

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Proses penulisan skripsi ini, penulis sedikit banyak terinspirasi dan dirujuk oleh penelitian-penelitian terdahulu yang berkaitan dengan latar belakang skripsi ini. Studi sebelumnya yang terkait dengan studi ini tercantum dalam Tabel 2.1:

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti	Judul	Parameter	Hasil
1`	Rodianto, Nur Imansyah, 2022	Sistem Informasi Penilaian Kinerja Pegawai Berbasis Web (Studi Kasus: Kantor Kecamatan Empang)	1. Kinerja pegawai 2. Kedisiplinan 3. Sikap 4. Pekerjaan	System ini dapat mengetahui tentang hasil kinerja
2	Mohamad Natsir, Yunita Sartika, Alamsyah, 2020	Aplikasi Sistem Informasi Penilaian Kinerja Karyawan Menggunakan Metode Profile Matching	1. Kehadiran 2. kinerja 3. Tanggung jawab	mengelola data karyawan dengan baik agar mengefektifkan dan mengefisiensikan proses penilaian kinerja di Perusahaan

No	Peneliti	Judul	Parameter	Hasil
		Berbasis Webkeluarga harapan		
3	Refni Wahyu ni, Yuda Irawan, 2021	Sistem penilaian kinerja karyawan berbasis web pada pt.wifiku indonesia	<ol style="list-style-type: none"> 1. kedisiplinan 2. kejujuran 3. kepribadian 4. kerjasama 	Sistem penilaian kinerja karyawan dibangun menggunakan berdasarkan kebutuhan sistem, dan dirancang untuk memudahkan proses seleksi.
4	Amelia Pratiwi, Aries Suharso ,Hannie , 2021	Klasifikasi Nilai Kepuasan Masyarakat Terhadap Pelayanan E-KTP Menggunakan Algoritma C4.5 (Studi Kasus: Kantor Kecamatan Rengasdengklok	<ol style="list-style-type: none"> 1. himpunan kasus 2. variable 3. jumlah partisi 4. jumlah kasus 	kepuasan masyarakat terhadap pelayanan e-KTP dengan tingkat kepuasan puas dan tidak puas dengan memakai metodologi KDD (knowledge Discovery of Database)
5	Tri Hartati,	Analysis and evaluation of	<i>1. Throughput</i>	Hasil dari analisa dan evaluasi

No	Peneliti	Judul	Parameter	Hasil
	Noer Hikmah, Leliyannah, 2022	employee performance assessment system using the pieces method at pt. Citra pesona gemilang	<p>2. <i>Respon Time</i></p> <p>3. <i>Audibilitas</i></p> <p>4. <i>Communication</i></p> <p>5. <i>Consistency</i></p>	terhadap sistem yang berjalan

2.2 Teori yang berhubungan dengan sistem secara umum

2.2.1 Pengertian Sistem

Sistem dapat didefinisikan dengan dua (2) pendekatan sistem, yaitu didefinisikan dengan pendekatan prosedur dan didefinisikan dengan pendekatan komponen. Pendekatan sistem yang didefinisikan dengan pendekatan prosedur ialah sebagai berikut :

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu[2]. Sedangkan pendekatan sistem yang didefinisikan dengan pendekatan komponennya ialah sebagai berikut:

Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditunjukkan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan.[3]

2.2.2 Pengertian Web

Menurut Elgamar, website adalah suatu media yang terdiri dari beberapa halaman yang saling berkaitan satu sama lain, dan berfungsi sebagai media untuk menampilkan suatu informasi, baik berbentuk gambar, video, teks, suara, ataupun gabungan dari semuanya[4]

Website bersifat multiplatform yang artinya dapat dibuka dari segala perangkat atau device yang terhubung dengan jaringan internet.

