

## BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Sebagai acuan dalam pengembangan Pembangunan Sistem Informasi Pengolahan Rapor Siswa Pada SDN Tritih Kulon 05 Cilacap Utara Berbasis Web dan Notifikasi Telegram, maka penulis mencari referensi beberapa penelitian terdahulu yang berkaitan secara langsung maupun tidak langsung dengan penelitian yang penulis kembangkan.

Penelitian terdahulu yang dilakukan oleh Evi Lestari Pratiwi, Muhammad Zaki Azhari, M.Robby Ariyandi (2022) yang berjudul "Sistem Informasi Raport Online Berbasis Web Pada SMA PGRI 1 Banjarmasin". Penelitian ini bertujuan untuk membangun sebuah sistem informasi pengolahan rapor *online* berbasis web di SMA PGRI 1 Banjarmasin. Sistem informasi rapor *online* berbasis web dirancang menggunakan bahasa pemrograman PHP dan basis data MySQL. Metode pengembangan yang dipergunakan adalah metode *Waterfall*. Penggunaan sistem menggunakan aplikasi XAMPP sebagai Web Server lokal yang dipergunakan pada saat sistem dikembangkan. Agar sistem dapat berjalan secara *online*, maka sistem harus diunggah ke *hosting*. Hasil dari penelitian ini yaitu sistem informasi rapor *online* berbasis web dirancang menggunakan bahasa pemrograman PHP dan basis data MySQL yang dapat meningkatkan efektivitas kerja seperti *input* nilai siswa, data siswa, guru, mata pelajaran dan laporan hanya memerlukan waktu singkat dan sistem memberikan informasi yang lebih jelas dan rinci serta membuat lebih hemat karena sedikit dalam penggunaan alat tulis [4].

Penelitian selanjutnya yang dilakukan oleh Dhea Arista Putri, M. Azhar Irwansyah, dan Enda Esyudha Pratama (2019) yang berjudul "Rancang Bangun Sistem Informasi Pengolahan Nilai Rapor Berbasis *Website* pada SMP Negeri 16 Pontianak" membahas tentang sistem informasi pengolahan nilai rapor berbasis *website* yang di dalamnya dapat manajemen seluruh data-data siswa, guru, nilai harian, nilai tugas, nilai ulangan tengah semester, nilai ulangan akhir semester, dan sebagainya. Metode yang digunakan yaitu dengan metode *waterfall* dimana hal ini menggambarkan pendekatan yang sistematis dan berurutan pada pengembangan perangkat lunak. Pengujian yang dilakukan terhadap sistem dengan metode UAT (*User Acceptance Test*). Dari hasil pengujian *User Acceptance Test* (UAT) yang telah dilakukan menunjukkan sistem

informasi berjalan dengan sangat baik sesuai fungsinya dan dapat diimplementasikan pada SMP Negeri 16 Pontianak [5].

Penelitian sejenis yang dilakukan oleh Aziza Femi Winanda, Ari Parulian, Dekra Wond Dwi Putra Dieta, Dinni Wahyuni, Ovie Julian Trivani, Muhammad Luthfi Hamzah (2022) yang berjudul “Perancangan Sistem Informasi E-Rapor Berbasis Web Pada Sekolah SMP 4 Muhammadiyah Pekanbaru Menggunakan Agile Method”. Penelitian ini bertujuan untuk mempermudah siswa dalam melihat nilai dan mendapatkan informasi seputar rapor pada SMP Muhammadiyah 4 Pekanbaru. Dengan pemanfaatan e-rapor tersebut merupakan solusi untuk menyelesaikan masalah tersebut. Dalam perancangan dan implementasi sistem e-rapor menggunakan metode agile, dimana Agile ini merupakan model pengembangan yang dilakukan secara iteratif, berulang ulang, dan dapat mengalami perubahan apabila diperlukan, dan menggunakan *software* lainnya seperti sublime text, bootstrap, mysql dan php. Hasil akhir dari penelitian ini merupakan sebuah rancangan *website* yang dapat menyelesaikan masalah penyampaian informasi rapor yang dapat dilaksanakan secara cepat efektif dan efisien [6].

