

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang relevan diantaranya adalah penelitian dengan judul “Perancangan Sistem Informasi Manajemen Sarana dan Prasarana Menggunakan Framework Codeigniter pada Akademi Ilmu Komputer Ternate” yang dilakukan oleh Ratna Jafar, MH Abdullah dan Mudar Safi pada tahun 2020. Tujuan dari penelitian ini adalah membangun sebuah sistem yang dapat membantu pihak manajemen AIKOM Ternate dalam mengelola sarana dan prasarana serta pelaporan. Sistem informasi dibangun menggunakan bahasa pemrograman PHP *framework* Codeigniter dan MySQL sebagai *database*. Metode pengembangan sistem yang diterapkan adalah metode Waterfall. Sistem ini memfokuskan pada data sarana, peminjaman dan laporan [4].

Penelitian berikutnya dilakukan oleh Algoniu, Wicaksono dan Afirianto pada tahun 2019 yang berjudul “Pengembangan Sistem Informasi Manajemen Sarana dan Prasarana (Studi Pada : SMK Muhammadiyah 1 Malang)”. Tujuan dari penelitian ini adalah membangun sebuah sistem informasi yang dapat digunakan untuk mempermudah dalam proses inventarisasi sarana dan prasarana di SMK Muhammadiyah 1 Malang yang semula masih menggunakan pembukuan dan bantuan Microsoft Excel. Sistem ini menerapkan *System Development Life Cycle (SDLC) Waterfall* dalam pengembangan sistemnya, tetapi tanpa melakukan tahap *deployment*. Dalam implementasinya menerapkan bahasa pemrograman PHP, *framework* Codeigniter, dan Mysql. Secara garis besar sistem ini berfokus pada alur pengadaan sarana dan prasarana saja [5].

Kemudian Muhammad Bakhar melakukan penelitian pada tahun 2019 dengan judul “Sistem Informasi Inventaris dan Perawatan Sarana dan Prasarana di Politeknik Harapan Bersama”. Pengelolaan data sarana dan prasarana tentu sudah menjadi tanggung jawab perguruan tinggi, adapun tujuan penelitian ini adalah membangun sistem informasi inventaris sarana dan prasarana untuk mempermudah dalam pengelolaan serta monitoring perawatan sarana dan prasarana di Politeknik Harapan Bersama. Penelitian ini merupakan penelitian eksperimen dimana proses awal dimulai dengan perencanaan yang matang, analisis data serta

design yang tersusun dan terarah. Dalam penelitian ini menggunakan metode pengumpulan data berupa observasi, wawancara dan studi literatur [6].

Susanto & Azzam S pada tahun 2020 juga melakukan penelitian berjudul “Pengembangan Aplikasi Sarana dan Prasarana Berbasis Web”. Peneliti melakukan penelitiannya di Universitas Teknologi Sumbawa, sebenarnya perguruan tinggi tersebut sudah memiliki sistem informasi untuk pengelolaan sarpras namun dirasa masih belum efisien dan maksimal sehingga perlu pengembangan lebih lanjut. Oleh karena itu, penelitian ini bertujuan untuk melakukan pengembangan sistem informasi sarana dan prasarana di Universitas Tekonologi Sumbawa berbasis web. Penelitian ini menggunakan metode pengembangan perangkat lunak spiral yang terdiri dari komunikasi, analisis risiko, rekayasa, konstruksi dan evaluasi. Dengan adanya pengembangan ini, diharapkan dapat meningkatkan atau memperbaiki proses sistem terutama dalam peminjaman, pengembalian dan pelaporan sarana dan prasarana [7].

Penelitian serupa dilakukan oleh Priyono, Krisbiantoro, dan Kusuma pada tahun 2020 dengan judul “Sistem Informasi Sarana dan Prasarana Sekolah Berbasis *Website* Smk Bakti Purwokerto”. Tujuan dilakukannya penelitian ini adalah mempermudah pengelolaan sarana dan prasarana yang ada di SMK Bakti Purwokerto yang semula masih menggunakan pencatatan dibuku. Sistem informasi ini berbasis *website* dengan metode pengembangan sistem prototype. Dengan adanya sistem informasi diharapkan dapat mempermudah staf Wakil Sarana dan Prasarana dalam melakuka pendataan, pengelompokan data sarana dan prasarana serta pencarian dengan QR Code [8].