2.3 Teori yang berhubungan dengan Topik yang diangkat

2.3.1 Pengertian Penilaian Kinerja

Penilaian kinerja adalah proses mengevaluasi seberapa baik karyawan melakukan pekerjaan mereka jika dibandingkan dengan seperangkat standar, dan kemudian mengkomunikasikan informasi tersebut kepada karyawan.[5]

Sedangkan menurut Siti dkk “Penilaian kinerja karyawan adalah serangkaian aktifitas evaluasi yang dilakukan secara sistematis mengenai performa seorang karyawan, dengan cara membandingkan antara kinerja aktual dengan kinerja standar yang sebelumnya telah ditetapkan oleh manajemen perusahaan dengan disertai pemberian umpan balik (feedback) dalam rangka pengembangan karyawan”. [6]

2.3.2 Pengertian RAD

Menurut Beynon-Davies, P., dan Holmes, S. *RAD* adalah pendekatan berulang dan kontingen untuk interaktif pengembangan perangkat lunak yang ditandai dengan jumlah besar keterlibatan pengguna, penggunaan prototipe tambahan dan manajemen proyek berbasis produk. Untuk membantu kami memahami dan menafsirkan materi yang kami kumpulkan saat ini kami mulai membandingkan dan membedakan aspek praktik *RAD* dengan bidang akademik yang memiliki kemiripan keluarga dengan *RAD* pendekatan - bidang desain partisipatif (PD) perangkat lunak sistem.[7]

2.3.3 Pengertian Kepuasan Masyarakat

Kepuasan masyarakat merupakan faktor utama yang harus diperhatikan oleh penyelenggara pelayanan publik, karena kepuasan masyarakat akan menentukan berhasil atau tidaknya pemerintah dalam mengelola pelayanan publik. Pengertian kepuasan masyarakat sering dikacaukan dengan pengertian kepuasan pelanggan atau kepuasan konsumen, penyedia dan apa motivasi untuk menyediakan layanan.

Menurut Atmaja, "Kepuasan sosial adalah perasaan senang atau kecewa yang dialami seseorang dengan membandingkan kinerja atau hasil yang dirasakan dari suatu produk atau jasa dengan harapannya.[8]

2.3.4 Pengertian *Framework* (Kerangka kerja)

Kecerdasan Kerangka kerja atau framework adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks.[9] Istilah ini sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali, serta dalam bidang manajemen untuk menggambarkan suatu konsep yang memungkinkan penanganan berbagai jenis atau entitas bisnis secara homogen.

2.3.5 Pengertian *Codeigniter*

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal.

CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.3.1.2 CodeIgniter. Tujuan dari pembuatan framework CodeIgniter ini menurut user manualnya adalah untuk menghasilkan framework yang akan dapat digunakan untuk pengembangan proyek pembuatan website secara lebih cepat

dibandingkan dengan pembuatan website dengan cara koding secara manual, dengan menyediakan banyak sekali pustaka yang dibutuhkan dalam pembuatan website, dengan antarmuka yang sederhana dan struktur logika untuk mengakses pustaka yang dibutuhkan. CodeIgniter membiarkan kita untuk memfokuskan diri pada pembuatan website dengan meminimalkan pembuatan kode untuk berbagai tujuan pembuatan website. Seperti sudah disebutkan penulis di atas, bahwa CodeIgniter menerapkan lingkungan pengembangan dengan metode MVC (Model View Controller). [10]

MVC memisahkan antara logika pembuatan kode dengan pembuatan template atau tampilan website. Penggunaan MVC membuat pembuatan sebuah proyek website menjadi lebih terstruktur dan lebih sederhana. Secara sederhana konsep MVC terdiri dari tiga bagian yaitu bagian Model, bagian View dan bagian Controller. Didalam website dinamis setidaknya terdiri dari 3 hal yang paling pokok, yaitu basis data, logika aplikasi dan cara menampilkan halaman website. 3 hal tersebut direpresentasikan dengan MVC yaitu model untuk basis data, view untuk cara menampilkan halaman website dan controller untuk logika aplikasi..

