

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Menurut Sri dalam (Rouf Ashidiq, 2019, p. 1) mengatakan bahwa sistem merupakan gabungan dari dua atau lebih komponen yang saling bekerja dan berhubungan untuk mencapai suatu tujuan tertentu. Sedangkan, menurut Susanto dalam (Astuti & Rapiyanta, 2019) sistem merupakan gabungan dari bagian atau komponen, baik fisik ataupun non fisik yang saling berkaitan satu sama lain dan bekerja sama dalam mencapai suatu tujuan tertentu.

Maka dengan demikian, sistem adalah suatu gabungan dari beberapa komponen yang saling bekerja sama dan berkaitan satu sama lainnya untuk mencapai suatu tujuan tertentu.

A. Model Pembelajaran Berbasis Web

Menurut Firmansyah dan Saidah dalam (Rouf Ashidiq, 2019, p. 1) dalam jurnalnya, mengatakan bahwa proses belajar mengajar pada dasarnya merupakan suatu proses komunikasi yang melibatkan penyampaian sebuah materi dari pemateri ke penerimanya. Dalam penyampaian tersebut, dibutuhkan suatu media agar pesan dapat diterima dengan baik. Media adalah salah satu bagian yang sangat penting dalam proses komunikasi. Tingkat dalam efektifitas media sangat berpengaruh pada sejauh mana peran komunikasi tersebut akan diterima oleh *audience* dengan cepat dan tepat ataupun sebaliknya. Perkembangan dari Teknologi Informasi dan Komunikasi (TIK) yang sangat cepat dan pesat ini mempengaruhi berbagai aspek kehidupan manusia, termasuk dalam interaksi antara guru dengan peserta didik.

Interaksi tersebut membutuhkan dukungan suatu media intruksional yang inovatif, kreatif, tepat dan efektif.

B. Sistem

1. Pengertian Sistem

Menurut Jogiyanto dalam (Aprilian, 2015, p. 60) mengatakan bahwa “sistem merupakan sebagai pendekatan prosedur dan pendekatan komponen. Sebagai pendekatan prosedur, sistem adalah suatu gabungan dan prosedur - prosedur yang memiliki suatu tujuan tertentu. Sedangkan, sebagai pendekatan komponen, sistem menekankan pada komponen yang merupakan gabungan dari komponen yang saling berkaitan antara satu dengan yang lainnya untuk membentuk suatu kesatuan untuk mencapai suatu tujuan tertentu.”

2. Pemrograman Berorientasi Objek (OOP)

Pemrograman berorientasi objek merupakan suatu pendekatan pemrograman yang menggunakan objek dan *class* (Rouf Ashidiq, 2019).

Sedangkan, menurut Sukamto dan Shalahuddin dalam (Boby Sanderson Pardede, 2019, p. 22) mengatakan bahwa “metodologi berorientasi objek merupakan suatu strategi pembangunan pada perangkat lunak yang mengorganisasikan perangkat lunak sebagai gabungan objek yang berisikan data dan operasi yang diberlakukan terhadapnya.”

C. Website

Menurut Yuhefizar dalam (Ali, 2018, p. 7) mengemukakan bahwa “*website* merupakan keseluruhan dari halaman - halaman *web* yang terdapat dari sebuah domain yang mengandung sebuah informasi.”

Menurut Abdulloh dalam (Sintawati & Suminten, 2019) mengatakan bahwa “*website* merupakan kumpulan halaman yang berisikan informasi data digital baik berupa teks, gambar, animasi, audio, video, atau gabungan dari semuanya yang disajikan melalui koneksi *internet* sehingga dapat diakses dan dilihat semua orang dimanapun dan kapanpun diseluruh dunia.”

Menurut Puspitosari dalam (Astuti & Rapiyanta, 2019), jenis - jenis dari *website* adalah sebagai berikut:

a. *Website Statis*

Merupakan *web* yang memiliki halaman yang informasinya jarang berubah atau bahkan tidak sama sekali. Jika ingin melakukan perubahan pada suatu halaman maka dilakukan dengan cara manual dengan mengubah kode yang ada pada struktur dari *web*-nya.

b. *Website Dinamis*

Merupakan sebuah *website* yang informasinya dapat diperbaharui atau di *update*. Contoh dari *website* dinamis ini seperti *web* berita.

c. *Website Interaktif*

Merupakan *web* yang *user*-nya berinteraksi dan beradu argumentasi mengenai apa saja yang menjadi topik pembicaraan mereka. Biasanya *website* tersebut memiliki moderator untuk mengatur agar topik yang diperbincangkan tidak keluar dari jalurnya.

