

BAB II TINJAUAN PUSTAKA

2.1 E-learning Space

E-learning Space atau ruang pengajaran cerdas menurut Badami *et al.*, (2018) mencakup ruang pengajaran fisik dan virtual dalam pengaturan institusi. Konsep *Smart Teaching Spaces* mendefinisikan furnitur penting dan spesifikasi teknis ruang kelas fisik yang ideal dan fitur paling penting dari platform pembelajaran virtual. Juga fasilitas dan fungsinya untuk ruang kuliah dan ruang pengajaran kolaboratif akan ditentukan dalam paket ini. Ruang Pengajaran Cerdas secara keseluruhan membawa fleksibilitas untuk belajar dan menjadikan pembelajaran itu ada di mana-mana.

E-learning Space merupakan ruang belajar elektronik dengan menyediakan wadah atau tempat belajar menggunakan sebuah aplikasi yang diterapkan dengan pemanfaatan teknologi. ruang belajar mengadopsi kelas pintar, juga pembelajaran virtual menggunakan sistem secara *online* dengan media internet

2.2 Mobile

Menurut Gunawan *et al.*, (2017), aplikasi *mobile* yaitu program siap pakai yang direkap untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ketempat yang lain.

Maka aplikasi *mobile* dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ketempat yang lain serta mempunyai ukuran yang kecil.

2.2.1 Mobile Web

Menurut Gunawan and Saputro (2018), *mobile web* adalah Halaman HTML berbasis *browser* yang diakses menggunakan perangkat *portable* (*smartphone* atau *tablet*) melalui jaringan seluler seperti 3G, 4G maupun Wifi. *Mobile*

Web dirancang untuk menampilkan data seperti teks, gambar, dan video dari sebuah website kedalam sebuah tampilan yang lebih kecil yakni perangkat *mobile*.

MobileWeb bertujuan membuat *web* dapat diakses dari sebuah perangkat *mobile* secara sederhana seperti mengakses *web* dari sebuah komputer desktop. Dalam membuat sebuah *mobile web* membutuhkan implementasi untuk perbaikan dari segi *interoperability*, *usability* dan *accessibility* pada sebuah *mobile web*. *Mobile Web* umumnya berukuran ringan disetiap halamannya yang ditulis dengan *Extensible Hypertext MarkupLanguage* (XHTML) atau *Wireless Markup Language* (WML) untuk mengirimkan konten ke perangkat *mobile*. Selain itu beberapa teknik seperti dengan menggunakan *Adobe Flash Lite* atau *Sun J2ME* yang memungkinkan untuk membuat perangkat *mobile* yang lebih bervariasi.

2.2.2 Web Based

Menurut Urbieta *et al.* (2019), *web Based* adalah Aplikasi yang dibuat berbasis web yang membutuhkan web server dan browser untuk menjalankannya.

Dengan membuat sistem berbasis web based ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin diupdate, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

2.2.3 Web View

Menurut Srijaya *et al.*, (2015), *web view* adalah sebuah *class* pada *Android* yang berfungsi sebagai semacam *sendbox* untuk menampilkan dan menjalankan

aplikasi *mobile* yang berbasis *web*, seperti HTML5, JQuery *Mobile*, dan sebagainya.

Cara mudahnya *WebView* itu seperti *Android browser*, tetapi tidak mempunyai *address bar* tempat memasukkan alamat URL. Karena URL yang mau kita buka sudah didefinisikan di aplikasi dan tidak bisa kita ganti.

2.2.4 CodeIgniter

Menurut Raharjo (2018), CodeIgniter adalah *Framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. CodeIgniter memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat mengembangkan dalam perangkat web, dekstop maupun mobile.

Sehingga *framework* tersebut sangat mudah diterapkan dan dikembangkan dengan kelebihan memiliki banyak referensi yang dapat digunakan sebagai acuan pengembangan sistem.

2.2.5 MySql

Menurut Sabar *et al.*, (2019), MySQL adalah basis data yang bersifat *open source* sehingga banyak di gunakan untuk media. Walaupun gratis, MySQL tetap berkualitas dan sudah cukup memberikan performance yang memadai. Penggunaan PHP MyAdmin lebih mudah digunakan karena menggunakan *interface* yang lebih mudah dipahami.