Penelitian serupa yang dilakukan oleh Anisah Nurul Salma, Heri Mulyono, Angгри Yulio Pernanda (2022) yang berjudul “Perancangan Sistem Informasi Pengolahan Nilai Rapor Berbasis Web”. Penelitian ini bertujuan untuk membuat sistem untuk mengumpulkan, mengolah, menyimpan dan menyediakan informasi. Salah satu pengolahan data di SMK adalah pengelolaan data siswa seperti data siswa, data prestasi siswa, data mata pelajaran, dan jurusan. Penelitian ini menggunakan metode *Software Development Life Cycle* (SDLC). Perancangan sistem dibuat dengan bahasa permodelan menggunakan *Unified Modeling Language* (UML). Sistem informasi pengolahan nilai rapor dibuat menggunakan bahasa pemrograman PHP, HTML dengan menggunakan *database* MYSQL serta pengujian *blackbox* dan pengujian *whitebox*. Hasil penelitian ini merupakan sistem informasi yang dirancang untuk dapat mempermudah guru dan *staff* tata usaha dalam pengelolaan nilai siswa, serta penyimpanan nilai rapor siswa [7].

Penelitian lainnya yang dilakukan oleh Ade Irma Kusuma Wardani dan Andi Iwan Nurhidayat (2019) yang berjudul “Rancang Bangun Sistem Informasi *Raport* Online Berbasis *Website* Menggunakan *Framework* Laravel”. Penelitian ini bertujuan membantu wali murid dan guru dalam hal pengolahan nilai siswa. Penelitian ini menggunakan metode *Waterfall* dan *Framework* Laravel. Hasil dari penelitian ini adalah

sistem informasi *raport online* yang dibuat dengan tujuan memudahkan dalam hal pengolahan siswa dan dapat dibuktikan dengan pengujian sistem yang dilakukan penyebaran kuisioner ke semua guru dan sebagian wali murid di SD Negeri Pagak Kabupaten Pasuruan. Hasil dari akumulasi kuisioner responden guru mendapatkan skor 70% dengan kategori kelayakan “baik”. Sedangkan hasil akumulasi kuisioner dari wali murid mendapatkan 82% dengan kategori kelayakan “sangat baik” [8].

Penelitian kali ini bertujuan untuk membangun sistem informasi pengolahan nilai rapor di SDN Tritih Kulon 05 Cilacap Utara berbasis web dan notifikasi Telegram. Sistem ini menggunakan PHP sebagai bahasa pemrograman dan Codeigniter sebagai *framework*. Metode penelitian yang digunakan adalah metode RAD dan teknik pengumpulan data yang digunakan adalah observasi, wawancara, dan studi pustaka. Kelebihan sistem ini yaitu menggunakan notifikasi Telegram untuk pemberitahuan kepada siswa bahwa nilai sudah dapat dilihat dalam sistem dan dapat menampilkan grafik kepada kepala sekolah untuk melihat perkembangan nilai siswa. Hasil dari penelitian ini berupa *website* sistem informasi yang dapat membantu mengelola data-data dan nilai siswa serta sebagai penyimpanan yang terkomputerisasi.

## **2.2 Landasan Teori**

### **2.2.1 Sistem**

Menurut Romney (2004), pengertian sistem merupakan suatu rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sedangkan menurut Hall (2007), sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama. Sistem terdiri dari tiga unsur yaitu *input* (masukan), proses dan *output* (pengeluaran) [9].

### **2.2.2 Informasi**

Menurut Kusriani dan Koniyo (2007) adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi.