2.4 Teori pendukung lainnya

Teori pendukung lainnya yakni meliputi Basis Data (*database*), HTML (*Hyper Text markup Language*), PHP, MySQL dan Visual Studio.

2.4.1 Basis Data (*database*)

1. Pengertian Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.[11]

Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang

direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data merupakan kumpulan data yang saling berhubungan. Relasinya biasanya biasanya ditunjukkan dengan kunci dari tiap *file* yang ada. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan dan menunjukkan dalam satu pengertian yang lengkap dalam satu *record*

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbaiss berkas.[10]

2.4.2 HTML (Hyper Text markup Language)

HTML (Hyper Text markup Language) merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman *web*. *HTML* berfungsi untuk mempublikasi dokumen online. *Statement* dasar dari mempubliasi dokumen *online*. *Statement* dasar dari *HTML* disebut *tags*. Sebuah *tag* dinyatakan dalam sebuah kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atay bagian dari suatu doumen haruslah dibuat berupa pasangan. Terdiri dari *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag* [13]. Dokumen *HTML* disusun oleh elemen-elemen. Elemen merupakan istilah bagi komponen-komponen dasar pembentuk dokumen *HTML*. Bberapa *HTML* contoh adalah: *head*, *body*, *table*, *paragraf*, dan *list*.

Untuk menandai berbagai elemen dalam suatu dokumen *HTML*, dapat menggunakan tag. Tag *HTML* terdiri atas sebuah kurung sudut kiri (<, Tanda lebih kecil), sebuah nama *tag*, dan sebuah kurung sudut kanan (>, tanda lebih besar). *Tag* umumnya berpasangan (misalnya <H1>

dengan `</HI>`, tag yang berpasangan selalu diawali dengan karakter garis miring (/). Tag-tag yang pertama menunjukkan *tag* awal yang berarti awal elemen, dan yang kedua menunjukkan tag akhir, berarti akhir elemen.

Elemen yang dibutuhkan untuk membuat suatu dokumen HTML dinyatakan dengan tag `<html>`, `<head>`, dan `<Body>` berikut *tag-tag* pasangannya. Setiap dokumen terdiri atas tag *head* dan *body*. Elemen *head* berisi informasi tentang dokumen tersebut, dan elemen *body* berisi tentang teks yang sebenarnya yang tersusun dari *link*, *grafik*, paragraf dan elemen lainnya.[13]

2.4.3 Pengertian *PHP*

PHP adalah bahasa pemrograman untuk dijalankan melalui halaman *web*, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain *PHP* adalah singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *webserver side* yang bersifat *open source* atau gratis. *PHP* merupakan script yang menyatu dengan *HTML* dan berada pada *server*. [14]

Seluruh aplikasi berbasis *web* dapat dibuat dengan *PHP*. Namun kekuatan yang paling utama *PHP* adalah pada konektivitasnya dengan sistem *database* didalam *web*. Kelebihan-kelebihan dari *PHP* diantaranya adalah:

- a. *PHP* mudah dibuat dan dijalankan, maksudnya *PHP* dapat berjalan dalam *web Server* dan dalam sistem operasi yang berbeda pula.
- b. *PHP* adalah *software open-source* yang gratis dan bebas didistribusikan kembali di bawah lisensi *GPL (GNU Public License)*. User dapat mendownload kode-kode *PHP* tanpa harus mengeluarkan uang atau khawatir dituntut oleh pihak pencipta *PHP*.
- c. *PHP* bisa dioperasi pada platform *Linux* ataupun *Windows*.
- d. *PHP* sangat efisien, karena *PHP* hanya memerlukan *resource system* yang sangat sedikit dibanding dengan bahasa pemrograman lain.

