

## BAB 2 LANDASAN TEORI

### 2.1 E-Learning

*E-Learning* merupakan sebuah sistem pembelajaran yang didasarkan pada sebuah pengajaran formal dengan sebuah bantuan sumber daya elektronik. Pada *E-Learning*, penggunaan internet dan juga sebuah *device* komputer menjadi sebuah komponen utama dari sebuah *e-learning*. *E-Learning* tentunya bersifat *online* dan para pengguna nya harus terhubung dengan internet agar dapat mengakses layanan *E-Learning*. Penggunaan teknologi internet ini yaitu untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan juga keterampilan. *E-Learning* berfokus pada penggunaan teknologi di bidang pendidikan dan pembelajaran [12]. *E-learning* dalam sebuah sekolah dapat diakses para pengguna nya menggunakan internet dan juga para pengguna mengakses nya melalui sebuah *website*, atau *e-learning* ini yaitu berbasis *website*. *E-learning* mengacu pada penggunaan teknologi komunikasi informasi yang canggih dalam proses pembelajaran dimana teknologi canggih tersebut terdiri dari media elektronik [12].

#### 2.1.1 Perbandingan aplikasi *e-learning* dengan yang lainnya

Terdapat perbandingan dari beberapa jenis aplikasi berbasis *e-learning*, yaitu :

##### A Edmodo

Edmodo merupakan sebuah situs *microblogging* yang dapat digunakan di dalam kelas, mupun rumah. Edmodo dapat membantu para guru untuk memberikan materi pembelajaran secara *online*. Edmodo memungkinkan para siswa untuk berbagi ide dengan mengunggah postingan melalui edmodo. Edmodo tidak dapat dimodifikasi oleh penggunanya, pengguna hanya dapat menggunakan fitur yang telah disediakan oleh edmodo. Hal tersebut membuat para guru harus mengikuti cara kerja dari edmodo dalam pembelajaran [13].

## B Moodle

Moodle adalah sebuah *Course Management System (CMS)* yang dapat digunakan secara gratis oleh penggunanya, dan dapat dimodifikasi oleh siapa saja dengan lisensi secara *General Public License (GNU)*. Moodle dapat disertakan *external plugins* seperti *wordpress*, dan lain sebagainya yang dapat memudahkan pengguna untuk membangun sistem sesuai dengan kebutuhan penggunanya [13].

Terdapat sebuah tabel perbandingan *e-learning* dengan yang lainnya pada tabel berikut :

Tabel 2.1. Tabel Perbandingan jenis sistem *e-learning*

No	Indikator	Edmodo	Moodle	E-Learning BNS
1	Konsep	Social Learning Network (SLC).	Learning Management System (LMS).	Learning Management System (LMS).
2	Pengembangan.	Tidak perlu melakukan instalasi apapun.	Perlu instalasi pada server sesuai dengan kebutuhan.	Tidak perlu melakukan instalasi apapun.
3	Kecepatan	Lebih cepat.	Sesuai dengan spesifikasi server yang digunakan.	Sesuai dengan spesifikasi server yang digunakan.
4	Tampilan	Tampilan tidak dapat diubah.	Tampilan dapat dimodifikasi.	Tampilan dapat dimodifikasi sesuai keinginan.
5	Informasi	Hanya dari guru.	Hanya dari guru.	Dari administrator dan guru.
6	Akses publik	Dapat diakses oleh semua orang, dan dapat digunakan oleh semua orang.	Dapat diakses oleh pengguna sistem sesuai dengan pengaturan developer.	Dapat diakses oleh pengguna sistem saja, sesuai dengan pengaturan developer.
Lanjut pada halaman berikutnya				

Tabel 2.1 Perbandingan jenis sistem *e-learning* (lanjutan)

