



BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

2.1.1. Sistem Pendukung Keputusan

Menurut Nofriansyah dan Sarjon (2017:2), “Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditunjukkan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur.”

Sedangkan menurut Little dalam Nofriansyah dan Sarjon (2017:1), “Sistem pendukung keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model.”

Dari berbagai pengertian di atas, dapat disimpulkan bahwa Sistem Pendukung Keputusan (*Decision Support Systems*) merupakan teknik dalam pengambilan keputusan yang berbasis komputer, baik untuk individu maupun kelompok. Dalam teorinya memiliki kriteria yang memiliki nilai-nilai atau bobot yang harus dimiliki oleh setiap alternatif, dimana sistem ini memberikan pilihan pada pengambilan keputusan yang lebih baik dan lebih konsisten dan lebih cepat.

2.1.2. Komponen Sistem Pendukung Keputusan

Komponen-komponen sistem pendukung keputusan (Latif, dkk., 2018:4) terdiri dari:

a. *Data Management*

Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).



b. *Model Management*

Melibatkan model finansial, statistik, *management science* atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.

c. *Communication (dialog subsystem)*

User dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.

d. *Knowledge Management*

Subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.1.3. Tahapan Sistem Pendukung Keputusan

Menurut Latif, dkk. (2018:5), ada 3 (tiga) tahapan dalam proses pengambilan keputusan. Diantaranya sebagai berikut:

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendekteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.



2.1.4. Tujuan Pemecahan Masalah Sistem Pendukung Keputusan

Menurut Latif, dkk. (2018:3), tujuan pemecahan masalah di bidang sistem pendukung keputusan, yakni:

1. Membantu manajer membuat keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung penilaian manajer bukan mencoba untuk menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan manajer daripada efisiensinya.

2.1.5. Elemen dalam Sistem Pendukung Keputusan

Menurut Latif, dkk. (2018:6), secara konsep ada 3 (tiga) elemen yang terkait dengan sistem pendukung keputusan, yakni:

1. Masalah, dalam sistem pendukung keputusan terdapat beberapa jenis masalah, seperti masalah terstruktur, masalah semi terstruktur dan masalah tidak terstruktur.
2. Solusi, dalam sebuah sistem pendukung terdapat beberapa jenis solusi pemecahan masalah diantaranya adalah *Multi Attribute Decision Making* (MADM), Metode *Simple Additive Weighting* (SAW), Metode *Weight Product* (WP), metode *Analytical Hierarchy Process* (AHP), Metode Topsis dan lainnya. Kemudian Metode *Multi Criteria Decision Making* (MCDM), seperti Metode *Promethee*, Metode *Electre*, Metode *Oreste*, Metode Entropi dan lainnya. Selain itu, terdapat juga Metode *Multi Factor Evaluation Process* (MFEP), Metode *Multi Attribute Utility Theory* (MAUT) serta Metode *Fuzzy Multi Attribute Decision Making* (FMADM) yang terdiri dari F-AHP, F-SAW dan lainnya.
3. Hasil, sistem pendukung keputusan akan menghasilkan sebuah keluaran berupa keputusan yang dapat dijadikan sebagai tolak ukur kebijakan dari sebuah masalah yang diteliti atau dibahas. Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah tersebut. Tindakan memilih strategi yang akan memberikan solusi terbaik bagi para manajer disebut dengan pengambilan keputusan.



2.1.6. Metode *Simple Additive Weighting* (SAW)

Menurut Nofriansyah dan Sarjon (2017:33), “Metode *Simple Additive Weighting* (SAW) dapat diartikan sebagai metode pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep metode ini adalah dengan mencari *rating* kerja (skala prioritas) pada setiap alternatif di semua atribut.”

Berdasarkan pengertian di atas, dapat disimpulkan bahwa metode *Simple Additive Weighting* (SAW) merupakan sebuah metode pembobotan dari banyaknya kriteria yang menyediakan berbagai alternatif yang ada. Kemudian dari berbagai alternatif tersebut akan memperlihatkan alternatif yang paling mendominasi dari pembobotan tersebut.

Adapun algoritma penyelesaian metode ini adalah sebagai berikut:

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan 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 preferensi pada setiap alternatif.
4. Langkah 4 : Melakukan pemeringkatan.

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

1. Menormalisasikan setiap alternatif (menghitung nilai *rating* kinerja).

$$R_{ij} = \left\{ \begin{array}{ll} \frac{X_{ij}}{\max_i X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i X_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{array} \right\}$$

2. Menghitung nilai bobot preferensi pada setiap alternatif.

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



Keterangan :

V_i = Nilai Bobot Preferensi dari setiap alternatif

W_j = Nilai Bobot Kriteria

R_{ij} = Nilai *Rating* Kinerja

2.2. Teori Khusus

2.2.1. Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin (2013:137), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Sehingga penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan untuk metodologi berorientasi objek.

