

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Sistem Informasi**

Menurut Krismiaji (2015) sistem informasi merupakan sekumpulan elemen yang saling terhubung atau berkaitan untuk tujuan tertentu, proses pengolahan data mentah menjadi informasi secara tersistem dapat menghasilkan informasi yang lebih sederhana dan mudah digunakan oleh pengguna.

### **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 padapendidikan 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 (Ahmar, 2019). 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.

### **2.3 Profile Matching**

Menurut Syahputra (2021) *Profile matching* adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati.

Langkah-langkah pada metode profil matching yaitu :

1. Menentukan Variabel-variabel Pemetaan Gap Kompetensi  
 Pada umumnya kriteria/aspek yang diukur dan nilai pembobotan yang digunakan adalah satu set variabel yang sudah ditentukan oleh pakar atau orang/lembaga yang berwenang.
  - a. Penentuan Aspek-aspek Penilaian  
 Aspek-aspek apa saja yang akan digunakan sebagai perbandingan.
  - b. Berdasarkan aspek-aspek dan faktor-faktor yang telah ditentukan sebelumnya, selanjutnya adalah menentukan nilai target untuk masing-masing point penilaian tersebut.
2. Pemilihan Kandidat  
 Masing-masing kandidat dinilai berdasarkan point-point faktor dalam aspek-aspek yang telah ditentukan sebelumnya.
3. Menghitung Bobot Dari Pemetaan Gap Kompetensi  
 Setelah proses pemilihan kandidat, proses berikutnya adalah menentukan kandidat mana yang paling cocok menduduki jabatan yang diajukan dapat ditunjukkan pada rumus di bawah ini:

$$\text{Gap} = \text{Profil}_{\text{Karyawan}} - \text{Profil}_{\text{Jabatan}} \quad \dots\dots [GAP-01]$$

Dengan menggunakan persamaan [GAP-01], dapat ditentukan nilai-nilai pemetaan GAP Kompetensi untuk tiap aspek dan alternatif yang diberikan.

4. Menghitung dan Mengelompokkan Core Factor dan Secondary Factor  
 Kemudian tiap aspek dikelompokkan menjadi 2 (dua) kelompok yaitu kelompok Core Factor dan Secondary Factor.

#### ***Core Factor***

Core factor merupakan aspek (kompetensi) yang paling menonjol atau paling dibutuhkan oleh suatu jabatan yang diperkirakan dapat menghasilkan kinerja optimal. Core factor dapat dirumuskan dalam persamaan berikut:

$$N_{CF} = \frac{\sum N_C(i,s,p)}{\sum I_C} \quad \dots [GAP-02]$$

Keterangan

1.  $N_{CF}$  : Nilai rata-rata *core factor*
2.  $N_C(i,s,p)$  : Jumlah total nilai *core factor* (kecerdasan,sikap kerja, perilaku)
3.  $I_C$  : Jumlah Item *core factor*

### ***Secondary Factor***

Secondary factor adalah item-item selain aspek yang ada pada core factor atau bisa juga disebut sebagai faktor pendukung. Secondary factor dapat dituliskan dalam rumusan sebagai berikut:

$$N_{SF} = \frac{\sum N_S(i,s,p)}{\sum I_S} \quad \dots \text{ [GAP-03]}$$

Keterangan

1.  $N_{SF}$  : Nilai rata-rata *secondary factor*
  2.  $N_S(i,s,p)$  : Jumlah total nilai *secondary factor* (kecerdasan,sikap kerja, perilaku)
  3.  $I_S$  : Jumlah Item *secondary factor*
5. Menghitung Nilai Total Tiap Aspek

Perhitungan Nilai Total dapat dilihat pada rumus di bawah ini:

$$x.N_{CF}(i,s,p) + y.N_{SF}(i,s,p) = N(i,s,p) \quad \dots \text{ [GAP-04]}$$

Keterangan

1.  $N_{CF}(i,s,p)$  : Nilai rata-rata *core factor*(kecerdasan,sikap kerja, perilaku)
2.  $N_{SF}(i,s,p)$  : Nilai rata-rata *secondary factor* (kecerdasan,sikap kerja, perilaku)
3.  $N(i,s,p)$  : Nilai Total dari aspek (kecerdasan,sikap kerja, perilaku)
4.  $x$  : Nilai prosentase yang diinputkan untuk *core factor*
5.  $y$  : Nilai prosentase yang diinputkan untuk *secondary factor*