1. Komponen - Komponen pada *Website*

Komponen - komponen yang ada pada *website* adalah sebagai berikut:

a. *Web Browser*

Menurut Sibero dalam (Ali, 2018, p. 7) mengatakan bahwa *web browser* adalah sebuah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan suatu sumber informasi *web*.

b. *Web Server*

Web server merujuk kepada perangkat keras (*server*) dan perangkat lunak yang memiliki penyediaan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun jenisnya (seperti FTP dan HTTPS) atas berkas - berkas yang terdapat dalam suatu URL ke pemakai (Rouf Ashidiq, 2019, p. 7).

c. *Internet*

Menurut Hidayatullah dan Kawistara dalam (Meidawati & Tofat Rapiyanta, 2018) mengatakan bahwa pengertian *internet* adalah suatu jaringan global yang menghubungkan komputer - komputer yang ada di seluruh dunia.

D. Basis Data

Menurut McLeod dan George P. Schell dalam (Santoso, 2016, p. 3) mengatakan bahwa “definisi umum dari basis data merupakan gabungan dari semua data yang berbasis komputer.

Sedangkan, menurut Sukamto dan Shalahuddin dalam (Meidawati & Tofat Rapiyanta, 2018, p. 96), basis data adalah sistem yang terkomputerisasi yang

memiliki tujuan utama, yaitu memelihara data yang telah diolah atau informasi dan membuat informasi tersedia jika dibutuhkan.

Dalam membuat dan mengelola basis data ada beberapa aplikasi yang digunakan, yaitu sebagai berikut:

1. MySQL (*My Structure Query Language*)

Menurut Bunafit dalam (Ramdhani, 2019), MySQL merupakan sebuah *software* atau program *database server*.

Sementara, menurut Sibero dalam (Ramdhani, 2019), MySQL dibaca “*MySekuel*” merupakan RDMS (*Relation Data-base Management System*) yang merupakan sebuah aplikasi sistem yang dijalankan untuk pengolahan data.

Maka dapat disimpulkan bahwa MySQL adalah sebuah *software database* yang berfungsi untuk mengolah data dan menyimpan data dalam bentuk tabel - tabel yang saling berkaitan.

2. XAMPP

Menurut Wahana Komputer dalam (Pramono, 2018), mengemukakan bahwa “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP, dan *Perl*. XAMPP merupakan *tools* yang memiliki paket perangkat lunak dalam satu buah paket.”

3. *PhpMyAdmin*

PhpMyAdmin merupakan sebuah aplikasi atau perangkat lunak yang sifatnya bebas (*open source*) yang ditulis dalam bahasa pemrograman PHP

yang berfungsi untuk menangani administrasi *database* pada MySQL melalui jaringan lokal maupun internet. *PhpMyAdmin* mendukung berbagai operasi dari MySQL seperti mengelola basis data, tabel - tabel, bidang (*fields*), relasi (*relations*), indeks, penggunaan (*user*), perijinan (*permissions*), dan lain sebagainya (Standisyah & Restu, 2017).

E. Model Pengembangan Perangkat Lunak

Metode yang digunakan dalam pengembangan perangkat lunak ini menggunakan model waterfall. Menurut Imroatus dkk. dalam (Nurfitriana et al., 2021, p. 38) model *waterfall* terdiri dari lima tahapan, yaitu:

1. Analisa Kebutuhan Sistem

Pada tahap awal ini melakukan analisis terhadap kebutuhan sistem yang dibutuhkan dalam mengembangkan program yang akan dibuat dan sesuai dengan kebutuhan sistem.

2. Desain

Pada tahap kedua ini sesuai dengan model *waterfall* yang digunakan dalam pengembangan perangkat lunak adalah beberapa proses desain database maupun desain sistem informasi yang akan dibangun. Adapun *tools* yang digunakan adalah *Entity Relationship Diagram* (ERD) dan *Logical Record Structure* (LRS) yang digunakan dalam desain database yang akan dibangun dan diagram UML yang terdiri dari *Use Case* dan *Activity Diagram* yang digunakan untuk desain sistem informasi yang akan dibuat.

3. Pengkodean

Adapun tahap ketiga ini adalah tahapan pengkodean dimana desain yang telah dibuat dan dirancang pada tahap sebelumnya ditranslasikan ke dalam bentuk *website* sistem informasi.

