



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Penerapan

Menurut Ali 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 Produksi

Menurut Sattar (2017:209) secara umum produksi diartikan sebagai suatu kegiatan atau proses yang mentransformasikan masukan (*input*) menjadi keluaran (*output*). Dalam pengertian umum inilah sekarang berkembang istilah industri, seperti industri manufaktur, industri pengolahan hasil-hasil pertanian atau agroindustri, industri pengolahan hasil-hasil pertambangan, dan industri angkutan.

Dalam arti sempit, pengertian produksi hanya dimaksudkan sebagai kegiatan yang menghasilkan barang, baik barang jadi atau barang setengah jadi, barang industri, suku cadang (*sparepart*) maupun komponen-komponen penunjang.

2.1.3 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 mencapai 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 perguruan 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.

2.1.4 Metode *Weighted Product* (WP)

Menurut Sari. (2017:57) “Metode *Weighted Product* (WP) merupakan bagian dari konsep *Multi Criteria Decision Making* (MCDM), merupakan teknik pengambilan keputusan dari beberapa pilihan alternatif yang ada. Metode ini memerlukan proses normalisasi pada perhitungannya. Dengan menggunakan metode *Weighted Product*, diharapkan dapat dikembangkan *software* sistem pendukung keputusan yang dapat digunakan oleh instansi, yang akan menjadi alternatif pemilihan dan memberikan nilai bobot pada perbandingan alternatif dan kriterianya. Metode ini mengevaluasi beberapa alternatif terhadap sekumpulan atribut atau kriteria, dimana setiap atribut saling tidak bergantung satu dengan yang lainnya.”

Menurut Lita Asyriati Latif, dkk. (2018:26) “Metode *Weighted Product* (WP) merupakan salah satu metode yang sederhana dengan perkalian untuk menghubungkan rating atribut dimana setiap atribut harus dipangkatkan dengan bobot atribut yang bersangkutan.”

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 (nilai vector).
3. Langkah 3 : Menghitung nilai bobot prefensi pada setiap alternatif.
4. Langkah 4 : Melakukan perangkaian.



Berikut rumus yang digunakan pada metode *Weighted Product* yaitu :

1. Normalisasi bobot menggunakan rumus

$$W_j = \frac{w_j}{\sum w_j}$$

2. Menentukan nilai vektor S_i menggunakan rumus

$$S_i = \prod_{j=1}^n x_{ij} w_j$$

3. Menghitung nilai prefensi V_i untuk setiap alternatif menggunakan rumus

$$V_i = \frac{S_1}{S_1 + S_2 + S_3} \text{ atau } V_i = \frac{\prod_{j=1}^n x_{ij} w_j}{\prod_{j=1}^n (x_{j*}) w_j}$$

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.

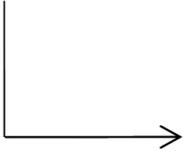
2.2.2 *Class Diagram*

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* :

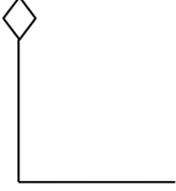
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



Tabel 2.1 Simbol Class Diagram

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

Lanjutan **Tabel 2.1**

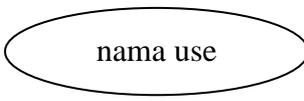
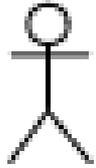
No	Simbol	Deskripsi
7	Agregasi/aggregation 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber : Sukamto dan Shalahuddin (2018:141)

2.2.3 Use Case Diagram

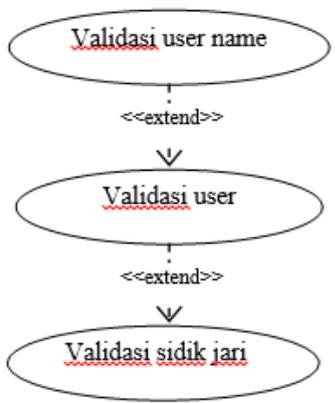
Menurut Sukamto dan Shalahuddin (2018:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* 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	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i>
2	Aktor / <i>actor</i> 	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.

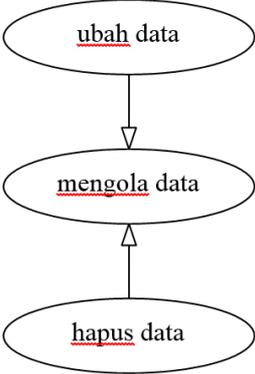
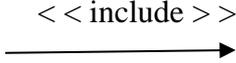


Lanjutan Tabel 2.2

No	Simbol	Deskripsi
3	Asosiasi / <i>association</i> —————	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interkasi dengan aktor
4	Ekstensi / <i>extend</i> << extend >> —————→	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal :</p>  <pre> graph TD A([Validasi user name]) -.-> <<extend>> B([Validasi user]) B -.-> <<extend>> C([Validasi sidik jari]) </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>



Lanjutan Tabel 2.2

No	Simbol	Deskripsi
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 umum dari lainnya, misalnya :  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
6	Menggunakan / <i>include</i> / <i>uses</i> 	Relasi <i>use case</i> tambahan ke 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 Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :

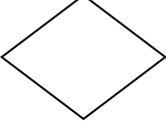
Sumber : Sukamto dan Shalahuddin (2018:155)



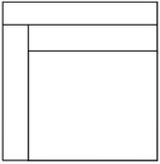
2.2.4 Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah 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 (<i>Initial State</i>) 	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/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	End / status akhir (<i>final state</i>) 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan **Tabel 2.3**