Pada penelitian ini penulis bermaksud membangun sebuah sistem informasi manajemen sarana dan prasarana sekolah berbasis *website* di MI Ya BAKII Kesugihan 02 menggunakan bahasa pemrograman PHP, *framework* Codeigniter dan *database* MySQL. Sistem ini bertujuan untuk mempermudah petugas sarana dan prasarana dalam proses pencatatan data sarana dan prasarana, pemeliharaan sarana dan prasarana, pencetakan data inventaris setiap ruangan, peminjaman dan pengembalian barang serta laporan. Dengan sistem tersebut petugas dapat membuat nomor sesuai dengan pedoman dan mencetak kode atau penomoran inventaris sarana dan prasarana tersebut. Adanya fitur pemeliharaan, sehingga pengecekan kondisi sarana dan prasarana

terstruktur dan apabila ditemukan kerusakan dapat segera dilakukan perbaikan.

2.2 Landasan Teori

2.2.1. Sistem

Sistem merupakan suatu kesatuan utuh yang terdiri dari bagian-bagian yang saling berhubungan dan berinteraksi untuk mencapai sebuah tujuan [9]. Sistem juga dapat diartikan sebagai sekumpulan unsur atau elemen yang saling berkaitan dan mempengaruhi untuk mencapai tujuan tertentu. Unsur-unsur sistem terdiri dari masukan (*input*), proses (*processing*) dan keluaran (*output*) [10].

2.2.2. Informasi

Informasi merupakan hasil dari data yang telah diolah menjadi bentuk yang lebih berguna, menggambarkan suatu kejadian nyata dan dapat digunakan sebagai alat untuk pengambilan sebuah keputusan [9]. Informasi adalah data yang telah diolah, dibentuk maupun dimanipulasi sesuai dengan kebutuhan penggunaannya sehingga hasilnya memiliki nilai yang lebih bermakna dan bermanfaat bagi penerimanya [10].

2.2.3. Sistem Informasi Manajemen

Secara etimologis, manajemen dalam bahasa Perancis kuno yaitu *management*, yang artinya seni melaksanakan dan mengatur. Sedangkan dalam bahasa Inggris manajemen berasal dari kata *to manage* yang berarti mengelola atau mengatur. Dengan demikian, dapat disimpulkan bahwa secara etimologis manajemen adalah sebuah kegiatan untuk mengelola. Manajemen adalah suatu kegiatan atau proses khusus yang meliputi perencanaan, pengorganisasian, pengarahan, serta pengendalian untuk mencapai suatu sasaran tertentu dengan memanfaatkan sumber daya manusia dan sumberdaya lainnya [11].

Sistem informasi dapat didefinisikan sebagai sistem yang diciptakan oleh manusia yang terdiri dari beberapa komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi juga diartikan sebagai kumpulan prosedur yang memberikan informasi bagi pengambil keputusan [12]. Berikut adalah komponen dari sistem informasi :

1. *Hardware* (perangkat keras) dan *Software* (perangkat lunak) yang berperan sebagai mesin.

2. *People* dan *procedures* merupakan manusia dan tatacara menggunakan mesin.
3. Data merupakan media penghubung antara manusia dan mesin sehingga terjadi suatu proses pengolahan data.

Dari uraian diatas sistem informasi manajemen dapat didefinisikan sebagai sistem berbasis komputer yang menyediakan informasi bagi para pengguna yang memiliki kebutuhan tertentu [13]. Sistem informasi manajemen menggabungkan kinerja ilmu komputer, ilmu manajemen, dan riset operasional berorientasi praktik guna mengembangkan solusi secara sistematis untuk memecahkan masalah di dunia nyata sekaligus mengelola sumber daya teknologi informasi [14].

