# BAB II

# TINJAUAN PUSTAKA

* 1. **Peneliti Terdahulu**

Dalam penelitian diperlukan dukungan dari hasil penelitian yang sudah ada sebelumnya dan berkaitan dengan penelitian tersebut.

Dari hasil penelitian (Hartanto, 2016) yang berjudul “Penggunaan *E-Learning* Sebagai MediaPembelajaran” menjelaskan bahwa beberapakonsekuensi logis yang terjadi antara lain adalah peserta didik dapat dengan mudah mengambil materi pembelajaran diamanapun tanpa terbatas lagi pada batasan tempat dan waktu. Peserta didik dapat dengan mudah berguru dan berdiskusi dengan para tenaga ahli atau pakar di bidang yang diminatinya. Materi pembelajaran bahkan dapat dengan mudah diambil di berbagai penjuru dunia tanpa bergantung pada dimana siswa belajar.

Dari hasil penelitian (Fuad et al., 2013) yang berjudul “Rancang Bangun Sistem Informasi E-Learning Berbasis Web di SMK Negeri 1 Tangerang” menjelaskan bahwaSistem Informasi *E-learning* SMK Negeri 1 Tangerang memudahkan siswa dalam memperoleh referensi pembelajaran. Karena disediakan halaman materi untuk melihat dan mendownload dokumen sesuai mata pelajaran yang di upload oleh guru mata pelajaran tersebut. Sehingga siswa termotivasi belajar secara mandiri. Sistem Informasi E-learning SMK Negeri 1 Tangerang memudahkan para guru untuk dapat mendistribusikan materi pelajaran dan tugas. Serta diberikannya waktu pengerjaan tugas untuk dapat mengontrol tugas yang diberikan. Adanya tugas menjadi pelatihan bagi siswa dalam pembelajaran yang akan meningkatkan pemahaman siswa tentang materi tersebut. Sistem Informasi E-learning SMK Negeri 1 Tangerang dirancang untuk membantu dan menjadi sarana dalam melakukan proses pembelajaran.

Dengan demikian, Aplikasi *E-Learning* yang dibangun bisa bermanfaat untuk proses pembelajaran secara *online.*

* 1. **Pengertian Aplikasi**

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata *apply* yang dalam Bahasa Indonesia berarti pengolah. Aplikasi adalah suatuprogram siap pakai yang dapat digunakan untuk menjalankan perintah dari pengguna aplikasi dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut (Priana & Fitriani, 2017).

* 1. **Pengertian *E-Learning***

*E-Learning* merupakan sebuah metode belajar mengajar secaraelektronik yang memungkinkan tersampaikannya bahan ajar dengan menggunakan internet atau media lainnya. Selain itu, *e-learning* sangat efektif dan fleksibel penggunaannya karena dapat diakses dimana saja dan menghemat waktu (Wassalam et al., 2017).

Menurut (Rohman & Suratno, 2019), manfaat yang terdapat pada *e-learning* yaitu:

1. Penggunaan *e-learning* untuk menunjang pelaksanaan proses belajar, dapat meningkatkan daya serap siswa atas materi yang diajarkan.
2. Meningkatkan partisipasi aktif dari siswa.
3. Meningkatkan kemampuan belajar mandiri siswa.
4. Meningkatkan kualitas materi pendidik dan pelatihan.

Menurut (Rohman & Suratno, 2019) , beberapa karakteristik yang terdapat pada *e-learning*:

1. Memanfaatkan jasa teknologi elektronik.
2. Memanfaatkan keunggulan komputer (media digital dan *computer* *networking*).
3. Menggunakan bahan ajar yang bersifat mandiri (*self learning* *materials*) kemudian disimpan di komputer, sehingga dapat diaksesoleh guru dan siswa kapan saja dan dimana saja.
4. Memanfaatkan jadwal pembelajaran, kurikulum, hasil kemajuan belajar, dan hal-hal yang berkaitan dengan administrasi pendidikan dapat dilihat setiap saat di komputer.
   1. **Pengertian *Website***

*World wide web* (www) atau lebih dikenal situs web adalah suatuhalaman web yang saling berhubungan antara satu dengan yang lainnya dimana data web tersebut berada pada server yang sama berisikan macam-macam kumpulan suatu informasi yang bisa disediakan perorangan, kelompok dan organisasi (Rahman & Ratna, 2018).

* 1. **Pengertian *CodeIgniter***

*CodeIgniter* adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal. Dalam situs resmi *codeigniter*, menyebutkan bahwa *codeigniter* merupakan *framework* PHPyang kuat dan sedikit bug. *Codeigniter* ini dibangun untuk para pengembang dengan bahasa pemrogram PHP yang membutuhkan alat untuk membuat web dengan fitur lengkap (Destiningrum & Adrian, 2017).

