

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian mengenai sistem informasi absensi pernah dilakukan oleh M. Fadhilur Rahman, absensi menggunakan *Qr-Code* berbasis *e-Confirmation* (Bot Telegram) dan *e-Notification* dengan Teknologi *Google Access* untuk mengetahui kehadiran guru dan siswa. Penelitian ini dilakukan pada tahun 2019 [4].

Penelitian sebelumnya dilakukan oleh Darma Setiawan Putra dkk, Aplikasi web ini menggunakan metode *Rapid Application Development* (RAD) bertujuan untuk mengetahui kehadiran mahasiswa, penelitian ini dilakukan tahun 2018 [5].

Penelitian selanjutnya dilakukan oleh Yodang dkk, penelitian ini dilakukan untuk pelatihan kompetensi guru dalam menggunakan *Qr-Code* berbasis Android guna memanfaatkan aplikasi *Qr-Code* presensi siswa secara online berbasis android, penelitian ini dilakukan tahun untuk presensi siswa secara online berbasis android dan dilakukan pada tahun 2021 [6].

Penelitian serupa dilakukan oleh Rastri Prativhi dkk, dengan judul Sistem Presensi Kelas menggunakan pengenalan wajah dengan metode *Haar Cascade Classifier* dan dilakukan tahun 2020 [7].

Penelitian selanjutnya dilakukan oleh Ali Aman dan Budi Triandi, penelitian ini bertujuan untuk memudahkan bawaslu dalam melakukan validasi atau melihat keaslian pelanggaran pemilu menggunakan algoritma Base64 dan *Qr-Code* dan dilakukan tahun 2020 [8]

Pada penelitian ini, akan dibuat sistem presensi rapat berbasis *Qr-Code* di Politeknik Negeri Cilacap. Aplikasi web ini dibuat menggunakan *prototype, native*, Bahasa pemrograman PHP, *Database Developer MySql, Web Server Apache* dan *tools editor* Notepad++. Terdapat fitur *email gateway* untuk memberikan notifikasi undangan rapat kepada peserta rapat. Hasil dari penelitian ini adalah sebuah sistem yang dapat memudahkan peserta rapat dalam melakukan presensi, serta memudahkan sekretaris direksi dalam pembuatan dan menginformasikan hasil MoM. Selain itu sistem ini dapat membantu proses pelaksanaan pembelajaran pembuatan suatu sistem bagi mahasiswa Politeknik Negeri Cilacap.

**Tabel 2. 1** Perbandingan Penelitian

No	Judul	Peneliti	Perbedaan	Keterbaruan
1.	Absensi <i>Qr-Code</i> Berbasis <i>e-Confirmation</i> (Bot Telegram) dan <i>e-Notification</i> dengan Teknologi Google <i>Access</i> (2019)	Muhammad Fadhilur Rahman	<ul style="list-style-type: none"> <li>- Bahasa pemrograman visual basic</li> <li>- Aplikasi berbasis desktop Tidak menggunakan <i>email gateway</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menggunakan <i>emailgateway</i></li> <li>- Aplikasi berbasis web</li> </ul>
2.	Perancangan Aplikasi Presensi Dosen Realtime Dengan Metode <i>Rapid Application Development</i> (RAD) Menggunakan <i>Fingerprint</i> Berbasis Web (2018)	Darma Setiawan Putra	<ul style="list-style-type: none"> <li>- Menggunakan <i>Fingerprint</i></li> <li>- Metode RAD</li> <li>- Tidak menggunakan <i>email gateway</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menggunakan <i>Fingerprint</i></li> <li>- Metode RAD</li> <li>- Tidak menggunakan <i>email gateway</i></li> </ul>
3.	Pelatihan Kompetensi Guru dalam Menggunakan <i>Qr-Code</i> Berbasis Android Pelatihan Kompetensi Guru dalam Menggunakan <i>Qr-Code</i> Berbasis Android untuk	Yodang	<ul style="list-style-type: none"> <li>- Tidak menggunakan <i>email gateway</i></li> <li>- Aplikasi berbasis <i>Android</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menggunakan <i>emailgateway</i></li> <li>- Berbasis Web</li> </ul>

	<i>Presensi Siswa (2021)</i>			
4.	Sistem Presensi Kelas Menggunakan Pengenalan Wajah Dengan Metode Haar Cascade Classifier (2020)	Rastri Pratihvi	<ul style="list-style-type: none"> <li>- Metode Haar Cascade Classifier</li> <li>- Tidak menggunakan email gateway</li> </ul>	<ul style="list-style-type: none"> <li>- Menggunakan emailgateway</li> <li>- Metode prototype</li> <li>- Database MySql</li> </ul>
5	Pada Bawaslu Sumatera Utara Menggunakan Qr-Code Dan Base64 (2020)	Ali Aman	<ul style="list-style-type: none"> <li>- Tidak menggunakan Email gateway</li> <li>- Menggunakan Algoritma Base 64</li> </ul>	<ul style="list-style-type: none"> <li>- Menggunakan emailgateway</li> <li>- Menggunakan metodeprototype</li> </ul>

## 2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

### 2.2.1 Sistem Informasi

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. Informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya dan untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan [9].

