



**BAB II**  
**TINJAUAN PUSTAKA**  
**LANDASAN TEORI**



## **BAB II**

### **TINJAUAN PUSTAKA LANDASAN TOERI**

#### **2.1 Tinjauan Pustaka**

Penelitian serupa pernah dilakukan Camelia Atmaja (2022) yang berjudul “Sistem Informasi Perpustakaan *Digital (Digital Library)* Pada Perpustakaan Smk Tamansiswa Bekasi”. Sistem yang dibuat bertujuan untuk mempercepat pengguna dalam memperoleh kebutuhan informasi dan membuat sistem layanan perpustakaan di Smk Taman Siswa tersistematis sehingga Sistem Informasi Perpustakaan *Digital (Digital Library)* pada Perpustakaan SMK TamanSiswa dapat lebih efektif dan efisien maka peneliti melakukan penelitian dengan judul “Sistem Informasi Perpustakaan *Digital (Digital Library)* Pada SMK TamanSiswa Bekasi”. Sistem ini dibangun menggunakan bahasa pemrograman *Python* dan *database MySQL*. [6]

Penelitian lainnya yang serupa pernah dilakukan oleh Florentina J.I Lo yang berjudul “Perancangan Sistem Informasi Perpustakaan Sekolah Berbasis *Web (Studi Kasus : SMP Negeri 1 Teluk Dalam)*”. Sistem yang dibuat bertujuan untuk pengolahan data secara cepat dan tepat, serta sistem dari perpustakaan itu sendiri bagaimana sehingga dapat mempermudah para pegawai perpustakaan dan membuat para peserta didik khususnya nyaman untuk belajar dan juga dalam melakukan peminjaman dan pengembalian buku di perpustakaan tersebut. Sistem ini dibangun menggunakan bahasa pemrograman *PHP* dan *database MySQL*. [7]

Penelitian lainnya yang serupa pernah dilakukan oleh Nadya Sari yang berjudul “Rancang Bangun Sistem Informasi Perpustakaan Berbasis *Website* Di Sman 3 Abdya” Sistem yang dibuat bertujuan untuk menciptakan sebuah sistem informasi perpustakaan berbasis *web* yang akan memberi kemudahan bagi anggota untuk mendapatkan buku-buku yang diperlukan, kemudahan untuk petugas perpustakaan mengolah data peminjaman, pengembalian, dan pemeliharaan koleksi buku. Sistem ini dibangun menggunakan bahasa pemrograman *PHP* dan *database MySQL*. [8]

Penelitian lainnya yang serupa pernah dilakukan oleh Dewi Wulan Dari “Rancang Bangun Aplikasi Perpustakaan Sekolah Berbasis *Website*” Sistem yang dibuat bertujuan untuk menghasilkan aplikasi perpustakaan

berbasis situs *web*, yang akan membantu petugas dalam mengelola data perpustakaan dan memfasilitasi pencarian data buku, data anggota dan meminjam dan mengembalikan buku. Sistem ini dibangun menggunakan bahasa pemrograman *PHP* dan *database MySQL*. [9]

Penelitian lainnya yang serupa pernah dilakukan oleh Nur Aini “Pembangunan Sistem Informasi Perpustakaan Berbasis Web menggunakan Metode *Rapid Application Development (RAD)* (Studi pada : SMK Negeri 11 Malang)” Sistem yang dibuat bertujuan untuk menghasilkan aplikasi perpustakaan SMKN 11 Malang yang berbasis *web* dengan tujuan agar staff perpustakaan dalam pengelolaan data – data perpustakaan, sistem tersebut bisa diakses kapanpun dan dimanapun. Sistem ini dibangun menggunakan bahasa pemrograman *PHP* dan *database MySQL*[10]

