

BAB 2 TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam perancangan pengerjaan dan penelitian penulis menggali informasi dan referensi dari jurnal, maupun skripsi yang sebelumnya sudah ada yang berkaitan dengan tema Logika *Fuzzy* dan teori Algoritma *Brute Force*.

Tabel 2.1 Penelitian Terdahulu

No.	Nama Peneliti	Judul	Tahun	Metode	Hasil
1.	Yuhendra	Rekayasa Pengetahuan Pakar Berbasis Aturan Untuk Identifikasi Hadware	2008	Menggunakan metode penalaran maju (<i>forwaed chaining</i>) dari pengambilan keputusan bersumber dari dialog antara pakar dan pemakai (<i>user</i>) <i>Hardware</i> .	Tampilan aplikasi berjalan dengan baik, sistem memberikan fasilitas berisi saran tips perawatan dan pemilihan terhadap kerusakan.
2.	Hastari Tri Rahayu	Sistem Pakar Pendeteksi Kerusakan pada Komputer dengan metode <i>Fuzzy</i> .	2014	Metode <i>fuzzy</i> memproses data tentang masalah kerusakan pada komputer yang menghasilkan informasi laporan-	Sistem sangat membantu kerusakan yang dialami pada komputer pengguna mendeteksi kerusakan,

				laporan manajerial yang dibutuhkan oleh pengguna	seperti kerusakan VGA, CD/DVD Room tidak membaca, Hardisk Tidak membaca, Infoormasi yang dihasilkan berupa laporan-laporan data diantaranya : laporan solusi kerusakan yang dihadapi oleh pengguna
3.	Ahmad Kamsyakawuni	Aplikasi Sistem Pakar untuk Diagnosa Penyakit Hipertiroid dengan metode Inferensi <i>Fuzzy</i>	2012	Pengunaan metode inferensi <i>fuzzy</i> mamdani yang inputanya berupa skor gejala untuk pengolahan pengetahuan agar diperoleh	Aplikasi sisem yang dibuat menggunakan MATLAB disajikan analisa Proses dan tampilan halaman validasi

		Mamdani		konsekuensi dan kesimpulan.	perhitungan akurasi sistem / keberhasilan sistemnya mengacu dari perbandingan antara hasil diagnosa dokter dan diagnose sistem yang pada akhir akan muncul kevalidtan akurasi prosentase mencapai 95,45 %.
--	--	---------	--	-----------------------------	--

Dari penelitian yang sudah dilakukan sebelumnya, referensi diatas melahirkan satu pembahasan yang sama dalam yang digunakan yakni pada pilihan topik sistem Pakar. Sedangkan penelitian ini perbedaannya, terletak pada metode yang digunakan Algoritma *Brute Force* dan model pengembangan sistemnya serta desain *Interface* yang berbeda dan tampilan menu yang akan lebih menarik.

1.2 Kecerdasan Buatan

Kecerdasan buatan merupakan suatu konsep atau aturan yang memiliki fungsi tertentu atau menyelesaikan suatu masalah tertentu secara cepat.

Kecerdasan buatan dinyatakan dalam bahasa program yang dapat dieksekusi pada sistem nyata. dapat Implementasi dari kecerdasan dalam bidang.

Logika merupakan salah satu teknik yang dapat digunakan untuk meneliti sebuah penalaran. Salah satu logika yang dipelajari dalam mata kuliah kecerdasan buatan adalah logika proposisi (Kusumadewi, 2003)

1.3 Logika *Fuzzy*

Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Peranan derajat keanggotaan pada himpunan *fuzzy* sangat penting, hal ini karena sebagai penentu keberadaan elemen dalam himpunan. Nilai keanggotaan atau derajat keanggotaan atau membership function menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut (Purnomo, 2010).

Sebagai perumpamaan Logika *fuzzy* dianggap sebagai lingkaran proses menghubungkan ruang input dengan ruang output mengolah menjadikan output ke dalam bentuk yang baik berasal dari data input.

Beberapa alasan mengapa banyak orang menggunakan logika *fuzzy*, yaitu:

1. Logika *fuzzy* dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
2. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
3. Konsep logika *fuzzy* mudah dimengerti yang mendasari penalaran.
4. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
5. Logika *fuzzy* sangat fleksibel.
6. Logika *fuzzy* mampu memodelkan fungsi-fungsi non linear yang sangat kompleks.
7. Logika *fuzzy* didasarkan pada bahasa alami.