Menurut Susanto (2004), terdapat tiga hal penting yang harus diperhatikan mengenai informasi yaitu informasi merupakan hasil pengolahan data, informasi mengandung makna atau arti, informasi berguna atau bermanfaat. Pada tahun 2008, Susanto menyatakan bahwa

menyatakan bahwa suatu informasi yang berkualitas harus memiliki ciri-ciri sebagai berikut:

1. Akurat, artinya informasi harus mencerminkan keadaan yang sebenarnya.
2. Tepat waktu, artinya informasi itu harus tersedia atau ada pada saat informasi tersebut diperlukan, tidak besok atau tidak beberapa jam lagi,
3. Relevan, artinya informasi yang diberikan harus sesuai dengan yang dibutuhkan oleh individu yang ada di berbagai tingkatan dan bagian dalam organisasi,
4. Lengkap, artinya informasi harus diberikan secara lengkap [9].

### **2.2.3 Sistem Informasi**

Menurut Jogiyanto (2005), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan oleh pihak luar tertentu [10].

### **2.2.4 Rapor**

Rapor siswa adalah penerapan berbagai macam cara dan penggunaan beragam alat penilaian untuk memperoleh informasi tentang sejauh mana ketercapaian kompetensi pada peserta didik. Hasil penilaian dapat berupa nilai kualitatif dan nilai kuantitatif, proses penilaian siswa adalah untuk mengambil keputusan dengan menggunakan informasi yang diperoleh melalui pengukuran hasil belajar yang baik menggunakan tes [11].

### **2.2.5 Nilai**

Fraenkel (1977:6) menyatakan bahwa idea atau konsep yang bersifat abstrak tentang apa yang dipikirkan seseorang atau dianggap penting oleh seseorang, biasanya mengacu kepada estetika(keindahan), etika pola perilaku dan logika benar salah atau keadilan [12].

### **2.2.6 Website**

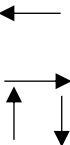


Menurut Ahmad Fadel (2018), *website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara


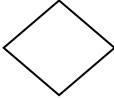
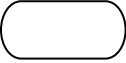


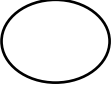
animasi sehingga lebih merupakan media informasi yang menarik untuk dikunjungi. Untuk membangun sebuah *website* yang lebih *advanced*, juga diharuskan untuk menyediakan unsur penunjang lainnya, di antaranya nama domain, *web hosting*, bahasa pemrograman, desain *website*, dan publikasi *website* [13].

### 2.2.7 Flowchart

*Flowchart* atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Analisis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun. Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung [14]. Berikut akan dijelaskan mengenai simbol-simbol *flowchart* seperti pada Tabel 2.1.

**Tabel 2. 1** Simbol-Simbol *Flowchart*

No.	Simbol	Nama	Fungsi
1.		Arus/Flow	Menyatakan jalannya arus suatu proses.
2.		Offline Connector	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
3.		Proses	Simbol yang menunjukkan setiap pengolahan yang akan dilakukan oleh komputer.

4.		Manual	Menyatakan suatu tindakan proses yang tidak dilakukan oleh komputer.
5.		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak.
6.		<i>Terminal</i>	Menyatakan awal atau akhir suatu program.
7.		<i>Input-output</i>	Menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
8.		<i>Document</i>	Mencetak laporan ke printer.
9.		<i>Connector (on-page)</i>	Menghubungkan proses ke proses lainnya dalam halaman/lembar yang sama.

### 2.2.8 UML (*Unified Modeling Language*)


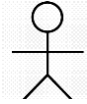



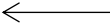

*UML (Unified Modeling Language)* adalah salah satu model untuk merancang pengembangan *software* yang berbasis *object-oriented* dan memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software* [15].

#### a. *Use Case Diagram*

*Use case* merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. *Use case* mendefinisikan apa yang akan diproses oleh sistem dan komponen-

komponennya. *Use case* bekerja dengan menggunakan skenario yang merupakan deskripsi dari urutan atau langkah-langkah yang menjelaskan apa yang dilakukan oleh *user* terhadap sistem maupun sebaliknya [16]. Seperti pada Tabel 2.2.




