ini dibuat untuk memudahkan dalam mengelola proses pencatatan data pengguna yang terlibat dalam sistem informasi (siswa, wali kelas, wali murid, guru BK, pihak tertib, Kepala sekolah), pencatatan data pelanggaran dan poin pelanggaran, pencatatan data sanksi pencatatan data proses bimbingan konseling. Pengembangan sistem menggunakan metode *Waterfall*. Penelitian ini bertujuan untuk memudahkan bimbingan konseling dalam pengelolaan proses administrasi layanan. Penelitian ini berfokus pada pengolahan data

pelanggaran dan poin pelanggaran, pengolahan data proses bimbingan konseling yang telah dilaksanakan serta pengiriman pesan secara langsung kepada orang tua melalui aplikasi pesan *whatsapp*^[5].

Penelitian keempat dilakukan oleh Rizwan Hanafi, Nur Cahyo Wibowo, Agung Brastama Putra (2020) dengan judul “Sistem Informasi Konseling berbasis Web (Studi Kasus : SMK Negeri 1 Mojoagung)”. Sistem ini dibuat untuk memudahkan pencatatan pelanggaran bagi guru BK. Pengembangan sistem menggunakan metode Waterfall. Penelitian ini bertujuan untuk memudahkan pencatatan pelanggaran bagi guru BK. Penelitian ini berfokus pada pencatatan pelanggaran dan prestasi siswa dan orang tua dapat memonitoring anaknya^[6].

Setelah mengevaluasi penelitian sebelumnya, penulis akan membuat sistem yang berjudul “Rancang Bangun Aplikasi Konseling Online berbasis Web (Studi Kasus : MAN 3 Banyumas)”. Sistem ini terdapat perbedaan dengan penelitian yang dilakukan sebelumnya. Sistem yang akan penulis rancang antara lain adanya *live chat* secara *real time* yang digunakan sebagai pelayanan Bimbingan dan konseling secara online, mengelola rekam data Bimbingan dan konseling, terdapat status pada guru BK untuk melakukan pelayanan Bimbingan dan konseling atau tidak. Sistem ini diharapkan dapat memberikan kemudahan bagi guru BK dan siswa untuk melakukan Bimbingan dan konseling dan dapat mengelola data Bimbingan dan konseling. Sistem yang akan dibangun menggunakan Bahasa pemrograman PHP dan MySQL berbasis Website menggunakan *waterfall*. Metode pengujian yang digunakan adalah *black box*.

2.2 Landasan Teori

2.2.1 Aplikasi

Aplikasi adalah proses / prosedur aliran data di suatu infrastruktur teknologi informasi yang dapat digunakan oleh para pengambil keputusan sesuai dengan tingkat dan kebutuhan yang relevan^[7]. Menurut Asropudin (2013) Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas.

Secara garis besar, aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut.

2.2.2 Website

Berikut adalah pengertian dan penjelasan mengenai jenis-jenis dalam pembuatan aplikasi berbasis *website*:

- a. Pengertian
Website adalah layanan penyajian informasi yang menggunakan konsep *hyperlink* untuk memudahkan *surfer* atau *browser* dalam mencari informasi melalui internet (Murya, 2012). Menurut Sibero (2013), *website* adalah sebuah sistem yang berhubungan dengan dokumen yang digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya internet.
- b. Jenis-jenis Website
 - 1) Website Dinamis
Website statis merupakan *website* yang berisikan data dan informasi yang tidak berubah-ubah. Dokumen *web* yang dikirim kepada client akan sama isinya dengan *web server*.
 - 2) Website Statis
Website dinamis merupakan *website* yang memiliki data dan informasi yang berbeda-beda tergantung input yang disampaikan oleh *client*.

