

BAB II LANDASAN TEORI

2.1 Tinjauan Studi

Dalam melaksanakan penelitian ini, diambil beberapa referensi sebagai dasar pelaksanaan penelitian tentang penilaian kinerja karyawan diantaranya :

Pada tinjauan studi yang pertama oleh M Fakhri Darmawan, Indra Waspada, Putut Sri Wasito.2011. *Pembuatan Sistem Penilaian Pegawai Berbasis Web bagi Pejabat dan Pelaksana di Lingkungan Universitas Diponegoro*. Pembuatan Sistem Penilaian Pegawai Berbasis Web bagi Pejabat dan Pelaksana di Lingkungan Universitas Diponegoro. Dihasilkan sebuah Sistem Penilaian Pegawai untuk memudahkan UPT Puskom Universitas Diponegoro dalam melakukan penilaian yang sesuai dengan Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011 sehingga dapat digunakan sebagai acuan dalam kenaikan dan penurunan jabatan dan peringkat bagi pejabat dan pelaksana di lingkungan Universitas Diponegoro. Sistem Penilaian Pegawai ini dikembangkan menggunakan *framework* PHP *CodeIgniter* yang berbasis Objek dan dapat digunakan di lingkungan Universitas Diponegoro melalui *web browser* didalam jaringan Universitas Diponegoro. Pengembangan sistem informasi ini menggunakan *Unified Process* sehingga memudahkan ketika diperlukan penelusuran dan modifikasi kebutuhan.

Pada tinjauan studi yang kedua oleh Ahmad Ruyani, Eneng Tita Tosida, Sufiatul Maryana, 2014. *Aplikasi penilaian kinerja kepegawaian Menggunakan metode perbandingan eksponensial (mpe) berbasis web*. Pola perhitungan yang digunakan untuk membangun aplikasi penilaian kinerja kepegawaian yaitu dengan menggunakan metode MPE dan teknik pembobotan *eckonrode*. dengan memangkatkan nilai hasil dari penghitungan sistem antar semua kriteria dan mencari skor hasil tertinggi. Sistem ini telah melalui tahap proses uji coba, diantaranya yaitu, uji coba struktural, uji coba fungsional, dan uji coba validasi yang telah berhasil dilakukan. Hasil uji coba nilai ekstrim pada sistem yaitu input nilai yang seragam atau sama antara 0 sampai 100 untuk seluruh input nilai pada setiap kriteria menghasilkan grafik

nilai yang tidak begitu signifikan perbedaannya, karena menurut rumus MPE No.1 yaitu nilai setiap kriteria dipangkatkan dengan bobot kriteria, sedangkan bobot kriteria sangat kecil yaitu 0,31 untuk bobot terbesar dan 0,07 untuk bobot terkecil karena $W_e(\text{total bobot keseluruhan}) = 1$, sehingga hasil perhitungan tidak begitu signifikan perbedaannya.

Pada tinjauan studi yang ketiga oleh Arie Eka S, 2014. *Sistem penilaian akademik berbasis web (studi kasus SMP Kartini Tamansidoarjo) menggunakan codeigniter*. Sistem informasi penilaian berbasis web yang telah dibuat dapat membantu pengelolaan data penilaian. Solusi yang diberikan sistem ini adalah mengurangi adanya kesalahan data yang dimasukkan ke dalam penilaian siswa. Dengan adanya sistem penilaian ini yaitu siswa dapat melihat informasi nilai ataupun informasi sekolah, dan dapat memudahkan atau efektif dan efisien dalam memberikan informasi nilai dan informasi tentang sekolah yang ada di sekolah SMP KARTINI taman sidoarjo. Dan mampu memberikan sebuah informasi kepada orang tua atau wali murid atas penilaian siswa/ siswi yang bersangkutan di sekolah tersebut melalui sms gateway.

2.2 Tinjauan Pustaka

2.2.1 XAMPP

Menurut Yogi wicaksono (2008:7) “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* dikomputer local”. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet.

XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. *Xampp* merupakan kepanjangan dari hurufnya yaitu :

X : Program ini dapat dijalankan dibanyak sistem operasi, seperti *Windows*, *Linux*, *Mac OS* dan juga *Solaris*.