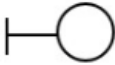


**Tabel 2.2** Simbol-Simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Fungsionalitas yang tersedia pada sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.		<i>Actor</i>	Men-spesifikasikan 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>	Men-spesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Men-spesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<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> ).
7.		<i>System</i>	Men-spesifikasikan paket yang menampilkan sistem secara terbatas

#### b. *Sequence Diagram*

*Sequence diagram* adalah sebuah diagram yang menggambarkan kolaborasi dari objek-objek yang saling berinteraksi antar elemen dari suatu *class* [17]. Simbol-simbol dalam *sequence* dapat dilihat pada tabel di bawah ini. Seperti pada Tabel 2.3.

**Tabel 2.3** Simbol -Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
2.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

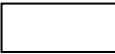
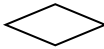


**c. *Entity Relationship Diagram (ERD)***

*Entity Relationship Diagram* adalah alat perancangan untuk memodelkan suatu basis data. Tujuan dibuatnya *Entity Relationship Diagram* adalah untuk pemodelan yang menunjukkan hubungan antar



data yang mempunyai relasi, juga mendokumentasikan data yang ada dengan cara menjelaskan tiap masing-masing data dan relasinya [18]. Simbol ERD ditunjukkan pada Tabel 2.4.

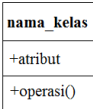
**Tabel 2. 4** Simbol-Simbol ERD





No	Simbol	Nama	Keterangan
1		<i>Entitas</i>	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2		Relasi	Menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
3		<i>Association</i>	Penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.
4		Atribut	Mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key yang diberi garis bawah).

#### d. *Class Diagram*

*Class* diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. Diagram kelas ini sesuai jika diimplementasikan ke proyek yang menggunakan konsep *object-oriented* karena gambaran dari *class* diagram cukup mudah untuk digunakan [19]. Simbol *class* diagram ditunjukkan pada Tabel 2.5

**Tabel 2. 5** Simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.

2.		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3.		Asosiasi berarah/ <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
5.		Kebergantungan <i>/ Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.

### 2.2.9 Rekayasa Perangkat Lunak

Menurut IEEE, rekayasa perangkat lunak adalah aplikasi dari sebuah pendekatan disiplin dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak yaitu aplikasi dari rekayasa perangkat lunak. Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat-alat bantu yang akan dipakai, kontrol, serta penyampaian yang dibutuhkan. Model-model proses untuk *software engineering* seperti model sekuensial linier atau *waterfall* model, model prototipe, model RAD, model inkremental, model spiral, model pengembangan terpadu, model metode formal, dan model teknik generasi keempat [20].

#### a. Metode Pengembangan Sistem

Pemodelan dalam pengembangan perangkat lunak harus dilakukan dengan tujuan efisiensi dan efektivitas proses, menjamin kualitas serta sesuai dengan keinginan pengguna. Metode yang digunakan pada

penelitian ini adalah model *Rapid Application Development (RAD)*. *Rapid Application Development (RAD)* adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental*(bertingkat) [21]. Metode ini menggunakan metode *iteratif*(berulang) dalam mengembangkan sistem dimana model kerja sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan pengguna. Berikut adalah tahapan dari metode pengembangan sistem RAD:

1. Perencanaan Syarat-Syarat (*Requirements Planning*)  
Fase perencanaan syarat-syarat merupakan tahap awal dari pengembangan sistem. Pada tahap ini peneliti harus mengumpulkan informasi di SDN Tritih Kulon 05 Cilacap Utara terkait sistem yang akan dikembangkan.
2. *Workshop* Desain RAD (*RAD Design Workshop*)  
Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Seorang analisis sistem melakukan proses desain menggunakan *Usecase Diagram*, *Sequence Diagram*, dan *Class Diagram*. Desain diimplementasikan menjadi perangkat lunak melalui bahasa-bahasa yang dimengerti oleh komputer menggunakan Codeigniter untuk *framework*, PHP untuk bahasa pemrograman, dan MySQL untuk menyimpan *database*.
3. Implementasi (*Implementation*)  
Implementasi merupakan fase dimana peneliti membutuhkan tanggapan akan sistem yang sudah dibuat dan mendapat persetujuan mengenai sistem tersebut.

