

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam membuat laporan akhir sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Berikut merupakan penelitian terdahulu berupa beberapa jurnal yang terkait dengan judul laporan akhir penulis.

Rujukan penelitian pertama yaitu penelitian yang dilakukan oleh Andika Ferdiansyah, Wirarama Wedashwara dan Lanang Tanaya pada tahun 2021 dalam jurnal yang berjudul “Sistem Informasi *Profile Company* Klinik Mitra Medistra Berbasis Web Dengan PHP Dan MySQL”. Penelitian ini dibangun menggunakan metode *waterfall* dengan PHP sebagai bahasa pemrograman-nya dan MySQL sebagai basis data-nya dan diuji dengan metode *blackbox*.

Rujukan penelitian kedua yaitu penelitian yang dilakukan oleh Bima Ady Prakoso dan Nugroho Eko Budiyanto pada tahun 2021 dalam jurnal yang berjudul “Sistem Informasi Klinik Berbasis Web (Studi Kasus Klinik dr. Susana Semarang)”. Penelitian ini dibangun menggunakan metode *waterfall* dengan PHP sebagai bahasa pemrograman-nya dan MySQL sebagai basis data-nya.

Rujukan penelitian ketiga yaitu penelitian yang dilakukan oleh Fahmi Aulia Rahman dan Syahbaniar Rofiah pada tahun 2019 dalam jurnal yang berjudul “Sistem Informasi Klinik Berbasis *Website*”. Penelitian ini dirancang dengan model pengembangan sistem *waterfall* dengan PHP sebagai bahasa pemrograman-nya, Codeigniter sebagai *framework* dan MySQL sebagai basis data-nya.

Rujukan penelitian keempat yaitu penelitian yang dilakukan oleh Devy Ferdiansyah pada tahun 2018 dalam jurnal yang berjudul “Penerapan Konsep *Model View Controller* Pada Rancang Bangun Sistem Informasi Klinik Kesehatan Berbasis Web”. Penelitian ini dirancang dengan konsep arsitektur MVC ini menggunakan metodologi SDLC dengan model pengembangan sistem *waterfall* dengan PHP sebagai bahasa pemrograman-nya dan MySQL sebagai basis data-nya

Rujukan penelitian kelima yaitu penelitian yang dilakukan oleh Iswanto Sulila, Agus Lahinta dan Mohamad Syafri Tuloli pada tahun 2021 dalam jurnal yang

berjudul “Sistem Informasi Pelayanan Kesehatan Berbasis Web Pada Klinik Gocare”. Penelitian ini dirancang dengan model pengembangan sistem Prototype dengan PHP sebagai bahasa pemrogramannya dan MySQL sebagai basis data-nya.

Dibawah ini adalah perbandingan penelitian diatas dengan penelitian yang akan dilakukan penulis yaitu “Implementasi Framework Laravel Pada Sistem Informasi Dan Profil Klinik Umum Dan Gigi Annisa Berbasis Website”.

**Tabel 2.1** Perbandingan Penelitian Terdahulu dengan Penelitian Sekarang

No	Penelitian	Persamaan	Perbedaan
1	Ferdiansyah, A., Wedashwara, W., & Tanaya, L. 2021. Sistem Informasi <i>Profile Company</i> Klinik Mitra Medistra. <i>Jurnal Begawe Teknologi Informasi</i> , 2(1).	<ul style="list-style-type: none"> <li>• Bahasa pemrograman PHP</li> <li>• Basis data MySQL</li> <li>• Pengujian <i>Black box</i></li> </ul>	<ul style="list-style-type: none"> <li>• Informasi bersifat statis</li> <li>• Fitur-fiturnya sangat <i>simple</i></li> </ul>
2	Prakoso, B. A., & Budiyanto, N. E. 2021. Sistem Informasi Klinik Berbasis Web. <i>Jurnal Informatika dan Rekayasa Perangkat Lunak</i> , 3(1), 47-51.	<ul style="list-style-type: none"> <li>• Bahasa pemrograman PHP</li> <li>• Basis data MySQL</li> <li>• Metode pengembangan sistem <i>waterfall</i></li> </ul>	<ul style="list-style-type: none"> <li>• Belum ada profil klinik hanya untuk <i>internal</i> klinik</li> </ul>
3	Rahman, F. A., & Rofiah, S. 2019. Sistem Informasi Klinik Berbasis <i>Website</i> . <i>Information Management For Educators And Professionals: Journal of</i>	<ul style="list-style-type: none"> <li>• Bahasa pemrograman PHP</li> <li>• Basis data MySQL</li> <li>• Metode</li> </ul>	<ul style="list-style-type: none"> <li>• Belum ada profil klinik hanya untuk <i>internal</i> klinik</li> <li>• Menggunakan <i>Framework</i></li> </ul>