**Tabel 2. 1** Penelitian Sebelumnya

No	Judul	Peneliti (tahun)	Tujuan	Metode	Hasil
1	Sistem Informasi Perpustakaan <i>Digital (Digital Library)</i> Pada Perpustakaan Smk Tamansiswa Bekasi	Camelia Atmaja, Mochammad Arief Sutisna (2022)	memprcepat pengguna dalam memperoleh kebutuhan informasi dan membuat sistem layanan perpustakaan di Smk Taman Siswa tersistematis dan dapat Lebih	<i>Rapid Application Development (RAD)</i>	Penelitian ini menghasilkan sistem informasi yang diberi nama “Sistem Informasi Perpustakaan <i>Digital (Digital Library)</i> Pada Perpustakaan Smk Tamansiswa

			Efektif dan efisien		wa Bekasi”.
2	Perancangan Sistem Informasi Perputakaan Sekolah Berbasis Web (Studi Kasus : SMPNegeri 1 Teluk Dalam)	Florentina J.I Loi, Rianto Sitanggang, Alexander F.K Sibero, Dini M Hutagalung (2021)	untuk pengolahan data secara cepat dan tepat, serta sistem dari perpustakaan itu sendiri bagaimana sehingga dapat mempermudah para pegawai perpustakaan dan membuat para peserta didik khususnya nyaman untuk belajar dan juga dalam melakukan peminjaman dan pengembalian buku di perpustakaan tersebut	<i>Rapid Application Development( RAD)</i>	Penelitian ini menghasilkan sistem informasi yang diberi nama “Perancangan Sistem Informasi Perputakaan Sekolah Berbasis Web (Studi Kasus : SMP Negeri 1 Teluk Dalam)”. memudahkan pegawai dan membuat para peserta didik khususnya nyaman untuk belajar

					dan juga dalam melakukan peminjaman dan pengembalian buku di perpustakaan tersebut.
3	Rancang Bangun Sistem Informasi Perpustakaan Berbasis Website Di Sman 3 Abdy	Nadya Sari (2023)	menciptakan sebuah sistem informasi perpustakaan berbasis web yang akan memberi kemudahan bagi anggota untuk mendapatkan buku-buku yang diperlukan, kemudahan untuk petugas perpustakaan mengolah data peminjaman,	<i>RnD (Research and Development)</i>	web ini memiliki buku pengunjung, terdapat menu pencarian buku serta letak buku di rak, memiliki akses pinjam dan pengembalian buku, laporan keanggotaan, laporan buku.

			pengembalian, dan pemeliharaan koleksi buku.		
4	Perancangan Sistem Informasi Perpustakaan Berbasis <i>Web</i> (Studi Kasus SMAN 1 Penengahan)	A. R. Kasmirin, M. Yusman, and I. Adipribadi (2020)	menghasilkan aplikasi perpustakaan berbasis situs <i>web</i> , yang akan membantu petugas dalam mengelola data perpustakaan dan memfasilitasi pencarian data buku, data anggota dan meminjam dan mengembalikan buku.	<i>Rapid Application Development (RAD)</i>	aplikasi perpustakaan berbasis situs <i>web</i> , yang akan membantu petugas dalam mengelola data perpustakaan dan memfasilitasi pencarian data buku, data anggota dan meminjam dan mengembalikan buku.
5	untuk memudahkan staff perpustakaan dalam pengelolaan data – data	Dewi Wulan Dari, Ani Oktarini Sari, Astrilya	untuk memudahkan staff perpustakaan dalam pengelolaan data – data	<i>Rapid Application Development (RAD)</i>	Penelitian ini menghasilkan sistem informasi yang diberi

	perpustakaan, sistem tersebut bisa diakses kapanpun dan dimanapun	na (2020)	perpustakaan, sistem tersebut bisa diakses kapanpun dan dimanapun . Selain itu, sistem informasi perpus berbasis web juga dapat memudahkan/membantu anggota perpustakaan ketika melakukan transaksi pinjam pada saat peminjaman buku		nama “Pembangunan Sistem Informasi Perpustakaan Berbasis Web Menggunakan Metode Rapid Application Development (Rad) (Studi Pada : Smk Negeri 11 Malang)”
--	---	-----------	--	--	--