1.4 Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan yaitu: Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan atau Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Himpunan *fuzzy* dipergunakan sebagai mengantisipasi hal tersebut. 2 himpunan yang berbeda dapat dimasuki oleh seseorang. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotannya. Himpunann *fuzzy* memiliki 2 atribut, yaitu:

1. *Numeris*, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 30, 29.
2. *Linguistic* adalah pemberian atau menamakan suatu grup.

2.5 Brute Force

Sebuah pendekatan yang memecahkan suatu masalah secara langsung (*straightforward*), biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *Brute Force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*) adalah Algoritma *Brute Force*.

2.5.1 Kelebihan dari Algoritma *Brute Force*

Berikut ini beberapa kelebihan yang dimiliki oleh *Brute Force*, yaitu:

1. Sederhana dan mudah dimengerti.
2. Sebagian besar masalah dapat diselesaikan dengan Algoritma *Brute Force*.
3. Menghasilkan algoritma baku (*standart*) untuk tugas-tugas komputasi seperti penjumlahan atau perkalian buah bilangan, menentukan elemen minimum atau maksimum ditabel.
4. Menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, perkalian *matriks*.

2.5.2 Kelemahan dari Algoritma *Brute Force*

Beberapa kelemahan yang dijabarkan oleh penulis, yaitu:

1. Tidak kreatif teknik pemecahan masalah lain.
2. Lambat sehingga tidak dapat diterima.
3. Jarang menghasilkan algoritma yang sesuai atau efektif.

2.6 Sistem Pakar

Sistem pakar adalah suatu aturan yang membentuk sebuah kecerdasan buatan dimana aturan ini dipergunakan untuk menyelesaikan atau mengatasi suatu masalah.

Banyak jenis aturan atau algoritma yang dapat diterapkan pada sebuah sistem pakar yang setiap aturan bisa disesuaikan pada masalah tertentu.

Beberapa komponen untuk membanguun sistem pakar (*expert system*) yang baik, antara lain (Iswanti, 2008).

1. Basis Pengetahuan (*Knowledge Base*).
2. Antar Muka Pengguna (*User Interface*).
3. Memori Kerja.
4. Mekanisme Inferensi (*Inference Machine*).
5. Pencocokan Pola (*Pattern Matching*).

Memori kerja merupakan data penyimpanan yang berisikan fakta-fakta yang diperoleh saat dilakukan proses konsultasi. Fakta-fakta inilah nantinya akan dijadikan mesin inferensi sebagai dasar pengetahuan untuk menentukan suatu keputusan pemecahan masalah. Serta fasilitas penjelasan dapat memberikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Sedangkan akuisisi pengetahuan merupakan proses perubahan dari keahlian seseorang ke dalam komputer.

2.7 PHP dan MySQL

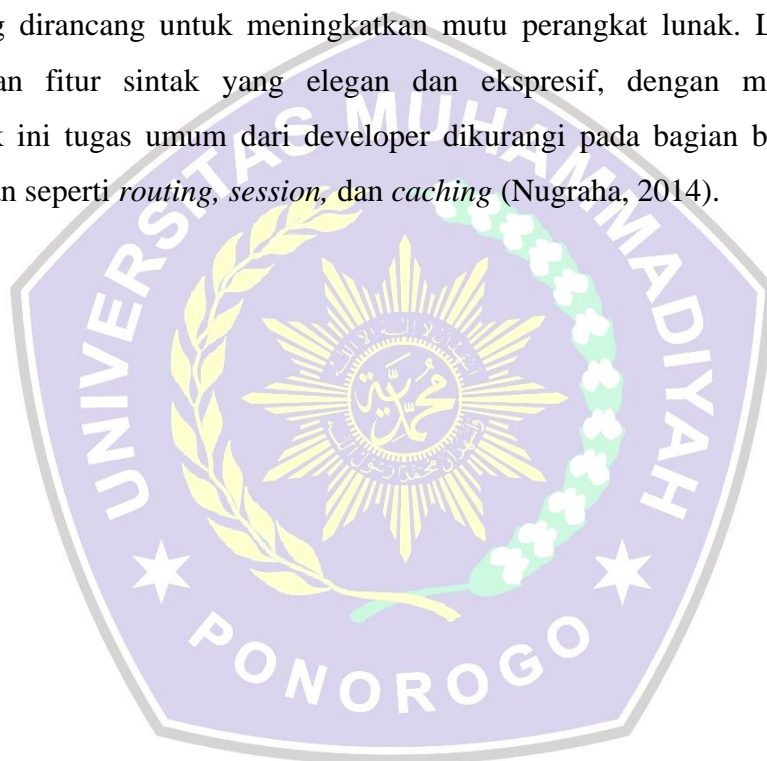
PHP (Hypertext Preprocessor) adalah bahasa *server side scripting* dijalankan pada server yang terintegasi dengan *HTML* yang dapat menciptakan halaman web yang Praktis. *PHP* sebagai bahasa yang sintaks dan perintah-perintah dieksekusi pada server selanjutnya akan masuk ke browser menggunakan format *HTML*. Arief (2011:43).