	<i>Information Management</i> , 3(2), 193-202.	<p>pengembangan sistem <i>waterfall</i></p> <ul style="list-style-type: none"> <li>• Menerapkan Konsep MVC</li> </ul>	Codeigniter
4	Ferdiansyah, D. (2018). Penerapan Konsep <i>Model View Controller</i> Pada Rancang Bangun Sistem Informasi Klinik Kesehatan Berbasis Web. <i>Jurnal Kajian Ilmiah</i> , 18(2), 195-205.	<ul style="list-style-type: none"> <li>• Adanya fitur cetak <i>invoice</i> pembayaran</li> <li>• Menerapkan Konsep MVC</li> </ul>	<ul style="list-style-type: none"> <li>• Belum ada profil klinik hanya untuk <i>internal</i> klinik</li> </ul>
5	Sulila, I., Lahinta, A., & Tuloli, M. S. 2021. Sistem Informasi Pelayanan Kesehatan Berbasis Web Pada Klinik Gocare. <i>Diffusion: Journal of Systems and Information Technology</i> , 1(1), 83-93.	<ul style="list-style-type: none"> <li>• Bahasa pemrograman PHP</li> <li>• Basis data MySQL</li> <li>• Pengujian Black box</li> </ul>	<ul style="list-style-type: none"> <li>• Metode Prototype</li> <li>• Fokus pada penjadwalan pelayanan</li> </ul>

## 2.2 Pengertian Implementasi

Pengertian implementasi dalam Kamus Besar Bahasa Indonesia (KBBI) secara umum adalah suatu tindakan atau pelaksanaan dari suatu rencana yang telah matang dan terperinci. Sedangkan menurut Totnanatzky dan Jhonshon Implementasi sebagai *the translation of any tool technique process or method of doing from knowledge to practice*. Dari dua pendapat diatas, implementasi dapat diartikan sebagai penerapan metode dari suatu rencana menjadi tindakan (Supriyadi dan Asih, 2021:13-14).

## 2.3 Pengertian Sistem Informasi

### 2.3.1 Sistem

Sistem adalah sesuatu yang terdiri dari obyek-obyek, atau elemen-elemen, atau komponen-komponen yang bertata kaitan dan bertata hubungan satu sama lain, sedemikian rupa sehingga elemen-elemen tersebut merupakan suatu unit atau

pemrosesan tertentu. Berdasarkan pendapat tersebut dapat disimpulkan bahwa sistem adalah kumpulan dari objek atau komponen atau prosedur atau elemen yang saling terintegrasi satu sama lain dan bekerja secara harmonis untuk mencapai tujuan atau maksud tertentu (Audrilia dan Budiman, 2020:5).

### **2.3.2 Informasi**

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi (Kristanto, 2018). Berdasarkan pendapat diatas dapat ditarik kesimpulan bahwa informasi adalah kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti agar bermanfaat dalam pengambilan keputusan bagi penggunaannya (Audrilia dan Budiman, 2020:5).

### **2.3.3 Sistem Informasi**

Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Kristanto, 2018). Berdasarkan pendapat diatas dapat ditarik kesimpulan bahwa sistem informasi adalah suatu sistem yang dibuat manusia yang terdiri dari kumpulan objek atau komponen atau prosedur atau elemen untuk mengolah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik untuk tujuan tertentu guna memecahkan masalah dan pengambilan keputusan (Audrilia dan Budiman, 2020:5).

## **2.4 Pengertian Website**

*Website* atau situs dapat didefinisikan sebagai kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau kombinasi dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling berkaitan, yang masing-masing dihubungkan dengan jaringan jaringan halaman (Novendri dkk, 2019:47).

