

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Teori Dasar**

#### **2.1.1 Pengertian Website**

Situs web (*website*) adalah kumpulan halaman web yang ditautkan ke file terkait lainnya. Sebuah website memiliki halaman pertama yang dilihat ketika user mengunjungi sebuah situs web, halaman tersebut disebut *Home Page*. Ketika user ingin pergi ke halaman lain yang terdapat dalam situs web tersebut, user dapat melakukan klik *hyperlink* (Jhonsen, 2015, p. 5).

Website berisikan informasi yang terdapat di semua halaman web. Halaman web yang terdiri dari banyak halaman yang saling berhubungan biasanya disebut situs web (*website*). Dalam halaman web terdapat *hyperlink* yang merupakan hubungan antara satu halaman web dengan yang lainnya, dan *hypertext* sebagai media koneksi yang digunakan pada teks (Yuhefizar, 2009, p.2).

#### **2.1.2 Pengertian E-Learning**

Kegiatan belajar mengajar dengan memanfaatkan internet atau media jaringan komputer lain dapat memungkinkan tersampainya materi pelajaran kepada pelajar, jenis kegiatan ini bisa disebut dengan *E-Learning*, (Hartley, 2001).

*E-Learning* menggunakan berbagai media untuk menyampaikan materi pelajaran atau konten melalui situs web di internet dengan menggunakan multimedia (ragam media yang dapat menyampaikan pesan teks, grafik, audio, video, dan animasi secara terintegrasi), televisi interaktif, kelas virtual (pembelajaran yang dimediasi komputer dan sinkronisasi internet secara nyata).

### 2.1.3 Pengertian Modul

Modul adalah suatu unit perangkat lunak yang dapat berdiri sendiri dan terdiri dari *model*, *view*, *controller* serta komponen lainnya. Modul terlihat sama dengan aplikasi, namun perbedaannya yaitu modul tidak dapat dijalankan langsung melainkan sebuah modul harus berada didalam aplikasi agar dapat berjalan dengan baik. Dalam aplikasi besar, modul dapat dibagi menjadi beberapa bagian untuk dikembangkan secara terpisah.

### 2.1.4 Pengertian *Systems Development Life Cycle* (SDLC)

*Systems Development Life Cycle* (SDLC) merupakan proses untuk mendukung suatu kebutuhan bisnis dengan cara memahami sebuah sistem kemudian melakukan perancangan, pembangunan dan menyampaikan hasilnya kepada pengguna (Dennis, Wixom, & Tegarden, 2015).

Ada 6 tahapan dalam pengembangan sistem SDLC yaitu:

1. *Planning* (Perencanaan)

Tahap perencanaan dilakukan untuk menentukan persyaratan project yang akan dikerjakan kedepannya. Tahap ini dilakukan untuk mendeskripsikan maksud dan ruang lingkup pengembangan serta untuk menemukan kesalahan sistem yang akan dilengkapi melalui pengembangan sistem.

2. *Analysis* (Analisis)

Pada tahap ini mulai melakukan analisa terhadap masalah yang dihadapi, tujuan dan juga sasaran dari pengembangan sistem untuk menentukan analisa kebutuhan sistem dan untuk membuat batasan sistem.

3. *Design* (Desain)

Pada tahap ini semua hasil analisis mengenai spesifikasi sistem akan dibentuk menjadi rancangan sebuah sistem. Dalam pembuatan perancangan desain akan dilakukan analisa interaksi objek dan fungsi pada sistem, lalu akan dilakukan analisa data untuk kemudian dibuat

skema database. Setelah itu akan dilakukan perancangan user interface aplikasi yang akan dibangun.

4. *Implementation* (Implementasi)

Tahap selanjutnya adalah implementasi yaitu untuk mengimplementasikan desain rancangan yang telah dibuat dari tahapan sebelumnya seperti membuat aplikasi berdasarkan desain sistem, pembuatan database sesuai dengan skema rancangan.

5. *Testing* (Pengujian)

Tahap pengujian sangat penting dilakukan untuk mengetahui apakah sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan yang diinginkan. Jika terdapat kesalahan pada tahap pengujian, maka harus segera dilakukan perbaikan.

6. *Maintenance* (Perawatan)

Tahap pemeliharaan dilakukan untuk menjaga dan memantau sistem agar dapat tetap beroperasi dengan baik. Jika terdapat kerusakan yang ditemukan pada sistem, maka harus segera dilaporkan dan diselesaikan.