Berdasarkan persamaan-persamaan [GAP-02],[GAP-03] dan [GAP-04] dapat dihitung nilai total untuk setiap aspek untuk masing-masing kandidat sebagai berikut :

## 6. Menghitung Hasil Akhir (Ranking)

Penentuan ranking mengacu pada hasil perhitungan tertentu, sebagai berikut:

$$Rangking = x.N_I + y.N_S + z.N_P$$

.. [GAP-05]

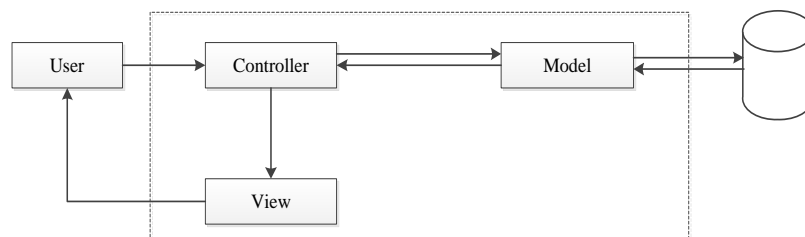
Keterangan

1.  $N_I$  : Nilai Kecerdasan
2.  $N_S$  : Nilai Sikap kerja
3.  $N_P$  : Nilai Perilaku
4.  $x$  : Nilai prosentase yang diinputkan untuk aspek Kecerdasan
5.  $y$  : Nilai prosentase yang diinputkan untuk aspek Sikap kerja
6.  $z$  : Nilai prosentase yang diinputkan untuk aspek Perilaku

## 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*.

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



**Gambar 2.1** Arsitektur MVC

### 2.4.1 Web Based

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

Menurut Purwati, *et al* (2018) 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 di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

#### **2.4.2 PHP**

Menurut Subagja (2018), PHP adalah bahasa *server-side-scripting* yang menyatudengan HTML untuk membuat halaman *web* yang dinamis.

Menurut Aryani, Setiadi and Alfiah (2015), berpendapat bahwa *PHP Hypertext Preprocessor* adalah bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis.

Sehingga menurut Maldhan (2019) PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

### 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 data.

Menurut (Amin, 2018) mendefinisikan *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.

### 2.5 Kamus Data

Menurut dosenpendidikan (2020) Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat difahami secara umum (memiliki standar cara penulisan).

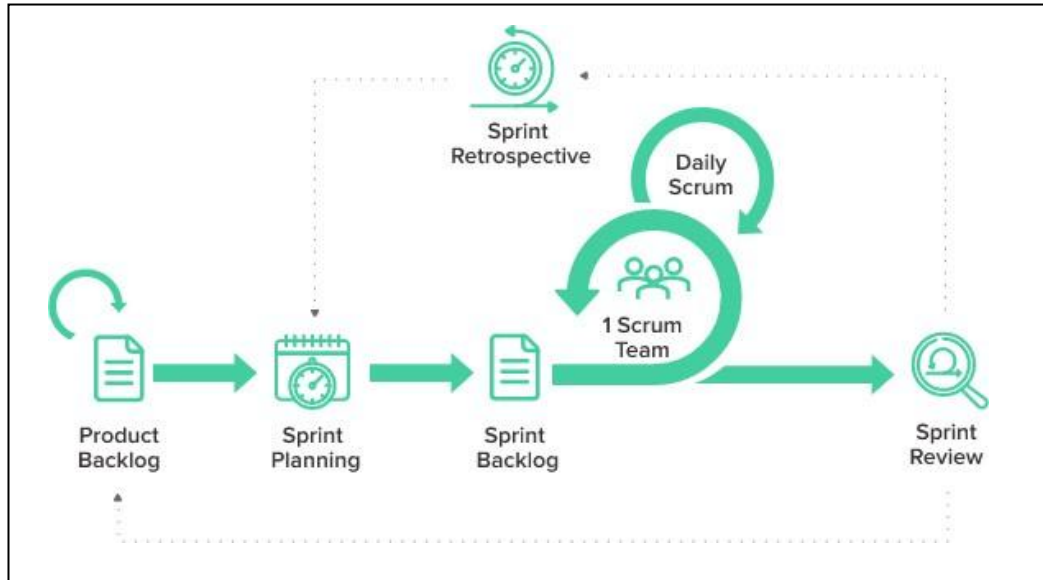
Menurut Nasution (2020) Kamus data selain digunakan untuk dokumentasi dan mengurangi redudansi, juga dapat digunakan untuk: memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan. Menentukan muatan data yang disimpan dalam file-file.