A : Apache, merupakan aplikasi web server. Tugas utama dari *Apache* adalah menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode PHP yang dituliskan oleh pembuat *web* atau *user*.

M : MySql, merupakan aplikasi data *server*. Perkembangannya disebut juga *Sql* yang merupakan kepanjangan dari *Structured Query Language*. *Sql* merupakan bahasa terstruktur yang digunakan untuk mengolah *database*.

P : PHP, merupakan bahasa pemrograman *web*, dimana *user* dapat menggunakan bahasa pemrograman ini untuk membuat *web* yang bersifat *server-side scripting*. **P : Perl**, yaitu merupakan bahasa pemrograman untuk segala keperluan, dan dikembangkan pertama kali oleh Larry Wall di mesin *Unix*.

2.2.2 *Hypertext Markup Language (HTML)*

Menurut Shelly, Woods, & Dorin (2008:8) *HTML* merupakan bahasa penulisan yang digunakan untuk membuat dokumen pada halaman *web*. *HTML* menggunakan sekumpulan instruksi khusus, yang dikenal dengan *tags* atau *markup* untuk mendefinisikan struktur dan susunan dari *web document* dan menetapkan apa yang akan ditampilkan pada *browser*.

2.2.3 **PHP**

Menurut Valade (2014:9) *PHP* akronim dari *Hypertext Preprocessor* adalah *open source* yang banyak digunakan sebagai tujuan utama *scripting language*. Di desain untuk digunakan pada pengembangan *website*. *PHP* berawal dari *personal home page tools*, yang di kembangkan oleh Rasmus Lerdorf untuk membantu *user* dengan *web page tasks*. *PHP* dibuktikan sangat berguna dan populer serta secara bertahap berkembang untuk menjadi *full-featured language*.

2.2.4 **MySQL**

Menurut King et.al (2009:4) *MySQL* adalah *open source database product* yang mendukung *key subsets SQL* di *Linux* dan *Unix systems*. *MySQL* digunakan secara gratis dan hanya menghabiskan sedikit biaya bagi periklanannya namun sangat populer. Tidak seperti *commercial database*, *MySQL* sangat terjangkau dan mudah digunakan.

Fitur yang terdapat di MySQL antara lain:

- *Openess*

MySQL sangat terbuka dalam setiap hubungan. *SQL dialect* yang digunakan adalah ANSI SQL2 sebagai pembangunnya. *Database engine* menjalankan *platform* yang tidak terhitung, termasuk Windows 2000, Mac OS X, Linux, FreeBSD, dan Solaris. Jika *binary* tidak tersedia untuk *platform*, user dapat mengaksesnya dari *source* untuk di *compile* ke *platform* tersebut.

- *Application Support*

MySQL memiliki API untuk semua bahasa pemrograman. *User* bisa menulis aplikasi *database* yang mengakses MySQL di C, C++, Eiffel, Java, Perl, PHP, Python, dan Tcl.

- *Cross-database joins*

User bisa mengkonstruksi MySQL *queries* yang bisa menggabungkan *tabel* dari *database* yang berbeda

- *Outer join support*

MySQL mendukung kiri dan kanan *outer joins* menggunakan ANSI dan sintaks ODBC.

2.2.5 Penilaian Kinerja

Pada organisasi yang modern, penilaian memberikan mekanisme penting bagi manajemen untuk digunakan dalam menjelaskan tujuan, standar kinerja dan untuk memotivasi kinerja individu di waktu berikutnya. Ini merupakan komponen kunci dalam proses pelaksanaan personalia dari sebagian besar perusahaan dan memberikan basis untuk keputusan-keputusan yang mempengaruhi gaji, promosi, pemberhentian, pelatihan, transfer, dan kondisi-kondisi kepegawaian lainnya.[4]

Menurut Henry Simamora (2012:338) penilaian kinerja adalah proses yang dipakai oleh organisasi untuk mengevaluasi pelaksanaan kerja individu karyawan. Penilaian kinerja merupakan salah satu fungsi mendasar personalia, kadang-kadang disebut juga dengan review kinerja, penilaian karyawan, evaluasi kinerja, evaluasi karyawan, atau *rating* personalia.