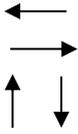
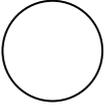
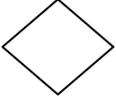
2.2.4. Sarana dan Prasarana Pendidikan

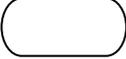
Sarana pendidikan merupakan perlengkapan yang digunakan dalam proses pendidikan secara langsung, sedangkan prasarana pendidikan merupakan fasilitas yang menunjang jalannya proses pendidikan secara tidak langsung [15]. Sehingga dapat disimpulkan bahwa sarana dan prasarana pendidikan adalah merupakan suatu alat atau fasilitas yang memiliki peran penting dalam memberikan kemudahan dan kelancaran penyelenggaraan proses pendidikan khususnya pada kegiatan pembelajaran seperti meja, kursi, halaman kebun, taman, dll.

2.2.5. Flowchart

Flowchart merupakan bagan yang memiliki arus dan menggambarkan langkah dalam penyelesaian suatu masalah. Bagan pada *flowchart* memperlihatkan urutan proses dalam sistem dengan menunjukkan alat media masukan, keluaran dan media penyimpanan dalam proses pengolahan data. Selain itu, pada bagan *flowchart* terdapat instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program. Simbol-simbol dalam *flowchart* dibagi menjadi 3 kelompok yaitu Simbol Penghubung/alur (*Flow Direction Symbols*), Simbol Proses (*Processing Symbols*) dan Simbol Input-Output (*Input –Output Symbols*) [12]. Simbol *flowchart* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Simbol *Flowchart*

No.	Simbol	Nama	Fungsi
1.		Arus/Flow	Menyatakan jalannya arus suatu proses.
2.		<i>Communication Link</i>	Menyatakan adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
3.		<i>Connector</i>	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
4.		<i>Offline Connector</i>	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
5.		Proses	Simbol yang menunjukkan setiap pengolahan yang akan dilakukan oleh komputer.
6.		Manual	Menyatakan suatu tindakan proses yang tidak dilakukan oleh komputer.
7.		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak.

8.		<i>Predefined Proses</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
9.		<i>Terminal</i>	Menyatakan awal atau akhir suatu program.
10.		<i>Keyring Operation</i>	Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.
11.		<i>Off-line storage</i>	Data dalam simbol ini akan disimpan ke suatu media tertentu.
12.		<i>Manual input</i>	Memasukan data secara manual dengan menggunakan online keyboard.
13.		<i>Input-output</i>	Menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
14.		<i>Punched card</i>	Menyatakan <i>input</i> berasal dari kartu atau output ditulis ke kartu.
15.		<i>Magnetic tap unit</i>	Menyatakan <i>input</i> berasal dari pita magnetic atau output disimpan ke pita magnetic.

16.		<i>Disk storage</i>	Menyatakan <i>input</i> berasal dari disk atau output ditulis ke disk.
17.		<i>Document</i>	Mencetak laporan ke printer
18.		Display	Menyatakan peralatan output yang digunakan berupa layar (video, komputer)

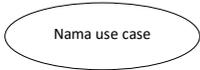
2.2.6. UML

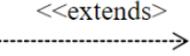
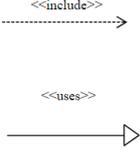
UML (*Unified Modeling Language*) merupakan bahasa berupa grafik/gambar yang berguna untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis *Object-Oriented* [16].

1. Use Case Diagram

Use case diagram merupakan pemodelan yang mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* diagram digunakan untuk mengetahui fungsi-fungsi pada sistem informasi dan siapa saja yang berhak menggunakan fungsi tersebut. Aktor dan *use case* merupakan dua hal utama dalam sebuah usecase diagram [17]. Simbol-simbol dalam *use case* dapat dilihat pada tabel 2.2.