Dilihat dari penelitian- penelitian terdahulu di atas, terdapat persamaan dengan penelitian “Sistem Informasi Perpustakaan SMA Negeri 1 Patikraja Berbasis *Website* (Studi Kasus SMA N 1 Patikraja)” yang diajukan, yaitu sistem berbasis *website*, menggunakan bahasa pemrograman *PHP* serta menggunakan *MYSQL* sebagai *database*. Terdapat juga perbedaannya yaitu dalam penelitian “Sistem Informasi Perpustakaan SMA Negeri 1 Patikraja Berbasis *Website* (Studi Kasus SMA N 1 Patikraja)” peneliti menggunakan metode *Waterfall* sebagai metode pengembangan sistem yang digunakan. Perbedaan lainnya adalah member dapat melakukan peminjaman untuk kedepannya dengan menaruh buku tersebut ke dalam keranjang, member dapat memberikan



penilaian hasil membacanya ke sistem, terdapat pelanggaran yaitu denda keterlambatan dan denda hilang atau rusak, superadmin dan kepala sekolah dapat mencetak laporan sesuai kebutuhan di antaranya per bulan, per semester dan per tahun.

## **2.2 Landasan Teori**

### **2.2.1 Perpustakaan**

Perpustakaan adalah institusi yang mengumpulkan pengetahuan tercetak dan terekam, mengelolanya dengan cara khusus guna memenuhi kebutuhan intelektualitas para penggunanya melalui beragam cara interaksi pengetahuan. Dalam arti tradisional, perpustakaan adalah sebuah koleksi buku dan majalah. Peneliti memilih perpustakaan dikarenakan dari hasil observasi peneliti menemukan permasalahan dalam sistem transaksi perpustakaan yang masih menggunakan tulis tangan di buku (manual). Salah satu contoh kegunaan perpustakaan bertugas dalam penyimpanan, mengelola serta melakukan transaksi buku untuk memberikan pengetahuan, mempunyai peranan yang penting dalam melaksanakan pelestarian buku [11].

### **2.2.2 Sistem informasi**

*Sistem Informasi* adalah *sistem* yang mengkombinasikan pekerjaan manusia dan penggunaan teknologi dalam upaya mendukung manajemen dan kegiatan operasional. *Sistem informasi* sering digunakan untuk mengumpulkan informasi dari petugas perpustakaan untuk membuat permasalahan yang nantinya akan dipecahkan oleh penelitian yang akan dilakukan dengan cara membuat laporan yang diperoleh dan dibutuhkan saat membangun sebuah pemrograman. Peneliti memilih menggunakan sistem informasi untuk menjalankan dan mengelola operasi program yang dibuatnya, berinteraksi dengan pengguna [12].

### **2.2.3 Website**

*Website* adalah halaman atau kumpulan halaman pada sebuah domain di internet yang dibuat dengan tujuan tertentu. *Website* umumnya berisi tampilan halaman berupa teks, gambar, animasi, audio, video atau gabungan satu dengan lainnya. Penggunaan *web* akan semakin mudah untuk menjalankannya sebab tidak terpaku dengan 1

perangkat untuk menjalankannya akan tetapi petugas wajib menggunakan internet untuk menjalankannya, *web* itu terdiri dari beberapa halaman yang sering disebut homepage. Peneliti memilih sistem informasi berbasis *web* karena memiliki beberapa kelebihan diantaranya mudah dikembangkan oleh seorang *web developer*, mudah diakses, mudah disalurkan dan lebih fleksibel [12].

#### **2.2.4 *framework CodeIgniter***

*CodeIgniter* adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis *web* yang disusun dengan menggunakan bahasa *PHP*. *framework* CI berfungsi untuk membantu peneliti dalam mengembangkan system berbasis *web*. *Codeigniter* adalah sebuah *framework PHP Open Source*, *Framework* ini digunakan peneliti dalam system pengolahan data pada perpustakaan yang siap pakai dan memungkinkan proses pembuatan aplikasi *web* menjadi lebih cepat [13].