4. Pengujian

Pada tahapan ini dilakukan pengujian yang dilakukan dengan cara memberikan gambaran hasil dengan menggunakan metode pengujian *blackbox testing* yang digunakan untuk meminimalisir terjadinya suatu kesalahan dan menilai apakah *website* yang telah dibuat tersebut sudah sesuai dengan kebutuhan *user* dan sesuai dengan rancangan sebelum diimplementasikan.

5. Implementasi

Pada tahapan ini merupakan tahap akhir setelah *website* diuji. Perancangan perangkat lunak ini direalisasikan sebagai sebuah rangkaian program atau unit program.

2.2. Teori Pendukung

A. *Entity Relationship Diagram (ERD)*

Menurut Indrajani dalam (Santoso, 2016), *Entity Relationship Diagram (ERD)* merupakan sebuah pendekatan dari *top-bottom* pada perancangan basis data yang dimulai dengan cara mengidentifikasi data - data terpenting yang disebut dengan *entitas* dan hubungan antara *entitas* tersebut digambarkan dalam suatu model.

1. Komponen - Komponen pada ERD

Menurut Sukamto dan Shalahuddin dalam (Pramono, 2018), mengatakan bahwa ERD mempunyai beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker dan Ian Palmer, Harry Ellis), dan beberapa notasi lainnya, namun yang banyak digunakan adalah notasi dari Chen.

Berikut merupakan simbol - simbol yang digunakan pada ERD (Pramono, 2018):

- a. Entitas (*entity*), merupakan data inti yang akan disimpan dalam tabel pada basis data.
- b. Atribut (*attribute*), merupakan *field* atau kolom data yang butuh untuk disimpan dalam suatu entitas.
- c. Relasi (*relation*), merupakan penghubung antara entitas yang biasanya diawali dengan kata kerja.
- d. Kardinalitas

Menurut Hutahaen dalam (Pramono, 2018), kardinalitas relasi yang terjadi antara dua himpunan entitas dapat berupa:

- 1) Satu ke satu (*one to one*)

Sebuah entitas pada A berhubungan paling banyak satu entitas pada B dan sebaliknya.

- 2) Satu ke banyak (*one to many*)

Sebuah entitas pada A berhubungan satu atau lebih entitas pada B. Namun, entitas pada B hanya dapat dihubungkan dengan paling banyak satu entitas pada A.

3) Banyak ke satu (*many to one*)

Sebuah entitas pada A berhubungan paling banyak satu entitas pada B. Namun, entitas pada B dapat berhubungan paling banyak satu atau lebih entitas pada A.

4) Banyak ke banyak (*many to many*)

Sebuah entitas pada A berhubungan paling banyak satu atau lebih entitas pada B dan sebaliknya.

2. Logical Structure Record (LRS)

Menurut Friyadie dalam (Boby Sanderson Pardede, 2019), LRS adalah permodelan *Entity Relationship* (ER) dengan atributnya sehingga dapat terlihat hubungan - hubungan dari entitasnya masing - masing.

B. Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin dalam (Nurfitriana et al., 2021), UML adalah sebuah standar bahasa yang digunakan dalam menganalisis, merancang, dan menggambarkan struktur program dalam pemrograman *object oriented*.

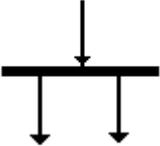
Adapun alat - alat pendukung dalam perancangan sistem berbasis UML adalah sebagai berikut:

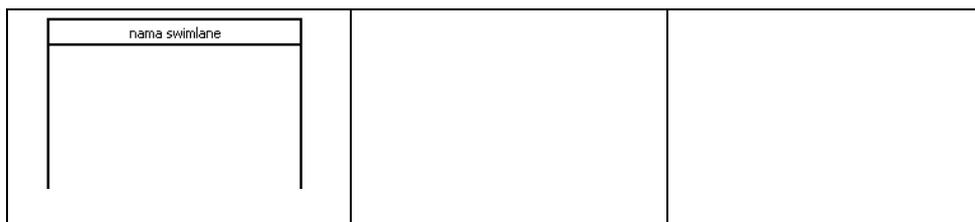
1. Diagram Aktifitas (*Activity Diagram*)

Menurut Fowler dan Martin dalam (Syafitri, 2016), *activity diagram* menggambarkan berbagai alir aktifitas dalam sistem yang tengah dirancang, diawali dengan bagaimana masing - masing alir berawal, keputusan yang mungkin dapat terjadi, dan bagaimana cara suatu aktifitas berakhir.