- e. Ada banyak *Web Server* yang mendukung PHP, seperti Apache, PWS, IIS, dan lain-lain.
- f. PHP juga didukung oleh database, seperti MySQL, PostgreSQL, interbase, SQL, dan lain-lain.
- g. Bahasa pemrograman PHP sintaknya sederhana, singkat dan mudah dipahami.
- h. *HTML-embedded*, artinya PHP adalah bahasa yang dapat ditulis dengan menempelkan pada sintak-sintak HTML.

2.4.4 Pengertian *MySQL*

MySQL adalah salah satu jenis database yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis. *MySQL* termasuk jenis *RDBMS (Relational Database Management Sistem)*. *MySQL* ini mendukung Bahasa pemrograman *PHP*. *MySQL* juga mempunyai query atau bahasa *SQL (Structured Query Language)* yang simple dan menggunakan escape character yang sama dengan *PHP* [13]

1. Keunggulan *MySQL*

Beberapa keunggulan dari *MySQL* yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya *MySQL* lebih cepat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan *MySQL*.
- b. Didukung oleh berbagai bahasa *Database server MySQL* dapat memberikan pesan error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman dan Italia.
- c. Mampu membuat tabel berukuran sangat besar 23.
- d. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan *MySQL* adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.

- e. Lebih murah *MySQL* bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk *UNIX platform*, *OS/2* dan *Windows platform*.

2.4.5 *Sublime Text*

1. Pengertian *Sublime Text*

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi *python API*. [14] Terciptanya aplikasi ini terinspirasi dari aplikasi *Vim*, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hamper di semua bahasa pemrograman yang didukung ataupun *dikembangkan* oleh komunitas seperti; *C*, *C++*, *C#*, *CSS*, *D*, *Dylan*, *Erlang*, *HTML*, *Groovy*, *Haskell*, *Java*, *Javascript*, *LateX*, *Lisp*, *Lua*, *Markdown*, *MATLAB*, *OCaml*, *Perl*, *PHP*, *Python*, *R*, *Ruby*, *SQL*, *TCL*, *Textile* dan *XML*. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukug secara *default* dapat lebih dimaksimalkan atau di dukung dengan menggunakan *add-ons* yang bisa di download sesuai kebutuhan *user*

2.5 Teori yang berhubungan dengan teknik analisa yang digunakan

Teori yang berhubungan dengan teknik analisa yang digunakan yakni meliputi UML.

2.5.1 UML (*Unified Manipulation Language*)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem” (Windu dan Grace, 2013). Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software (<http://www.omg.org>). Diagram Unified Modelling Language.[12]

Unified Manipulation Language (UML) salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Berikut dibawah ni beberapa diagram *Unified Manipulation Language (UML)*.

1. Diagram *Use Case (Use Case Diagram)*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rossa dan Shalahuddin, 2013). Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada suatu hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

a. Aktor


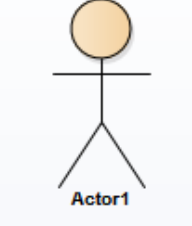
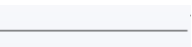
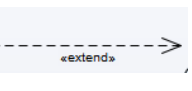
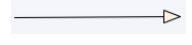
Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang.

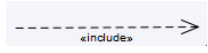
b. *Use case*

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut dibawah ini simbol dan penejelasan yang ada pada diagram *use case*.

Tabel 2. 2 Simbol - simbol Use Case Diagram

No	Simbol	Nama	Deskripsi
1.		<i>UseCase</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar proses antar unit atau aktor. Biasanya
2.		Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang , tapi aktornya belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase aktor.
3.		Asosiasi / <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Ekstensi / <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.		Generalisasi / <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih dari umum dari lainnya.

No	Simbol	Nama	Deskripsi
6.		<i>Include</i>	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Asa dua sudut pandang yang cukup besar mengenai include di <i>use case</i> .