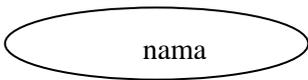
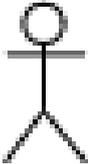
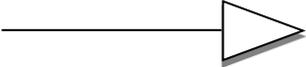
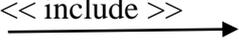
2.2.2. Use Case Diagram

Menurut Sukamto dan Shalahuddin (2013:155), “*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. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam suatu sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tabel 2.1. Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar



		<p>pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i>.</p>
2.	<p>Aktor / <i>actor</i></p> 	<p>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; biasanya dinyatakan menggunakan kata benda di awal fase nama aktor</p>
3.	<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
4.	<p>Ekstensi / <i>extend</i></p> <p><< extend >></p> 	<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 pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i></p>
5.	<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><< include >></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

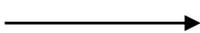
(Sumber: Sukanto dan Shalahuddin, 2013:155)



2.2.3. Activity Diagram

Menurut Sukamto dan Shalahuddin (2013:161), “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.

Tabel 2.2. Simbol *Activity Diagram*

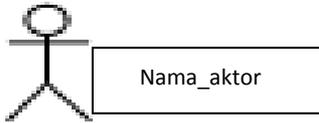
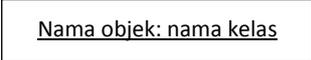
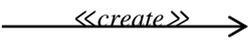
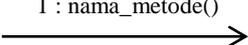
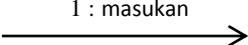
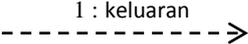
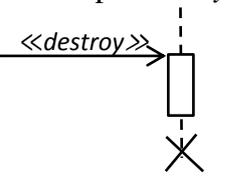
No	Simbol	Deskripsi
1.		<i>Start</i> atau <i>intial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup
2.		<i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem.
3.		<i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin.
4.		Sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu yang memicu perubahan tersebut.

(Sumber: Sukamto dan Shalahuddin, 2013:161)

2.2.4. Sequence Diagram

Menurut Sukamto dan Shalahuddin (2013:165), “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstantiasi menjadi objek itu dan juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2.3. Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	<p>Aktor</p>  <p>atau</p>	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 gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2	<p>Garis Hidup/<i>Lifeline</i></p> 	Menyatakan kehidupan suatu objek
3	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
4	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5	<p>Pesan tipe <i>create</i></p> 	Menyatakan sebuah objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat
6	<p>Pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7	<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8	<p>Pesan tipe <i>return</i></p> 	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	<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>



(Sumber: Sukamto dan Shalahuddin, 2013:165)

2.2.5. Class Diagram

Menurut Sukamto dan Shalahuddin (2013:143), “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 metode atau operasi.”

Tabel 2.4. 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 antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang saat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

(Sumber: Sukamto dan Shalahuddin, 2013:143)



2.3. Teori Program

2.3.1. PHP

Menurut Winarno, dkk. (2014:49), “PHP atau PHP *Hypertext Preprocessor* adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode PHP dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*)”.

Dari pengertian di atas dapat disimpulkan bahwa PHP merupakan sebuah bahasa pemrograman yang digunakan untuk membuat aplikasi *web*.

2.3.2. Database

Menurut Jubilee (2017:1), “*Database* adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap *database* mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari dan menyalin data yang ada di dalamnya.”

Menurut Pamungkas (2017:2), “Basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *database* atau basis data merupakan sekumpulan informasi yang disimpan di dalam sebuah komputer yang tersusun secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk mendapatkan informasi dari basis data tersebut.

2.3.3. MySQL

Menurut Jubilee (2017:3), “MySQL adalah *Relational Database Management System* (RDBMS) yang cepat dan mudah digunakan, serta sudah banyak dipakai untuk berbagai kebutuhan. MySQL dikembangkan oleh MySQL AB Swedia. Hampir sebagian besar aplikasi *website* yang ada di internet dikembangkan menggunakan MySQL dan bahasa pemrograman lainnya, seperti PHP.”



Menurut Mundzir (2018:217) dalam buku *Pemrograman Web Seri PHP* mengatakan bahwa MySQL adalah sistem manajemen *database* SQL yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database* MySQL mampu mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system* (DBMS).

Berdasarkan pengertian di atas, dapat disimpulkan bahwa MySQL merupakan sebuah *software* sistem manajemen basis data SQL yang berfungsi untuk menampung berbagai informasi di dalam suatu program komputer.

2.3.4. XAMPP

Menurut Jubilee (2017:94), “XAMPP merupakan *server* yang paling banyak digunakan dan diminati karena selain dapat untuk mengubah komputer menjadi *MySQL Server*, *module Apache*-nya juga dapat dipakai untuk membantu pemrograman PHP.”

