BAB II
TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Penerapan

Menurut Muhammad (2017:51), "Implementasi (penerapan) merupakan penyediaan sarana untuk melaksanakan sesuatu yang menimbulkan dampak atau akibat terhadap sesuatu".

Sedangkan menurut Browne dan Wildavsky yang dikutip oleh Firdianti Arinda (2018:19), "Implementasi (Penerapan) adalah perluasan aktivitas yang saling menyesuaikan."

2.1.2 Sistem Pendukung Keputusan

Menurut Asyriati, dkk. (2018:1) dalam buku Sistem Pendukung Keputusan Teori dan Implementasi. Sistem merupakan kumpulan sub-sub sistem (elemen) yang saling berkorelasi satu dengan yang lainnya untuk mencapai tujuan tertentu. Sebagai contoh: sebuah perusahaan memiliki sistem manajerial yang terdiri dari *bottom management*, *middle management*, dan *top manajemen* yang memiliki tujuan untuk mecapai kemajuan masyarakat. Sistem pendukung keputusan dapat diartikan sebagai suatu sistem yang di rancang yang digunakan untuk mendukung manajemen dalam pengambilan keputusan.

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada tahun 1971 oleh Scoot dalam Turban (2001) dengan istilah *Management Decision System*, kemudian sejumlah perusahaan, lembaga penelitian dan pergurusan tinggi mulai melakukan penelitian dan membangun Sistem Pendukung Keputusan, sehingga dari produksi yang dihasilkan dapat disimpulkan bahwa sistem ini merupakan suatu sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur.

2.1.3 *Simple Additive Weighting* (SAW)

Menurut Asyriati, dkk. (2018:21) "Metode *Simple Additive Weighting* merupakan pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan dengan mencari rating kinerja (skala prioritas) pada setiap alternatif di semua atribut."

Adapun algoritma penyelesaian metode ini adalah sebagai berikut :

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan di jadikan sebagai tolak ukur penyelesaian masalah.
2. Langkah 2 : Menormalisasi setiap nilai alternatif pada setiap atribut dengan cara menghitung nilai rating kinerja.
3. Langkah 3 : Menghitung nilai bobot prefensi pada setiap alternatif.
4. Langkah 4 : Melakukan perangkaian.

Berikut rumus yang digunakan pada metode *Simple Additive Weighting* yaitu :

1. Menormalisasikan setiap nilai alternatif pada setiap atribut dengan cara menghitung nilai rating kinerja

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\underset{i}{Max}} \; Jika\ j\ adalah\ atribut\ keuntungan\ \ (benefit) \\ \frac{\underset{i}{Min}\ X_{ij}}{X_{ij}} \; Jika\ j\ adalah\ atribut\ biaya\ (cost) \end{cases}$$

2. Menghitung nilai bobot prefensi pada setiap alternatif

$$V_i = \sum_{j=1}^{n} W_j\, r_{ij}$$

Keterangan :

V_i = Nilai Bobot Preferensi dari setiap alternative

$W_{j=}$ Nilai Bobot Kinerja

R_{ij}=Nilai Rating Kinerja

2.2 Teori Khusus

2.2.1 *Unified Modelling Language* (UML)

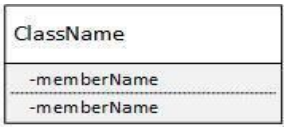
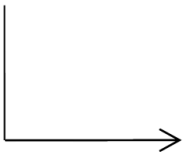
Menurut Sukamto dan Shalahuddin (2018:13), *Unified Modeling Language* (UML*)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

2.2.2 *Class Diagram*

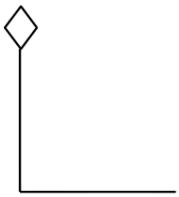
Menurut Sukamto dan Shalahuddin (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.1 Simbol *Class Diagram*

No	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem
2	Antarmuka/*interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
3	Asosiasi/*association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
4	Asosiasi berarah/ *directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus)
6	Kebergantungan/*dependensi* 	Relasi antar kelas dengan makna kebergantungan antar kelas

Lanjutan **Tabel 2.1**

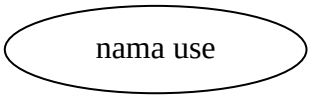
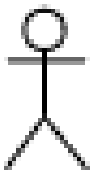
No	Simbol	Deskripsi
7	Agregasi/*aggregation* 	Relasi antar kelas dengan makna semua bagian (*whole-part*)