#### **2.2.5 *Xampp***

*XAMPP* adalah sebuah paket kumpulan software yang terdiri dari *PHP MyAdmin*, *Apache*, *PHP*, *MySQL*, *Perl*, *Filezilla* dan lainnya.” *Xampp* berfungsi memudahkan instalasi *PHP* untuk pengembangan *web MySQL*, *PhpMyAdmin*, *PHP Apache*, dan *software untuk membuat web*. Untuk aplikasi dapat di *extract* dan diinstal memilih *xampp* sesuai sistem operasinya [14]

#### **2.2.6 *PHP***

*PHP* adalah bahasa *skrip* sisi *server* yang digunakan untuk membuat halaman *web* yang dinamis dan interaktif. Selama pengembangan dengan *PHP* sederhana, logika bisnis dicampur dengan *query* basis data dan tag presentasi. Karena campuran ini mode pengembangan, pemeliharaan dan skalabilitas aplikasi menjadi sulit. *PHP* telah membawa perbedaan kerangka pengembangan untuk memecahkan masalah ini. Kerangka kerja *PHP* membantu pengembang membangun aplikasi *web* lebih cepat dan lebih mudah dengan menyediakan model kerangka kerja dasar [15]

### 2.2.7 **Black-box**

program perangkat lunak atau sistem yang diuji diamati sebagai "kotak hitam". Pilihan test case untuk pengujian fungsional didasarkan pada persyaratan atau spesifikasi desain entitas perangkat lunak yang diuji. Contoh hasil yang diharapkan terkadang disebut test oracle, termasuk persyaratan/spesifikasi desain, nilai yang dihitung dengan tangan, dan hasil simulasi. Pengujian fungsional terutama berfokus pada perilaku eksternal entitas perangkat lunak[5].

### 2.2.8 **Waterfall**

Metode *waterfall* adalah hal yang menggambarkan pendekatan secara sistematis dan juga berurutan (*step by step*) pada sebuah pengembangan perangkat lunak. Tahapan dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan yaitu *planning*, permodelan, konstruksi, sebuah system dan penyerahan sistem kepada pengguna, dukungan pada perangkat lunak lengkap yang dihasilkan[16].

### 2.2.9 **DML**

*DML (Data Manipulation Language)* merupakan basis data yang digunakan untuk melakukan modifikasi dan pengambilan data pada suatu *database*, pengolahan atau modifikasi ini meliputi: *Insert*, digunakan untuk melakukan penambahan data. ? *Select*, digunakan untuk pengambilan data ? *Update*, digunakan untuk melakukan perubahan data ? *Delete*, digunakan untuk melakukan penghapusan data. ? *Drop database*, digunakan untuk menghapus *database*. ? *Drop table*, digunakan untuk menghapus tabel. ? *Group by*, digunakan untuk mengelompokkan *data*. ? *Order by*, digunakan untuk menampilkan semua data secara urut berdasarkan abjad nya[17]

### 2.2.10 **DDL**

*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*: ? *Create*, digunakan untuk membuat *database*, tabel, dan objek lain dalam *database*. ? *Alter*, digunakan untuk *memodifikasi tabel*, seperti contoh mengubah nama tabel, *field*, menambah nama *field*. ? *Drop*, digunakan untuk menghapus *database*, tabel, dan objek lain dalam *database*[17].

### 2.2.11 ERD

“*entity relationship diagram (ERD)* adalah sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data”. merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh Sistem Analisis dalam tahap analisis persyaratan proyek pengembangan sistem[16].

### 2.2.12 Flowchart

*Flowchart* atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. seorang *analisis sistem* menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *programmer*. 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. Dengan adanya *flowchart*, setiap urutan proses dapat digambarkan menjadi lebih jelas. Selain itu, ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan *flowchart* ini. Setelah proses membuat *flowchart* selesai, maka giliran *programmer* yang akan menerjemahkan desain logis tersebut kedalam bentuk program dengan berbagai bahasa pemrograman yang telah disepakati.[18].

### 2.2.13 UML

*UML (Unified Modeling Language)* adalah sebuah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak (Pressman). *UML* yang saya gunakan yaitu (*behavioral diagram*) biasa digunakan untuk visualisasi, spesifikasi, konstruksi dan dokumentasi beberapa bagian-bagian dari system yang ada dalam perangkat lunak. seperti halnya

seorang arsitek dalam membuat dokumen cetak biru yang digunakan oleh perusahaan konstruksi untuk membangun sebuah bangunan, arsitek perangkat lunak membuat diagram-diagram *UML* untuk membantu *programmer / developer* membangun perangkat lunak. Untuk selanjutnya, semakin kita mengetahui beberapa kosakata yang digunakan *UML*, kita akan semakin mudah dalam memahami spesifik.[19].