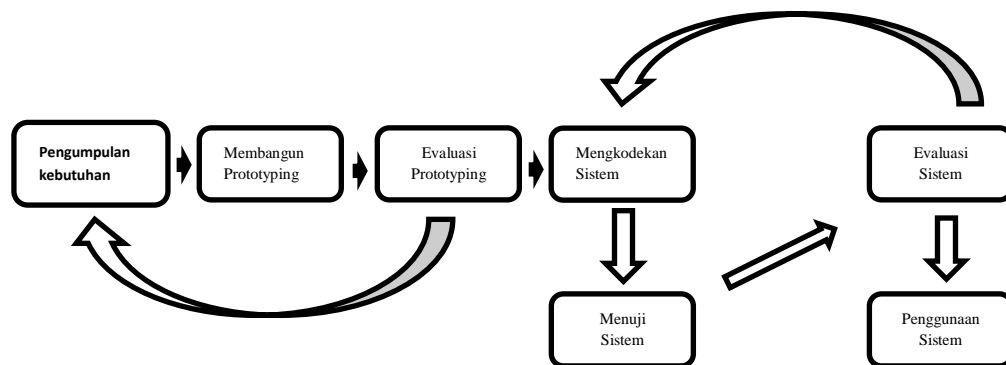
MySQL adalah nama sebuah *database server* yang menangani akses database yang selaludalam bentuk pernyataan SQL (*Structured Query Language*) yaitu suatu bahasa yang digunakan untuk mengakses *database* relasional.

2.3 Metode Pengembang Sistem

Metode pengembang sistem merupakan metode yang digunakan sebagai alur proses dalam pengembangan, sehingga penelitian dapat di kembangkan sesuai tahapan dari metode pengembang sistem.

2.3.1 Prototype

Metode Prototype atau sering disebut juga dengan prototyping merupakan sebuah metode pengembangan sistem yang didasarkan pada konsep working model. Penelitian lain pun mengatakan prototype didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah prototype disebut prototyping (Yusniar Nur Syarif Sidiq, Rd. Nuraini Siti Fathonah, S.S., M.Hum. Noviana Riza, S.Si., M.T. 2020).



Gambar 1.1 Metode *Prototype*

Dalam setiap metode tentunya memiliki kelebihan dan kekurangannya termasuk metode prototype. Adapun kelebihan dan kekurangan yang dimiliki metode prototype dapat dilihat pada Tabel 2.1.

Tabel 1.1 kelebihan dan kekurangan metode prototype

Kelebihan	Kekurangan
Pengguna (User) berperan aktif dalam pengembangan sistem.	Kualitas aplikasi belum teruji dan belum mencantumkan pemeliharaan jangka panjang.
Waktu yang digunakan lebih efisien.	Algoritma dan bahasa yang digunakan sederhana.
Adanya komunikasi antara user dengan pengembang.	Teknik rancangan tidak baik dilihat dari hubungan pelanggan dengan computer.
Pengembang dapat bekerja lebih baik.	

Tahap-tahap dalam pembuatan prototyping

1. Pengumpulan Kebutuhan

Klien (Pelanggan) dan pengembang bersama-sama mendefinisikan format software, mengidentifikasi kebutuhan dan garis besar sistem yang dibuat.

2. Membangun Prototype / Prototyping

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (contoh membuat input dan format output).

3. Evaluasi Prototyping

Tahap ini dilakukan oleh klien(pelanggan), apakah prototyping yang dibangun, sudah sesuai dengan keinginan dan kebutuhan pelanggan atau belum. Jika tidak sesuai, prototyping akan direvisi dengan mengulangi langkah-langkah sebelumnya. Tapi jika sudah sesuai, maka langkah selanjutnya akan dilaksanakan dan diambil..

4. Mengkodekan Sistem

Di tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji Sistem

Setelah sistem sudah menjadi suatu software yang siap pakai, maka software harus di tes dahulu sebelum digunakan. Hal ini bertujuan untuk meminimalisir kesalahan software tersebut. Pengujian dilakukan dengan Black Box, White box, Pengujian arsitektur, Basis path dan lain-lain.