Tabel II.1

Simbol - Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point</i>	Merupakan awal dari sebuah aktivitas.
	<i>End Point</i>	Akhir dari sebuah aktivitas.
	<i>Fork / percabangan</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join (penggabungan) / rake</i>	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan suatu keputusan, <i>true or false</i> .
	<i>Swimline</i>	Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.



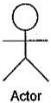
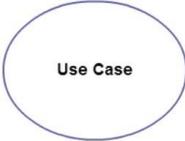
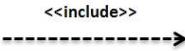
Sumber : Ade Hendini dalam (Fitri Ayu and Nia Permatasari, 2018)

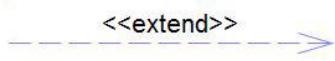
2. Use Case Diagram

Use case merupakan gambaran fungsi pada sebuah sistem dari perspektif penggunanya. *Use case* bekerja dengan cara menggambarkan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui alur bagaimana sebuah sistem menyediakan fungsinya (Santoso, 2016).

Tabel II.2

Simbol - Simbol Use Case Diagram

Simbol	Nama	Keterangan
	Aktor	Merupakan penggunaan dari sistem. Penamaan aktor menggunakan kata benda.
	<i>Use Case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> menggunakan kata kerja.
	<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.

	<p style="text-align: center;"><i>Extend</i></p>	<p>Hubungan antara <i>use case</i> dengan <i>use case</i>, <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.</p>
	<p style="text-align: center;">Asosiasi</p>	<p>Hubungan antara aktor dengan <i>use case</i>.</p>

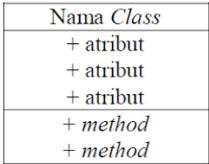
Sumber : (Fitri Ayu and Nia Permatasari, 2018)

3. *Class Diagram*

Menurut M. Shalahuddin dalam (Sitanggang et al., 2017), *class diagram* mendeskripsikan struktur dari bagian pendefinisian kelas - kelas yang dibuat untuk membangun sebuah sistem.

Tabel II.3

Simbol - Simbol *Class Diagram*

Simbol	Nama	Keterangan
	<p style="text-align: center;"><i>Class</i></p>	<p>Kelas pada struktur sistem.</p>
	<p style="text-align: center;">Antarmuka / <i>Interface</i></p>	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
	<p style="text-align: center;">Asosiasi / <i>Association</i></p>	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>.</p>
		<p>Relasi antar kelas dengan makna kelas</p>

	Asosiasi berarah / <i>Directed Association</i>	yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Kebergantungan / <i>dependency</i>	Relasi antar kelas dengan makna ketergantungan antar kelas.
	Agregasi / <i>Agregation</i>	Relasi antar kelas dengan makna semua bagian..

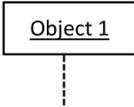
Sumber : (Fitri Ayu and Nia Permatasari, 2018)

4. *Sequence Diagram*

Menurut Munawar dalam (Rouf Ashidiq, 2019), *sequence diagram* dimanfaatkan untuk menggambarkan perilaku dalam sebuah skenario.

Tabel II.4

Simbol - Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Objek / Aktor	Sebuah objek yang berasal dari kelas. Atau dapat dinamakan dengan kelas saja. Aktor termasuk juga objek. garis putus - putus menunjukkan garis hidup suatu objek.

	Aktivasi	Menunjukkan masa hidup dari objek.
<p>Message 1</p> 	Pesan	Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditunjukkan pada bagian operasi pada diagram kelas.
<p>Message 2</p> 	<i>Return</i>	Pesan kembali dari komunikasi antar objek.

Sumber : Feri Sulianti dalam (Fitri Ayu and Nia Permatasari, 2018)

C. Pengujian Web

Terdapat dua jenis pengujian dalam proses pengujian *web*, yaitu *black-box testing* dan *white-box testing*. Namun dalam tahapan pengujian yang dilakukan penulis saat ini adalah menggunakan *black-box testing*.

Menurut Rizky dalam (Ali, 2018), *black-box testing* merupakan tipe pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya tersebut. Sehingga para penguji memandang perangkat lunak tersebut seperti “kotak hitam” yang tidak terlalu penting dilihat isinya, tapi cukup dikenai proses pengujian pada bagian luarnya.