

BAB II **TINJAUAN PUSTAKA**

2.1 Sistem Informasi

Definisi menurut Nas (2018) Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.2 Penilaian Kinerja Guru

Berdasarkan UU No. 14 Tahun 2005 tentang guru dan dosen, guru adalah pendidik profesional dengan tugas utama mendidik, mengajar, membimbing, mengarahkan, melatih, menilai, dan mengevaluasi peserta didik pada pendidikan anak usia dini jalur pendidikan formal, pendidikan dasar, dan pendidikan menengah. Dalam Pasal 4 disebutkan bahwa kedudukan guru sebagai tenaga profesional sebagaimana dimaksud dalam Pasal 2 ayat (1) berfungsi untuk meningkatkan martabat dan peran guru sebagai agen pembelajaran berfungsi untuk meningkatkan mutu pendidikan nasional.

Menurut Kementerian Pendidikan dan Kebudayaan (Kemdikbud), Penilaian Kinerja Guru (PKG) merupakan tindak lanjut dari uji kompetensi guru (UKG) dalam rangka menghasilkan potret kompetensi guru. PKG dilakukan untuk menunjang mutu kompetensi guru dan sebagai upaya pembinaan dan pengembangan profesi guru. Pada akhirnya, PKG dapat meningkatkan proses pembelajaran yang berkualitas di sekolah. Penilaian kinerja guru dapat diklasifikasikan menjadi beberapa bagian seperti Komponen Penilaian PKG Matapelajaran, Bimbingan Konseling, TIK Teknologi Informasi dan Komunikasi, Tambahan Kepala Perpustakaan, Tambahan Wakil Kepala Sekolah, Kepala

Sekolah dan Guru produktif. Berdasarkan beberapa bagian dalam penilaian guru, berikut komponen yang digunakan :

- a. Mengenal Karakteristik Peserta Didik
- b. Menguasai teori belajar dan prinsip-prinsip pembelajaran yang mendidik
- c. Pengembangan kurikulum
- d. Kegiatan pembelajaran yang mendidik
- e. Memahami dan mengembangkan potensi
- f. Komunikasi dengan peserta didik
- g. Penilaian dan evaluasi
- h. Bertindak sesuai dengan norma agama, hukum dan sosial
- i. Menunjukkan pribadi yang dewasa dan teladan
- j. Etos kerja, tanggungjawab yang tinggi, dan rasa bangga menjadi guru
- k. Bersikap inklusif, bertindak objektif, serta tidak diskriminatif
- l. Komunikasi dengan sesama guru, tenaga pendidik, dan orang tua peserta didik
- m. Penguasaan materi struktur konsep dan pola pikir
- n. Penguasaan materi struktur konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu
- o. Komponen penilaian pkg guru produktif tambahan ketua program keahlian.

2.2.1 Indikator Penilaian Kinerja Guru

Indikator penilaian kinerja guru dilakukan untuk acuan terhadap indikator yang digunakan seperti berikut :

Tabel 2.1 Indikator Penilaian Kinerja Guru

No.	Indikator
1	Menguasai karakteristik peserta didik
2	Menguasai teori belajar dan prinsip-prinsip pembelajaran yang mendidik
3	Pengembangan kurikulum
4	Kegiatan pembelajaran yang mendidik
5	Pengembangan potensi peserta didik
6	Komunikasi dengan peserta didik
7	Penilaian dan evaluasi

8	Bertindak sesuai dengan norma agama, hukum, sosial dan kebudayaan nasional
9	Menunjukkan pribadi yang dewasa dan teladan
10	Etos kerja, tanggung jawab yang tinggi, rasa bangga menjadi guru
11	Bersikap inklusif, bertindak obyektif, serta tidak diskriminatif
12	Komunikasi dengan sesama guru, tenaga kependidikan, orang tua, peserta didik, dan masyarakat
13	Penguasaan materi, struktur, konsep dan pola pikir keilmuan yang mendukung mata pelajaran yang diampu
14	Mengembangkan keprofesionalan melalui tindakan yang reflektif
15	Menerapkan kreatifitas dan inovasi kepada peserta didik

2.2.2 Skor Penilaian Kinerja Guru

Skor penilaian kinerja guru digunakan sebagai acuan terhadap penilaian yang dilakukan oleh penilai seperti berikut:

