

BAB II

LANDASAN TEORI

2.1. Tinjauan Terhadap Penelitian Sebelumnya

Berikut beberapa peneliti terkait yang menjadi referensi pada penelitian ini: Joko S Dwi Rahardjo , Sutarman , Hilmi Hidayat dalam skripsinya yang berjudul: “Diagnosis Penyakit Pada Burung *Lovebird* Dengan Algoritma *Fuzzy*” Penelitian ini menghasilkan penerapan algoritma Fuzzy pada sistem pakar; khususnya, prosedurnya melibatkan pemilihan fakta atau gejala yang dialami oleh burung *lovebird*, yang kemudian dianalisis oleh sistem. Hasilnya, termasuk data gejala, kategori penyakit, dan potensi penyembuhan, kemudian disajikan.

Berikut beberapa peneliti terkait yang menjadi referensi pada penelitian ini: rifki fahrial zainal, Syariful Alim, dan Muhammad Hamza Syaiful Islam dalam skripsinya yang berjudul: “Sistem Pakar untuk Klasifikasi dan Diagnosa Penyakit Burung Murai Batu Menggunakan Metode Dempster-Shafer” Sebuah aplikasi sistem pakar untuk klasifikasi dan diagnosis penyakit pada burung robin murai menggunakan teknik Dempster-Shafer dikembangkan sebagai hasil dari penelitian ini. Anda juga dapat menggunakan aplikasi ini untuk mendiagnosis gejala pada burung robin murai dan mengetahui persentase kemungkinan penyakitnya dengan berharap aplikasi ini dapat memberikan diagnosis yang akurat.

2.2. Sistem

Semua bagian, peralatan, spesifikasi model, operasi, dan koneksi yang membentuk suatu sistem. Setiap susunan terstruktur dari orang, komputer, program, data, dan jaringan disebut sistem informasi (Indrajani, 2009).

2.3. Sistem Pakar

Sistem yang dirancang untuk mereplikasi pengetahuan manusia demi pemecahan masalah dengan cara yang mirip dengan penalaran manusia disebut sistem pakar. Untuk mengatasi tantangan tertentu, sistem

pakar yang efektif meniru perilaku individu yang berpengetahuan. (Ahmad, 2012).

2.4. Pemrograman Aplikasi

2.4.1 HTML

Bahasa dasar untuk menampilkan berbagai komponen web adalah *Hypertext Markup Language* (HTML). Tim Berners-Lee adalah orang pertama yang menciptakan HTML. Tujuan utama pengembangan HTML, sebagaimana dinyatakan dalam jurnal Achmad Solichin (2019), adalah untuk membangun tautan antar halaman web. Sederhananya, HTML adalah kode yang mendasari keseluruhan web. Kemudahan implementasi HTML berasal dari sintaksisnya yang lugas.

Peramban web mengambil kode HTML yang baru dihasilkan dan mengubahnya menjadi format yang dapat ditampilkan. Semua peramban web dapat membaca dan menampilkan kode HTML, tetapi tampilan halamannya bervariasi. *World Wide Web Consortium* (W3C), sebuah organisasi standar dunia yang terspesialisasi, bertanggung jawab atas desain dan regulasi HTML.

Hal ini disebabkan oleh fakta bahwa standar universal diperlukan karena berbagai program peramban web memiliki interpretasi kode HTML yang berbeda-beda.

2.4.2 CSS

Cara yang elegan dan menarik secara visual untuk menata dan memperindah halaman web adalah dengan *Cascading Style Sheets* (CSS). *World Wide Web Consortium* (W3C) merekomendasikan CSS pada tahun 1996 sebagai teknologi internet. Selain menentukan warna dan tata letak font, programmer dan desainer web menggunakan CSS untuk mengontrol penyajian konten di situs web mereka. menurut Jurnal yang di tulis oleh (Didik Setiawan, 2018). Ada dua sifat CSS yaitu internal dan eksternal. Jika internal yang dipilih, maka skrip itu dimasukkan secara langsung ke

halaman *website* yang akan didesain. Kalau halaman *web* yang lain akan didesain dengan model yang sama, maka skrip CSS itu harus dimasukkan lagi ke dalam halaman *web* yang lain itu. Sifat yang kedua adalah eksternal dimana skrip CSS dipisahkan dan diletakkan dalam berkas khusus. Nanti cukup gunakan semacam tautan menuju berkas CSS itu jika halaman *web* yang didesain akan dibuat seperti model yang ada di skrip tersebut.