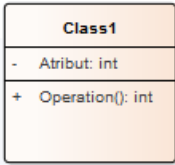
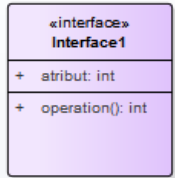
2. Diagram Kelas (*Class Diagram*)

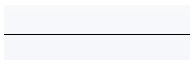
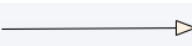

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [12]. Kelas memiliki apa yang disebut atribut dan metode atau operasi :

- 1) Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- 2) Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh kelas.

Berikut dibawah ini adalah simbol-simbol pada *Class Diagram* :

Tabel 2. 3 Simbol - simbol *Class Diagram*

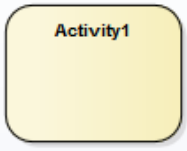
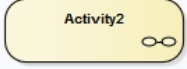
No	Simbol	Nama	Deskripsi
1.		<i>Class</i>	Kelas pada struktur sistem.
2.		Antarmuka / <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek




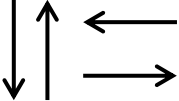
No	Simbol	Nama	Deskripsi
3.		Asosiasi / <i>association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
5.		Kebergantungan / <i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.

3. Activity diagram

Activity diagram atau disebut diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2. 4 Simbol - simbol *Activity Diagram*

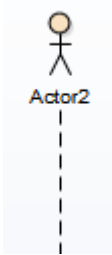




No	Simbol	Nama	Deskripsi
1.		<i>Activity/ Aktivitas</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2.		State	State dari sistem yang mencerminkan eksekusi dari suatu aksi.



No	Simbol	Nama	Deskripsi
3		<i>Initial node/Status awal</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity final node/ Status Akhir</i>	Objek dibentuk untuk diakhiri.
5		<i>Decesion</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
6		<i>Line connector</i>	Digunakan untuk menghubungkan suatu simbol dengan simbol lainnya.

4. *Sequence Diagram*

Sequence Diagram atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

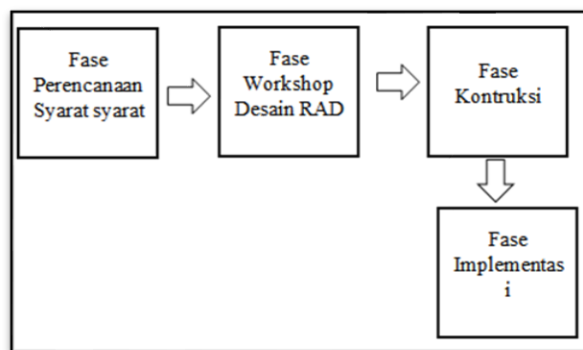
Tabel 2. 5 Simbol - simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat jalan itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		Garis hidup <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi
4		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	1 : keluaran 	Pesan tipe	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu

No	Simbol	Nama	Deskripsi
		return	kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
7		Objek	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
8	1 : masukan 	Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

2.6 Metode pengembangan sistem

Dalam mengembangkan sistem informasi Penilaian kinerja yang akan dilakukan pada peneliti ini yaitu menggunakan metode *Rapid And Development*



Gambar 2. 1 Metode RAD (*Rapid Application Development*) (kendal ,2003)

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat lunak.[15]

Perancangan sistem penilaian kinerja berbasis web ini menggunakan metode *Rapid Application Development (RAD)*, terdapat empat fase yang ada pada metode *RAD* dan pada tahap penilaian melibatkan penganalisis dan pengguna Empat Fase tersebut yaitu :

a. *Requirements Planing* (Perencanaan syarat syarat).

