



BAB II

TINJAUAN PUSAKA

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Pada penelitian yang dilakukan oleh Hartatik, Iskandar Jati Cahyaningsih, Agus Purnomo, Rudi Hartono dan Sahirul Alim Tri Bawono (2017) dengan judul **“Pengembangan Aplikasi E-Learning Sekolah Menengah Atas”** Secara umum, peran e-learning dalam proses pembelajaran dapat dikelompokkan menjadi dua, yaitu komplementer dan substitusi. Model komplementer menerapkan pembelajaran dengan pertemuan tatap muka masih berjalan tetapi ditambah dengan model e-learning, model substitusi menerapkan sebagian besar proses pembelajaran.

Kemudian Roni Yunis, Kristian Telaumbanua (2017) melakukan penelitian berjudul **“Pengembangan E-Learning Berbasis LMS untuk Sekolah, Studi Kasus SMA/SMK di Sumatera Utara”** e-learning dikembangkan, disusun rencana implementasi, yaitu dengan melakukan pelatihan terkait penggunaan e-learning. Hasil peningkatan kemampuan pengguna terhadap rencana penerapan e-learning melalui pelatihan akan dilakukan melalui evaluasi atau pengukuran, sehingga diharapkan dapat dijadikan sebagai dasar dalam pengambilan keputusan terkait dengan implementasi e-learning untuk sekolah. Hasilnya diharapkan dapat meningkatkan proses belajar mengajar yaitu dengan memanfaatkan e-learning yang sudah dikembangkan, dan diimplementasikan di beberapa sekolah yang ada di Sumatera Utara.

Kemudian menurut Farida Nur Kumala, Cicilia Ika Rahayu Nita, Alvied Vernanda Agustin (2020) melakukan penelitian berjudul **“Pengembangan Multimedia E-Learning Materi Ekologi Siswa Sekolah Dasar”** Penelitian ini dilatarbelakangi perlunya media pembelajaran berupa Multimedia E-Learning materi Ekologi Siswa Sekolah Dasar. Tujuan dari penelitian ini untuk mengembangkan media pembelajaran khususnya pada materi ekologi siswa sekolah dasar kelas V dan mengetahui kelayakan media yang dikembangkan. Penelitian ini menggunakan model ADDIE dengan pengumpulan data menggunakan lembar validasi. Teknik analisis data menggunakan analisis data kuantitatif. E-Learning materi Ekologi Siswa Sekolah Dasar yang dihasilkan dinyatakan layak atau valid dengan skor rata-rata 89% yang menunjukkan kategori baik dan layak digunakan. Sehingga Multimedia E-Learning materi Ekologi Siswa Sekolah Dasar layak digunakan

II.2. Landasan Teori

Sebelum membuat sebuah *E-Learning* ada beberapa teori yang harus di mengerti terlebih dahulu, meliputi teori dari para ahli dan juga teori umum yang berhubungan dengan *E-Learning*

II.2.1. E-Learning.

(Rusman 2011: 265) menyatakan bahwa E-Learning memiliki empat karakteristik utama yaitu interaktivitas, kemandirian, aksesibilitas, dan pengayaan. Keempat karakteristik ini merupakan hal yang membedakan E-Learning dengan kegiatan pembelajaran secara konvensional. Media *E-Learning* sendiri dimaksudkan sebagai media pembelajaran yang digunakan sebagai strategi

pembelajaran

II.2.2. Website

Website atau web merupakan suatu dokumen berupa sekumpulan halaman yang berisi berbagai informasi berbentuk digital. Informasi itu bisa berupa teks, gambar, animasi, video atau gabungan dari semuanya yang disediakan melalui internet dan dapat diakses oleh banyak orang di seluruh dunia selama memiliki koneksi internet.

kumpulan dari halaman – halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di dalam Internet. Sebuah halaman web biasanya berupa dokumen yang ditulis dalam format HTML (Hyper Text Markup Language), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser (Bilal Syaid : 2021)

II.2.3. PHP

PHP adalah singkatan dari Personal Home Page yang merupakan bahasa standar yang digunakan dalam dunia website. PHP adalah bahasa pemrograman yang berbentuk script yang diletakkan didalam web server. PHP dapat diartikan sebagai Hypertext Preeprocessor. Ini merupakan bahasa yang hanya dapat berjalan pada server yang hasilnya dapat ditampilkan pada klien. Interpreter PHP dalam mengeksekusi kode PHP pada sisi server disebut server side, berbeda dengan mesin maya Java yang mengeksekusi program pada sisi klien. (Peranginangin, 2006).