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 [10].

### 2.2.2 Presensi dan *Qr-Code*

Presensi adalah suatu bentuk kegiatan ataupun ritinitas yang telah dilakukan oleh seseorang individu yang dapat berguna untuk membuktikan bahwa dirinya hadir atau tidak hadir. Sistem presensi yang pada awal mulanya hanya dilakukan secara manual seperti halnya tanda tangan diatas kertas, pada era sekarang menjadi lebih maju yaitu dilakukan dengan cara menggunakan alat elektronik berupa penggunaan presensi [11].

*Qr-Code* atau *Quick Response Code* merupakan sebuah kode yang mampu menyimpan dan memberikan data (angka/numerik, alpha numerik, biner, kanji/kana) dengan respon yang cepat, kode ini berupa matriks dua dimensi. *Qr-Code* biasanya berbentuk persegi putih kecil dengan bentuk geometris hitam, meskipun saat ini banyak yang telah

berwarna dan digunakan sebagai brand produk. Informasi yang dikodekan dalam *Qr-Code* dapat berupa URL, nomor telepon, pesan SMS, *V-Card* atau teks apapun [12].

### 2.2.3 PHP Mailer

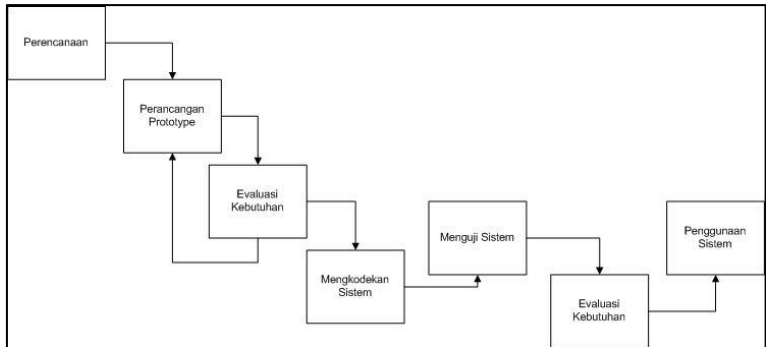
PHP Mailer merupakan salah satu *library* yang disediakan dan bersifat free. Library ini dapat dijalankan di berbagai macam *framework PHP*, *Library* ini memungkinkan *user* dapat memanfaatkan e-mail sebagai media interaksi antara user dengan sistem. PHP Mailer adalah salah satu *library PHP open source* 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 pemberitahuan jadwal rapat untuk melakukan rapat pimpinan. PHP Mailer digunakan juga sebagai notifikasi undangan rapat telah dibuat dan dikirimkan kepada peserta rapat [13].

### 2.2.4 Rekayasa Web

Rekayasa Web merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin [14]. Berikut adalah metode dan tools yang digunakan:

#### A. Metode Pengembangan Sistem

Metode dapat diartikan sebagai menyediakan cara bagaimana secara teknis membangun perangkat lunak yang harus berada pada sebuah komitmen dasar menuju kualitas. Metode yang digunakan pada penelitian ini adalah metode *prototype*. Metode ini dibuat secara terstruktur dan memiliki beberapa tahap-tahap yang harus dilalui dalam pembuatannya namun jika tahap final dinyatakan bahwa sistem yang telah dibuat belum sempurna maka sistem dievaluasi kembali [15].



**Gambar 2. 1** Pemodelan prototype [15].

Metode *prototype* memiliki 6 tahapan, yaitu [15] :

1. Perencanaan

tahap ini merupakan tahap dari perencanaan yang akan dibuat dimulai dari penyebaran kuisioner yang diisi oleh pengguna sistem.

2. Perancangan *prototype*

Pada tahap ini merupakan bagian untuk perancangan tampilan *prototype* dengan membuat perancangan sementara dengan menggunakan aplikasi *balsamic mockup* yang berfokus pada penyajian kepada pengguna (misalnya dengan membuat format *input* maupun *output*).

3. Evaluasi kebutuhan

Pada tahap ini merupakan bagian untuk mengevaluasi *prototype*, selain itu pada tahap ini juga dilakukan oleh pengguna apakah *prototype* yang sudah dibangun sudah sesuai dengan keinginan pengguna.

4. Mengkodekan sistem

Pada tahap ini *prototype* yang sudah disepakati maka akan diterjemahkan kedalam bahasa pemrograman yang sesuai.