2.4.3 PHP

Menurut Supono & Putratama (2018:1), mengemukakan bahwa “PHP ((PHP:*hypertext preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang ditambahkan ke HTML”. Selain itu, ada kode yang dapat dieksekusi, yang tersedia gratis daring dan dapat dijalankan langsung di sistem operasi (Nurmalina, 2017)

2.4.4 JavaScript

Menurut menurut Sunyoto (2017:17), adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer* (EI), *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan tag SCRIPT.

Beberapa hal tentang *Javascript*:

- a. *Javascript* didesain untuk menambah interaktif suatu *web*.
- b. *Javascript* merupakan sebuah bahasa *scripting*.
- c. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
- d. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*).
- e. *Javascript* biasanya disisipkan (*embedded*) dalam halaman HTML.
- f. *Javascript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi).

2.4.5 Bootstrap

Bootstrap adalah *framework front-end* yang intuitif dan *powerful* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah. *Bootstrap* menggunakan HTML, CSS, dan *Javasript*. *Bootstrap* memiliki fitur-fitur komponen *interface* yang bagus seperti *Typografi, Forms, Buttons, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel*, dan lain sebagainya. Dengan demikian dalam membuat *website* kita bisa menghemat waktu, fitur yang *responsive*, dan memiliki *design* yang konsisten menurut Jurnal yang di tulis oleh (Gregorius Agung, 2018). *Bootstrap* adalah kumpulan aturan dan komponen untuk kelas yang dapat digunakan untuk membangun situs web. *Bootstrap* memiliki antarmuka yang menarik, bersih, ringan, dan ramah pengguna, serta memungkinkan untuk memodifikasi dan menambahkan kelas sesuai keinginan. Penggunaan *Bootstrap* juga memberi fleksibilitas selama pengembangan situs web, sehingga dapat mengurangi waktu dan upaya yang dibutuhkan untuk pengembangan situs web. Kita selalu dituntut untuk melakukan tugas secara efisien dan efektif, sehingga dapat memilih framework *Twitter Bootstrap* saat perlu membuat situs web untuk diri sendiri atau pelanggan.

2.4.6 SQL

Menurut Jurnal yang di tulis oleh (R.H. Sianipar, 2018) dalam Buku “Membangun *Web* dengan PHP & MYSQL untuk Pemula & Programmer”. MySQL bukan termasuk bahasa pemrograman. MySQL merupakan salah satu *database* populer dan mendunia. MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database, field*, ataupun *index* untuk menambah atau menghapus data jurnal (Sianipar, R.H. 2015).

2.4.7 JQuery

“Jquery merupakan *library* atau kumpulan fungsi-fungsi *JavaScript* yang dapat digunakan untuk mempermudah pembuatan program yang dibuat dengan *JavaScript*” (Abdulloh,2017:24). Menurut (Wahyudi, 2107), JQuery adalah kumpulan program *Javascript* yang sangat sederhana yang ditulis dalam berbagai modul.

2.5. Algoritma Sistem

2.5.1. Metode *Dempster Shafer*

Model ketidakpastian dengan banyak probabilitas pada satu probabilitas awalnya diuji oleh Dempster, yang kemudian mengusulkan pendekatan Shafer. Sebuah buku diterbitkan oleh Shafer pada tahun 1976 yang memuat teori Dempster berjudul “*Mathematical Theory of Evident. Dempster-Shafer Theory of Evidence*”. Dengan menggunakan bukti, teori ini menunjukkan cara memberikan kepercayaan pada suatu keyakinan. Teori ini membedakan antara ketidaktahuan dan ketidakpastian, dan mengandung fitur-fitur yang konsisten dengan pemikiran para ahli dan memiliki landasan matematika yang kuat (Elyza Gustru Wahyuni & Widodo Prijodiprojo. 2013).

Untuk setiap bukti yang mendukung suatu teori, teknik *Dempster Shafer* memberinya skor keyakinan. Bobot ini diberikan kepada elemen-elemen dalam himpunan. Secara teori, kita dapat membedakan antara ketidaktahuan dan ketidakpastian dengan menggunakan pendekatan ini (Hartati & Iswanti, 2008).