## 2.5 Pengertian Visual Studio Code

Visual Studio Code telah menjadi alat pengembangan *cross-platform* pertama dalam Microsoft Visual Studio *family* yang berjalan di Windows, Linux, dan macOS secara gratis dan *open source* di (<https://github.com/Microsoft/vscode>), aplikasi ini jelas merupakan kode editor yang memudahkan untuk mengedit file kode dan sistem proyek berbasis *folder* serta menulis web *cross-platform* dan aplikasi *mobile* dengan dukungan terintegrasi untuk sejumlah besar bahasa dan fitur pengeditan yang kaya seperti *IntelliSense*, *finding symbol references*, *quickly reaching a type definition*, dan banyak lagi lagi (Sole, 2019:2).

## 2.6 Composer



**Gambar 2. 1** Logo Composer

(Sumber: [www.deviantart.com](http://www.deviantart.com))

Composer digunakan untuk menginstal dan menambahkan *dependency* PHP pada Laravel. Banyak sekali *plugin* yang dapat digunakan untuk meningkatkan fungsionalitas laravel yang disediakan oleh pihak ketiga. *Plugin-plugin* tersebut dapat diinstal dengan mudah menggunakan Composer (Abdulloh, 2018:5).

## 2.7 XAMPP

XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Xampp merupakan alat yang menyediakan paket perangkat lunak dalam satu buah paket (Novendri, 2019:48).

1. **X** : Program ini dapat dijalankan dibanyak sistem operasi, seperti Windows, Linux, Mac OS dan juga Solaris.

2. **A** : Apache merupakan aplikasi web server. Tugas utama dari Apache adalah menghasilkan halaman web yang benar untuk pengguna berdasarkan kode PHP yang dituliskan oleh pembuat web atau *user*.
3. **M** : MySQL, merupakan aplikasi data *server*. Perkembangannya disebut juga SQL yang merupakan singkatan dari *Structured Query Language*. SQL adalah bahasa terstruktur yang digunakan untuk memproses *database*.
4. **P** : PHP, merupakan bahasa pemrograman web, dimana pengguna dapat menggunakan bahasa pemrograman ini untuk membuat web yang bersifat *server-side scripting*.
5. **P** : Perl, yaitu merupakan bahasa pemrograman untuk segala keperluan, dan dikembangkan pertama kali oleh Larry Wall di mesin Unix.

## 2.8 PHPMyAdmin

PHPMyAdmin adalah aplikasi untuk mengelola *database* MySQL dan *database* MariaDB dengan mudah melalui antarmuka (*Interface*) grafis yang lengkap dan menarik. Aplikasi web ini ditulis menggunakan bahasa pemrograman PHP dan unsur HTML XHTML CSS serta kode JavaScript.

Keberadaan PHPMyAdmin yang dianggap sangat penting dan juga sifatnya terbuka menjadikannya salah satu aplikasi yang selalu ada di dalam cPanel (aplikasi populer untuk mengontrol *website*) hal ini menunjukkan bahwa penyedia *web hosting* dan pengguna seperti kita menaruh kepercayaan yang sangat besar pada PHPMyAdmin sebagai salah satu aplikasi web yang dipasang (*install*) di *server* (Yudhanto, 2019:19).

## 2.9 HTML (*Hyper Text Markup Language*)

HTML adalah singkatan dari *Hyper Text Markup Language* yang merupakan bahasa pemrograman dasar dalam pembuatan *website*, HTML terdiri dari *Head*, *Body* dan di dalamnya terdapat TAG dan *attribute*, walaupun dikatakan sebagai bahasa pemrograman, tetapi HTML belum dapat dikatakan sebagai bahasa pemrograman karena HTML tidak memiliki hal-hal yang dibutuhkan oleh bahasa pemrograman yaitu logika, HTML hanya memberikan output, maka dari itu HTML

di ibaratkan sebagai pondasi atau struktur dari *web* dan yang menjadi bahasa pemrogramannya yaitu PHP dan Javascript (Marlina dkk, 2021).