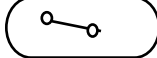


### 2.2.14 Activity Diagram


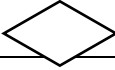


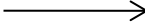

*Activity Diagram* adalah bagian penting dari *UML* yang menggambarkan aspek dinamis dari sistem. *Logika prosedural*, proses bisnis, dan aliran kerja suatu bisnis dapat dengan mudah dideskripsikan dalam *activity diagram*. Tujuan dari *activity diagram* adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktivitas ke aktivitas lain[20].


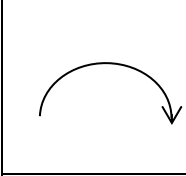
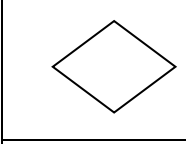
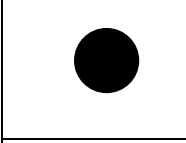
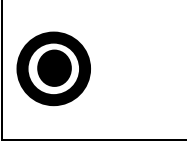
Secara umum *Activity Diagram* digunakan untuk menggambarkan diagram alir yang terdiri dari banyak aktivitas dalam sistem dengan beberapa fungsi tambahan seperti percabangan, Aliran *parallel* dan *swim lane*

Tujuan dari *activity diagram* bisa digambarkan sebagai berikut  
Menggambarkan aliran aktivitas dari sistem  
Menggambarkan urutan aktivitas dari satu aktivitas ke lainnya  
Menggambarkan paralelisme, percabangan dan aliran konkuren dari sistem. Simbol *Activity Diagram*[20].

**Tabel 2. 2** Activity Diagram

Gambar	Nama	Fungsi
	<i>State</i>	Menambahkan <i>State</i> untuk suatu Objek
	<i>Activity</i>	Menambahkan Aktivitas baru pada diagram
	<i>Start state</i>	Memperlihatkan dimana aliran kerja Berawal

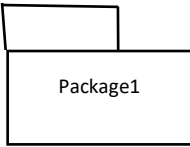
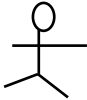


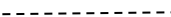
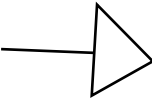
	<i>End state</i>	Memperlihatkan dimana aliran kerja berakhir
	<i>State Taransition</i>	Menambah transisi dari suatu aktivitas yang lainnya
	<i>Decisions Points</i>	Menambah titik keputusan pada aliran kerja
	<i>Horizontal Synchronization</i>	Menambahkan sinkronisasi <i>Horizontal</i> pada diagram
	<i>Vercital Synchronization</i>	Menambahkan sinkronisasi <i>Vertikal</i> pada diagram
	<i>State transition</i>	Menambahkan transisi dari suatu aktivitas ke aktivitas yang lainnya
	<i>Horizontal synchronization</i>	Menambahkan sinkronisasi horisontal pada diagram

	<i>Vertical synchronization</i>	Menambahkan sinkronisasi vertikal pada diagram
	<i>Transition to self</i>	Menambahkan transisi rekursif
	<i>Decisions points</i>	Menambahkan titik keputusan pada aliran kerja
	<i>Start state</i>	Memperlihatkan dimana aliran dimulai
	<i>End state</i>	Memperlihatkan dimana aliran diakhiri

### 2.2.15 Use Case Diagram

*Use case diagram* tersebut menggambarkan secara garis besar interaksi yang terjadi antara pengguna (*user*) dengan sistem. Aktor (*user*) yang terlibat dengan sistem bisa disebut sebagai admin pada tempat usaha persewaan. *Admin* berinteraksi dengan sistem dan dapat melakukan fitur-fitur yang tersedia pada sistem perangkat lunak. Sedangkan *customer*, atau owner tidak memiliki akses terhadap sistem perangkat lunak, namun admin nantinya akan memberikan laporan sebagai output dari sistem perangkat lunak yang menggunakan pemodelan *UML*. *Symbol Use Case Diagram* Pada tabel dibawah ini[21].