Fase ini merupakan fase pertemuan antara penganalisis dan pengguna untuk mengidentifikasi tujuan dari sistem yang akan di bangun serta mengidentifikasi syarat syarat informasi yang akan timbul untuk mencapai tujuan tersebut serta menganalisa semua sistem yang dibutuhkan oleh pengguna. Fase ini berorientasi pada penyelesaian masalah yang ada di Desa Kutamendala.

b. *Design Workshop (Whorkshop Desain RAD)*

Fase ini merupakan fase dalam bentuk workshop *desain RAD* antara penganalisis dan pemrograman untuk merancang sebuah sistem yang akan di bangun. Penganalisis dan pemrogram saling bekerja sama dalam membangun sistem dan menunjukkan representasinya dalam bentuk visual desain dan pola kerjanya kepada pengguna sistem. Pada fase ini juga pengguna merespon *prototype* yang telah dirancang. Penganalisis dan pemrogram dapat memperbaiki serta menganalisis modul modul yang dirancang berdasarkan dari respon pengguna sistem.

c. *Fase Construction* (Kontruksi)

Fase kontruksi merupakan fase eksekusi dalam pembuatan script program dan merupakan kelanjutan dari fase kedua. Pada fase ini juga menunjukkan *platform , hardware , dan software* yang digunakan. Setiap desain yang dibuat pada fase sebelumnya, akan di tingkatkan dengan menggunakan perangkat *RAD*. Setelah fungsi baru tersedia, fungsi baru tersebut ditunjukkan kepada pengguna untuk mendapatkan interaksi dan

revisi selanjutnya penganalisis akan melakukan perubahan dalam setiap desain aplikasi berdasarkan intruksi pengguna.

d. *Fase implementation* (implementasi)

Fase ini penganalisis bekerja sama dengan para pengguna secara intensif selama workshop berlangsung, dan merancang beberapa aspek dan nonteknis yang dibutuhkan. Segera setelah aspek aspek ini disetujui dan sistem sistem dibangun dan di saring, sistem sistem baru atau bagian dari sistem ujicoba dan kemudian diperkenalkan kepada organisasi.

2.7 Weight Sum Model (WSM)

Metode Weighted Sum Model merupakan metode yang sangat umum, dan banyak diterapkan untuk membantu pengambil keputusan dalam mengambil suatu keputusan. WSM merupakan salah satu metode yang paling sederhana dan mudah dipahami penerapannya.[16]

Merupakan bagian dalam metode MCDM(Multi-Criteria Decison Making) dalam mengevaluasi nilai pada setiap alternatif.

$$A_i^{WSM-score} = \sum_{j=1}^n w_j x_{ij} , for i = 1,2,3,...$$

Dimana:

n = jumlah kriteria

W_j = bobot dari setiap kriteria

X_{ij} = Nilai Matrik x

Penerapan metode Weighted Sum Model ini merupakan metode yang sangat sederhana dengan hanya beberapa langkah untuk dapat memberikan hasil Penilaian Kinerja Perangkat Desa. Keputusan yang dihasilkan dngan penerapan WSM bukan merupakan keputusan akhir/mutlak.

2.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum. .Singkat kata,

pengujian adalah aktivitas untuk menemukan dan menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya. Berikut merupakan penjelasan lebih rinci dari pengujian perangkat lunak baik itu teknik maupun metode yang digunakan didalam pengujian perangkat lunak.

2.8.1 Pengujian *black-box*

Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai kebutuhan fungsional yang telah di definisikan .[17] Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain input dan output program. Metode *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data akses basis data eksternal, kesalahan kerja, inisialisasi dan kesalahan terminasi.[18]

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Fungsi-fungsi yang tidak benar atau hilang
3. Kesalahan *interface*
4. Kesalahan dalam struktur data atau akses database eksternal
5. Kesalahan kerja
6. Inisialisasi dalam kesalahan terminasi.