### 2.1.5 Pengertian Agile

Metode *Agile* adalah sebuah metode yang digunakan untuk pengembangan bertahap yang berfokus pada pengembangan yang cepat, perangkat lunak yang dirilis selangkah demi selangkah, mengurangi dalam pemrosesan, dan juga dalam pembuatan kode dengan kualitas yang bagus dan dalam pemrosesan berkembangnya sistem berdasarkan *requirement* dari user (Sommerville, 2011).

*Agile Software Development* merupakan seperangkat teknik pengembangan perangkat lunak yang berdasarkan pengembangan iterative, yang menciptakan persyaratan dan solusi pengembangan melalui kolaborasi teroganisir antar tim (Pressman, 2010).

### 2.1.6 Pengertian *Object Oriented Analysis and Design*

*Object Oriented Analysis and Design* adalah pendekatan berorientasi objek yang digunakan untuk mengidentifikasi dan melakukan perancangan sistem. *Object* dapat diartikan menjadi sebuah identitas, ataupun perilaku dalam sebuah sistem (Mathiassen). Dalam analisis identitas suatu *object* mewakili bagaimana pengguna membedakan dari objek lain, dan perilaku objek dijelaskan oleh peristiwa yang dilakukannya. Dalam perancangan, objek yang berada pada sistem akan dijelaskan dan bagaimana objek - objek tersebut akan bekerja sehingga dapat menjadi sebuah perilaku yang dapat diakses dan digambarkan menjadi sebuah objek yang dapat mengatur dan mempengaruhi objek lainnya.

Metode berorientasi objek atau *object oriented* adalah pola baru dalam pengembangan *software* yang berinteraksi satu sama lain sebagai kumpulan objek (Sholih, 2006).

### 2.1.7 Pengertian *Unified Modelling Language*

*Unified Modelling Language* adalah bahasa untuk menjelaskan persyaratan, pembuatan analisis dan desain serta menggambarkan arsitektur yang banyak digunakan dunia industri dalam pemrograman berorientasi objek (Rosa dan Shalahuddin, 2013). UML merupakan sarana untuk perancangan sistem berorientasi objek yang merepresentasikan penggambaran secara visual. Berikut adalah jenis-jenis diagram UML:

#### a. *Use Case Diagram*

*Use Case* merupakan suatu gambaran singkat untuk melihat apa yang bisa dijalankan oleh aktor dan untuk menunjukan aktor siapa saja yang menggunakan sistem. *Use Case Diagram* merupakan suatu penggambaran dari kelakuan sistem informasi yang akan dikembangkan, dengan menggunakan *Use Case* maka tugas yang diberikan akan lebih mudah untuk dipahami ketika menjalankan tugas tersebut (Sukanto dan Shalahuddin, 2014). Berikut adalah komponen-komponen yang terdapat pada *Use Case Diagram*:








Tabel 2. 1 Komponen *Use Case Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Mengambarkan peran pengguna yang berinteraksi dengan sistem.
2.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem untuk dijalankan oleh Aktor.
3.		<i>Association</i>	Penghubung antara aktor dengan <i>Use Case</i> .
4.		<i>Generalization</i>	Hubungan generalisasi dengan spesialisasi antara dua buah usecase dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya
5.		<i>Dependency</i>	Relasi yang menggambarkan ketergantungan antara <i>Use Case</i> satu dengan <i>Use Case</i> lainnya.
6.		<i>Include</i>	Menspesifikasikan bahwa <i>Use Case</i> sumber secara eksplisit.
7.		<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya apabila satu kondisi terpenuhi.
8.		<i>System</i>	Menspesifikasi paket yang menampilkan sistem secara terbatas.

b. *Class Diagram Diagram*

*Diagram* kelas atau *Class Diagram Diagram* mendefinisikan kelas yang akan dibuat dalam pembangunan sistem untuk menggambarkan struktur sistem (Sukamto dan Shalahuddin, 2013, p.141). Berikut adalah komponen-komponen yang terdapat pada *Class Diagram Diagram*:






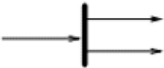
Tabel 2. 2 Komponen *Class Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class Diagram</i>	Kelas pada struktur sistem.
2.		<i>Association</i>	Relasi antar <i>Class Diagram</i> dengan arti umum.
3.		<i>Dependency</i>	Menunjukkan operasi pada suatu <i>Class Diagram</i> yang menggunakan <i>Class Diagram</i> lain.
4.		<i>Generalization</i>	Menghubungkan antar kelas dengan arti umum-khusus.
5.		<i>Aggregation</i>	Relasi antar <i>Class Diagram</i> dengan makna semua bagian.
6.		<i>One to Many</i>	Menggambarkan relasi antar table yang bersifat <i>one to many</i> .
7.		<i>One to One</i>	Menggambarkan relasi antar table yang bersifat <i>one to one</i> .