Endang Lestari dan Emiliya Artha melakukan studi menggunakan teknik *Dempster Shafer* untuk menganalisis gangguan layanan di PT. Pada tahun 2017, Telkom Megelang memperkenalkan layanan Indihome. Eksekusi strategi ini berjalan dengan sangat lancar. Setiap individu dapat menyampaikan

gejalanya melalui antarmuka sistem untuk memudahkan diagnosis gangguan. Melalui penyajian dua contoh, yang masing-masing menggambarkan dua gejala yang berbeda, metode *Dempster Shafer* secara efektif menunjukkan perhitungannya yang selaras dengan evaluasi para ahli. Tingkat akurasi 73% dicapai ketika mengidentifikasi gangguan sebagai kabel telepon UTP yang putus, dengan gejala awal menunjukkan nilai keyakinan 0,8 dan gejala selanjutnya menunjukkan nilai keyakinan 0,9.

Dengan cara tertentu, teori umum *Dempster Shafer* diartikulasikan dalam interval [*Belief*, *Plausibility*]:

1. *Belief* (Bel) berfungsi sebagai indikator kekokohan bukti (*evidence*) yang mendasari suatu proposisi. Nilai 0 menandakan tidak adanya bukti, sementara nilai 1 menunjukkan adanya bukti (*evidence*) atau kepastian. Nilai *belief* (bel) didefinisikan berkisar antara 0 hingga 0,9.

2. *Plausability* (Pl) dinotasikan sebagai:

$$Pl(s)=Bel(-s)$$

Plausability juga bernilai 0 sampai 1. Jika yakin $\neg s$, maka dapat dikatakan bahwa:

$$Bel = (\neg s) = 0.$$

Seperti yang diuraikan oleh Giarratano dan Riley, konsep Kepercayaan (*Belief*) dapat diartikulasikan dan direpresentasikan melalui persamaan (1):

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

Dan *Plausibility* dinotasikan pada persamaan (2):

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

Dimana:

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

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

$$m(X) = mass\ function\ dari(X)$$

$m(Y)$ = *mass function* dari (Y)

Menurut teori *Dempster Shafer*, simbol (Θ) dapat digunakan untuk merepresentasikan kerangka diskresi. Kerangka diskresi merupakan sekumpulan hipotesis, sehingga sering disebut lingkungan, yang dapat ditunjukkan dalam persamaan (3):

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_N\} \quad (3)$$

Dimana:

Θ = *frame of discrement* atau *environment*

$\theta_1, \dots, \theta_N$ = Konstituen yang ada di *environment* hanya ada satu elemen lingkungan yang sesuai dengan respons yang diperlukan, namun semuanya mewakili respons potensial. $P(\Theta)$ adalah simbol untuk kemungkinan-kemungkinan ini, yang juga dikenal sebagai himpunan pangkat dalam teori *Dempster-Shafer*. Nilai elemen-elemen dalam himpunan pangkat ini berkisar dari nol hingga satu.

$$m: P(\Theta) \rightarrow [0,1]$$

Kemudian dapat dirumuskan pada persamaan (4):

$$\sum_{X \in P(\Theta)} m(X) = 1 \quad (4)$$

Dimana:

$P(\Theta)$ = *power set*

$m(X)$ = *mass function* (X)

Mass function (m) dalam kerangka teori *Dempster-Shafer* merepresentasikan derajat keyakinan yang terkait dengan suatu *evidence* (atau gejala), yang umumnya disebut sebagai *evidence measure*, dan dilambangkan dengan (m). Tujuannya adalah untuk menghubungkan berbagai komponen kepercayaan, yang dilambangkan sebagai θ . Tidak semua bukti (*evidence*) secara langsung menguatkan masing-masing komponen tersebut.

Dengan demikian, dari kebutuhan tersebut muncul fungsi kerapatan probabilitas (m). Mark m Tidak hanya menggambarkan elemen-elemen θ , tetapi juga mencakup semua himpunan bagiannya. Jika θ terdiri dari n elemen, maka jumlah himpunan bagian dari θ adalah 2^n . Total semua m dalam himpunan bagian θ adalah 1. Jika tidak ada informasi yang tersedia, mohon sebutkan dengan tepat:

$$m\{\theta\} = 1,0$$

Mengingat X merupakan himpunan bagian dari θ , yang dicirikan oleh fungsi kerapatan m_1 , dan Y juga merupakan himpunan bagian dari θ dengan fungsi kerapatannya sendiri m_2 , maka dapat disimpulkan bahwa kombinasi m_1 dan m_2 dapat menghasilkan fungsi kerapatan baru m_3 , seperti yang diartikulasikan dalam persamaan (5):