Tabel 2. 2 Simbol *Use Case Diagram*

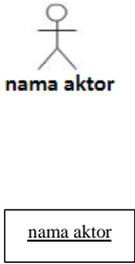
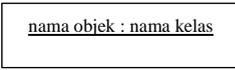
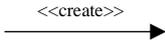
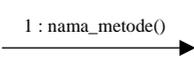
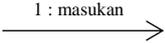
No.	Simbol	Nama	Fungsi
1.		<i>Use case</i>	Fungsionalitas yang tersedia pada sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.		Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar

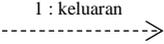
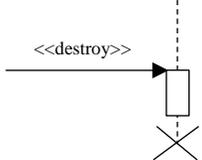
			sistem informasi yang akan dibuat itu sendiri.
3.		Asosiasi	Penghubung komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
4.		Ekstensi	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan tersebut.
5.		Generalisasi	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> .
6.		<i>Include</i> atau <i>uses</i>	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. <i>Use case</i> tambahan akan mengecek apakah <i>use case</i> yang ditambahkan sudah dijalankan sebelum menjalankan <i>use case</i> tambahan.

2. *Sequence Diagram*

Sequence merupakan diagram yang menggambarkan perilaku objek pada *use case* dengan mendefinisikan waktu hidup serta pesan yang dikirimkan maupun diterima antar objek. *Sequence diagram* digambar minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. [17]. Simbol-simbol dalam *sequence* dapat dilihat pada tabel 2.3.

Tabel 2. 3 Simbol *Sequence Diagram*

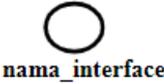
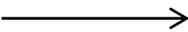
No.	Simbol	Nama	Fungsi
1.		Aktor Tanpa waktu aktif	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu orang walaupun simbol dari aktor adalah gambar orang.
2.		Garis hidup atau <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan
4.		Waktu aktif	Menyatakan semua objek yang terhubung dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek lain.
6.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil metode pada objek lain atau dirinya sendiri.
7.		Pesan tipe <i>send</i>	Menyatakan suatu objek mengirimkan data ke objek lainnya.

8.		Pesan tipe <i>return</i>	Menyatakan suatu objek telah menjalankan suatu operasi atau metode dan menghasilkan kembalian pada objek tertentu.
9.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain.

3. *Class Diagram*

Class Diagram merupakan salah satu jenis diagram UML yang menggambarkan struktur sistem dari sudut pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, didalam kelas terdapat deskripsi *class*, atribut, metode, dan hubungan dari setiap objek [17]. Simbol-simbol dalam *class diagram* dapat dilihat pada tabel 2.4.

Tabel 2. 4 Simbol *Class Diagram*

No.	Simbol	Nama	Fungsi
1.		Kelas	Kelas pada struktur sistem
2.		Antarmuka/ <i>interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.		Asosiasi/ <i>association</i>	Relasi antar kelas dengan makna umum.
4.		Asosiasi berarah/ <i>direct association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain.

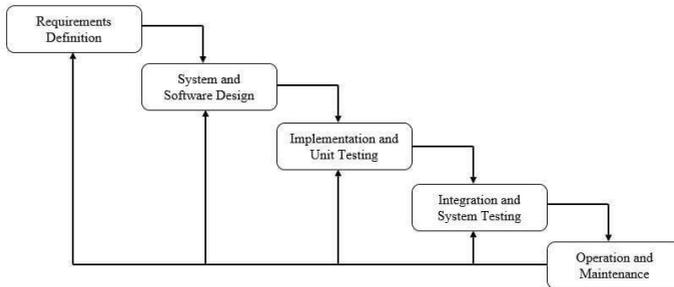
5.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.		Kebergantungan/ <i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.		Agregasi/ <i>aggregation</i>	Relasi antarkelas dengan makna semua bagian.

2.2.7. Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak merupakan suatu bidang profesi menekuni cara-cara pengembangan perangkat lunak termasuk didalamnya pembuatan, pemeliharaan, manajemen, organisasi pengembangan perangkat lunak, dan sebagainya [17]. RPL juga dapat didefinisikan sebagai disiplin ilmu yang mengulas seluruh aspek produksi perangkat lunak mulai dari tahap spesifikasi hingga pemeliharaan sistem.