### 2.6 Metode Scrum

“*Scrum Method Implementation in a Software Development Project Management*” dalam (Julianto, 2019) menyatakan bahwa *Scrum* pertama kali dikembangkan oleh Schwaber dan Sutherland pada tahun 1993 dan tujuannya adalah menjadi metodologi pengembangan yang mengikuti prinsip-prinsip metodologi *Agile*.

*Scrum* adalah suatu metodologi atau kerangka kerja yang terstruktur untuk mendukung pengembangan produk yang kompleks. *Scrum* terdiri dari sebuah tim

yang memiliki peran dan tugas masing-masing. Setiap komponen dalam kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan *Scrum* (Schwaber and Sutherland, 2017).



**Gambar 2.2** Tahapan-Tahapan Metode Scrum

### 2.6.1 Tahapan-Tahapan Metode *Scrum*

Adapun tahapan-tahapan dalam *Scrum* oleh (Schwaber and Sutherland, 2017) adalah sebagai berikut ini:

#### 1. **Product Backlog**

Peneliti sistem akan mengumpulkan dan menyusun semua kebutuhan sistem dan permintaan pengguna terhadap sistem, misalnya fitur-fitur yang dibutuhkan oleh pengguna terhadap sistem. *Product backlog* berada dalam tanggung jawab *product owner*. Setelah targetnya ditetapkan, semua kebutuhan dan permintaan akan dibagikan menjadi poin-poin kecil yang mana setiap poin tersebut mempunyai tingkat layak untuk dikembangkan.

#### 2. **Sprint Planning**

*Sprint Planning* merupakan sebuah langkah yang wajib dilaksanakan setiap saat akan memulainya sprint baru. Pada langkah tersebut peneliti akan menyusun pekerjaan-pekerjaan apa saja yang harus diselesaikan dalam 1 *sprint*.

### 3. **Sprint Backlog**

Perencanaan *sprint* dilakukan dalam pertemuan/*meeting* antara pemilik produk dan tim developer, yang akan berkolaborasi untuk memilih produk yang akan dikembangkan *backlog* untuk dimasukkan kedalam proses *sprint*. Hasil dari pertemuan tersebut adalah *sprint backlog*.

### 4. **Sprint**

Dalam *Scrum*, *Sprint* adalah sebuah kerangka waktu yang berdurasi maksimal 1 bulan untuk mengembangkan produk yang berpotensi untuk dirilis. Dalam *Sprint* terdapat 2 bagian pekerjaan, yaitu:

#### a. **Pertemuan Harian (*Daily Standup Meeting*)**

Merupakan pertemuan dimana setiap 24 jam (1 hari), tim pengembang bertemu untuk membahas proses pengembangan produk.

#### b. **Refleksi *Sprint***

Merupakan pertemuan yang dilakukan setiap bulannya, yang bertujuan untuk membahas hal dari *Sprint Backlog* yang telah berjalan dan telah berhasil dikerjakan, serta dapat memperbaiki dan meningkatkan kualitas produk pada *Sprint* yang berikutnya.

### 5. ***Working Increment (Sprint Rivew)***

*Increment* merupakan hasil dari seluruh hal dalam *product backlog* yang telah selesai dikerjakan pada seluruh *sprint*.