Tabel 2.2 Skor Penilaian Kinerja Guru

Tidak ada bukti (Tidak terpenuhi)	Terpenuhi sebagian	Seluruhnya terpenuhi
0	1	2

Berdasarkan rentang nilai untuk memberikan skor dilakukan dengan cara memberikan kuisioner kepada guru dan akan dilakukan penilaian terhadap syarat yang dipenuhi oleh guru, setelah semua pertanyaan diberikan skor kemudian dilakukan perhitungan seperti berikut:

1. Persentase = $(\text{total skor} / \text{skor maksimal}) \times 100\%$
2. Nilai untuk kompetensi
($0\% < X \leq 25 = 1$; $25\% < x \leq 50\% = 2$; $50 < x \leq 75\% = 3$; $75\% < x \leq 100\% = 4$)

Skor akhir yang diperoleh dengan menghitung nilai skor total dibagi skor maksimal dan dikalikan 100 % akan dicocokkan dengan nilai kompetensi seperti diatas sesuai rentang nilai skor persentase akhirnya.

2.3 Key Performance Indicator (KPI)

Key Performance Indicator merupakan serangkaian indikator kunci yang bersifat terukur dan memberikan informasi sejauh mana sasaran strategi yang dibebankan kepada suatu organisasi sudah berhasil dicapai. Unsur-unsur yang terdapat pada KPI terdiri atas tujuan strategis, indikator kunci yang relevan dengan sasaran strategis tersebut, sasaran yang menjadi tolok ukur, dan kerangka waktu atau periode berlakunya KPI tersebut (Soemohadiwidjojo, 2015).

Didalam organisasi pendidikan misi pertama adalah dapat melakukan sertifikasi yang diharapkan dapat memperoleh hasil guru yang berkompeten. Terdapat tiga pokok penting dalam evaluasi ini yaitu tes, pengukuran dan penilaian yang menjadi acuan dalam pembuatan *key performance indicator*.

2.4 Penilaian Menggunakan Skala Likert

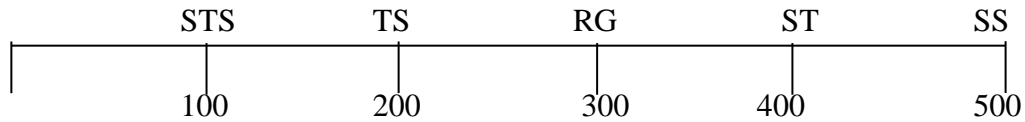
Menurut Sugiyono (2018), *skala likert testing* adalah Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel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

Dengan *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. Pengujian menggunakan 5 kategori jawaban dengan bobot yang berbeda untuk setiap jawabannya seperti pada Tabel 2.3.

Tabel 2.3 Bobot Jawaban *Usability*

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

Secara hasil kesimpulan dari hasil kuisioner yang diperoleh dapat menggunakan pengukuran dari hasil maksimal nilai tertinggi, berikut merupakan gambar aspek *usability* pada Gambar 2.5.



Gambar 2.1 Hasil Pengukuran Aspek *Usability*

Kemudian hasil perhitungan yang didapatkan dari angket, selanjutnya dibandingkan dengan rentang kriteria interpretasi skor untuk menyatakan hasil yang didapatkan dengan rentang pada Tabel 2.4.

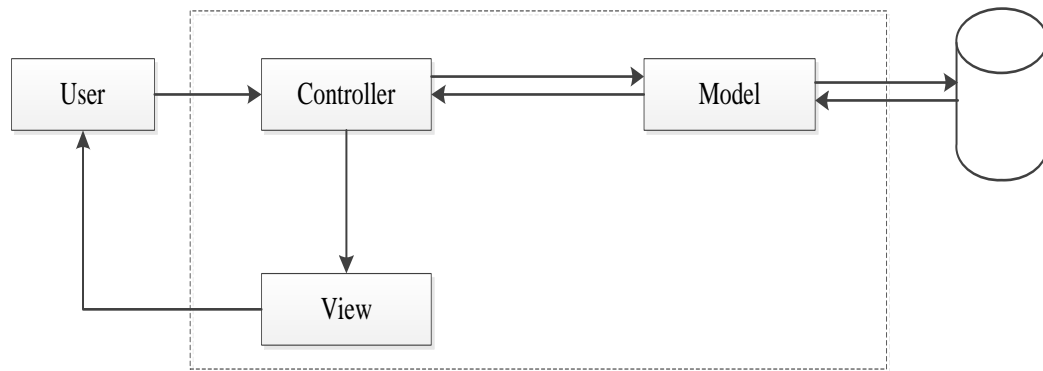
Tabel 2.4 Rentang *Criteria Interpretasi*

No	Rentang Kriteria	Kriteria
1	0% - 20%	Sangat Tidak Baik
2	21% - 40%	Tidak Baik
3	41% - 60%	Kurang Baik
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

(Sumber: Sugiyono, 2018)

2.5 CodeIgniter

CodeIgniter menurut Raharjo (2018) merupakan *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*. *CodeIgniter* memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.