MySQL merupakan sistem pengelolaan basis data relasional. Ini didasarkan pada *query* struktur (*SQL*), yang digunakan untuk melakukan menambah, menghapus, dan memodifikasi informasi pada database. Perintah *SQL* standar,

sistem dapat diartikan menjadi kelompok dari prosedur-prosedur yang mempunyai maksud. (Mustakini, 2009).

2.8 FRAMEWORK LARAVEL

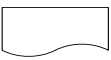




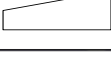

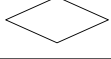

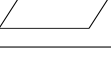



Menurut Rahmat Awaludin (Awaludin, 2016) bahwa Laravel merupakan *framework PHP* yang Rancang Bangun Aplikasi Kerusakan Komputer dan Berbasis Web Menggunakan Fraamework Laravel diluncurkan dibawah surat izin *MIT* pertama dikembangkan oleh Taylor Otwell, dibuat menggunakan *MVC* (*Model View Controller*). Laravel adalah peningkatan dari rancangan sebelumnya *MVP* yang dirancang untuk meningkatkan mutu perangkat lunak. Laravel juga memberikan fitur sintak yang elegan dan ekspresif, dengan menggunakan framework ini tugas umum dari developer dikurangi pada bagian besar proyek web dengan seperti *routing*, *session*, dan *caching* (Nugraha, 2014).



2.9 Data flow diagram (*FLOWCHART*)

Bagan yang di pergunakan untuk menerangkan suatu proses. Proses yang dimaksud berupa alur sistem alur konsep pada suatu sistem.

Tabel 2.2 Daftar Simbol *Flowchart* (Bagan Alir)

No	Gambar	Nama	Keterangan
1		Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
2		Proses Manual	Merupakan proses manual dalam <i>flowchart</i> .
3		Simbol Proses Komputerisasi	Menunjukkan kegiatan proses dari operasi program komputer.
4		<i>File Hardisk/ Database</i>	Menunjukkan kegiatan <i>input</i> atau <i>output</i> menggunakan <i>hardisk</i> .
5		<i>Offline Storage</i>	Menunjukkan tfile non-komputer yang diarsip urut tanggal (cronological)
6		Simbol <i>Keyboard</i>	Merupakan <i>input</i> data yang menggunakan <i>online keyboard</i> .
7		Arus dokumen/ pemrosesan	Menunjukkan arus dari proses.
8		Keputusan	Menunjukkan tahapan pembuatan keputusan
9		Terminal	Menunjukkan awal dan akhir dari bagan alir dokumen.
10		<i>Input/Output</i>	Mewakili data <i>input/output</i> .
11		Penjelasan	Menunjukkan penjelasan dari suatu proses
12		<i>Connector</i>	Menunjukkan penghubung ke halaman yang sama atau ke halaman lain
13		Arus dari Jaringan	Data melalui channel komunikasi

2.10 (Data Flow Diagram) DFD

Alat yang dipergunakan sebagai perancangan pembuatan alur sistem. Proses yang diterapkan di DFD dengan menghubungkan sebagai alur hubungan antar sistem yang tersusun secara rapi, mudah dan cepat untuk di pahami dengan alur data secara terstruktur, dapat digunakan sebagai analisa perancangan sistem yang dengan mudah dikomunikasikan secara professional dan efisien sistem kepada penggunanya atau pembuat program.

Tabel 2.3 Contoh Simbol-simbol DFD

Elemen Data Flow Diagram	Field tipikal yang biasa digunakan	Simbol Gene & Sarson	Simbol De Marco & Jourdan
Setiap Proses Memiliki: Nomor Nama Deskripsi proses Satu/lebih output Data flow Satu/lebih input <i>Flow</i>	Label (nama) Type (proses) Deskripsi Nomor proses		
Setiap data flow memiliki: Nama Deskripsi Satu/Lebih konteks Ke suatu proses	Label Type Deskripsi Alias Komposisi (Deskripsi dari elemen-elemen data)		
Setiap data store memiliki: Nomor Nama Deskripsi Satu/lebih output Data flow Satu/lebih input Data flow	Label(nama) Type Deskripsi Alias Komposisi Catatan		
Setiap entitas eksternal memiliki nama deskripsi	Label Tipe Deskripsi Alias Deskripsi entitas		

(Sumber: Hanif Al – Fatta, 2007)