II.2.4. MYSQL

MYSQL menurut Raharjo (2011), merupakan RDBMS (server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user. Sedangkan Menurut Kadir (2008) MySQL adalah sebuah software open source yang digunakan untuk membuat sebuah database. Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah Suatu software atau program yang digunakan untuk membuat sebuah basis data yang bersifat open source.

II.2.5. XAMPP.

Menurut Frangky Rawung (2017;7) XAMPP adalah perangkat lunak bebas (free software), yang mendukung untuk banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri sebagai server yang berdiri sendiri (localhost), yang terdiri dari beberapa program, antara lain Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan dan dapat menampilkan halaman web yang dinamis.

Komponen dan fungsi XAMPP dijelaskan pada penjelasan berikut ini :

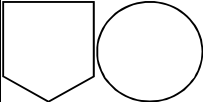
- 1) **Htdoc** adalah folder dimana dapat meletakkan file yang akan dijalankan, seperti file PHP, HTML, dan script lainnya.

- 2) **phpMyAdmin** adalah bagian untuk mengelola database MySQL yang ada di komputer. Untuk membukanya, buka browser dan ketik alamat <http://localhost/phpMyAdmin>, halaman phpMyAdmin akan muncul.
- 3) **Control panel**, yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti stop service (berhenti) atau start (mulai).

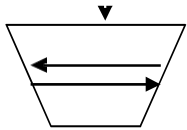
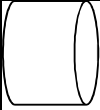
II.2.6. FlowChart.

Bagan Alir (*Flowchart*) adalah bagan (*chart*) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowchart* adalah unit symbol gambar (*chart*) yang menunjukkan aliran (*flow*) dari proses terhadap data. *Flowchart* suatu gambar yang menjelaskan urutan pembacaan data, pemrosesan data, pengambilan keputusan akhir dan penyajian hasil pemrosesan data (Rita Irviani, 2017:65).

Tabel II.2. Simbol Flow Chart

NO	Simbol	Nama	Keterangan
1		Proses komputerisasi	Untuk proses pengolahan data secara komputerisasi.
2		Penghubung	Digunakan untuk menghubungkan sambungan
3		Aliran Sistem	Untuk arah pengaliran data

II.2.6. UML.

4		Proses manual	Untuk proses pengolahan data secara manual
5		Dokumen	Digunakan untuk operasi input.
6		Basis Data	Untuk media penyimpanan

UML (Unified Modeling Language) adalah Spesifik standar yang digunakan untuk mendokumentasikan, menspesifikasi dan membangun aplikasi. UML merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP). Namun demikian model - model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain: *Usecase Diagram*, *Sequence Diagram*, *Activity Diagram* dan *Class Diagram* (Sri Mulyani 2016; 35).

1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh SchullerJ. (2004), “*The purpose of the diagram sisto present multipleviews of a system; this set of multipleviews is called a model*”. Berikut tujuan utama dalam desain UML adalah (Haviluddin; 2015:4) :

- a. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan *visual* yang ekspresif sehingga mereka dapat

mengembangkan dan melakukan pertukaran model data yang bermakna.

- b. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
- c. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
- d. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
- e. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
- f. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
- g. Memiliki integrasi praktik terbaik.

2. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

Struktur diagram dalam UML terdiri atas:

a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam

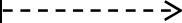
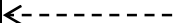
sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel

II.2 di bawah ini :

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target
	sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas- tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .

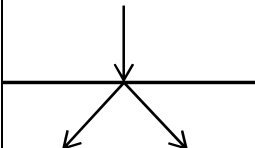
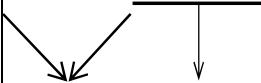
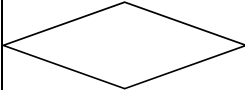
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain,
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain.

(Sumber: Urva; 2015)

1. Activity Diagram

Menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* merupakan *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*) (Dharwiyanti ; 2003).

Tabel II.2 Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .

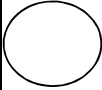
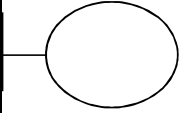
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.
---	--

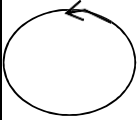
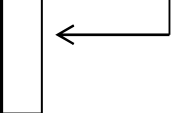
(Sumber : Urva ; 2015)

2. *Sequence Diagram*

Menggambarkan interaksi antar objek didalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* digunakan untuk 9 menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu (Dharwiyanti ; 2003).

Tabel II.3 Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang
	membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.

	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Urva ; 2015)

3. *Class Diagram*

Adalah sebuah spesifikasi yang jika diintansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan deain *beorientasi objek*. *Class* menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan

layanan untuk memanipulasi keadaan tersebut (metode/fungsi) (Dharwiyanti ; 2003).

Tabel II.4. Tabel *Multiplicity Class Diagram*

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva ; 2015)