Struktur dasar lengkap pada HTML seperti berikut :

```

<html>
  <head>
    ..... } Deskripsi Dokumen
  </head>
  <body>
    ..... } Isi Dokumen
  </body>
</html>

```

## 2.10 CSS (*Cascading Style Sheet*)

CSS merupakan kependekan dari *Cascading Style Sheet* yang berfungsi untuk mengatur tampilan dengan kemampuan jauh lebih baik dari tag maupun atribut standar HTML (*Hypertext Markup Language*). CSS sebenarnya adalah suatu kumpulan atribut untuk fungsi format tampilan dan dapat digunakan untuk mengontrol tampilan banyak dokumen secara bersamaan. Keuntungan menggunakan CSS yaitu jika ingin mengubah format dokumen, maka tidak perlu mengedit satu persatu (Novendri, 2019:47).

## 2.11 Java Script

Java script adalah bahasa skrip yang digunakan untuk mengontrol tindakan-tindakan yang diperlukan di halaman web. Java script adalah bahasa yang berfungsi membuat skrip-skrip program yang dapat dikenal dan dieksekusi oleh web browser. Pemrosesnya dilakukan di sisi *client*. Peran javascript dalam membuat website adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan event yang dilakukan oleh pengguna website (Novendri, 2019:49).

## 2.12 PHP (*Hypertext Preprocessor*)

PHP merupakan akronim dari *Hypertext Preprocessor* yang merupakan sebuah bahasa *script* tingkat tinggi yang dipasangkan pada dokumen HTML. Sebagian besar

sintaks PHP mirip dengan bahasa C, java, dan perl. Namun, PHP terdapat beberapa fungsi yang lebih spesifik. PHP digunakan untuk merancang yang sifatnya dinamis dan dapat bekerja secara otomatis.

Web dinamis yang bisa dibuat dengan menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung kebutuhan atau situasi. Web dinamis juga dapat menyimpan data ke dalam *database*, membuat halaman yang berubah-ubah sesuai dengan permintaan pengguna, dan lain sebagainya. PHP disebut juga sebagai *scripting language* karena memiliki fitur tersebut (Novendri, 2019:47). Bentuk skrip penulisan PHP, sebagai berikut :

```
<?php      //awal tag
.....
..... } Tuliskan script PHP disini
.....
?>        //akhir tag
```

Dua fungsi- fungsi array dalam PHP yang di gunakan dalam *website* antara lain:

#### 1. Fungsi Explode

Fungsi `explode()` memecah string menjadi array dengan ketentuan tanda pemisah yang kita inginkan. Fungsi ini memiliki dua buah argument wajib dan sebuah argument pilihan atau opsional. Berikut formatnya:

```
<?php
explode(string pemisah, string yang akan dipisah, [limit])
?>
```

Parameter ketiga bersifat opsional, boleh tidak ditulis.

#### 2. Fungsi Implode

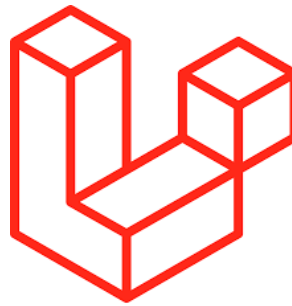
Fungsi `implode()` akan menyatukan array ke dalam string dengan memberikan tanda pemisah yang kita inginkan. Fungsi ini memiliki dua buah argument wajib. Berikut formatnya:

```
<?php
implode(string penghubung, array yang ingin digabung)
?>
```



*Note* : Tanda pemisah tidak harus simbol, spasi pun dapat menjadi tanda pemisah (Majid, 2018:48-49).

### 2.13 Framework Laravel



**Gambar 2. 2** Logo Laravel  
(Sumber: id.m.wikipedia.org)

#### 2.13.1 Framework

Pengertian *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau Kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan data. Salah satu keuntungan menggunakan *framework* adalah memberi struktur yang baik dalam program yang dibuat karena *framework* memiliki *library* atau fungsi yang bisa langsung digunakan. Selain itu, *framework* memudahkan dalam pekerjaan program secara tim. Salah satu *framework* yang banyak digunakan oleh *programmer* adalah *framework* laravel (Naista, 2017).