#### **b. Metode Pengembangan Sistem**

Pengujian sangat dibutuhkan untuk mengetahui sistem yang dibuat sudah sesuai tanpa adanya kesalahan dan untuk memperbaiki jika terjadi kesalahan. Metode yang digunakan dalam pengujian ini yaitu *Blackbox Testing*. *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detailnya [22]. *Blackbox Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program [23].

### 2.2.10 Pemrograman Berorientasi Objek(PBO)

Pemrograman Berorientasi Objek (PBO) adalah sebuah bahasa pemrograman yang memandang segala sesuatu menjadi sebuah objek. Paradigma dari PBO adalah menyelesaikan masalah dengan merepresentasikan masalah ke model objek. Karakteristik dari PBO adalah:

1. *Class* dan Objek.

*Class* dapat dikatakan sebagai cetak biru (*blue print*) dari sebuah objek ataupun beberapa objek. *Class* diartikan juga menggambarkan ciri-ciri objek secara umum. Sebagai contoh Nokia, Samsung, Blackberry dan Xiaomi merupakan objek dari *class handphone*. Nokia dan objek yang lainnya mempunyai atribut yang sama diantaranya memiliki tipe, tahun rilis, dimensi, berat, fitur dan lain sebagainya. Selain atribut, objek-objek tersebut memiliki *methode* yang sama diantaranya kunci tombol, menampilkan teks dilayar, kirim sms dan lain sebagainya.

2. *Encapsulation/Enkapsulasi*.

Disebut juga dengan pembungkusan, yaitu melindungi program dan data yang sedang diolah agar tidak diakses secara sembarangan oleh program lain. Dalam *java*, dasar enkapsulasi adalah *class*. Variabel atau *methode* pada sebuah *class* tidak dapat diakses oleh *class* lain dengan menjadikan *class* tersebut bersifat *private*, atau menjadikan *class* tersebut bersifat *protected*, yang hanya bisa diakses oleh turunannya (*inheritance*) atau menjadikan *class* tersebut bersifat *public*, sehingga bisa diakses oleh sembarang *class*.

3. *Inheritance/Inheritansi*

Disebut juga dengan turunan. Prinsipnya adalah sebuah *class* dapat diturunkan dari *class* yang lain. *Class* yang menurunkan ke *class* lain disebut dengan superclass, parent class atau *base class* atau kelas induk, sedangkan *class* yang merupakan turunan disebut sebagai *subclass*, *child class* atau *derived class* atau *class* turunan. *Class* turunan secara otomatis memiliki sifat (*variable*) dan kelakuan (*behavior, method*) yang dimiliki oleh super *class*-nya. *Class* turunan dapat menambahkan fitur atau *behavior* dengan mendefinisikan suatu *method* di dalam *class* turunan tersebut.

4. *Polimorphisme/Polimorfisme*.

Polimorfisme secara bahasa dapat diartikan dengan memiliki banyak bentuk. kegunaan dari polimorfisme adalah agar dapat mendefinisikan beberapa konstruktor atau metode dengan

karakteristik yang berbeda-beda agar nantinya dapat digunakan untuk kasus-kasus yang berbeda. Method atau perilaku yang sama tapi implementasinya/caranya yang berbeda-beda [24].