Gambar 2.2 Arsitektur *Model-View-Controller* (MVC)

Berdasarkan arsitektur tersebut diketahui konsep *Model-View-Controller* yaitu:

1. *Model*

Model merupakan suatu fungsi yang digunakan mengelola database pada sistem untuk menangani struktur data dari *controller*.

2. *View*

View merupakan bagian untuk mengelola tampilan dari *website* atau dapat disebut sebagai *user interfac* yang diatur bagian *controller*.

3. *Controller*

Controller merupakan kunci dalam konsep MVC dengan fungsi untuk menghubungkan *model* dengan *view*.

2.4.1 *Web Based*

Web Based menurut Urbietta *et al.* (2019) 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 Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).

2.4.2 *Perl Hypertext Preprocessor (PHP)*

Menurut Subagja (2018) *Perl Hypertext Preprocessor* adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis.

Jadi *Perl Hypertext Preprocessor* merupakan bahasa pemrograman *web server-side* yang bersifat *open source* atau mudah dikembangkan. *Perl Hypertext Preprocessor* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*.

2.4.3 *MySql*

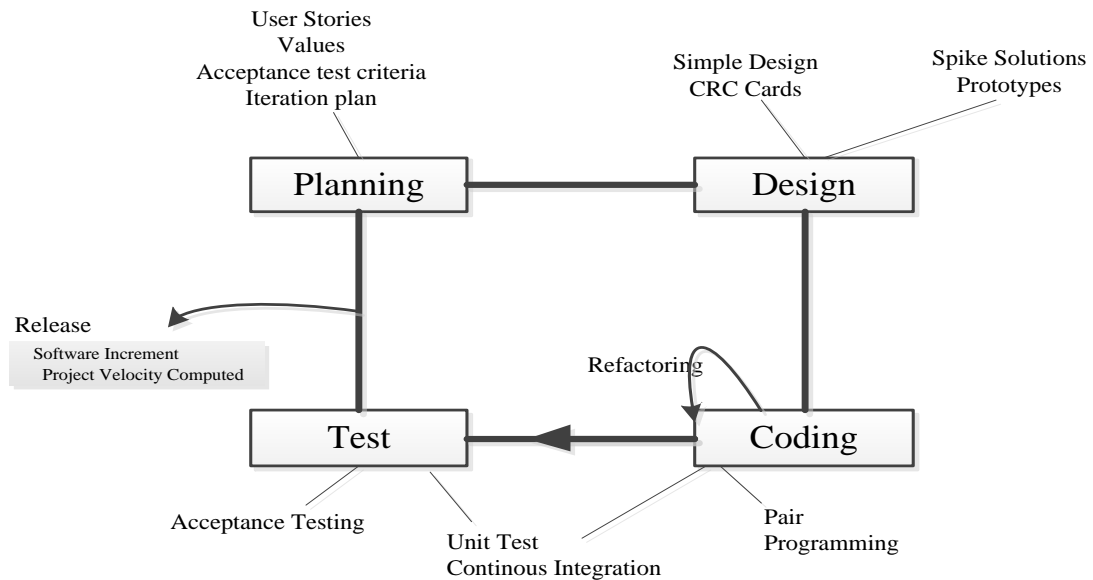
Menurut MySQL (2018) *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan basis data.

Menurut Amin (2018) mendefinisikan *mysql* *MYSQL* adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

2.6 Metode Extreme Programming

Menurut Suryantara (2017) *extreme programming* berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing*. Berikut merupakan konsep *Extreme programming*.



Gambar 2.3 *Extreme Programming*

Sumber: (Suryantara, 2017)

2.6.1 Kerangka Kerja Extreme Programming

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning*, *design*, *coding* dan *testing*.

1. *Planning*/Perencanaan

Tahap ini dimulai dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.

2. *Design*/Perancangan

Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas yang akan digunakan pada diagram UML.