2.2.6 Motivasi

Motivasi adalah proses yang menyebabkan intensitas (*intensity*), arah (*direction*), dan usaha terus-menerus (*persistence*) individu menuju pencapaian tujuan. Intensitas menunjukkan seberapa keras seseorang berusaha untuk mencapai tujuannya. Motivasi memiliki dimensi usaha terus menerus. Motivasi merupakan ukuran berapa lama seseorang dapat menjaga usaha mereka.[5]

2.2.7 Kinerja Karyawan

Kinerja merupakan suatu pencapaian berhasil atau tidaknya tujuan organisasi yang telah ditetapkan. Informasi tentang kinerja organisasi adalah hal yang sangat penting yang digunakan untuk mengevaluasi apakah proses kinerja yang dilakukan organisasi selama ini sudah sejalan dengan tujuan yang diharapkan atau belum mendefinisikan kinerja sebagai hasil kerja yang dicapai oleh individu yang disesuaikan dengan peran atau tugas individu tersebut dalam suatu perusahaan pada suatu periode waktu tertentu, yang dihubungkan dengan suatu ukuran nilai atau standar tertentu dari perusahaan dimana individu tersebut bekerja.[6]

2.2.8 Unified Modelling Language (UML)


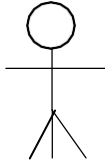
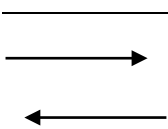
Menurut Whitten dan Bentley (2007: 371) *UML (Unified Modelling Language)* merupakan satu kumpulan konvensi pemodelan yang digunakan untuk merincikan atau menjelaskan sistem piranti lunak yang dijadikan sebagai standar pemodelan objek. *UML* dapat dijadikan panduan bagi *developer* dalam mengetahui sudut pandang sebuah sistem dan dikomunikasikan kepada pihak – pihak yang terkait dengan sistem dalam bentuk diagram. *UML* merupakan bahasa kesatuan yang mendeskripsikan model sebuah sistem secara efektif.

1) Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem sendiri melalui sebuah cerita dimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan

urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dalam pembicaraan tentang use case, pengguna biasanya disebut dengan actor. Actor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Pada notasi use case diagram dapat menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan sistem atau sub sistem boundary. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case[13].

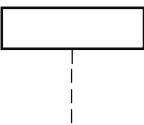
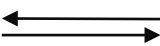
Tabel 2.1 Simbol *Use Case*

No.	Simbol	Nama	Deskripsi
1.		<i>Case</i>	Menggambarkan proses / kegiatan yang dapat dilakukan oleh actor
2.		<i>Actor</i>	Menggambarkan entitas / subjek yang dapat melakukan suatu proses
3.		<i>Relation</i>	Relasi antara case dengan actor ataupun case dengan case lain.

2) *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan (*message*) yang diletakkan diantara objek-objek ini didalam use case. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segi empat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progres vertical[13]






Tabel 2.2 Simbol *Sequence Diagram*

No.	Simbol	Nama	Deskripsi
1.		<i>Object</i>	Menggambarkan pos-pos objek yang pengirim dan penerima <i>message</i>
2.		<i>Message</i>	Menggunakan aliran pesan yang dikirim oleh pos-pos objek

1) Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peranan seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa[13].

Tabel 2.3 *Activity Diagram*

No.	Simbol	Nama	Deskripsi
1.		<i>Action State</i>	Menggambarkan keadaan elemen dalam suatu aliran
2.		<i>State</i>	Menggunakan kondisi suatu elemen
3.		<i>Flow</i>	Menggambarkan aliran aktifitas dari suatu elemen ke elemen lain
4.		<i>Initial State</i>	Menggambarkan titik awal siklus hidup suatu elemen
5.		<i>Final State</i>	Menggambarkan titik akhir yang menjadi kondisi akhir suatu elemen

