

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Belipati, dkk (2021) mengenai Penerapan Metode *Dempster Shafer* untuk Menganalisis Kepuasan Mahasiswa Universitas San Pedro, penelitian Belipati, dkk menggunakan metode *Dempster Shafer* untuk menganalisis kepuasan mahasiswa, sedangkan penelitian ini menggunakan metode *Dempster Shafer* untuk mengetahui tingkat belajar.

Berdasarkan penelitian yang dilakukan oleh Hairani, dkk (2021) mengenai Metode *Dempster Shafer* untuk Diagnosis Dini Jenis Penyakit Gangguan Jiwa Skizofrenia Berbasis Sistem Pakar, penelitian Hairani, dkk menggunakan metode *Dempster Shafer* untuk diagnosa dini jenis penyakit gangguan jiwa skizofrenia, sedangkan penelitian ini menggunakan metode *Dempster Shafer* untuk mengetahui tingkat belajar murid.

Berdasarkan penelitian yang dilakukan oleh Hastari dan Bimantoro (2018) mengenai Sistem Pakar untuk Mendiagnosis Gangguan Mental Anak Menggunakan Metode *Dempster Shafer*, penelitian Hastari dan Bimantoro menggunakan metode *Dempster Shafer* untuk diagnosa gangguan mental anak, sedangkan penelitian ini menggunakan metode *Dempster Shafer* untuk mengetahui tingkat belajar.

Berdasarkan penelitian yang dilakukan oleh Rahmadhani, dkk (2020) mengenai Sistem Pakar Deteksi Dini Kesehatan Mental Menggunakan Metode

Dempster Shafer, penelitian Rahmadhani, dkk menggunakan metode *Dempster Shafer* untuk deteksi dini kesehatan mental, sedangkan penelitian ini menggunakan metode *Dempster Shafer* untuk mengetahui tingkat belajar murid.

Berdasarkan penelitian yang dilakukan oleh Sinaga (2021) mengenai Sistem Pakar Tes Kepribadian Menggunakan Metode *Dempster Shafer*, penelitian Sinaga menggunakan metode *Dempster Shafer* untuk tes kepribadian, sedangkan penelitian ini menggunakan metode *Dempster Shafer* untuk mengetahui tingkat belajar murid.

II.2. Landasan Teori

Landasan teori dikutip dari beberapa jurnal penelitian terdahulu yang berkaitan dengan penelitian ini.

II.2.1. Penerapan

Penerapan merupakan proses penterjemahan dari pemodelan menjadi bentuk aplikasi sesuai dengan kebutuhan informasi pengguna. Proses implementasi adalah proses penterjemahan dari pemodelan ke dalam pengkodean atau pembentukan antarmuka. (Oktaviani dan Sauda, 2019 : 182).

Penerapan (implementasi) merupakan suatu proses penerapan ide, konsep, kebijakan atau inovasi dalam suatu tindakan praktis sehingga memberikan dampak baik berupa perubahan pengetahuan, keterampilan maupun nilai dan sikap. implementasi diartikan sebagai pelaksanaan atau penerapan. Implementasi merupakan aktivitas yang saling menyesuaikan. Implementasi merupakan sistem rekayasa. Pengertian-pengertian tersebut memperlihatkan bahwa kata implementasi

bermuara pada aktivitas, adanya aksi, tindakan atau mekanisme suatu sistem. Ungkapan mekanisme mengandung arti bahwa implementasi bukan sekedar aktivitas tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan

II.2.2. Sistem Pakar

Sistem pakar adalah salah satu cabang dari *Artificial Intelligence* yang membuat penggunaan secara *knowledge* (pengetahuan) yang khusus untuk menyelesaikan masalah tingkat manusia yang pakar (ahli). Seorang pakar adalah orang memiliki keahlian dalam bidang tertentu, yaitu pakar yang mempunyai pengetahuan atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. (Sinaga, 2021 : 169).

Sistem pakar merupakan pemindahan pengetahuan ahli ke komputer untuk menyelesaikan permasalahan tertentu. Permasalahan yang dituntaskan dapat menggunakan sistem pakar misalnya diagnosa penyakit, deteksi kerusakan dan diagnosa kejiwaan. (Hairani, dkk, 2021 : 281).

II.2.3. Metode Dempster Shafer

Metode *dempster shafer* adalah suatu teori matematika untuk pembuktian berdasarkan *belief functions* (fungsi kepercayaan) dan *plausible reasoning* (pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval :

[*Belief, Plausibility*]

Belief (Bel) merupakan ukuran kepercayaan *evidence* untuk mendukung suatu himpunan. Nilai *Belief* diberikan oleh pakar berdasarkan ilmu pengetahuan pakar terhadap jenis *evidence*. Nilai Bel ini berada dalam kisaran $[0 \dots 1]$, Jika nilai $Bel = 0$ artinya tidak ada *evidence* dan $Bel = 1$ artinya kepastian. Fungsi belief dapat diformulasikan dan ditunjukkan pada persamaan (1).