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X).m_2(Y)}{\sum_{X \cap Y = \emptyset} m_1(X).m_2(Y)} \quad (5)$$

Dimana:

$m_3(Z)$ = *mass function* dari *evidence* (Z)

$m_1(X)$ = *mass function* dari *evidence* (X), yang diperoleh dari nilai keyakinan suatu *evidence* dengan dikalikan dengan nilai *disbelief* dari *evidence* tersebut

m_2 = *mass function* dari *evidence* (Y), yang diperoleh dengan mengalikan keyakinan yang ditandai terkait dengan *evidence* dengan *disbelief* yang ditandai terkait dengan bukti tersebut.

$\sum_{X \cap Y = Z} m_1(X).m_2(Y)$ = Merupakan nilai kekuatan dari *evidence* Z yang diperoleh dari kobinasi nilai keyakinan sekumpulan *evidence*

2.5.2. Metode *Fuzzy*

Suatu nilai bisa benar dan salah karena *fuzzy* adalah istilah bahasa yang berarti ambigu atau membingungkan. Derajat keanggotaan dalam logika *fuzzy* dapat memiliki nilai antara nol dan satu. Batas antara benar dan salah dalam logika *fuzzy* kabur. Setiap nilai dalam teori logika *fuzzy* bisa benar atau salah pada saat yang bersamaan. Namun, bobot komponen-komponennya menentukan derajat kebenaran atau ketidakbenarannya.

Derajat keanggotaan dalam logika *fuzzy* dapat berkisar dari nol hingga satu, dan ini menunjukkan seberapa benar atau tidak benarnya suatu nilai. Logika *fuzzy* adalah cara yang tepat untuk memetakan ruang masukan ke dalam ruang keluaran dengan nilai-nilai kontinu. Tingkat keanggotaan dan tingkat kebenaran merupakan blok-blok pembangun logika *fuzzy*. Ini berarti bahwa segala sesuatu mungkin memiliki beberapa kebenaran, tetapi juga memiliki beberapa kesalahan (Kusumadewi, 2004).

Banyak contoh ambiguitas dapat ditemukan dalam kehidupan sehari-hari. Seorang pendidik, misalnya, mungkin meminta setiap siswa di kelas yang memiliki sepeda untuk mengangkat tangan. Bersepeda menjadi moda transportasi umum bagi banyak siswa di kampus. Namun, ketika pendidik mendorong siswa yang cerdas untuk berpartisipasi, muncul ketidakpastian mengenai keterlibatan mereka dalam kelompok intelektual yang terhormat. Perbedaan antara memiliki sepeda dan tidak memilikinya sangat jelas dan definitif; namun, garis yang memisahkan siswa cerdas dari yang tidak cerdas jauh lebih samar dan kompleks. Perbedaan antara siswa yang menunjukkan kecerdasan dan yang tidak tampak agak ambigu. Akibatnya, diperlukan leksikon ilmiah baru yang dapat secara efektif merangkum nuansa yang melekat dalam bahasa sehari-hari (Susilo, 2006).

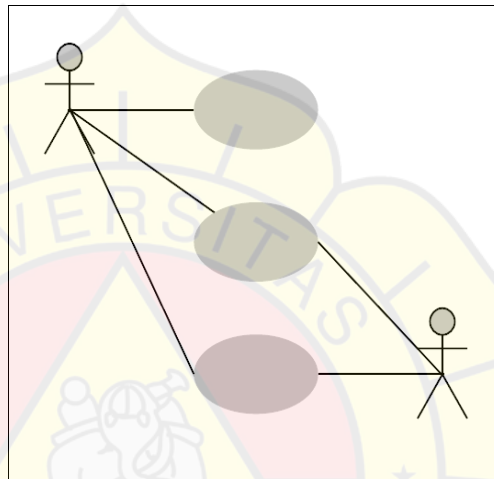
Bahasa semacam ini dikembangkan oleh Lotfi Asker Zadeh, seorang profesor terkemuka di Universitas California, AS, pada