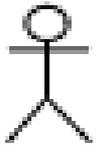
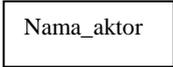
No	Simbol	Deskripsi
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Sukamto dan Shalahuddin (2018:161)

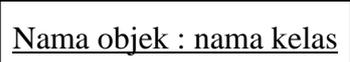
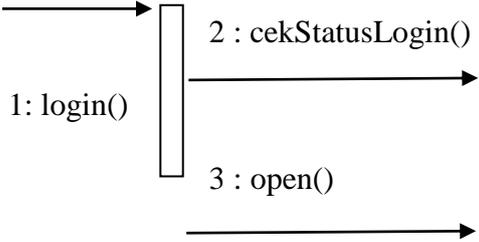
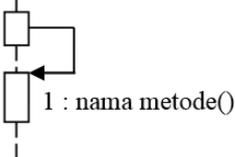
2.2.5 Sequence Diagram

Menurut Sukamto dan Shalahuddin (2018:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* 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

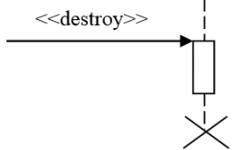
No	Simbol	Deskripsi
1	Aktor  atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
2	Garis hidup/ <i>lifeline</i>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

Lanjutan **Tabel 2.4**

No	Simbol	Deskripsi
3	Objek 	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>
5	Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6	Pesan tipe <i>call</i> 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 <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe <i>return</i> 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 <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i>

Sumber : Sukamto dan Shalahuddin (2018:165)

2.3 Teori Program

2.3.1 Hypertext Processor (PHP)

Menurut Yudhanto, dkk. (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*.

2.3.2 *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* berjalan stabil pada berbagai sistem operasi seperti *windows,linux* dan lainnya serta dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan.

2.3.3 *XAMPP*

Menurut Haqi, dkk. (2019:8) *XAMPP* adalah perangkat lunak bebas (*free software*) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi *XAMPP* sendiri sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari beberapa program, antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemahan yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.



Sedangkan menurut Aryanto (2016:4) *XAMPP* merupakan sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman seperti; *Apache HTTP Server*, *MySQL database*, bahasa pemrograman *PHP* dan *Perl*.

Dari pengertian diatas, dapat disimpulkan *XAMPP* merupakan sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman.

2.3.4 HTML (Hyper Text Markup Language)

Menurut Rintho (2018:18) *HTML (Hyper Text Markup Language)* adalah sebuah *text* biasa dapat berfungsi lain, kita dapat membuatnya menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik *text* tersebut.

Menurut Jubilee (2016:7) *HTML (Hyper Text Markup Language)* disebut *Hypertext* karena di dalam *script HTML* anda bisa membuat agar sebuah teks menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik teks tersebut.

Dari pengertian diatas, dapat disimpulkan *HTML* merupakan sebuah teks biasa yang bisa dibuat menjadi *link* untuk berpindah halaman.

2.4 Penelitian Terdahulu

Menurut penelitian Sucipto (2017) telah ditemukan sebuah sistem informasi dengan metode *Weighted Product (WP)* untuk memfasilitasi mahasiswa dalam merekomendasikan dosen pembimbing sesuai dengan kebutuhannya. Pada proses rekomendasi dosen pembimbing praktek kerja lapangan tersebut dilakukan dengan penilaian lima variabel. Variabel yang diperoleh berasal dari penilaian kinerja dosen.

Marisa, dkk. (2016) melakukan sebuah penelitian yang menyebutkan, metode *Weighted Product (WP)* dapat menyelesaikan masalah dengan cara perkalian untuk menghubungkan *rating atributte* bobot yang bersangkutan. Hasil



dari penelitian ini berupa aplikasi sistem pemilihan *smartphone android* berbasis *web* yang dapat memberikan rekomendasi pada *user* sebagai bahan pertimbangan.

Dalam penelitian Firdyana, dkk. (2016) mengatakan bahwa metode *Weighted Product* (WP) lebih efisien karena waktu yang dibutuhkan dalam penelitian lebih singkat. Metode ini dipilih karena dapat menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perengkingan yang akan menghasilkan pilihan penerima bantuan raskin yang sesuai dengan kriteria.

Anwar, dkk. (2017) melakukan sebuah penelitian yang menyebutkan, metode *Weighted Product* (WP) metode ini memberikan nilai bobot setiap kriteria dan selanjutnya dilakukan perengkingan. Dalam penelitian ini ada dua program bantuan yang pertama SWT parameter yang digunakan yaitu pekerjaan, penghasilan, kendaraan, harga kendaraan, dan jumlah anak, yang kedua lansia parameternya yaitu umur, kesehatan, tempat tinggal, dan kondisi hunian.

Dalam penelitian Fajarianto, dkk. (2017) mengatakan bahwa penelitian bertujuan untuk memberikan penilaian seleksi calon karyawan baru masih dilakukan secara manual sehingga semua data seleksi calon karyawan baru tidak memiliki bobot yang tetap dan sering terjadi kesalahan sehingga ada calon karyawan yang tidak memenuhi standar proses seleksi. Untuk itu dalam penelitian ini penulis menerapkan metode *Weighted Product* (WP), karena metode *Weighted Product* (WP) adalah metode penyelesaian masalah dengan menggunakan perkalian untuk menghubungkan *rating* atribut, dimana *rating* harus dipangkatkan terlebih dahulu.