#### 2.13.2 Laravel

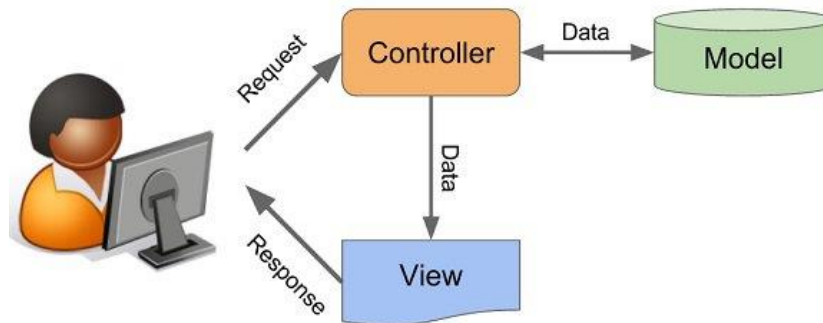
Laravel adalah Framework berbasis PHP yang sifatnya *open source* dan menggunakan konsep *Model View Controller*. Proyek Laravel dimulai pada April 2011 yang dikembangkan oleh Taylor Otwell, Laravel merupakan salah satu *framework* PHP terbaik yang *up-to-date* dengan versi PHP. Oleh karena itu, Laravel mensyaratkan PHP versi 5.3 ke atas (Naista, 2017).

## 2.14 Bootstrap

Bootstrap merupakan *framework open source* yang dapat digunakan secara gratis. Bootstrap adalah *framework front-end* yang intuitif dan *powerful* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah. Bootstrap memungkinkan desain sebuah website menjadi responsive sehingga dapat dilihat dari berbagai macam ukuran device dengan tampilan tetap menarik.. Bootstrap telah didukung oleh hampir semua browser baik pada desktop maupun mobile (Agustiansyah dan Solikin, 2022).

## 2.15 Konsep MVC (*Model-View-Controller*)

MVC adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) untuk membuat satu jenis paket data jaringan menjadi jenis data lainya bersama dengan pemrosesan (*Model*), dari proses manipulasi (*Controller*) dan tampilan (*View*) untuk dipresentasikan pada sebuah *user interface* (Dalis dkk, 2019).



**Gambar 2. 3** Konsep MVC (*Model-View-Controller*)

(Sumber: stekom.ac.id)

### 1. *Model*

*Model* adalah bagian kode program yang menangani *database*, isi dari model merupakan bagian (fungsi-fungsi) yang berhubungan langsung dengan *database* untuk mengelola data seperti memasukkan data, pembaruan data, hapus data, dan lain-lain, namun tidak dapat berhubungan langsung dengan bagian *view*.

## 2. *View*

Tampilan (*View*). Bagian ini mengandung keseluruhan detail dari implementasi *user interface*. *View* adalah bagian kode program yang mengatur tampilan *website*. *View* biasanya berupa file skrip HTML. *View* juga berfungsi menampilkan data serta inputan *user*, jadi *view* merupakan halaman web.

## 3. *Controller*

Cara pemrosesan (*Controller*). *Controller* merupakan bagian yang menghubungkan *model* dan *view*. *Controller* berisi perintah-perintah yang bertanggung jawab untuk memproses suatu data dan mengirimkannya ke halaman web. *Controller* berfungsi untuk menerima request dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

### 2.16 *Database*

*Database* atau basis data adalah sekumpulan data-data yang saling berhubungan dan diorganisasikan sedemikian rupa sehingga data tersebut dapat diakses dengan mudah dan cepat oleh program menjadi sebuah informasi yang lebih bermanfaat bagi pengguna. Dalam basis data, data yang ada tidak hanya disimpan begitu saja dalam sebuah media penyimpanan, tetapi dikelola oleh sebuah sistem basis data.

Sistem basis data adalah koleksi dari data-data yang terstruktur sehingga data tersebut mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan tertentu, serta dihapus). Sistem ini sering disebut SMDB (Sistem Manajemen Basis Data) (Novendri dkk, 2019:48).