3. *Coding*/Pengkodean

Hal utama dalam pengembangan menggunakan XP yaitu *pari programming* (Proses pembuatan program melibatkan 2 atau lebih programmer).

4. *Testing*/Pengujian

Tahap ini fokus pada pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

2.6.2 Keuntungan dan Kerugian *Extreme Programming*

Keuntungan pada penerapan metode XP yaitu:

- a. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi
- b. Saling menghargai antar developer dan meningkatkan komunikasi
- c. Dapat menjadi pembelajaran bagi orang lain
- d. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
- e. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat
- f. Developer tidak berkerja secara berlebihan
- g. Dengan XP dapat membuat keputusan yang bersifat teknikal

2.6.3 Class Responsibility Collaboration (CRC) Card

Menurut Rosa and Shalahuddin (2019) *Class Responsibility Collaboration (CRC) Card* merupakan tools yang digunakan untuk mendefinisikan behavior dan responsibility dari masingmasing class dan hubungan kolaborasi antara *class-class* tersebut. *CRC Card* dikembangkan dengan interaksi antara analis dan pengguna. Masing-masing orang diminta untuk mendeskripsikan logika yang diperlukan untuk memenuhi suatu *responsibility*, dan informasi apa saja yang dibutuhkan tetapi tidak dimilikinya. *Class* lain yang memiliki informasi yang dibutuhkan akan menjadi *collaborator* untuk *behavior* tersebut.


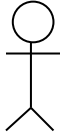
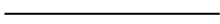
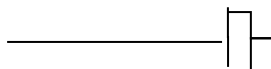
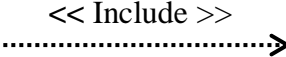
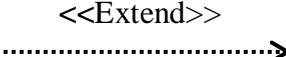
2.7 Alat Pengembang Sistem (Unified Modelling Language)

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut (Rosa and Shalahuddin, 2019) UML (*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 UML (*Unified Modelling Language*).

2.7.1 Use Case Diagram

Menurut Rosa and Shalahuddin (2019) *Use Case* adalah *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 2.5 berikut ini:

Tabel 2.5 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		Use case 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 Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.


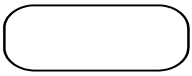
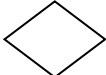

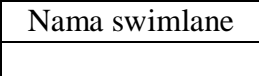

Sumber : (Rosa and Shalahuddin, 2019)

2.7.2 Activity Diagram

Menurut Rosa and Shalahuddin (2019) *activity diagram* adalah *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 *activitydiagram* dapat dilihat pada tabel 2.6 berikut ini :

Tabel 2.6 Simbol *Activity Diagram*

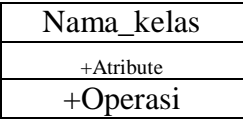
No.	Simbol	Keterangan
1.		Status awal aktivitas sitem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari 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.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.


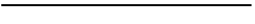
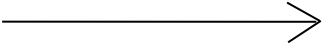
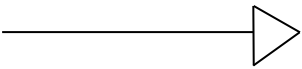
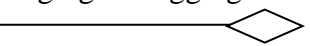
Sumber : (Rosa and Shalahuddin, 2019)

2.7.3 Class Diagram

Menurut Rosa and Shalahuddin (2019) *Class Diagram* adalah *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 2.7 berikut ini :

Tabel 2.7 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.

2.	Antar Muka/Interface  Nama_Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / Association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / Directed Association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / aggregation 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa and Shalahuddin, 2019).

2.8 User Acceptance Test

Menurut Ayunda *et al.*, (2017), *User Acceptance Test* merupakan proses verifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai untuk pengguna. Proses ini berbeda dengan pengujian sistem (memastikan *software* tidak *crash* dan sesuai dengan dokumen permintaan pengguna), melainkan memastikan bahwa solusi dalam sistem tersebut akan bekerja untuk pengguna (yaitu, tes bahwa pengguna menerima solusi di dalam sistem).

User Acceptance Test umumnya dilakukan oleh klien atau pengguna akhir, biasanya tidak fokus pada identifikasi masalah sederhana seperti kesalahan ejaan, maupun di cacat *showstopper*, seperti *crash* perangkat lunak. Penguji dan pengembang mengidentifikasi dan memperbaiki masalah ini selama tahap awal pengujian fungsionalitas, pengujian saat integrasi dan pada tahap sistem testing.