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \dots \dots \dots (1)$$

Plausibility (Plau) juga bernilai 0 sampai 1, jika kita yakin terhadap X, maka dapat dikatakan bahwa $Bel(X)=1$, dan $Pls(X)=0$. *Plausibility* mengurangi tingkat kepercayaan dari *evidence*. *Plausibility* (Pls) dinotasikan pada persamaan (2).

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \subseteq X} m(Y) \dots \dots \dots (2)$$

Dimana :

$$Bel(X) = Belief(X).$$

$$Pls(X) = Plausibility(X).$$

$$m(Y) = mass\ function.$$

Pada teori *Dempster-Shafer* adanya *frame of discernment* yang dinotasikan dengan θ dan *mass function* yang dinotasikan dengan m . *frame of discernment* adalah semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment*. Sedangkan *mass function* (m) dalam teori *Dempster Shafer* adalah tingkat kepercayaan dari suatu *evidence* (instrumen kesehatan mental), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m). Pada sistem ini, terdapat sejumlah *evidence* (instrumen kesehatan mental) yang akan digunakan psada faktor ketidakpastian dalam menentukan hasil deteksi dini. Untuk

mengatasi sejumlah *evidence* tersebut pada teori *Dempster Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination* pada persamaan (3).

$$m_z(Z) = \frac{\sum_{X \cap Y = Z} m_x(X) \cdot m_y(Y)}{1 - K} \dots \dots \dots (3)$$

Dimana :

$m_z(Z)$ = *mass function* dari *evidence* z.

$m_x(X)$ = *mass function* dari *evidence* x.

$m_y(Y)$ = *mass function* dari *evidence* y.

$\sum_{X \cap Y = Z} m_x(X)$.

$m_y(Y)$ = jumlah dari irisan pada perkalian m_x dan m_y .

K = perkalian dari *mass function* yang mengalami konflik *evidence* bila irisannya kosong. (Rahmadhani, dkk, 2020 : 40).

II.2.4. Tingkat Belajar

Belajar merupakan kegiatan yang kompleks, setelah belajar orang memiliki keterampilan, pengetahuan, sikap dan nilai. Timbulnya kapabilitas tersebut dari (i) stimulasi yang berasal dari lingkungan dan (ii) proses kognitif yang dilakukan oleh pembelajar. Jadi belajar merupakan seperangkat proses kognitif yang mengubah sifat stimulasi lingkungan, melewati pengolahan informasi menjadi kapabilitas baru. Hasil belajar adalah bila seseorang telah belajar akan terjadi perubahan tingkah laku pada orang tersebut, misalnya dari tidak tahu menjadi tahu, dan dari tidak mengerti menjadi mengerti. Hasil belajar adalah kemampuan yang dimiliki siswa setelah ia menerima pengalaman belajarnya. Jadi hasil belajar merupakan

suatu kemampuan atau keterampilan yang dimiliki oleh siswa setelah siswa tersebut mengalami aktivitas belajar. Disimpulkan bahwa tingkat belajar merupakan ukuran capaian kegiatan yang kompleks, setelah belajar orang memiliki keterampilan, pengetahuan, sikap dan nilai. (Djonomiarjo, 2019 : 42).

II.2.5. Murid

Murid atau peserta didik merupakan orang yang belum dewasa dan memiliki sejumlah potensi dasar (fitrah) yang perlu dikembangkan. Peserta didik merupakan “*Raw Material*” (Bahan Mentah) dalam prosestransformasi dan implementasi, menempati posisi yang sangat penting untuk melihat signifikasinya dalam menemukan keberhasilan sebuah proses. Peserta didik adalah makhluk individu yang mempunyai kepribadian dengan ciri-ciri yang khas yang sesuai dengan pertumbuhan dan perkembangannya. Pertumbuhan dan perkembangannya dipengaruhi oleh lingkungan dimana ia berada. Peserta didik adalah anggota masyarakat yang berusaha mengembangkan potensi diri melalui proses pembelajaran yang tersedia pada jalur, jenjang dan jenis pendidikan tertentu. (Fauzi, 2019 : 62).

II.2.6. Web

Situs *web* merupakan salah satu media yang dapat digunakan untuk memberikan layanan informasi. *Web Browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*. Sumber

informasi web diidentifikasi dengan *Uniform Resource Identifier* (URL) yang dapat terdiri dari halaman *web*, *video*, gambar, ataupun konten lainnya. (Choiri, dkk, 2021 : 198).