c. *Activity Diagram*

*Activity Diagram* merupakan gambaran aktivitas yang dilakukan oleh sistem bukan yang dilakukan oleh aktor. *Diagram* aktivitas atau *Activity Diagram* merupakan bentuk khusus dari *state machine* yang bertujuan memodelkan komputasi-komputasi dan aliran kerja yang terjadi dalam sistem atau perangkat lunak yang sedang dikembangkan (Nugroho, 2010). Berikut adalah komponen-komponen yang terdapat pada *Activity Diagram*:


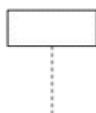

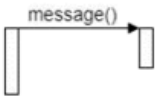
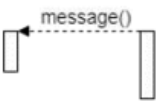

Tabel 2. 3 Komponen *Activity Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Activity</i>	Aktivitas yang dilakukan oleh sistem.
2.		<i>Decision</i>	Percabangan terjadi jika ada pilihan lebih dari satu.
3.		<i>Start Point</i>	Menggambarkan awal dari aktivitas.
4.		<i>End Point</i>	Menggambarkan akhir dari aktivitas.
5.		<i>Transition</i>	Menggambarkan aliran perpindahan control antara <i>state</i> .
6.		<i>Fork</i>	Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan.

d. *Sequence Diagram*

*Sequence Diagram* adalah penggambaran kolaborasi sejumlah objek untuk menunjukkan pesan yang akan dikirim antara objek. *Sequence Diagram* menunjukkan alur penyampaian pesan yang terjadi pada aplikasi, sebagai bentuk interaksi dengan pengguna (Pratama, 2014). Berikut adalah komponen-komponen yang terdapat pada *Sequence Diagram*:

Tabel 2. 4 Komponen *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Mengambarkan peran pengguna yang berinteraksi dengan sistem.
2.		<i>Life Line</i>	Menghubungkan objek selama pesan dikirim atau diterima dan aktivitasnya.
3.		<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi.
4.		<i>Message Entry</i>	Menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian.
5.		<i>Message to Self</i>	Menggambarkan pesan/hubungan objek itu sendiri yang menunjukkan urutan kejadian.
6.		<i>Message Return</i>	Menggambarkan hasil dari pengiriman <i>message</i> .

## 2.2 Tinjauan Studi

Tinjauan studi dilakukan penulis untuk mendukung penulisan tugas akhir, tinjauan studi yang dibuat berkaitan dengan topik yang dibahas. Berikut adalah beberapa referensi tinjauan studi:

1. Jurnal penelitian yang disusun oleh Mario Stefanus dan Johanes Fernandes Andry, di tahun 2020 dengan judul **“PENGEMBANGAN APLIKASI *E-LEARNING* BERBASIS WEB MENGGUNAKAN MODEL WATERFALL PADA SMK STRADA 2 JAKARTA”**. Jurnal ini membahas mengenai proses pembelajaran yang terjadi di sekolah masih menggunakan metode tradisional sehingga penyampaian materi kepada siswa masih sangat terbatas. Siswa pun mulai kehilangan minat untuk belajar. Untuk menangani masalah tersebut maka dikembangkanlah aplikasi *E-Learning* yang diharapkan dapat memfasilitasi kebutuhan pembelajaran menjadi lebih mudah. Aplikasi



ini dikembangkan dengan metode *System Development Life Cycle* (SDLC) *Waterfall* karena merupakan metode klasik yang sederhana.