## 2.7 ***Unified Modelling Language (UML)***

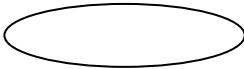
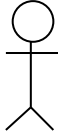

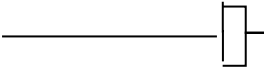
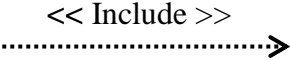

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram (Agarina and Karim, 2019). Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut Rosa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



### 2.7.1 Use Case Diagram

Menurut Rosa and Shalahuddin (2019) *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 2.1** Simbol *Use Case Diagram*



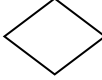

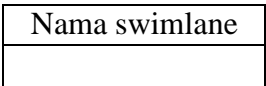

No	Simbol	Deskripsi
1.		Use case Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan 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 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 2.2:

**Tabel 2.2** Simbol *Activity Diagram*

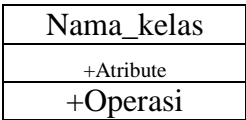


No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan ( <i>Decision</i> ) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari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

**Sumber :** (Rosa and Shalahuddin, 2019)

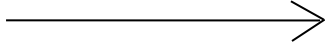
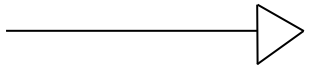
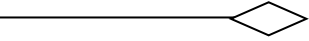
### 2.7.3 *Class Diagram*

Menurut Rosa and Shalahuddin (2019) *Class diagram* 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 2.3** 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 / Asociation 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol

**Tabel 2.3** Simbol *Class Diagram* (Lanjutan)

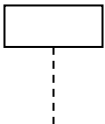

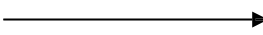
No.	Simbol	Deskripsi
4.	Asosiasi Berarah / Directed Association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / aggregation 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

**Sumber:** (Rosa and Shalahuddin, 2019).

#### 2.7.4 *Sequence Diagram*

Diagram rangkaian menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah use-case atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi pada Tabel 2.4.

**Tabel 2.4** Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<i>Object lifeline</i> 	Menggambarkan panjang kehidupan suatu objek selama scenario sedang di buat contohnya
2.	<i>Activation</i> 	Dimana proses sedang dilakukan oleh <i>object</i> atau <i>class</i> untuk memenuhi pesan atau perintah
3.	<i>Message</i> 	Sebuah anak panah yang mengindikasikan pesan diantara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri

**Sumber:** (Rosa dan Salahuddin, 2019)

## 2.8 Pengujian *Black Box Testing*

*Black box testing* menurut Rosa dan Salahuddin (2019) yaitu pengujian perangkat lunak dari segi pendefinisian fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian yang dilakukan dengan membuat kasus yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji dilakukan harus dibuat dengan benar dan salah, seperti proses *login* “Jika user memasukan *username* dan *password* yang benar maka dapat *login*?”.

## 2.9 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan perbandingan dan acuan. Selain itu, untuk menghindari anggapan kesamaan dengan penelitian ini. Maka dalam kajian pustaka ini peneliti mencantumkan hasil-hasil penelitian terdahulu sebagai berikut:

1. Hasil Penelitian Axel Reinno Fabiyanto, Yusi Tyroni Mursityo dan Djoko Pramono (2019)

Penelitian oleh Axel Reinno Fabiyanto, Yusi Tyroni Mursityo dan Djoko Pramono dengan judul “Pengembangan Sistem Informasi Penilaian Kinerja Guru Menggunakan Metode Rational Unified Process (RUP) Berbasis Web”. Penelitian ini merupakan penelitian yang menggunakan Metode Rational Unified Process (RUP). Hasil penelitian yaitu Dengan adanya sistem informasi penilaian kinerja guru dapat menyelesaikan masalah seperti proses penilaian kinerja guru menjadi lebih cepat dan efisien serta mempermudah proses pembuatan laporan.

2. Hasil Penelitian Deiky Arizal, Muhammad Nur Cholis dan Laura Saraswati Nusantara (2016)

Penelitian oleh Deiky Arizal, Muhammad Nur Cholis dan Laura Saraswati Nusantara dengan judul “Perancangan Aplikasi Penilaian Kinerja Guru Di Upt Sdn Kebonagung”. Penelitian ini merupakan penelitian yang menggunakan Metode *Waterfall*. Hasil penelitian yaitu Aplikasi yang telah

dirancang dan dibangun memudahkan Tim Penilai di UPT SDN kebonagung dalam menilai kinerja guru dan menjadi fasilitas alternatif untuk menilai kinerja guru.