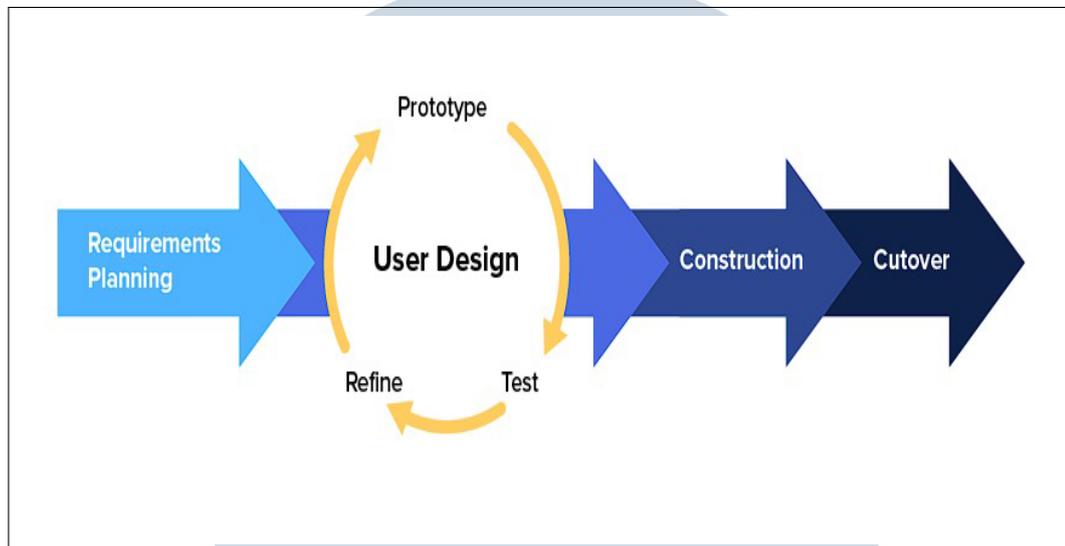
No	Indikator	Edmodo	Moodle	E-Learning BNS
7	Pengaturan	Tidak banyak pengaturan yang tersedia.	Perlu konfigurasi seperti instalasi pada server, tampilan, dan lain sebagainya.	Tidak banyak pengaturan, dan dapat disesuaikan oleh developer
8	Layanan	Quiz, materi, chat	Quiz, materi, chat	Quiz, materi, timeline, kelas, mata pelajaran, dan masih dapat ditambahkan layanan sesuai dengan pengembang

## 2.2 Rapid Application Development

*Rapid Application Development (RAD)* adalah jenis pengembangan perangkat lunak *incremental* yang didasarkan pada serangkaian siklus pengembangan iteratif pendek. Model *RAD* dianggap sebagai model yang lebih cepat dibandingkan dengan *waterfall*. Model *RAD* digunakan untuk proyek dengan struktural yang jelas dan dipahami. Model *Rapid Application Development* merupakan metodologi pengembangan yang cepat dan berulang di mana umpan balik pemilik produk merupakan elemen yang diperlukan, sebelum merilis versi final. Pengembangan menggunakan *Rapid Application Development* memerlukan waktu 2 - 3 bulan. Pada setiap iterasi, pengembang dapat memvalidasi persyaratan yang telah diselesaikan dan mengerjakan *task* yang disarankan [10].

*Rapid Application Development* digunakan sebagai metode dalam penelitian ini untuk mengembangkan aplikasi dengan cepat. *Rapid Application Development (RAD)* merupakan sebuah pendekatan berorientasi objek untuk pengembangan sistem yang mencakup metode pengembangan untuk perangkat lunak [14]. Metode *rapid application development* digunakan dalam siklus pengembangan aplikasi yang cepat dan juga memberikan kualitas perangkat lunak yang baik dibandingkan

dengan pendekatan metode tradisional [9]. Terdapat model dari metode *Rapid Application Development* yang diuraikan pada bagan dibawah ini [15] :



Gambar 2.1. *Rapid Application Development (RAD) Model*

Terdapat tahapan dari model pengembangan *rapid application development* [16], yaitu :

1. Perencanaan Kebutuhan

Tahap perencanaan kebutuhan ini dilakukan untuk menentukan menganalisis kebutuhan masalah untuk pembuatan sistem yang ingin dibuat, dan juga menentukan syarat-syarat yang diperlukan untuk pembuatan sistem tersebut.

2. *User Design*

Tahap perancangan ini dilakukan perancangan seperti pembuatan *mockup*, *flowchart*, dan lain sebagainya sebagai alat bantu untuk melakukan proses perancangan, desain, dan perbaikan jika terdapat ketidaksesuaian dari sistem yang telah dibuat.

3. Konstruksi

Pada tahap konstruksi ini dilakukan pengkodean aplikasi atau program dari rancangan yang telah dibuat atau didefinisikan. Pengkodean dilakukan dengan jangka waktu tertentu sesuai dengan analisis perancangan yang telah dibuat. Pada tahap ini juga dilakukan pengujian terhadap sistem yang dibangun.