Sumber : Sukamto dan Shalahuddin (2018:141)

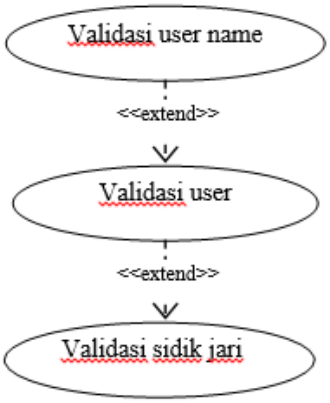
2.2.3 *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2018:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

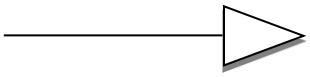
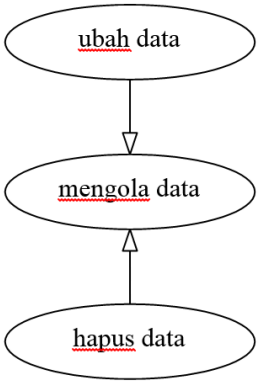
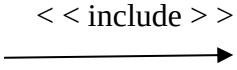
Tabel 2.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1	*Use case* 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anta unit atau aktor; biasanya dinyatakan dengan menggunkana kata kerja di awal fase nama *use case*
2	Aktor / *actor* 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

Lanjutan **Tabel 2.2**

No	Simbol	Deskripsi
3	Asosiasi / *association* _____	Komunikasi antara aktor dan *use case* yang berpatisipasi pada *use case* atau *use case* memiliki interkasi dengan aktor
4	Ekstensi / *extend* < < extend> > ⟶	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *ingeritance* pada pemograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misal :  Arah panah mengarah pada *use case* yang ditambahkan, biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.

Lanjutan **Tabel 2.2**

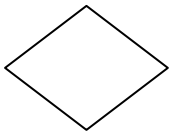
No	Simbol	Deskripsi
5	Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  arah panah mengarah pada *use case* yang menjadi generalisasinya (umum)
6	Menggunakan / *include* / *uses* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini Ada dua sudut pandang yang cukup besar mengenai *include* di *use case* : 1. *Include* berarti *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan, missal pada kasus berikut :

Sumber : Sukamto dan Shalahuddin (2018:155)

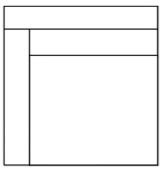
2.2.4 *Activity Diagram*

Menurut Sukamto dan Shalahuddin (2018:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.3 Simbol Activity Diagram

No	Simbol	Deskripsi
1	Start / status awal (*Intial State)* 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	End / status akhir (*final state)* 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan **Tabel 2.4**

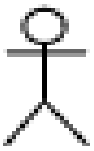
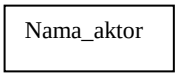
No	Simbol	Deskripsi
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggunga jawab terhadap aktivitas yang terjadi.

Sumber : Sukamto dan Shalahuddin (2018:161)

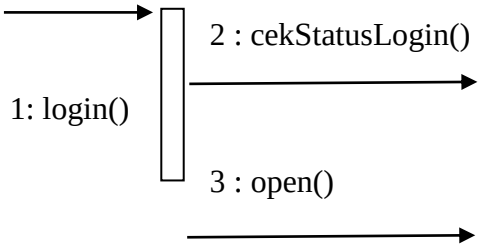
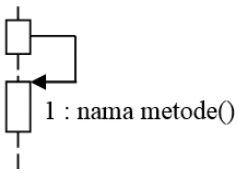
2.2.5 *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *massage* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada diagram sekuen :

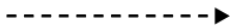
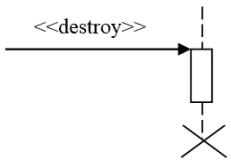
Tabel 2.4 Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor  atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
2	Garis hidup/*lifeline*	Menyatakan kehidupan suatu objek.

Lanjutan **Tabel 2.4**

No	Simbol	Deskripsi
3	Objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubungdengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif
5	Pesan tipe *create* <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6	Pesan tipe *call* 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, 

Lanjutan **Tabel 2.4**

No	Simbol	Deskripsi
		Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7	Pesan tipe *send* 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe *return* 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada *create* maka ada *destroy*

Sumber : Sukamto dan Shalahuddin (2018:165)

2.3 Teori Program

2.3.1 Pengertian *CodeIgniter*

Menurut Budiyanto dan Yosef Murya (2018), *CodeIgniter* dapat disebut sebagai *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP, dengan kerangka kerja sehingga menjadi sistematis.