6. Evaluasi Sistem

Di tahap ini klien(Pelanggan) mengevaluasi sistem yang sudah dibuat sudah sesuai yang diinginkan. Jika tidak, maka pengembang akan mengulangi langkah ke 4 dan 5. Tapi jika iya, maka langkah ke 7 akan dilakukan.

7. Menggunakan Sistem

Software yang telah diuji dan diterima klien(pelanggan) siap untuk digunakan.

2.4 Alat Pengembang Sistem

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram (Agarina and Karim,

2019). Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*.


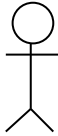
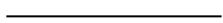
2.4.1 *Unified Modelling Language* (UML)

Menurut Rosa dan Salahuddin (2019), *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language*.

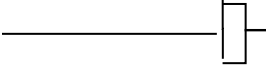
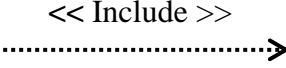
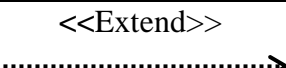
1. *Use Case Diagram*

Menurut Rosa dan Salahuddin (2019), *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1 :

Tabel 1.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Usecase</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi(<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)


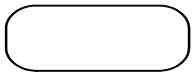
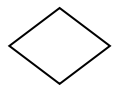

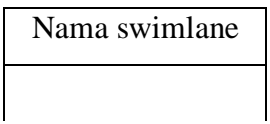
No	Simbol	Deskripsi
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

Sumber: (Rosa dan Salahuddin, 2019)


2. *Activity Diagram*

Menurut Rosa dan Salahuddin (2019), *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.2:

Tabel 1.3 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.

Tabel 2.2 Simbol *Activity Diagram* (Lanjutan)

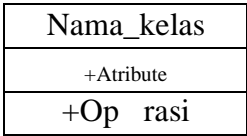


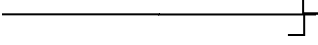
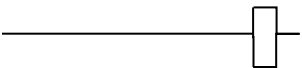
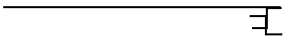
No.	Simbol	Keterangan
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber: (Rosa dan Salahuddin, 2019)

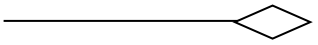
3. *Class Diagram*

Menurut Rosa dan Salahuddin (2019), *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.3:

Tabel 1.4 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/Interface</p>  <p>Nama_Interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / Association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / Directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / dependency</p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.

Tabel 2.3 Simbol *Class Diagram* (Lanjutan)

No.	Simbol	Deskripsi
7.	Agregasi / aggregation 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa dan Salahuddin, 2019)

2.5 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya.

2.5.1 *Black Box Testing*

(Uus Rusmawan, 2019) Menurut Rizky dalam penelitian Nina Rahayu (2014:42), black box testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar.

Beberapa keuntungan yang diperoleh dan jenis testing ini antara lain:

1. Anggota tim tester tidak harus dan seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dan perangkat lunak atau pun seringkali diternukan oleh komponen tester yang berasal dan pengguna.
3. Hasil dad black box testing dapat memperjelaskan kontradiksi atau pun kerancuan yang mungkin ditimbulkan dad eksekusi perangkat lunak.
4. Proses testing dapat dilakukan lebih cepat dibandingkan white box testing.

2.5.2 *Skala Likert*

Menurut Sugiyono (2018) Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian”.

Skala *likert* maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator variabel tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan.

Jawaban setiap item instrumen yang menggunakan Skala *Likert* mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata-kata seperti Sangat Setuju, Setuju, Ragu-ragu, Tidak setuju, Sangat Tidak Setuju.

Keperluan analisis kuantitatif, maka jawaban itu dapat diberi skor, misalnya:

- | | | |
|------------------------|-------------|---|
| a. Sangat setuju | diberi skor | 5 |
| b. Setuju | diberi skor | 4 |
| c. Ragu-ragu/netral | diberi skor | 3 |
| d. Tidak setuju | diberi skor | 2 |
| e. Sangat tidak setuju | diberi skor | 1 |

Instrumen penelitian yang menggunakan *Skala Likert* dapat dibuat dalam bentuk *checklist* ataupun pilihan ganda. Pengujian menggunakan 5 kategori jawaban dengan bobot yang berbeda untuk setiap jawabannya seperti pada tabel berikut.