Web merupakan kumpulan halaman *web* yang saling berkaitan. *Web* adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video dan dapat diakses melalui perangkat yang biasa disebut *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019 : 151).

II.2.7. *Hyper Text Markup Language* (HTML)

HTML adalah "bahasa" untuk membuat dokumen yang bisa diakses lewat web. HTML memuat dua bagian yakni isi dan pemformat. Isi adalah materi yang akan disampaikan, sedangkan bagian pemformat berwujud "tag" untuk menjadikan isi bisa diterjemahkan sebagai paragraf, ganti baris, heading, unordered list, ordered list, table. Dokumen HTML merupakan dokumen teks biasa yang bisa dibuka dengan pengedit teks seperti Notepad. Bila dibuka dengan Notepad akan terlihat bahwa yang tersaji di web browser sebenarnya sudah ada di sana. Tetapi di Notepad terlihat ada tambahan di sana-sini berupa "tag" sehingga di browser terlihat seperti itu. Anatomi dokumen HTML terdiri dari dua bagian, yakni HEAD dan BODY. Bagian HEAD biasanya memuat informasi tentang atribut dari dokumen itu. Sedangkan bagian BODY memuat informasi utama yang akan dikomunikasikan lewat web. Adanya link dalam dokumen HTML menjadikan dokumen ini bisa menunjuk ke dokumen-dokumen HTML lain baik di folder yang sama, di server yang sama, maupun di server lain. Inilah alasan mengapa dokumen ini disebut

hiperteks. Satu dokumen HTML juga bisa "memuat" dan menampilkan file-file lain seperti gambar, audio, Word, PDF, dan sebagainya. (Huzaeni, dkk, 2019 : 140).

Hypertext Markup Language atau HTML adalah bahasa yang digunakan pada dokumen web. Dengan kata lain, HTML merupakan bahasa pemrograman berupa tagtag yang digunakan untuk membuat dan mengatur penampilan dari halaman *website*. (Irmanawan, 2020 : 44).

II.2.8. *Hypertext Preprocessor (PHP)*

PHP adalah bahasa standar yang digunakan dalam *website*. PHP adalah bahasa program yang berbentuk skrip yang diletakan di dalam *server web*. (Irawan, 2020 : 46).

PHP singkatan dari *Hypertext PreProcessor* merupakan bahasa pemogram yang skript bersifat *Open Source*. Program ini bersifat *server side*, artinya tanpa adanya *server* yang berjalan disisinya script program PHP tidak dapat dijalankan PHP adalah *script* yang ditanamkan dalam HTML untuk membuat halaman *website* dinamis yang berkerja secara otomatis dan berfungsi sebagai pengolahan data pada *server* di mana *script* tersebut di jalankan. (Huzaeni, 2019 : 140).

II.2.9. *My Structure Query Language (MySQL)*

MYSQL merupakan salah satu database yang bersifat open source yang populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna diseluruh dunia MYSQL turunan dari SQL (Structured Query Language) yang sering digunakan oleh para programmer dalam pembuatan program. Sistem database

MYSQL selain mudah memahami sintaks dan pengaksesan juga di dukung beberapa fitur seperti Multithreaded, multi-user dan SQL database managemen sistem. (Huzaeni, dkk, 2019 : 141).

MySQL (*My Structured Query Language*) atau yang biasa dibaca mal-se-kuel adalah sebuah program pembuat dan pengelola database atau yang sering disebut dengan DBMS (*Database Management System*), sifat dari DBMS ini adalah open source. (Imaniawan, 2020 : 45).

II.2.10. *Unified Modeling Language (UML)*

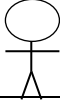
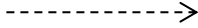
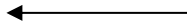
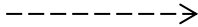
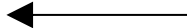
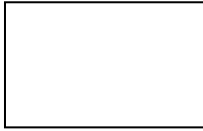
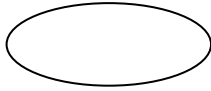
UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. use case diagram dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol Use Case

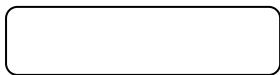
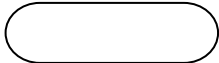
Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol Activity Diagram

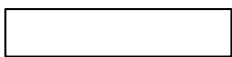
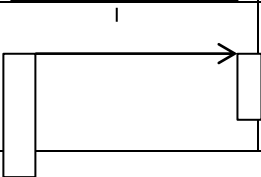
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

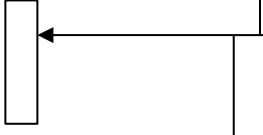
(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol Sequence Diagram

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

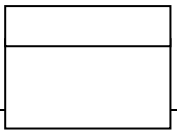
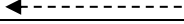
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
---	----------------	---

(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen

		mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
_____	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Andikos, 2019 : 39)