

1. DDL (*Data Definition Language*) merupakan perintah yang dipakai untuk mendefinisikan suatu struktur data, misalnya membuat *database*, *field*, dan lain sebagainya (Yuhendra, 2014).
2. DML (*Data Manipulation Language*) merupakan perintah yang dipakai untuk membuat CRUD (*Create, Read, Update, Delete*) yang digunakan untuk memanipulasi sebuah data (Yuhendra, 2014).

Kelebihan dalam menggunakan MySQL ini yaitu:

1. Mempunyai kemampuan untuk dikembangkan lagi dengan arti lain yaitu bersifat *open source* (Sahertian, Setiabudi & Santoso, 2015).
2. Bahasa yang digunakan berstandar bahasa dunia dalam mengolah data yaitu bahasa SQL (Sahertian et al., 2017).
3. Mampu dijalankan dalam berbagai sistem operasi (Sahertian et al., 2017).
4. Dapat digunakan oleh beberapa pengguna dalam waktu yang sama tanpa terjadi konflik (Sahertian et al., 2017).

Struktur dasar pembentuk SQL yaitu *select*, *from* dan *where* sedangkan struktur untuk memodifikasi data yaitu *delect*, *insert* dan *update* (Sovia & Febio, 2017).


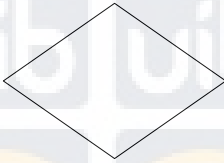


2.2.8 *Entity Relationship Diagram (ERD)*

ERD merupakan suatu model atau teknik yang menyatakan hubungan suatu model. Dalam hubungan ini paling utama yaitu objek dari suatu data yang disebut dengan entitas dan hubungan yang disebut juga dengan *relation*. Jadi ERD ini merupakan hubungan antar objek data (Fridayanthie & Mahdiati, 2018).

Fungsi dari ERD yaitu menggambarkan pemodelan basis data relasional dimana bisa kita lihat bagaimana hubungan relasi antar tabel yang ada pada

database (Sari & Yulianti, 2015). Sedangkan manfaat dari menggunakan ERD yaitu untuk mengetahui kunci relasi antar tabel (Haryati, Khairul, & Nadriati, 2018).

Tabel 1 dibawah ini menunjukkan simbol dari ERD:

No	Gambar	Nama	Keterangan
1		Entitas	Menandakan sebuah objek atau entitas
2		Relation	Menandakan sebuah relasi antar entitas
3		Link	Menandakan kardinalitas antar relasi
4		Atribut	Menandakan atribut yang dimiliki oleh suatu entitas




Tabel 1 Simbol ERD

2.2.9 Unified Modelling Language (UML)

Unified Modelling Language adalah bahasa yang digunakan untuk merancang sebuah sistem. UML dijelaskan sebagai sebuah bahasa visual untuk menjelaskan suatu model dari sebuah sistem. Macam-macam UML adalah sebagai berikut:

1. *Use Case Diagram* merupakan sebuah fungsionalitas yang diprediksikan dalam sebuah sistem. *Use Case* juga digunakan untuk menjelaskan interaksi antar pengguna dengan sistem.

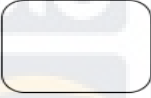



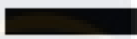
Tabel 2 dibawah ini menggambarkan simbol-simbol yang terdapat pada *use case diagram*.

NO	GAMBAR	NAMA	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 (<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 <i>eksplisit</i> .
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>	Apa 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 aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama 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

Tabel 2 Simbol *Use Case Diagram*

2. *Activity Diagram* adalah diagram khusus yang menjelaskan alir-alir aktifitas dalam sebuah sistem yang akan dirancang, bagaimana alir dimulai dan keputusan yang kemungkinan akan terjadi serta bagaimana alir tersebut berakhir. Berikut ini merupakan simbol-simbol yang digunakan untuk membuat *activity diagram*:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Tabel 3 Simbol *Activity Diagram*