3. Hasil Penelitian Moh Fadik Pratama S.Sadjar, Sitti Suhada, S.Kom.,MT, Edi Setiawan,M.Kom (2020)

Penelitian oleh Moh Fadik Pratama S.Sadjar, Sitti Suhada, S.Kom.,MT, Edi Setiawan,M.Kom dengan judul “Sistem Informasi Kinerja Sekolah Dan Guru Di Dinas Kabupaten Gorontalo Berbasis Web”. Penelitian ini merupakan penelitian yang menggunakan metode waterfall. Hasil penelitian yaitu Aplikasi ini dapat Harapkan membantu pihak sekolah dalam menilai Guru dan Accessor dalam menilai Akreditasi sekolah.

4. Hasil penelitian olah Ritong, A'id dan Megayanti, 2021

Penelitian oleh Roy Amrullah Ritonga, Asep Maburur A'id, Anita Megayanti dengan judul “Implementasi Metodologi Scrum Dalam Pengembangan Aplikasi Eregitrasi Vendor (Studi Kasus : Krakatau IT)”. Penelitian ini merupakan penelitan yang menggunakan Scrum. Hasil penelitian yaitu penggunaan metodologi scrum pada implementasi eregstrasi vendor yang dirancang dan dikembangkan bertujuan untuk memudahkan dapat membantu bagian perencanaan dalam proses pendaftaran vendor secara efesien dan efektif pada saat pencarian vendor yang expire dokumen tidak boleh mengikuti pengadaan serta memudahkan para vendor untuk mendaftar cukup dengan mengakses link eregistrasi tidak perlu datang langsung ke lokasi perusahaan.

**Tabel 2.5** Penelitian Terdahulu

<b>Nama</b>	<b>Judul</b>	<b>Variabel</b>	<b>Metode Analisis</b>	<b>Hasil Analisis</b>
Axel Reinno Fabiyanto, Yusi Tyroni Mursityo dan Djoko Pramono (2019)	Pengembangan Sistem Informasi Penilaian Kinerja Guru Menggunakan Metode Rational Unified Process (RUP)	Sistem, Informasi, Penilaian Kinerja Guru	RUP	Hasil penelitian yaitu Dengan adanya sistem informasi penilaian kinerja guru dapat menyelesaikan

Nama	Judul	Variabel	Metode Analisis	Hasil Analisis
	Berbasis Web			n masalah seperti proses penilaian kinerja guru menjadi lebih cepat dan efisien serta mempermudah proses pembuatan laporan.
Deiky Arizal, Muhammad Nur Choliz dan Laura Saraswati Nusantara (2016)	Perancangan Aplikasi Penilaian Kinerja Guru Di Upt Sdn Kebonagung	Aplikasi, Penilaian	Metode <i>Waterfall</i>	Hasil penelitian yaitu Aplikasi yang telah dirancang dan dibangun memudahkan Tim Penilai di UPT SDN kebonagung dalam menilai kinerja guru dan menjadi fasilitas alternatif untuk menilai kinerja guru
Moh Fadik Pratama S.Sadjar, Sitti Suhada, S.Kom.,MT, Edi Setiawan,M.Kom (2020)	Sistem Informasi Kinerja Sekolah Dan Guru Di Dinas Kabupaten Gorontalo Berbasis Web	Kinerja Sekolah Dan Guru	Metode <i>Waterfall</i>	Hasil penelitian yaitu Aplikasi ini dapat Harapkan membantu pihak sekolah dalam menilai Guru dan Accessor dalam menilai Akreditas sekolah

**Tabel 2.5** Penelitian Terdahulu

Nama	Judul	Variabel	Metode Analisis	Hasil Analisis
Roy Amrullah Ritonga, Asep Maburur A'id, Anita Megayanti (2021)	Implementasi Metodologi Scrum Dalam Pengembangan Aplikasi Eregitrasi Vendor (Studi Kasus : Krakatau It)	Framework CodeIgniter, Metode Scrum, Product Backlog	Scrum	Penggunaan metodologi scrum pada implementasi eregstrasi vendor yang dirancang dan dikembangkan bertujuan untuk memudahkan dapat membantu bagian perencanaan dalam proses pendaftaran vendor secara efisien dan efektif pada saat pencarian vendor yang expire dokumen tidak boleh mengikuti pengadaan serta memudahkan para vendor untuk mendaftar cukup dengan mengakses link eregistrasi tidak perlu datang langsung ke lokasi perusahaan.