2. Jurnal penelitian yang disusun oleh Wahyu Eko Susanto dan Yoanna Galuh Ayu Astuti, di tahun 2017 dengan judul “**PERANCANGAN E-LEARNING BERBASIS WEB PADA SMP NEGERI 3 PATUK GUNUNGGIDUL YOGYAKARTA**”. Jurnal ini membahas mengenai perkembangan teknologi informasi yang semakin pesat dibutuhkan di berbagai macam bidang, salah satunya bidang pendidikan. Sudah banyak sekolah yang menggunakan teknologi untuk menunjang proses pembelajaran. Salah satunya SMP Negeri 3 Patuk Gunungkidul yang sudah memiliki laboratorium komputer dan juga akses internet namun belum memiliki sarana pendukung seperti aplikasi *E-Learning*. Sehingga proses pembelajaran antara siswa dan guru masih terbatas. Maka dibangunlah aplikasi *E-Learning* untuk mendukung pembelajaran antara pengajar dan pelajar yang dapat dilakukan setiap waktu. Metode yang digunakan untuk perancangan aplikasi *E-Learning* adalah metode waterfall, mulai dari analisis kebutuhan, membuat desain, membuat kode program dan melakukan pengujian.
3. Jurnal penelitian yang disusun oleh Tb. Dedy Fu’ady dan Waliadi Gunawan, di tahun 2018 dengan judul “**ANALISA SISTEM PERANCANGAN PENILAIAN RAPORT SISWA BERORIENTASI OBJEK UML (UNIFIED MODELING LANGUAGE) DI SMK INFORMATIKA SUKMA MANDIRI CILEGON**”. Jurnal ini membahas mengenai pelayan pada bagian administrasi sekolah masih menggunakan cara manual dalam mengatur dokumennya, hal ini tentunya menghabiskan waktu yang cukup lama. Kegiatan administrasi di sekolah seperti penilaian raport pada siswa merupakan salah satu kegiatan rutin yang dilakukan. Oleh karena itu dibuatlah rancangan sistem penilaian raport siswa menggunakan model

UML (Unified Modeling Language) yang diharapkan dapat mengatasi keterbatasan dalam pengelolaan administrasi sekolah.

4. Jurnal penelitian yang disusun oleh Mukhlisoh Syaukati Robbi dan Yulianti, di tahun 2019 dengan judul “**PERANCANGAN APLIKASI E-LEARNING BERBASIS WEB DENGAN MODEL PROTOTYPE PADA SMPN 7 KOTA TANGERANG SELATAN**”. Jurnal ini membahas mengenai pembelajaran yang dilakukan antara siswa dan guru harus dilakukan dengan baik agar informasi yang disampaikan dapat diterima dengan baik juga. Namun jika guru yang mengajar saat itu berhalangan untuk hadir, maka proses penyampaian materi akan terhambat. Sama halnya dengan siswa yang berhalangan hadir akan tertinggal materi pembelajaran di hari itu. Maka dibuatlah *E-Learning* yang diharapkan dapat memudahkan proses belajar mengajar di sekolah. Metode pengembangan sistem yang digunakan adalah metode prototype, karena metode prototype merupakan salah satu metode siklus hidup sistem yang berdasarkan konsep model bekerja. Terdapat tiga tahap dalam metode prototype yaitu pengumpulan kebutuhan, membangun prototype dan uji coba.
5. Jurnal penelitian yang disusun oleh Julian Chandra Wibawa dan Elid Edah, di tahun 2015 dengan judul “**APLIKASI E-LEARNING DI SMP NEGERI 46 BANDUNG**”. Jurnal ini membahas mengenai masalah yang terdapat pada SMPN 46 Bandung dalam proses pembelajaran, seperti keterbatasan waktu dengan banyaknya materi yang disampaikan sehingga proses pembelajaran menjadi kurang optimal. Kemampuan siswa dalam melakukan pemanfaatan terhadap penggunaan internet juga masih dinilai kurang mengeksplorasi. Untuk menangani permasalahan ini, maka aplikasi *E-Learning* yang diharapkan mempermudah kegiatan pembelajaran seperti mendapat materi, mengirim tugas dan juga mengikuti ujian secara *online*. Metode

yang digunakan untuk mengembangkan aplikasi *E-Learning* adalah metode prototype.

6. Jurnal penelitian yang disusun oleh M. Elfin Noor, Wahyu Hardyanto dan Hari Wibawanto di tahun 2017 dengan judul “**Penggunaan *E-Learning* dalam Pembelajaran Berbasis Proyek di SMA Negeri 1 Jepara**”. Jurnal ini membahas mengenai interaksi dan komunikasi antara pelajar dengan guru harus lebih banyak dalam pembelajaran berbasis proyek. Solusi yang digunakan yaitu menggunakan *E-Learning* untuk meningkatkan interaksi pelajar dan guru. Pembelajaran berbasis proyek bertujuan untuk mengetahui hasil dari penggunaan aplikasi *E-Learning Schoology* dan Edmodo, mengkaji perbedaan dalam penggunaan aplikasi *E-Learning Schoology* dan Edmodo, mengkaji perbedaan hasil belajar, dan mengkaji hubungan antara pengguna perempuan dan laki-laki. Manfaat pembelajaran berbasis proyek yang dapat dirasakan oleh pelajar yaitu bahan ajar dan materi pembelajaran dapat diakses kapan saja dan dimana saja.