*Framework Codeigniter* dikembangkan oleh Rick Ellis, CEO Ellislab,Inc. Kelebihan dari *framework codeigniter* jika dibandingkan dengan *framework* lain adalah sebagai berikut:

1. Gratis (*Open-Source*)

Kerangka kerja *Codeigniter* memiliki lisensi dibawah Apache/BSD *open-source* sehingga bersifat bebas atau gratis.

1. Berukuran kecil

Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan *framework* lain yang berukuran besar dan membutuhkan *resource* yang besar dan juga dalam eksekusi maupunpenyimpanannya.

1. Menggunakan konsep M-V-C

*Codeigniter* merupakan konsep M-V-C (*Model- View-Controller*)yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Dengan konsep ini kode PHP, *query MySQL*, *Javascript* dan CSS dapat saling dipisah-pisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.

* 1. Model Kode merupakan program (berupa OOP *class*) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (*input-edit-delete*).
  2. View merupakan kode program berupa *template* atau PHP untuk menampilkan data pada browser.
  3. Controller merupakan kode program (berupa OOP *class* ) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan *view.*
  4. **Pengertian *Email***

Pada pengembangan sistem ini, penulis menggunakan notifikasi *email* untuk membantu memberikan informasi kepada *user*. *Email* merupakansurat elektronik yang penggunanya menggunakan internet yang berbasis file teks, namun dengan perkembangan teknologi, *email* lebih atraktif terhadap penggunanya. Tidak hanya dapat mengirim file teks, tetapi juga dapat mengirim file audio, video, foto dan file ekstensi lainnya (Horiyah, Bambang Sugantoro, 2016).

* 1. **Pengertian PHP (*Hypertext Prepocessor*)**

PHP adalah salah satu bahasa pemrograman web berbasis server yang mampu memparsing kode PHP dengan ekstensi PHP sehingga menghasilkan tampilan website yang dinamis (Sitohang, 2018).

* 1. **Pengertian HTML (*Hyper Text MarkupLanguage*)**

HTML merupakan bahasa pemrograman yang digunakan untuk menampilkan dokumen pada browser dalam sebuah web. Bertujuan untuk mendefiniskan struktur dokumen web dan tata letak tampilan (Jayanti, Dwi & Iriani, 2014).

* 1. **Pengertian *Xampp***

*Xampp* adalah salah satu paket instalasi apache, PHP, dan MySQLsecara instant yang dapat digunakan untuk menghubungkan antara ketiga tersebut (Sitohang, 2018).

* 1. **Pengertian MySQL (*My Structure Query Language*)**

MySQL ( *My Structure Query Language* ) adalah sebuah software database dengan tipe data relasional yang artinya penyimpanan datanya dalam bentuk tabel yang saling berhubungan (Sitohang, 2018).

* 1. **Pengertian PHPMyAdmin**

PHPMyAdmin adalah aplikasi web yang dibuat oleh phpmyadmin.net.phpMyAdmin digunakan untuk administrasi database MySQL (Simangunsong, 2018).

* 1. **Pengertian *Database***

*Database* atau basis data adalah sekumpulan data yang berelasiditunjukkan dengan kata kunci dari tiap data yang ada. Basis data menampilkan data dalam bentuk tabel-tabel, dimana tabel-tabel tersebut saling berhubungan oleh nilai-nilai yang sama pada kolom terkait (Anisah & Mayasari, 2016).

* 1. **Alat Bantu Perancangan Sistem**
     1. ***UML(Unified Modelling Language)***

*UML* (*Unified Modelling Language*) adalah salah satu alatbantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2018).

* + 1. ***Use Case Diagram***

*Use case diagram* merupakan salah satu diagram yangmenggambarkan perilaku sistem. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna (Munawar, 2018).

Berikut ini simbol/notasi/komponen dan penjelasan dari use case diagram yang ada pada Tabel 2.1

Tabel 2.1. Simbol *Use Case Diagram* (Munawar, 2018)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Aktor | Mewakili peran Orang, sistem yang lain, atau alat ketika berkomnikasi dengan *usecase.* |
|  | *Use Case* | Abstraksi dan interaksi antara sistem dan aktor. |
|  | *Association* | Abstraksi dari penghubung antara aktor dengan *use case.* |
|  | *Generalisasi* | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan *use case.* |
|  | *Include* | Menunjukkan bahwa suatu *use case*seluruhnya merupakan fungsionalitas dari *usecase* lainnya. |
|  | *Extend* | Menunjukkan bahwa suatu usecasemerupakan tambahan fungsional dari usecase lainnya jika suatu kondisi terpenuhi. |

* + 1. ***Class Diagram***

*Class diagram* adalah diagram statis. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak. *Class diagram* menggambarkan atribut, *operation* dan juga *constraint* yang terjadi pada sistem. *Class diagram* juga dikenal sebagai diagram struktural (Munawar, 2018).