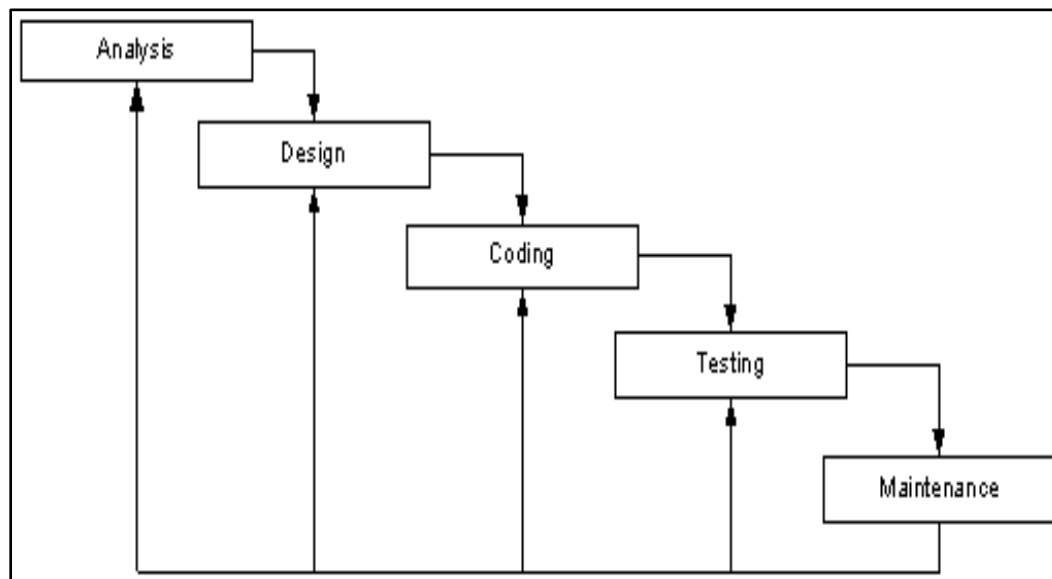
### 2.17 *MySQL (Structured Query Language)*

MYSQL atau disebut juga SQL merupakan singkatan dari *Structured Query Language*. MYSQL adalah sebuah Sistem Manajemen Basis Data (SMDB) yang bersifat *open source* dan *relational* artinya, data yang dikelola dalam *database* yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. MYSQL dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar. Sedangkan SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. SQL merupakan

bahasa pemrograman yang dirancang khusus untuk mengirimkan suatu perintah query (pengaksesan data berdasarkan pengalamatan tertentu) terhadap sebuah *database*. Singkatnya, SQL adalah perintah atau bahasa yang melekat di dalam SMBD seperti MySQL sebagai suatu bahasa permintaan (Novendri, 2019:48-49).

### 2.18 Metode *Waterfall*

*Waterfall* model merupakan sebuah *software development lifecycle* yang memiliki jenis linear. Memiliki beberapa fase seperti contohnya *requirement analysis* (analisis kebutuhan), *design* (perancangan), *coding* (pembuatan kode program), *testing* (pengujian) dan *maintenance* (pemeliharaan) dimana umumnya satu fase tidak diulang kembali dan berlanjut ke fase berikutnya sampai fase sebelumnya selesai (Prihandoyo, 2018).



**Gambar 2. 4** *Flowchart* Metode *Waterfall*

### 2.19 Metode *Blackbox Testing*

*Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Pengujian Aplikasi dengan Metode ini memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Jaya, 2018:45).


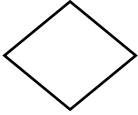
Keuntungan penggunaan metode *Blackbox Testing* adalah:



1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. *Programmer* dan *tester* keduanya saling bergantung satu sama lain.

## 2.20 ERD (*Entity Relational Diagram*)

ERD adalah suatu diagram yang digunakan untuk memodelkan struktur data dan hubungan antar data. ERD berfungsi untuk menggambarkan relasi dari dua *file* atau dua tabel yang dapat di golongkan dalam tiga macam bentuk relasi, yaitu satu-satu, satu-banyak dan banyak-banyak (Novendri dkk, 2019:51-52).

**Tabel 2. 1** Simbol-Simbol ERD (*Entity Relational Diagram*)

No	Simbol	Keterangan
1		<u>Entitas (<i>Entity</i>)</u> adalah objek data prinsip tentang informasi yang dikumpulkan. Suatu objek yang dapat didefinisikan lingkungan pemakaian dalam konteks sistem yang telah dibuat.
2		<u>Relasi (<i>Relationship</i>)</u> Relasi adalah suatu asosiasi antara dua tabel atau lebih. 1) Relasi satu ke satu ( <i>One-to-One Relationship</i> ) adalah suatu hubungan dimana entitas pertama hanya mempunyai 1 hubungan pada entitas kedua. Hubungan ini dinotasikan dengan 1 ke 1 atau 1:1. 2) Relasi satu ke banyak ( <i>One-to-Many Relationship</i> ) adalah suatu hubungan dimana satu entitas pertama bisa mempunyai banyak hubungan pada entitas kedua. Hubungan ini dinotasikan dengan 1 ke M atau 1:M 3) Relasi banyak ke satu ( <i>Many-to-One Relationship</i> ) adalah suatu hubungan dimana banyak entitas