2.2.3 Bimbingan Konseling

Bimbingan dan konseling adalah upaya proaktif dan sistematis dalam memfasilitasi individu mencapai tingkat perkembangan yang optimal, pengembangan perilaku yang efektif, pengembangan lingkungan, dan peningkatan fungsi atau manfaat individu dalam lingkungannya. Semua perubahan perilaku tersebut adalah proses perkembangan individu, yakni proses hubungan antara individu dengan lingkungan melalui hubungan yang sehat dan produktif. Bimbingan dan konseling memegang tugas dan tanggung jawab yang penting untuk mengembangkan lingkungan, membentuk interaksi dinamis antara individu dengan lingkungan, membelajarkan individu untuk mengembangkan, merubah dan memperbaiki perilaku.

Pelayanan bimbingan dan konseling di sekolah sangat diperlukan. Karena setiap siswa memiliki masalah, baik masalah pribadi maupun belajar di sekolah, maka masalah yang dihadapi oleh semua siswa tentunya berbeda-beda^[8]. Pelayanan bimbingan dan konseling memfasilitasi pengembangan peserta didik secara individual, kelompok, potensi, bakat dan klasikal, sesuai dengan kebutuhan, potensi, bakat, minat, perkembangan, kondisi, serta peluang-peluang yang dimiliki.

Guru BK berperan penting terkait dengan pengembangan diri melalui layanan konseling kepada siswa. Adapun peran Guru BK sebagai berikut :

- a. Pengembangan kehidupan pribadi, yaitu bidang layanan yang membantu siswa memahami dan mengevaluasi bakat dan minat mereka.
- b. Pengembangan kehidupan sosial, yaitu bidang pelayanan yang membantu siswa dalam memahami dan mengevaluasi dan mengembangkan kemampuan hubungan sosial yang harmonis, dinamis, adil dan bermartabat.
- c. Pengembangan kemampuan belajar, yaitu bidang pelayanan yang membantu siswa dalam mengembangkan kemampuan dan keterampilan belajarnya agar dapat mengikuti pendidikan di sekolah atau madrasah secara mandiri.
- d. Pengembangan karir, yaitu bidang pelayanan yang membantu siswa dalam memahami dan menilai informasi, serta memilih dan mengambil keputusan karir.
- e. Pengembangan kehidupan beragama, yaitu bidang pelayanan yang membantu siswa untuk memberikan bimbingan rohani sesuai dengan kepercayaan dan keyakinan masing-masing^[9].

2.2.4 PHP Mailer

PHP *Mailer* adalah teknologi yang digunakan untuk mengirimkan informasi tertentu terhadap *user* yang terhubung dengan *website* tertentu dan cara kerja *PHP Mailer* seperti pengiriman *email* secara otomatis tanpa menggunakan *user* untuk mengirimkan *email* kepada *user* lain dengan fungsi yang dimiliki oleh *PHP* yaitu fungsi *mail()*. Hal ini memberikan kemudahan dalam proses pengiriman *email* serta mengurangi waktu untuk melakukan pengiriman *email* secara manual. Hal ini bergantung dengan koneksi internet yang digunakan oleh *user* karena jika kecepatan internet lambat dapat mempengaruhi proses pengiriman kepada *user*^[10].

PHP *Mailer* adalah salah satu library PHP open-source yang digunakan untuk mengirim *e-mail* dari localhost. PHP *Mailer* dapat menjalankan fungsinya sebagai *e-mail* jika kita mensupportnya dengan *Simple Mail Transfer Protocol* (SMTP). SMTP adalah suatu protocol yang diperlukan untuk mengirim dan menerima *e-mail*. Karena itulah kita harus menggunakan SMTP sebagai layanan mengirim *e-mail*. Layanan

ini dapat digunakan untuk keperluan seperti memverifikasi *e-mail*, contohnya ketika mendaftarkan di Twitter atau Facebook. Setelah mendaftar maka kita diharuskan membuka e-mail dan memverifikasinya.

PHP *Mailer* dapat digunakan sebagai notifikasi ketika ada bimbingan masuk untuk guru BK, sehingga guru BK dapat mengetahui jika ada bimbingan masuk. PHP *Mailer* digunakan sebagai notifikasi jika ada balasan bimbingan masuk untuk siswa, sehingga siswa akan mengetahui kapan mereka menerima balasan.