2.2.9 *CodeIgniter*

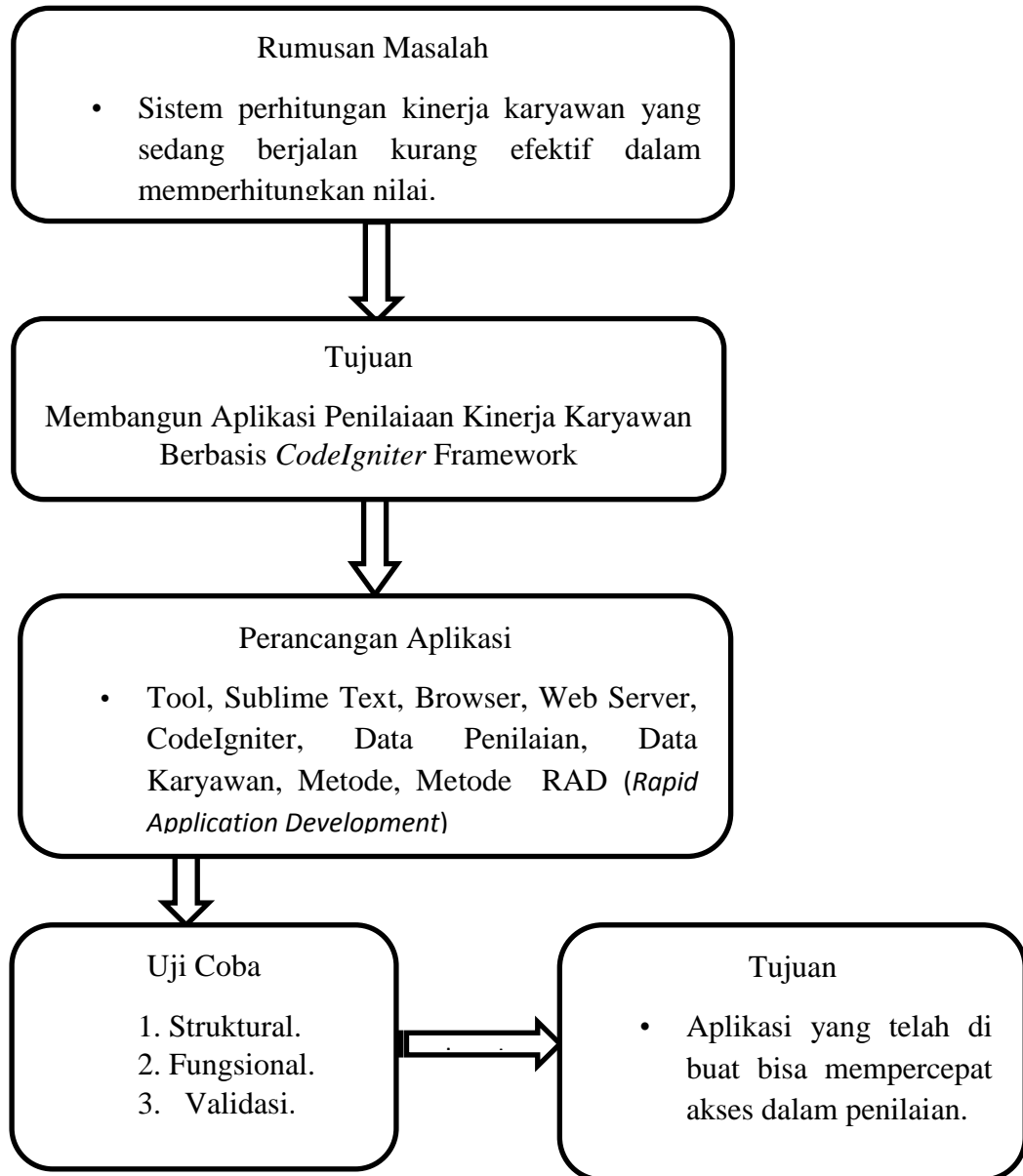
CodeIgniter (CI) merupakan aplikasi *open source* yang berupa *framework* dengan model *MVC* (*Model, View, Controller*) untuk membangun website dinamis menggunakan PHP. Tiga jenis komponen yang membangun suatu *MVC pattern* dalam suatu aplikasi antara lain :

1. **View** adalah merupakan bagian yang menangani *presentation logic* pada suatu aplikasi web, bagian ini biasanya berupa file template HTML yang diatur oleh *controller*.
2. **Model** yaitu biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*) juga menangani validasi

dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

3. **Controller** merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *Controller* berfungsi untuk menerima *request* dan data dari *user*, kemudian menentukan apa yang akan diproses oleh aplikasi. [3]

2.3 Kerangka Pemikiran



Tabel 2.4 Kerangka Pemikiran Penulis