Untuk simbol-simbol dan penjelasan dari class diagram, saya sudah saya sediakan di tabel 2.2.

Tabel 2.2. Simbol *Class Diagram* (Munawar, 2018)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Generalization | Hubungan dimana objek turunan (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk. |
|  | Nary Association | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
|  | Class | Himpunan dari objek-objek berbagi atribut serta operasi yang sama. |
|  | Collaboration | Deskripsi dari urutanaksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | Realization | Operasi yang benar-benar dilakukkan suatu objek. |
|  | Dependency | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | Association | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

* + 1. ***Activity Diagram***

*Activity diagram* adalah bagian penting dari *UML* yangmenggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis, dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity* *diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akantetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa (Munawar, 2018).

Inilah simbol atau komponen dari activity diagram, antara lain:

Tabel 2.3. Simbol *Activity Diagram* (Munawar, 2018)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Start Point | Bagaimanan objek dibentuk atau diawali. |
|  | End Point | Bagaimana objek dibentuk atau diakhiri. |
|  | Activity | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
|  | Fork | Percabangan yang menunjukkan aliran pada *activity diagram.* |
|  | Join | Penggabungan yang menjadi arah aliran pada *activity diagram*. |
|  | Decision | Pilihan untuk mengambil keputusan. |
| Swimlane |  | Sebuah cara untuk mengelompokkan *activity* berdasarkan aktor (mengelompokkan *activity* dalam urutan yang sama. |

* + 1. ***Sequence Diagram***

*Sequence diagram* digunakan untuk menggambarkan perilaku padasebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*. Komponen utama sequence diagram terdiri atas obyek yangdituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal (Munawar, 2018).

Berikut adalah komponen - komponen yang terdapat di dalam Sequence Diagram.

Tabel 2.4. Simbol *Sequence Diagram* (Munawar, 2018)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | LifeLine | Objek entity, antarmuka yang saling berinteraksi. |
|  | Entity Class | Menggambarkan hubungan yang akan dilakukan. |
|  | Boundary Class | Menggambarkan sebuah penggambaran dari form. |
|  | Control Class | Menggambarkan penghubung antara *boundary* dengan tabel. |
|  | Message | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yanng terjadi. |

* 1. **Pengertian Pengujian Sistem**

Pengujian sistem yaitu elemen kritis dari jaminan perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain, dan pengodean. Selain itu, pengujian juga dapat diartikan sebagai sebuah proses eksekusi suatu program dengan maksud menentukan kesalahan (Destiningrum & Adrian, 2017).

Pengujian untuk validasi sistem memiliki beberapa pendekatan sebagai berikut :

a. *White-Box Testing* (Pengujian Kotak Putih)

*White-Box Testing* yaitu salah satu cara untuk menguji suatuaplikasi atau *software* dengan cara meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak. Jika model yang sudah dihasilkan berupa *output* yang tidak sesuai dengan yang diharapkan maka akan dikompilasi ulang dan dicek kembali kode-kode tersebut hingga sesuai dengan yang diharapkan (Suprapti et al., 2017). Langkah penyelesaian *White-Box* sebagai berikut:

1. Menganalisa sistem berdasarkan alur *flowchart* aplikasi *e-learning*.
2. Membuat *flowgraph* berdasarakan alur *flowchart*.
3. Menentukan jalur independen berdasarkan gambar *flowgraph*.
4. Menghitung kompleksitas siklomatis berdasarkan jalur independen yang dilalui. Untuk menghitung kompleksitas siklomatis ada 3 cara yaitu :
   1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis.
   2. Kompleksitas siklomatis V(G) untuk grafik alir G ditentukan sebagai V(G) = E – N + 2 , dimana E adalah jumlah *edge* grafik alir dan N adalah jumlah simpul grafik alir.
   3. Kompleksitas siklomatis V(G) untuk grafik alir G ditentukan sebagai V(G) = P + 1 , dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir G.
5. Melakukan *test case.*

b. *Black-Box Testing* (Pengujian Kotak Hitam)

*Black-Box Testing* yaitu pengujian yang terfokus pada spesifikasifungsional dari *software,* penguji dapat mendefinisikan sekumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Iskandar & Hamdani, 2017) Menurut (Nurudin et al., 2019) , pengujian ini memberikan gambaran atas sekumpulan kondisi masukan dan melakukan pengujian pada uraian fungsional program. *Black-Box* *Testing* digunakan untuk mendeteksi permasalahan berikut:

1. Fungsi yang salah atau hilang.
2. Kesalahan pada *interface*.
3. Kesalahan struktur data dan basis data.
4. Kesalahan Fungsi.
5. Kesalahan deklarasi dan terminasi.