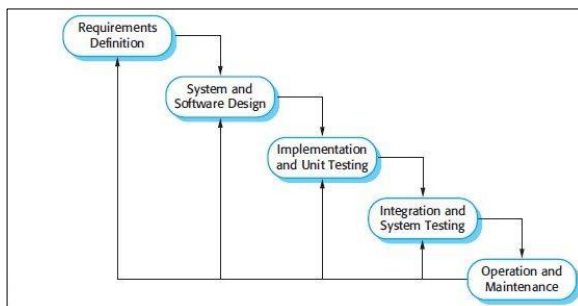
2.2.5 Rekayasa Web

Rekayasa web adalah proses yang digunakan untuk membuat aplikasi web yang berkualitas tinggi. Rekayasa web mengadaptasi rekayasa perangkat lunak dalam hal konsep dasar yang menekankan pada kegiatan teknis dan manajemen.

1. Metode *Waterfall*

Metode *Waterfall* merupakan metode pengembangan sistem terjadi secara berurutan dari satu fase ke fase lain 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^[11]. Metode *Waterfall* menurut Ian Sommerville dipilih karena pendekatannya yang sistematis dan terurut serta dapat melakukan pengulangan beberapa tahap atau seluruh tahap.

Model air terjun menjalankan siklus proses, seperti yang ditunjukkan pada Gambar 2.1. Dalam model *waterfall* menurut Ian Sommerville, ada cara untuk kembali ke tahap sebelumnya jika terjadi kegagalan atau perbaikan.



Gambar 2.1 Metode *Waterfall* menurut Ian Sommerville

Adapun tahap-tahapnya yaitu :

1. Requirements Analysis and Definition
Tahap awal dimana adanya analisis untuk menentukan kebutuhan, batasan dan tujuan dari pengembangan perangkat lunak harus paham mengenai fungsi yang dibutuhkan oleh *user*.
2. System and Software Design
Tahap *system and software design*, hasil analisa kebutuhan perangkat lunak pada sistem ini akan dideskripsikan ke dalam diagram, antara lain *use case diagram*, *sequence diagram*, dan ERD.
3. Implementation and Unit Testing
Pada tahap ini, penulis melakukan penulisan kode program atau bahasa yang dimengerti oleh mesin komputer dengan menggunakan bahasa pemrograman yang beracuan pada tahapan desain sistem sebelumnya.
4. Integration and System Testing
Pada tahap *integration and system testing*, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada.

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* merupakan pengujian yang menggunakan data uji dan memeriksa fungsional dari perangkat lunak^[12]. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak untuk memperoleh serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang hilang atau salah,
2. Kesalahan desain antarmuka (*interface*) atau tampilan,
3. Kesalahan dalam struktur data atau akses database eksternal,
4. Kesalahan performa,
5. Kesalahan inisialisasi dan terminasi^[13].

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.

5. Operation and Maintenance

Tahapan *operation and maintenance* mencakup koreksi dari tahap sebelumnya, perbaikan atau implementasi dari persyaratan-persyaratan baru yang ditambahkan, tetapi tidak termasuk pembuatan program perangkat lunak baru^[14].

2.2.6 Tools /Alat Penelitian

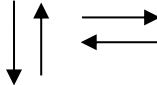
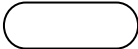





1. *Flowchart*

Flowchart adalah diagram yang menunjukkan alur (*flow*) kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem^[15].

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*)

Tabel 2.1 Simbol Flowchart

No.	Simbol	Keterangan
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> .
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>Input-output Symbol</p>	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
7.	 <p>Document Symbol</p>	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah hasil kolaborasi antara konsorsium berbagai organisasi yang berbeda dan telah menjadi standar baku dalam OOAD (*Object Oriented Analysis dan Design*). Macam-macam dari *Unified Modeling Language (UML)* antara lain^[16] :



a. *Use Case*









Use Case adalah pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang berhak menggunakan fungsi-fungsi tersebut^[17]. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita tentang bagaimana sistem digunakan.