Berdasarkan pengertian di atas, dapat disimpulkan bahwa XAMPP merupakan perangkat lunak (*software*) bebas yang mendukung banyak sistem operasi. Fungsinya sendiri adalah sebagai penyedia (*server*) yang berdiri sendiri, yang terdiri dari berbagai program seperti MySQL.

2.4. Referensi Penelitian Sebelumnya

Berdasarkan penelitian yang telah dilakukan oleh Andreas, dkk. pada tahun 2017 dengan judul Penerapan Algoritma *Simple Additive Weighting* untuk Membantu dalam Menentukan Calon Penerima Bantuan Pangan Non Tunai (BPNT) dengan ISSN : 2356-4393, konsep dasar dari metode SAW adalah mencari penjumlahan terbobot dari *rating* kinerja pada setiap alternatif dari semua atribut, metode SAW membutuhkan normalisasi ke skala yang dapat diperbandingkan dengan semua *rating* alternatif yang ada. Kriteria yang digunakan seperti luas lahan, dinding, lantai, penghasilan dan aset. Untuk perancangan analisis menggunakan metode *waterfall* yang memiliki tahapan seperti *requirement analysis*, *system design*, *implementation*, *testing*, *deployment* dan *maintenance*. Hasil dari pengujian yang dilakukan melalui aplikasi adalah



menampilkan hasil penerimaan BPNT dengan menggunakan data yang sama. Sehingga dapat disimpulkan bahwa sistem pendukung keputusan ini layak dijadikan sebagai aplikasi rekomendasi calon penerima BPNT.

Menurut penelitian yang telah dilakukan oleh Pahu, dkk. pada tahun 2018 yang berjudul Sistem Pendukung Keputusan Menentukan Calon Penerima Raskin Menggunakan Metode *Simple Additive Weighting* dengan ISSN : 2615-224X, ditemukan bahwa pembobotan kriteria mempengaruhi hasil perhitungan. Dengan pekerjaan, penghasilan, jumlah tanggungan, luas bangunan, kondisi rumah, sinetasi rumah dan aliran listrik yang menjadi kriteria, didapatkan hasil akhir melalui metode pembobotan SAW dengan penerapan metode *database* dan pemrograman *web* dalam menentukan pendataan calon penerima raskin.

Berdasarkan penelitian pada tahun 2018 yang berjudul Sistem Pendukung Keputusan Pemilihan Penerima Bantuan Perbaikan Rumah Menggunakan Metode *Simple Additive Weighting* (SAW) Studi Kasus Kelurahan Tambelan Sampit Kota Pontianak dengan ISSN : 2302-8149 dan E-ISSN : 2540-9719 oleh Salim. Penerimaan bantuan perbaikan ini memerlukan 13 kriteria, diantaranya adalah material dinding, kondisi dinding, material atap, material lantai, luas rumah hingga jumlah penghuni rumah. Bahasa Pemrograman yang digunakan adalah *JavaScript* (JS) dan *Hypertext Preprocessor* (PHP). PHP sendiri merupakan bahasa *script server-side* yang digunakan dalam pengembangan *web* dengan cara disisipkan pada dokumen HTML. *Web* dapat dibuat dinamis menggunakan PHP, sehingga perbaikan *web* tersebut menjadi lebih mudah dikerjakan dan lebih efisien. Sistem ini menggunakan *database* MySQL untuk penyimpanan data.

Menurut jurnal penelitian dengan judul Sistem Pendukung Keputusan Penilaian Kinerja Karyawan PT. Harjamukti Jaya Mandiri Menggunakan Metode *Simple Additive Weighting* pada tahun 2018 oleh Sokibi dan Apriyanto dengan ISSN : 2407-4322 dan E-ISSN : 2503-2933. Didapatlah nama karyawan dengan ranking teratas yang berhak menjadi karyawan yang terbaik setelah melalui proses perhitungan dengan pembobotan kriteria dengan menggunakan metode SAW. Adapun kriterianya adalah kedisiplinan kerja, pendidikan terakhir,



pengalaman kerja, kerjasama, dan keaktifan. Dengan menggunakan sistem pendukung keputusan penilaian kinerja karyawan dengan metode SAW dapat memudahkan *top manager* dalam menghitung penilaian para karyawan.

Berdasarkan jurnal penelitian yang dibuat oleh Rinianty dan Sukardi pada tahun 2018 dengan ISSN : 1978-8282 yang berjudul Sistem Pendukung Keputusan Penerima Karyawan Menggunakan Metode SAW pada CV. Green Advertising yang memberikan kriteria berupa pendidikan terakhir, pengalaman kerja, keahlian, dan kelengkapan berkas yang diberi bobot tersendiri dengan perhitungan menggunakan metode SAW, didapatkan nilai yang tertera berupa nilai akhir dari alternatif karyawan yang memiliki ranking teratas. Pengembangan aplikasi ini menggunakan bahasa pemrograman PHP dan MySQL.