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 *waterfall*. Model *waterfall* merupakan salah satu siklus hidup klasik (*classic life cycle*) dalam pengembangan suatu perangkat lunak. Metode air terjun ini menggambarkan pendekatan yang cukup sistematis dan berurutan pada pengembangan perangkat lunak, metode ini disebut *waterfall* karena tahapan-tahapan yang dilalui harus menunggu selesainya tahap sebelumnya [2]. Dalam proses pengembangannya dilakukan bertahap satu persatu sehingga meminimalisir kesalahan yang mungkin terjadi. Berikut adalah tahapan dari metode pengembangan sistem *waterfall* dapat dilihat pada gambar 2.1.



Gambar 2. 1 Metode *Waterfall*
 Sumber : Ian Somerville 2011 [18]

1. **Analisa Kebutuhan (*Requirements Definition*).**
 Pada tahap analisa kebutuhan pengembang harus memahami kebutuhan perangkat lunak yang diinginkan oleh pengguna beserta dengan batasan-batasannya. Informasi tersebut dapat diperoleh melalui wawancara, observasi maupun diskusi.
2. **Desain Sistem (*System and Software Design*)**
 Tahap selanjutnya setelah menyelesaikan analisis adalah perancangan sistem. Tahap desain bertujuan memberikan gambaran lengkap terkait apa yang harus dikerjakan dan bagaimana tampilan sistem yang dibutuhkan pengguna. Sehingga pada tahap ini sangat membantu dalam menentukan arsitektur sistem secara keseluruhan.
3. **Penulisan Kode Program (*Implementation and Unit Testing*)**
 Merupakan tahap pembuatan perangkat lunak sesuai dengan perancangan sistem yang telah dibuat pada tahap sebelumnya. Proses penulisan kode berada di tahap ini, dimana pembuatan perangkat lunak dipecah menjadi beberapa modul kecil yang nantinya akan digabungkan. Pada tahap ini juga dilakukan pemeriksaan untuk mengetahui apakah modul yang dibuat telah memenuhi fungsi atau belum.
4. **Pengujian Program (*Integration and System Testing*)**
 Pada tahap ini dilakukan penggabungan modul-modul yang dibuat pada tahap sebelumnya. Kemudian dilakukan pengujian terhadap keefektifan dan kemampuan sistem bertujuan untuk mengetahui apakah sistem sudah berjalan dengan baik atau masih terdapat kesalahan. Jika masih terdapat kesalahan, maka perlu pengkajian

ulang serta perbaikan perangkat lunak hingga sesuai yang diharapkan pengguna.

5. Penerapan Program dan Pemeliharaan (*Operation and Maintenance*)

Tahap ini merupakan tahap terakhir pada metode pengembangan model waterfall. Perangkat lunak yang sudah jadi akan dioperasikan oleh penggunanya, kemudian dilakukan pula pemeliharaan perangkat lunak termasuk didalamnya perbaikan terhadap kesalahan-kesalahan implementasi sistem.

B. Metode Pengujian Sistem

Pentingnya pengujian sangat dibutuhkan dalam penilaian kualitas perangkat lunak supaya fungsi yang ada didalamnya dapat bekerja sesuai dengan harapan pengguna. Metode yang digunakan dalam pengujian ini adalah metode *black-box*. Metode *black-box* merupakan metode yang bertujuan untuk melihat apakah program tersebut sudah sesuai dengan fungsionalitas yang diharapkan pengguna tanpa mengetahui kode program yang dipakai. Pengujian *black-box* pada perangkat lunak bertujuan untuk mengetahui kesalahan-kesalahan pada fungsi yang tidak benar atau hilang, *interface*, struktur data atau pengaksesan *database*, kesalahan kinerja, inisialisasi dan kesalahan terminasi [3].

2.2.8. PHP

PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming* karena keseluruhan dari proses dijalankan pada server. PHP juga merupakan bahasa pemrograman *Open Source* atau pengguna pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya masing-masing [19]. Dengan memanfaatkan basis data, bahasa pemrograman PHP dapat menghasilkan halaman *website* yang dinamis. Sistem manajemen basis data yang biasa digunakan bersamaan dengan PHP adalah MySQL. Selain itu, PHP juga dapat dijalankan diberbagai macam sistem operasi seperti windows, mac os, linux, dll [20].