5. Evaluasi sistem

Pada tahap ini merupakan tahap pengguna mengevaluasi apakah sistem yang sudah dibangun sudah sesuai dengan yang diharapkan. Jika ya, langkah 6 dilakukan, jika tidak, ulangi langkah 4 dan 5. pada tahap evaluasi sistem ini pengembang menampilkan hasil dari mengkodekan sistem, dan menguji sistem dengan melibatkan pengguna sistem.

## 6. Menggunakan sistem

Tahap terakhir dari metode penelitian ini adalah menggunakan sistem. Pada tahap ini sistem yang telah dievaluasi siap untuk digunakan.

## B. Metode Pengujian Sistem

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 [16] :

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 kontrol, maka perhatian berfokus pada domain informasi.

Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut [17].

1. Bagaimana validitas fungsional diuji?
2. Kelas *input* apa yang akan *test case* menjadi baik?
3. Apakah sistem sangat sensitif terhadap harga *input* tertentu?
4. Bagaimana batasan dari suatu data diisolasi?
5. Kecepatan data apa dan volume data apa yang dapat ditolelir oleh sistem?
6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

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. 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.

## C. *Tools/ Alat Bantu Penelitian*

### 1. *Flowchart*

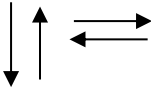


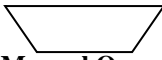



*Flowchart* adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi [17]. Ada beberapa jenis-jenis *flowchart* diantaranya :

- 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. 2 Simbol Flowchart [17]

No	Simbol	Keterangan
1	 <p><b>Flow Direction Symbol</b></p>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
2	 <p><b>Terminator Symbol</b></p>	Yaitu simbol untuk permulaan ( <i>start</i> ) atau akhir ( <i>end</i> ) dari suatu kegiatan.
3	 <p><b>Processing Symbol</b></p>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4	 <p><b>Manual Operation Symbol</b></p>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
6	 <p><b>Decision Symbol</b></p>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
7	 <p><b>Input-Output Symbol</b></p>	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
6	 <p><b>Document Symbol</b></p>	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

## 2. *Unified Modeling Language (UML)*

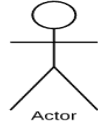

*Unified Modeling Language (UML)* adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design (OOAD)*. Macam-macam dari *Unified Modeling Language (UML)* antara lain [17]: *use case* diagram, *sequence* diagram dan *class* diagram.





### a. *Use Case*





*Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor* [17].

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. 3** Simbol *Use Case Diagram* [17]

No	Gambar	Nama Gambar	Keterangan
1.		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case
2.		Dependency	Hubungan dimana perubahan yang

			terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri ( <i>dependent</i> ).
3.		Generalization	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk
4.		Include	Menspesifikasikan bahwa use case sumber secara eksplisit
5.		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6.		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

7.		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi suatu aktor.
9.		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

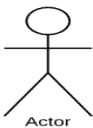
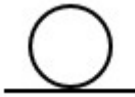
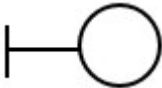


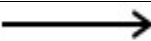
**b. Sequence Diagram**

*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 obyek 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 [17]. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2. 4** Simbol Sequence Diagram [17]

No	Gambar	Nama Gambar	Keterangan
1.	 A stick figure with a circle for a head, a horizontal line for shoulders, and two diagonal lines for legs. The word "Actor" is written below the figure.	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.	 A circle with a horizontal line underneath it.	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
3.	 A circle with a horizontal line extending to the left from its center.	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari <i>form</i>
4.	 A circle with a curved arrow pointing clockwise from the top.	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel
5.	 A vertical rectangle with dashed lines extending from its top and bottom centers.	<i>A Focus of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>message</i>
6.	 A horizontal arrow pointing to the right.	<i>A Message</i>	Menggambarkan Pengiriman Pesan

### 2.2.5 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis [18].

Sistem berorientasi objek merupakan 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.

### 2.2.6 Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk penginformasian suatu perusahaan instansi, dalam bahasan tertentu [19].

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)*.

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;

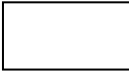

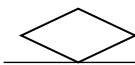

- c. UPDATE: perintah update digunakan untuk memperbaharui data pada sebuah tabel. Contoh: UPDATE nama\_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;
- d. DELETE: perintah delete digunakan untuk menghapus data dari sebuah tabel. Contoh: DELETE FROM nama\_tabel WHERE kolom=data;

### 2.2.7 Entity Relationship Diagram (ERD)

ERD adalah 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 hubpeungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya [17].

Berikut merupakan simbol-simbol dari ERD :




**Tabel 2. 5** Simbol ERD[17]

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. 6** Macam-Macam *Kardinalitas* [17]

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.		Relasi Banyak ke Banyak ( <i>Banyak to Many</i> )	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya

*~Halaman Ini Sengaja Dikosongkan~*