**Tabel 2. 3** *Use Case Diagram*

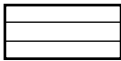




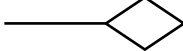
Gambar	Nama	Fungsi
	<i>Package</i>	Menambah paket baru dalam diagram
	<i>Actor</i>	Menambah Actor dalam Diagram
	<i>Use Case</i>	Menambah <i>Use Case</i> pada Diagram
	<i>Unidirection association</i>	Mengambarkan Relasi antara aktor dengan <i>Use Case</i>
	<i>Dependencies or instatiantes</i>	Mengambarkan kebergantungan( <i>dependencies</i> ) antar item dalam digram
	<i>Generalization</i>	Mengambarkan relasi lanjut antara <i>use case</i> atau menggambarkan struktur pewarisan antara Ator

### 2.2.16 *Class Diagram*

*Class Diagram* merupakan salah satu jenis diagram *UML* yang digunakan untuk menggambarkan koneksifitas basis data. *Symbol Class Diagram* Pada tabel tabel dibawah ini[6]



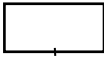

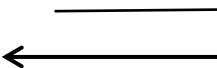


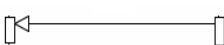
**Tabel 2. 4** *Class Diagram*

Gambar	Nama	Fungsi
	<i>Class</i>	Menambahkan kelas baru pada <i>Diagram</i>
	<i>Interface</i>	Menambah Kelas antarmuka ( <i>Interface</i> ) pada diagram
	<i>Association</i>	Mengambar Relasi Asosiasi
	<i>Generalization</i>	Mengambar Relasi Generalisasi
	<i>Association Class</i>	Menghubungkan asosiasi kelas ( <i>Association Class</i> ) pada suatu Relasi
	<i>Agregation</i>	Mengambarkan Relasi Agregasi

### 2.2.17 *Sequence Diagram*

*Sequece diagram* merupakan salah satu jenis diagram pada *UML* yang dapat menjelaskan urutan waktu pemrosesan sistem. Pada gambar-gambar dibawah ini, penulis mendeskripsikan urutan sistem per bagian objek. *Symbol Sequence Diagram* Pada tabel dibawah ini[22]

Tabel 2. 5 *Sequence Diagram*

Gambar	Nama	Fungsi
	<i>Objet</i>	Menambahkan objek baru pada Diagram
	<i>Object Message</i>	Mengambarkan Pesan ( Message) antara dua Objek
	<i>Message to Self</i>	Menggambar pesan
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

### 2.2.18 Object Oriented Programming (OOP)

Pemrograman *berorientasi* objek atau *object oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*. *OOP* memberikan kemudahan dalam pembuatan sebuah program, keuntungan yang didapat apabila membuat Program berorientasi objek atau *object oriented programming* (OOP)

antara lain : 1) *Reusability*, kode yang dibuat dapat digunakan kembali, 2) *Extensibility*, pemrogram dapat membuat metode baru atau mengubah yang sudah ada sesuai yang diinginkan tanpa harus membuat kode dari awal, 3) *Maintainability*, kode yang sudah dibuat lebih mudah untuk dikelola apabila aplikasi yang dibuat berskala besar yang memungkinkan adanya error dalam pengembangannya hal tersebut dapat diatasi dengan OOP karena pemrograman OOP sudah menggunakan konsep modularitas. Keuntungan Pemrograman *berorientasi objek* atau *object oriented programming (OOP)* yang lain adalah : 1) Alamiah, 2) Dapat diandalkan (*reliable*), 3) Dapat dipakai kembali (*reusable*), 4) Mudah dirawat (*maintainable*), 5) Dapat diperluas (*extendable*), 7) Efisiensi waktu[23].

### 2.2.19 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. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system, DBMS*). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan di mana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. *MySQL* adalah *DBMS* yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi

*MySQL* adalah database server yang gratis dengan lisensi *GNU General Public License (GPL)* sehingga dapat digunakan untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada[24]