Urutan langkah-langkah yang menerangkan antara pengguna dan sistem adalah skenario sedangkan pengguna disebut aktor. Aktor adalah sebuah peran yang biasanya dimainkan oleh pengguna dalam berinteraksi dengan sistem. Model *use case* merupakan bagian dari model kebutuhan. Definisi *use case* lainnya adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* dibuat berdasarkan kebutuhan aktor.

Berdasarkan definisi diatas, maka dapat disimpulkan bahwa *use case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna potensial yang terdiri dari sekumpulan scenario dan aktor. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analisis dan pengguna serta analisis dan pelanggan. 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 use case.
2.		<i>Dependency</i> , Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).



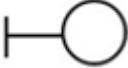



3.		<i>Generalization</i> , Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i> , Menspesifikasikan bahwa <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 suatu hasil yang terukur bagi suatu <i>actor</i> .
9.		<i>Collaboration</i> , Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (<i>sinergi</i>).
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 skenario. Diagram ini menunjukkan sekumpulan objek dan *message* yang ditempatkan di antara objek-objek dalam *use case*.

Komponen utama *sequence diagram* terdiri dari 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 antara objek. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambar hubungan yang akan dilakukan.
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form.
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
	<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message.
	<i>A message</i>	Menggambarkan pengiriman pesan.

2.2.7 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengatur perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek adalah suatu cara untuk membangun sistem perangkat lunak melalui pendekatan objek yang sistematis.

Sistem berorientasi objek adalah sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, komponen lainnya dan dapat berinteraksi satu sama lain.

2.2.8 Basis Data

Basis Data adalah kumpulan tabel-tabel yang berisi sekumpulan data yang fakta sebagai sumber informasi yang disimpan dalam media penyimpanan secara digital dan dapat diperiksa menggunakan suatu program komputer yang berguna untuk memudahkan kegiatan untuk pengumpulan informasi^[18].

MySQL adalah 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).

MySQL juga dapat didefinisikan sebagai sebuah database server dan juga dapat juga berperan 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*. Popularitas 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 dari salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah suatu bahasa (*language*) yang digunakan untuk mengakses data dalam sebuah database relasional. *SQL* sering juga disebut sebagai *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 membuat kerangka kerja untuk 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** digunakan 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** digunakan 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) *Data Control Language (DCL)*

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi *user* dan hak akses (*priviledges*). Yang termasuk perintah dalam DCL ada 2 (dua), yaitu :

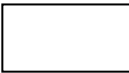

- a. GRANT : perintah ini digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna. Hak akses tersebut bisa berupa hak membuat (CREATE), mengambil data (SELECT), menghapus data (DELETE), mengubah data (UPDATE), dan hak khusus lainnya yang berhubungan dengan sistem database.
- b. REVOKE : perintah ini digunakan untuk mencabut hak akses yang telah diberikan kepada user. Dalam ini merupakan kebalikan dari perintah GRANT^[19].

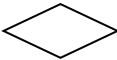

2.2.9 *Entity Relationship Diagram (ERD)*

ERD merupakan suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya^[20].

Berikut merupakan simbol-simbol dari ERD :

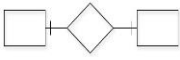
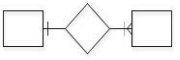
Tabel 2.4 Simbol ERD

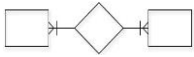
No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.

3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	Link		Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya.

ERD memiliki derajat relasi atau biasa disebut kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.

Tabel 2.5 Macam-macam Kardinalitas

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke Satu (<i>One to One</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya.
2.		Relasi Satu ke Banyak (<i>One to Many</i>)	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya.

3.	 <p>The diagram shows two rectangular entities connected by a diamond-shaped relationship. Each connection line has a crow's foot notation (a vertical line with a perpendicular tick mark) at the entity end, indicating a many-to-many relationship.</p>	Relasi Banyak ke Banyak (<i>Banyak to Many</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaiknya.
----	---	---	---