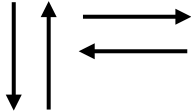
		<p>pertama bisa mempunyai satu hubungan pada entitas kedua. Hubungan ini dinotasikan dengan M ke 1 atau M:1</p> <p>4) Relasi banyak ke banyak (<i>Many-to-Many Relationship</i>) adalah setiap entitas pertama dapat mempunyai banyak hubungan pada entitas yang kedua, begitu juga yang kedua bisa memiliki banyak hubungan pada entitas pertama. Hubungan ini dinotasikan dengan M ke M atau M:M.</p>
3		<u>Atribut</u> adalah properti atau karakteristik yang dimiliki oleh suatu entitas data yang berfungsi endekripsikan karakter dari entitas.
4		<u>Garis</u> Menunjukkan garis.




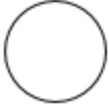




(Sumber : Novendri dkk, 2019)

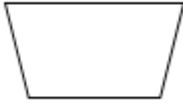


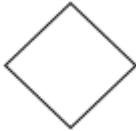



### 2.21 Flowchart

*Flowchart* adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Santoso, 2017:86).

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

NO	Simbol	Keterangan
1.	<p><i>Flow Direction</i></p> <p><i>Symbol</i></p> 	Untuk menghubungkan antara simbol yang satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses.

2.	Terminal (mulai atau berhenti) 	Simbol ini digunakan untuk menunjukkan awal kegiatan ( <i>start</i> ) atau akhir dari suatu kegiatan ( <i>stop</i> ).
3.	<i>Input dan Output</i> 	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
4.	Proses (Pengolahan) 	Untuk menunjukkan pengolahan yang dilakukan oleh komputer atau PC.
5.	<i>Connector</i> 	Simbol suatu keluaran atau masukan prosedur atau proses dalam lembar atau halaman yang sama.
6.	<i>Offline Connector</i> 	Simbol untuk keluaran atau masukan prosedur atau proses dalam lembar atau halaman yang berbeda.
7.	<i>Document</i> 	Untuk menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.
8.	<i>Manual Input</i> 	Berfungsi untuk pemasukan data secara manual <i>on-line keyboard</i> .
9.	<i>Preparation</i> 	Berfungsi untuk mempersiapkan penyimpanan yang sedang/ akan digunakan sebagai tempat pengolahan didalam <i>storage</i> .

10.	<p><i>Manual Operation</i></p> 	Berfungsi untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer atau PC.
11.	<p><i>Multiple Document</i></p> 	Sama seperti <i>symbol document</i> , hanya saja dokumen yang digunakan lebih dari satu dalam simbol ini.
12.	<p><i>Predefined</i></p> 	Untuk pelaksanaan suatu bagian (subprogram) / prosedur.
13.	<p><i>Decision (Keputusan)</i></p> 	Menunjukkan suatu perbandingan yang harus dibuat bila hasilnya “ya”, maka alir data menunjukkan ke suatu tempat, bila “tidak” maka akan menuju ke tempat lain.
14.	<p><i>Display</i></p> 	Simbol yang digunakan untuk menyatakan perangkat <i>output</i> yang digunakan.
15.	<p><i>Database</i></p> 	Simbol yang digunakan untuk menyatakan data disimpan dalam <i>database</i> .
16.	<p><i>On-line storage</i></p> 	Simbol yang menyatakan input yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i> .

(Sumber : Santoso, 2017)


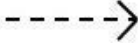




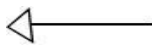





## 2.22 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu (Prihandoyo, 2018):

- a) **Use Case** : Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

**Tabel 2. 3** Simbol-Simbol UML *Use Case*

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 <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerjasama 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

(Sumber : Prihandoyo, 2018)