Tabel 1.5 Bobot Jawaban *Usability*

No.	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.	Tampilan pada sistem mudah digunakan	5	4	3	2	1

Dengan cara pengumpulan data angket, maka instrument tersebut misalnya diberikan kepada 100 orang pengguna yang diambil secara *random*, dari 100 orang tersebut telah dianalisis, misalnya:

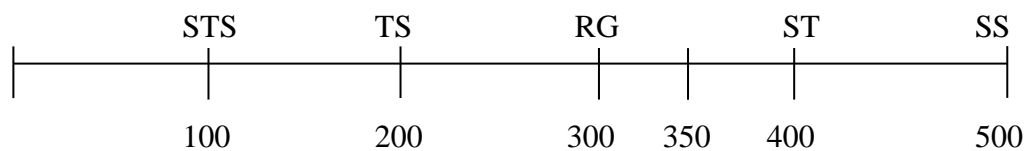
- | | |
|-------------------|-----|
| 25 orang menjawab | SS |
| 40 orang menjawab | ST |
| 5 orang menjawab | RG |
| 20 orang menjawab | TS |
| 10 orang menjawab | STS |

Berdasarkan analisis data tersebut maka dapat ditetapkan dari nilai skor sebagai berikut:

- | | |
|-------------------------------------|-------------|
| Jumlah skor untuk 25 orang menjawab | = 25x5 =125 |
| Jumlah skor untuk 40 orang menjawab | = 40x4 =160 |

$$\begin{array}{rcl}
 \text{Jumlah skor untuk 5 orang menjawab} & = & 5 \times 3 = 15 \\
 \text{Jumlah skor untuk 20 orang menjawab} & = & 20 \times 2 = 20 \\
 \text{Jumlah skor untuk 10 orang menjawab} & = & 10 \times 1 = 10 \\
 \hline
 & & = 350
 \end{array}$$

Jumlah skor ideal (Kriterium) untuk seluruh item = $5 \times 100 = 500$ (Seandainya semua menjawab SS). Jumlah skor yang diperoleh dari angket = 350, jadi didapat hasil sebesar $=(350:500) \times 100\% = 70\%$. Secara kontinum dapat digambarkan seperti berikut :



Gambar 1.2 Skala Hasil Pengukuran

2.6 Penelitian Terdahulu

Penelitian terdahulu dapat dilihat pada Table 2.6.

Tabel 1.6 Penelitian Terdahulu

No	Nama	Judul	Terbit/Tahun	Keterangan
1	Hamdan Husein Batubara	Pengembangan Media Pembelajaran Matematika berbasis Android untuk Siswa SD/MI	VOL. 3, NO. 1, OKTOBER, 2017	Penelitian ini bertujuan untuk menghasilkan media pembelajaran matematika berbasis android untuk siswa SD/MI dengan melibatkan pakar pada proses pengembangan dan penilaian produknya.
2	Joko Kuswanto , Ferri Radiansah	Media Pembelajaran Berbasis	Jurnal Media Infotama Vol. 14 No. 1,	menghasilkan produk berupa media

		Android Pada Mata Pelajaran Sistem Operasi Jaringan Kelas XI	Februari 2018	pembelajaran yang sesuai dengan kebutuhan siswa kelas XI serta sesuai dengan kurikulum dan silabus yang berlaku, sehingga dapat bermanfaat secara teoretis maupun praktis.
3	Dania Ayu Wulandari, Agus Murnomo, Hari Wibawanto, Agus Suryanto	Pengembangan Mobile Learning Berbasis Android Pada Mata Pelajaran Rekayasa Perangkat Lunak Di Smk Sultan Trenggono Kota Semarang	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK) Vol. 6, No. 5, Oktober 2019, hlm. 577-584	Berdasarkan uji coba pada pembelajaran di kelas, aplikasi mobile learning mendapat penilaian terhadap kemudahan dan kemanfaatan penggunaan aplikasi