2.2.9. Basis Data (*Database*)

Basis Data (*database*) merupakan kumpulan dari suatu data yang terhubung dan disimpan secara bersama pada suatu media,

diorganisasikan berdasarkan skema atau struktur tertentu, serta dapat dilakukan manipulasi untuk kegunaan tertentu dengan bantuan *software*. Basis data juga dapat didefinisikan sebagai kumpulan data yang disusun dalam beberapa tabel yang saling memiliki relasi satu sama lain atau berdiri sendiri [21].

DBMS (*Database Management System*) merupakan *software* yang didesain agar user dapat mendefinisikan, membuat, memelihara dan menyediakan akses terkontrol pada data [21]. Salah satu server yang menyediakan layanan sistem manajemen basis data adalah MySQL. Bahasa dalam basis data dibedakan menjadi 2 yaitu *Data Definision Language* (DDL) yang berisi perintah membuat, mengubah dan menghapus serta *Data Manipulation Language* (DML) yang berisi kumpulan perintah *query* untuk manipulasi data pada tabel basis data.

2.2.10. Framework

Framework merupakan sekumpulan instruksi dalam sebuah *class* dan *function*, yang memiliki fungsi masing-masing untuk memudahkan pengembang memanggilnya tanpa harus menulis kode program yang sama secara berulang-ulang sehingga mempercepat dalam penyelesaian suatu program. Tersedia berbagai macam *framework* PHP yang sangat membantu dalam pengembangan *website* atau aplikasi, salah satunya adalah Codeigniter. Codeigniter merupakan salah satu *framework* PHP bersifat *open source* menggunakan metode MVC (Model, View dan Controller) untuk memisahkan data, tampilan dan prosesnya sehingga pembuatan dan pengelolaan program akan lebih cepat dan mudah [22].

2.2.11. Usability Testing

Usability Testing adalah salah satu kategori metode dalam evaluasi usability yang digunakan untuk mengevaluasi sebuah produk dengan mengujinya langsung pada pengguna. Tujuannya adalah untuk mengidentifikasi masalah uji ketergunaan seperti, mengumpulkan data kualitatif dan kuantitatif, mengukur kemudahan, mengukur efisiensi dan menentukan kepuasan pengguna dengan produk [23].

Pertanyaan pada kuesioner *usability testing* harus memuat komponen di bawah ini [24] :

1. *Learnability*, adalah tingkat kemudahan bagi seorang pengguna sistem yang dapat dilihat berdasarkan pemakaian fitur dan fungsi yang terdapat dalam sistem

2. *Efficiency*, adalah tingkat kecepatan seorang pengguna dalam melakukan fungsi-fungsi yang terdapat dalam sistem.
3. *Memorability*, adalah tingkat kemampuan seorang pengguna dalam mengingat dan mempertahankan pengetahuannya setelah beberapa waktu tidak mengoperasikan sistem.
4. *Errors*, adalah tingkat atau jumlah *error* atau kesalahan yang dilakukan oleh seorang pengguna sistem, tingkat kebosanan dalam menghadapi error, dan cara memperbaikinya.
5. *Satisfaction*, adalah tingkat kepuasan pengguna setelah berinteraksi dengan sistem

Setiap pertanyaan pada pengujian *usability* menggunakan skala Likert, dan memiliki interval jawaban 1- 4. Skala likert digunakan untuk mengukur pendapat, persepsi, atau sikap dari perorangan atau sekelompok orang terhadap fenomena sosial tertentu. Adapun jawaban setiap pertanyaan yang digunakan dapat dinilai dengan skor seperti tabel 2.5.

Tabel 2. 5 Skala Likert

Jawaban	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Setuju	3
Sangat Setuju	4

Hasil dari kuesioner yang telah dijawab oleh responden kemudian dihitung menggunakan rumus berikut:

$$U = \frac{np}{nm} \times 100\%$$

~Halaman Ini Sengaja Dikosongkan~