#### 4. *Cutover*

Pada tahap *cutover* ini yaitu sistem lama di ubah dengan sistem yang baru, termasuk pengujian, konversi data, dan lain-lain terkait sistem yang dibuat. Pada tahap ini, dilakukan pengujian yang meliputi aspek *usability* untuk memastikan perancangan berjalan dengan baik.

### 2.3 User Acceptance Testing

*User Acceptance Testing* digunakan sebagai metode pengujian dalam pembuatan aplikasi *e-learning* berbasis *website*. *User Acceptance Testing (UAT)* adalah sebuah pengujian yang melibatkan pengguna akhir atau *end user* untuk dapat berinteraksi langsung menggunakan sistem dan juga memverifikasi terkait fungsionalitas dari aplikasi tersebut [17].

Terdapat beberapa manfaat dan tujuan dari penggunaan *User Acceptance Testing*, yaitu :

#### A Manfaat

1. Untuk meningkatkan kepercayaan klien mengenai persyaratan perangkat lunak yang sudah terpenuhi.
2. Mendapatkan sistem yang sesuai dengan spesifikasi fungsional sistem.
3. Kepuasan klien meningkat, karena mereka lebih yakin bahwa sistem sudah memenuhi persyaratan.
4. Mendapatkan sistem yang sesuai dengan persyaratan dan fungsional sistem.

#### B Tujuan

1. Menguji kesesuaian sistem berdasarkan persyaratan dan fungsional sistem.
2. Memberikan keyakinan terhadap klien bahwa sistem telah memenuhi persyaratan dan berjalan dengan baik
3. Melengkapi persyaratan tambahan yang telah disetujui [13]

## 2.4 Black Box Testing

*Black Box Testing* merupakan sebuah metode pengujian fungsional dengan melakukan pengujian terhadap *input* dan *output* dari aplikasi dan memastikan bahwa aplikasi sudah sesuai dengan *requirements* yang sudah diterapkan[18]. Metode *Black Box Testing* ini dilakukan pada setiap proyek pengembangan aplikasi untuk dapat memastikan bahwa fungsional sistem sudah berjalan dengan baik secara keseluruhan.

## 2.5 Fisher-Yates Algorithm

*Fisher-Yates algorithm* diambil dari nama Ronal Fisher dan Frank Yates dan dikenal dengan nama *Knuth Suffle* yang diadopsi dari nama Donald Knuth. *Fisher-Yates algorithm* merupakan sebuah algoritma yang menghasilkan sebuah permutasi acak dari suatu set data yang terbatas, atau dapat diartikan yaitu untuk mengacak suatu himpunan tersebut. Hasil dari *Fisher-Yates algorithm* tidak akan berat sebelah sehingga permutasi tetap akan memiliki kemungkinan yang sama jika diimplementasikan dengan benar[19]. Penggunaan *Fisher-Yates algorithm* dalam sistem ini yaitu untuk melakukan pengacakan soal kuis agar tidak menampilkan urutan soal yang sama pada setiap siswa.

Penerapan *Fisher-Yates Algorithm* dilakukan terhadap sistem yang dibangun untuk mengacak urutan soal pada suatu set data terbatas. Terdapat tahapan atau langkah-langkah yang digunakan pada implementasi *Fisher-Yates Algorithm* sebagai berikut:

1. Tuliskan soal dari soal nomer 1 sampai dengan soal nomer ke N untuk kuis siswa.
2. Pilih sebuah soal acak K diantara 1 sampai dengan jumlah soal kuis yang belum dicoret.
3. Dihitung dari bawah, coret soal K yang belum dicoret, dan tuliskan soal tersebut di lain tempat.
4. Ulangi langkah 2 dan langkah 3 sampai semua soal kuis sudah tercoret.
5. Urutan soal kuis yang dituliskan pada langkah 3 merupakan permutasi acak dari soal kuis awal.

## 2.6 USE Questionnaire

*Usefulness, Satisfaction, and Ease of Use* atau *USE Questionnaire* merupakan sebuah paket kuesioner yang diusulkan oleh Lund untuk dapat dipakai sebagai pengukuran *usability* dan mencakup tiga standar pengukuran berdasarkan ISO 9241, yaitu efisiensi, efektivitas, dan kepuasan[20]. Penggunaan *USE Questionnaire* mempunyai 30 pertanyaan yang disajikan dan dikelompokkan ke dalam empat dimensi, yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction*[21]. Terdapat daftar pertanyaan yang disesuaikan dan telah dikelompokkan dalam bahasa Indonesia, yaitu[22] :