### 2.2.11 Basis Data (*Database*)

*Database* atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Hal ini bertujuan untuk menyederhanakan interaksi antara pengguna dengan sistemnya [25]. Salah satu server yang menyediakan layanan sistem manajemen basis data adalah MySQL.

### 2.2.12 MySQL

MySQL adalah sebuah perangkat lunak yang terdapat di dalam sistem manajemen basis data SQL (*database management system*) atau yang biasa disebut DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia [26].

### 2.2.13 DDL (*Data Definition Language*)

DDL (*Data Definition Language*) merupakan suatu perintah yang digunakan untuk menciptakan struktur data, atau untuk membangun *database*. DDL mempunyai tugas untuk membuat objek SQL dan menyimpan definisinya dalam tabel. Contoh objek tersebut yaitu tabel, *view*, dan *index*. DDL mempunyai fungsi untuk melakukan perubahan struktur tabel, mengubah nama tabel. Berikut perintah-perintah yang ada dalam golongan DDL:

- a. *Create*, digunakan untuk membuat *database*, tabel, dan objek lain dalam *database*.
- b. *Alter*, digunakan untuk memodifikasi tabel, seperti contoh mengubah nama tabel, *field*, menambah nama *field*.
- c. *Drop*, digunakan untuk menghapus *database*, tabel, dan objek lain dalam *database* [27].

### 2.2.14 DML (*Data Manipulation Language*)

DML (*Data Manipulation Language*) merupakan basis data yang digunakan untuk melakukan modifikasi dan pengambilan data pada suatu *database*, pengolahan atau modifikasi ini meliputi:

- a. *Insert*, digunakan untuk melakukan penambahan data.

Contoh:               INSERT               INTO               kepalasekolah  
(nama,no\_induk,jenis\_kelamin)               VALUES               ('Kartika  
Wahyuningsih', '19880910 201001 2 013', 'Perempuan');

- b. *Select*, digunakan untuk pengambilan data  
Contoh: `SELECT * FROM kepalasekolah;`
- c. *Update*, digunakan untuk melakukan perubahan data  
Contoh: `UPDATE kepalasekolah SET nama='Kartika Wahyuningsih,S.Pd.SD' WHERE id = '1';`
- d. *Delete*, digunakan untuk melakukan penghapusan data.  
Contoh: `DELETE FROM kepalasekolah WHERE id = "1";`
- e. *Drop database*, digunakan untuk menghapus database.  
Contoh: `DROP DATABASE siroline;`
- f. *Drop table*, digunakan untuk menghapus tabel.  
Contoh: `DROP TABLE kepalasekolah;`
- g. *Group by*, digunakan untuk mengelompokkan data.  
Contoh: `GROUP BY nama, no_induk;`
- h. *Order by*, digunakan untuk menampilkan semua data secaraurut berdasarkan abjad nya  
Contoh : `ORDER BY nama;` [27].

### 2.2.15 Framework

*Framework* adalah sebutan untuk sebuah kerangka kerja yang digunakan para developer aplikasi atau *software* untuk mempermudah mereka dalam membuat maupun mengembangkan sebuah *software* atau aplikasi. *Framework* sendiri berisikan fungsi dasar dan perintah yang lazim dipakai untuk membuat dan mengembangkan sebuah *software* atau aplikasi, dengan harapan aplikasi yang dibuat bisa dibangun secara lebih terstruktur, lebih cepat serta lebih tersusun dengan cukup rapi [28].

### 2.2.16 Codeigniter

Menurut Septian (2011), CodeIgniter merupakan aplikasi *open source* berupa *framework* PHP dengan model MVC (Model, View, Controller) untuk membangun aplikasi web dinamis dengan cepat dan mudah. CodeIgniter ini memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Artinya bahwa CodeIgniter masih memberi kebebasan kepada para pengembang untuk menuliskan bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau dengan *syntax* umum di dalam PHP tidak harus menggunakan aturan penulisan kode di CodeIgniter [29].