awal tahun 1965. Ia mengubah kerangka teori himpunan tradisional menjadi ranah inovatif teori himpunan *fuzzy*. Teori ini dapat diterapkan di berbagai bidang, seperti algoritma kontrol, diagnosis medis, sistem pendukung keputusan, ekonomi, teknik, psikologi, studi lingkungan, keamanan, dan penelitian ilmiah (Setiadji, 2009). Misalnya, seorang supervisor gudang mengomunikasikan kepada supervisor produksi jumlah inventaris yang diperkirakan tersedia pada akhir minggu, sehingga supervisor produksi dapat memastikan jumlah inventaris yang dibutuhkan untuk dihasilkan keesokan harinya. Ilustrasi lain melibatkan seorang individu yang melaksanakan tanggung jawabnya dengan sangat baik, yang menghasilkan penghargaan dari supervisornya yang sejalan dengan kualitas kontribusi individu tersebut. Dengan memanfaatkan teori himpunan *fuzzy*, seseorang dapat menggambarkan logika bahasa melalui suatu wilayah yang menandakan tingkat keanggotaannya (Kusumadewi, 2004).

2.6. Pemodelan UML

Dalam hal penilaian, pembuatan, dan pendokumentasian sistem perangkat lunak, *Unified Modelling Language (UML)* kini menjadi standar de facto. Dalam hal perancangan dan pemodelan sistem, UML menyediakan standarnya. Program yang menggunakan pemodelan prosedural, seperti Visual Basic atau C, juga dapat memanfaatkan UML. (Yuni Sugiarti, 2018).

2.5.1 UseCase Diagram

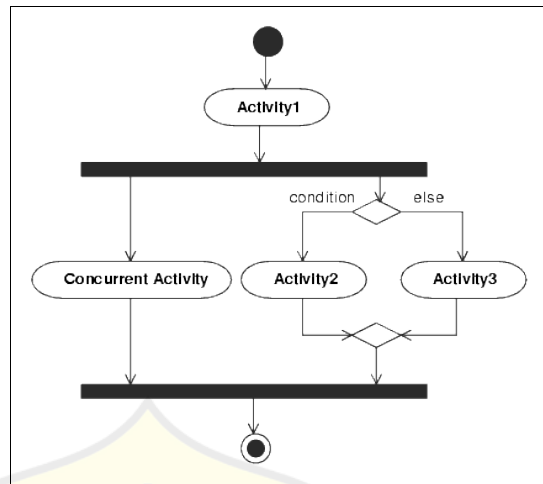


Gambar 2.1 Use Case Diagram

Sumber: Jurnal (Yuni Sugiarti, 2018)

Contoh *Use Case Diagram* adalah interaksi antara sistem masa depan dan satu atau lebih aktor. Jurnal (Yuni Sugiarti,2018).

2.5.2 Activity Diagram

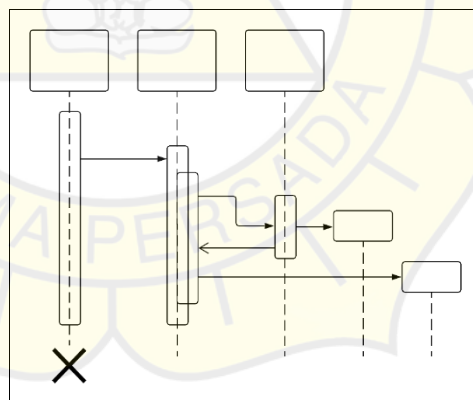


Gambar 2. 2 Activity Diagram

Sumber: Jurnal (Yuni Sugiarti,2018)

Activity Diagram adalah representasi grafis dari alur kerja suatu proses atau sistem. Perlu diingat bahwa diagram aktivitas menunjukkan kemungkinan-kemungkinan yang dapat dicapai suatu sistem, alih-alih tindakan individ.. (Yuni Sugiarti,2018).

2.5.3 Sequence Diagram



Gambar 2. 3 Sequence Diagram

Sumber: Jurnal (Yuni Sugiarti,2018)

Dengan merinci masa pakai objek dan pesan yang dikirim dan diterima di antara objek-objek tersebut, *Sequence Diagram* menggambarkan perilaku objek dalam suatu kasus penggunaan. Setelah semua *Use case* diidentifikasi dan *Sequence Diagram* telah mencakup interaksi alur pesan, jumlah *Sequence Diagram* yang

perlu dihasilkan berbanding lurus dengan jumlah *Use case* yang memiliki prosesnya masing-masing.. (Yuni Sugiarti,2018)