### 2.6.1 Usefulness

1. Sistem ini membantu saya menjadi lebih efektif
2. Sistem ini membantu saya menjadi lebih produktif
3. Sistem ini bermanfaat
4. Sistem ini membantu saya terhadap aktivitas yang saya lakukan
5. Sistem ini membuat hal-hal yang ingin saya capai lebih mudah untuk dilakukan
6. Sistem ini menghemat waktu saya ketika menggunakannya
7. Sistem ini sesuai dengan kebutuhan saya
8. Sistem ini bekerja sesuai apa yang saya harapkan

### 2.6.2 Ease of Use

1. Sistem ini mudah digunakan
2. Sistem ini praktis untuk digunakan
3. Sistem ini mudah dipahami oleh pengguna
4. Langkah-langkah pengoperasian sistem ini praktis
5. Sistem ini dapat disesuaikan dengan kebutuhan

6. Saya merasa tidak kesulitan menggunakan sistem ini
7. Saya dapat menggunakannya tanpa instruksi tertulis
8. Saya tidak melihat adanya ketidak konsistenan selama menggunakan sistem ini
9. Pengguna yang jarang maupun rutin menggunakan akan menyukai sistem ini
10. Saya dapat kembali dari kesalahan dengan cepat dan mudah
11. Saya dapat menggunakan sistem ini dengan berhasil setiap kali saya menggunakannya

### **2.6.3 Ease of Learning**

1. Saya belajar menggunakan sistem ini dengan cepat
2. Saya mudah mengingat bagaimana cara menggunakan sistem ini
3. Sistem ini mudah untuk dipelajari cara penggunaannya
4. Saya cepat menjadi terampil dengan sistem ini

### **2.6.4 Satisfaction**

1. Saya puas dengan sistem ini
2. Saya akan merekomendasikan sistem ini kepada rekan
3. Sistem ini menyenangkan untuk digunakan
4. Sistem ini bekerja seperti yang saya inginkan
5. Sistem ini sangat bagus
6. Saya merasa harus memiliki / menggunakan sistem ini
7. Sistem ini nyaman untuk digunakan

## 2.7 Skala Likert

*Skala likert* adalah skala yang digunakan untuk mengukur persepsi seseorang, pendapat, dan juga sikap [23]. Penggunaan *skala likert* yaitu menggunakan variabel yang menjadi sebuah indikator pengukuran terhadap pertanyaan, maupun pernyataan.

Tabel 2.2. Tabel Skala Likert

No	Kriteria	Skor
1	Sangat Setuju (SS)	5
2	Setuju (S)	4
3	Ragu-Ragu (R)	3
4	Tidak Setuju (TS)	2
5	Sangat Tidak Setuju (STS)	1

### 2.7.1 Mencari Total Skor

Mencari total skor yaitu dengan cara mengalikan setiap skor dari setiap kategori dengan total responden yang memilih kategori tersebut. Perhitungan mencari total skor dapat dilihat pada rumus berikut :

$$TotalSkor = (1 * STS) + (2 * TS) + (3 * KS) + (4 * S) + (5 * SS) \quad (2.1)$$

Keterangan:

- STS = jumlah sangat tidak setuju
- TS = jumlah tidak setuju
- KS = jumlah kurang setuju
- S = jumlah setuju
- SS = jumlah sangat setuju

### 2.7.2 Mencari Skor Maksimum

Skor maksimum didapat dari hasil mengalikan jumlah responden, skor maksimum *likert*, dan juga jumlah pertanyaan yang diajukan kepada responden.

Mencari skor maksimum dapat dilakukan dengan perhitungan berikut :

$$SkorMaks = JumlahResponden * SkorMaksLikert * JumlahPertanyaan \quad (2.2)$$

### 2.7.3 Mencari Nilai Indeks (%)

Nilai indeks diperoleh dari hasil bagi total skor dan juga skor maksimum, kemudian dikalikan dengan 100. Pencarian nilai indeks dapat dilakukan dengan perhitungan berikut :

$$Indeks = \frac{TotalSkor}{SkorMaks} * 100\% \quad (2.3)$$

### 2.7.4 Menentukan Interval Penilaian

Penentuan interval penilaian ditentukan seperti yang dijabarkan pada Tabel 2.3

Tabel 2.3. Interval Penilaian

<b>Indeks</b>	<b>Kategori</b>
0% - 19,99%	Sangat Buruk
20% - 39,99%	Buruk
40% - 59,99%	Kurang Baik
60% - 79,99%	Baik
80% - 100%	Sangat Baik

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A