2.3.2 *Hypertext Processor* (PHP)

Menurut Yudhanto dan Prasetyo (2019:9) "PHP atau *Hypertext Processor* adalah Bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web."

Sedangkan menurut Heru (2019:5) "PHP (*Hypertext Processor*) adalah Bahasa pemrograman yang digunakan untuk membuat *website* atau situs dinamis dan menangani rangkaian Bahasa pemrograman antara *client side scripting* dan *server side scripting*."

Dari pengertian-pengertian diatas, dapat disimpulkan PHP atau *Hypertext Processor* adalah Bahasa pemrograman yang digunakan untuk pemrograman *script server side*.

2.3.3 *Mysql*

Menurut Sumarlinda (2015:20) "MySQL adalah *multiuser database* yang menggunakan bahasa *structured query language* (SQL), MySql dalam operasi *client-server* melibatkan *server daemon* MySql disisi *server* dan berbagai macam program serta *library* yang berjalan disisi client.

Menurut Sidik (2017:301) "MySQL merupakan *software database* yang termasuk paling populer dilingkungan *Linux*, kepopuleran ini karena ditunjang karena performasi *query* dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah. MySQL telah tersedia juga dilingkungan *Windows*".

Berdasarkan pengertian diatas, bahwa dapat disimpulkan MySQL adalah sebuah database yang digunakan untuk menyimpan data dalam tabel terpisah, mysql dapat berjalan stabil pada berbagai sistem operasi seperti windows,linux dan lainnya serta dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan.

2.4 Penelitian Terdahulu

Menurut penelitian yang dilakukan oleh Tanto (2018) telah ditemukan sebuah Perancangan Sistem Pendukung Keputusan Pemberian Kredit Pemilihan Rumah (KPR) menggunakan metode *Simple Additive Weighting*. Metode ini dipilih karena merupakan metode yang sederhana secara perhitungan (komputasinya sederhana). Metode *Simple Additive Weighting* merupakan metode yang paling banyak digunakan dalam memecahkan permasalahan, seperti dalam SPK penentuan kelayakan nasabah penerima KPR.

Ishak, dkk (2017) melakukan sebuah penelitian yang menyebutkan, Sistem Pendukung Keputusan kelayakan sertifikasi guru menggunakan metode *Simple Additive Weighting* (SAW). Sehingga dengan penerapan metode *Simple Additive Weighting* didalam sistem pendukung keputusan dalam menentukan proses kelayakan guru yang layak disertifikasi diharapkan dapat bekerjasama dengan instansi pendidikan tinggi yang kompeten, yang diakhiri dengan pemberian sertifikat pendidik kepada guru yang telah dinyatakan memenuhi standar profesional.

Dalam penelitian yang dilakukan oleh Muslihudin (2017) mengatakan bahwa Sistem Pendukung Keputusan menentukan kelayakan penerimaan bantuan pengusaha ayam petelur oleh dinas peternakan kabupaten pesawaran menggunakan metode *Simple Additive Weighting*. Guna menentukan penerimaan bantuan yang tepat untuk di jadikan modal usaha peternakan pada kabupaten pesawaran, maka dibuatlah sebuah sistem penunjang pengambilan keputusan sebagai masukan bagi Pengusaha ayam petelur. SAW mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut.

Parida (2017) melakukan penelitian yang menyatakan penerapan metode *simple additive weight* (SAW) dan AHP dalam sistem pendukung keputusan penentuan penilaian karyawan berprestasi. diharapkan dapat membantu dalam pengambilan keputusan yang tepat dan cermat untuk menentukan penilaian karyawan berprestasi.

Dalam penelitian yang dibuat oleh Fahmi, dkk (2018) mengatakan bahwa Pemberian beasiswa khususnya universitas sudah banyak ditawarkan kepada

mahasiswa, beasiswa yang diberikan yaitu berupa beasiswa PPA dan BBM. hasil studi penerapan metode saw dalam menentukan penerima beasiswa DSS baru ada beberapa perbedaan dalam hasil urutan-urutan kandidat dan perbedaan waktu pelaksanaan masing-masing metode. perbedaan urutan peringkat metode ini adalah karena efek dari nilai alternatif, kriteria bobot, dan metode perhitungan.