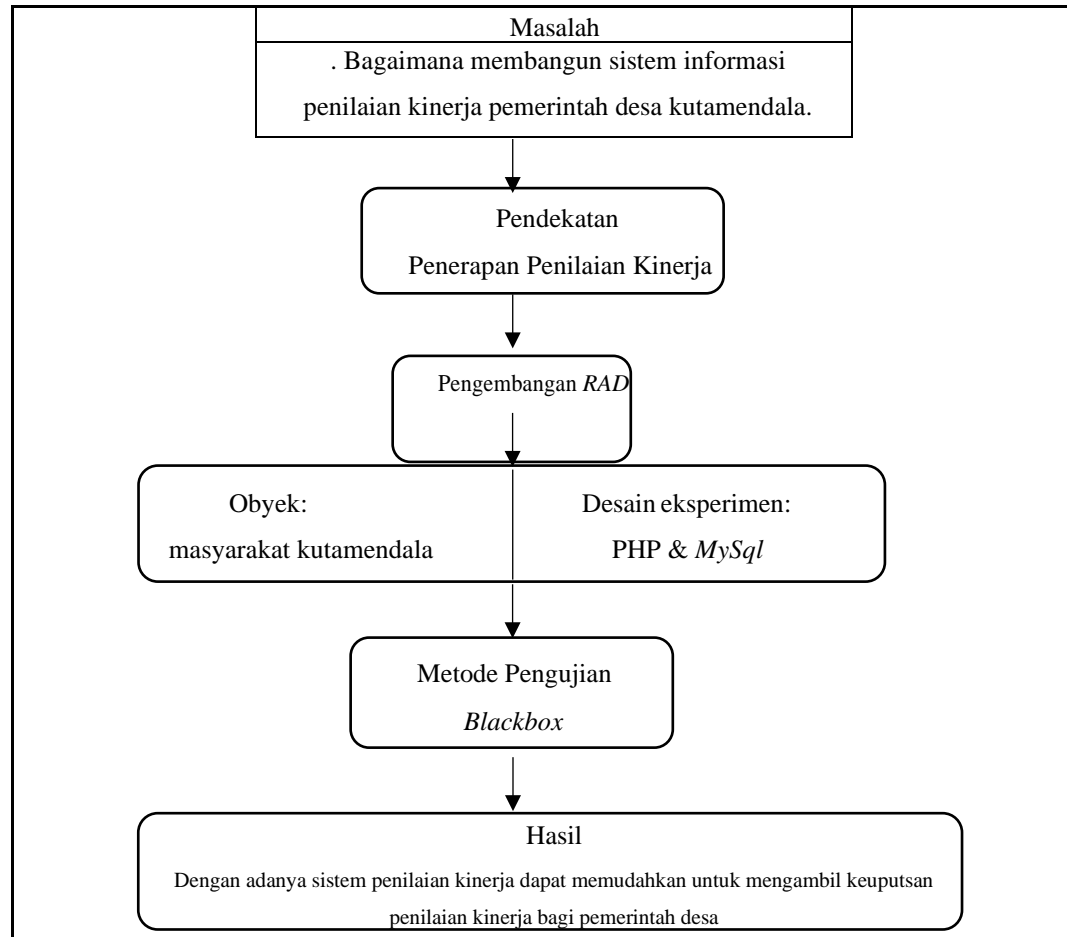
Dengan mengaplikasikan teknik *black-box*, maka user menarik serangkaian *test case* yang memenuhi kriteria berikut ini:

1. *Test case* yang mengurangi, dengan harga lebih dari satu, jumlah test case tambahan yang harus di desain untuk mencapai pengujian yang dapat di pertanggung jawabkan.
2. *Test case* memberitahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada

2.8.2 Pengujian UAT

Pengujian UAT adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa sistem yang dikembangkan dapat diterima atau tidaknya oleh pengguna, apabila hasil pengujian sudah bisa dianggap memenuhi kebutuhan dari pengguna maka aplikasi dapat diterapkan.[19] Pengujian dengan UAT dilakukan dengan mengajukan beberapa pertanyaan terhadap pegawai dan anggota perpustakaan yang bertindak sebagai pengguna, pengujian ini melibatkan 10 pengguna sistem. Hasil user acceptance test dinilai dengan 5 kategori, yaitu SS (Sangat Sesuai), S (Sesuai), C (Cukup), TS (Tidak Sesuai) Berikut ini rincian hasilnya.

2.9 Kerangka berfikir



Gambar 2. 2 Kerangka Berfikir