

# Sistem Pakar Diagnosa dan Troubleshooting Kerusakan Website Menggunakan Metode Dempster-Shafer

## *Expert System for Diagnosing and Troubleshooting Website Damage Using the Dempster-Shafer Method*

Yusuf Hendra Pratama<sup>1</sup>, Firmansyah<sup>2</sup>

<sup>1,2</sup>Universitas Islam Al-Azhar

[Yshendra.tm@gmail.com](mailto:Yshendra.tm@gmail.com)<sup>1</sup>, [firmansyah@gmail.com](mailto:firmansyah@gmail.com)<sup>2</sup>

### Informasi Artikel:

Diterima: 28 April, 2023, Direvisi: 03 Juni, 2023, Disetujui: 06 Juni, 2023

---

#### Abstrak-

**Latar Belakang:** Troubleshooting website merupakan proses untuk mengidentifikasi, menganalisis, dan memecahkan masalah sebuah website. Permasalahan dalam penelitian ini adalah kurangnya informasi dan lamanya waktu dalam mengetahui kerusakan website sehingga pengguna kesulitan dalam melakukan troubleshooting.

**Tujuan:** Tujuan penelitian adalah membuat sistem pakar yang dapat digunakan untuk mendiagnosa kerusakan website dan mudah diakses.

**Metode:** Metode yang digunakan pada penelitian ini adalah metode *Dempster-Shafer* yang merupakan metode pembuktian berdasarkan *belief function* dan *plausible reasoning* dalam mengkombinasikan bukti untuk mengkalkulasikan kemungkinan dari suatu kerusakan pada *website*.

**Hasil:** Hasil penelitian ini adalah sebuah sistem pakar untuk diagnosa kerusakan *website* yang memiliki 15 gejala dan 5 kerusakan. Sistem yang dikembangkan menunjukkan nilai akurasi sebesar 84% dengan uji coba sebanyak 25 kasus.

**Kesimpulan:** Sistem pakar yang dihasilkan dapat digunakan untuk mendiagnosa kerusakan website berdasarkan gejala dengan tingkat akurasi yang baik.

**Kata Kunci:** Sistem Pakar, Dempster-Shafer, Troubleshooting, Website

---

#### Abstract-

**Background:** Website troubleshooting is the process of identifying, analyzing, and solving a website's problems. The problem in this study is the lack of information and the length of time in knowing the website damage, so users have difficulty troubleshooting.

**Objective:** The purpose of the research is to create an expert system that can be used to diagnose website damage and is easily accessible.

**Methods:** The method used in this research was the Dempster-Shafer method, which is a method of proof based on the belief function and plausible reasoning in combining evidence to calculate the probability of damage to the website.

**Result:** The results of this study were an expert system for diagnosing website crashes which had 15 symptoms and 5 crashes. The developed system showed an accuracy value of 84% with a trial run of 25 cases.

**Conclusion:** The resulting expert system can be used to diagnose website damage based on symptoms with a good degree of accuracy.

**Keywords:** Expert System, Dempster-Shafer, Troubleshooting, Website

---

#### Penulis Korespondensi:

Yusuf Hendra Pratama,

Program Studi Ilmu Komputer, Fakultas Teknik, Universitas Islam Al-Azhar, Mataram, Indonesia,

Email: [yshendra.tm@gmail.com](mailto:yshendra.tm@gmail.com)

---

**How to Cite:** Y. H. Pratama, Firmansyah, "Sistem Pakar Diagnosa dan Troubleshooting Kerusakan Website Menggunakan Metode Dempster-Shafer". *Jurnal Bumigora Information Technology (BITe)*, vol. 5, no. 1, pp. 31~42, 2023.

This is an open access article under the CC BY-SA license (<https://creativecommons.org/licenses/by-sa/4.0/>)

## 1. PENDAHULUAN

Troubleshooting merupakan alur pencarian suatu akar permasalahan yang dilakukan secara sistematis guna mendapatkan solusi dari permasalahan tersebut. *Troubleshooting* juga dapat didefinisikan sebagai suatu proses menghilangkan penyebab potensial dari suatu permasalahan yang umumnya sering kali ditemukan pada berbagai bidang seperti pada bidang informatika, sistem administrasi, serta bidang elektronika dan kelistrikan [1]. Salah satu contoh *troubleshooting* dibidang informatika adalah *troubleshooting website*. *Troubleshooting* pada *website* merujuk pada proses mengidentifikasi, menganalisis, dan memecahkan masalah yang terjadi pada sebuah *website*. Hal ini melibatkan langkah-langkah untuk menemukan dan memperbaiki penyebab masalah, sehingga *website* dapat berfungsi dengan baik dan memberikan pengalaman pengguna yang optimal [2].

*Troubleshooting website* dilakukan guna mengetahui akar permasalahan dalam kondisi kerusakan *website*. Permasalahan kerusakan *website* dikategorikan menjadi 5 bagian yaitu kerusakan yang disebabkan pada sisi *server*, *database*, penulisan baris kode program (*scripting*), jaringan dan kerusakan yang disebabkan dari sisi komputer pengguna *website* [2]. Pada saat terjadi kerusakan *website*, *browser* akan menampilkan beberapa pesan *error* pada halaman *website* yang diakses. Hal ini menyebabkan pengguna *website* kesulitan untuk menemukan akar permasalahan yang terjadi sehingga ditampilkan pesan *error* tersebut. Umumnya untuk mengetahui kerusakan pada *website* pengguna akan langsung menanyakan kepada layanan penyedia *server* untuk memastikan apakah *server* yang digunakan berjalan baik atau tengah dalam kondisi *down*. Selain itu, pengguna melakukan pengecekan satu persatu baik dari sisi *server*, *database*, kode program, jaringan maupun dari sisi komputer yang digunakan dalam melakukan akses *website* tersebut. Dalam melakukan penelusuran tiap-tiap sisi *website* memerlukan waktu yang cukup lama hingga dapat mengidentifikasi kendala yang ada. Untuk mempermudah pengguna dalam mengetahui penyebab utama kerusakan pada *website* maka dibutuhkan suatu sistem pakar yang digunakan untuk memberikan kemudahan akses pengguna dalam melakukan identifikasi kerusakan *website* secara mandiri.

Sistem pakar (*expert system*) adalah program berbasis pengetahuan yang mencakup pengetahuan dan pengalaman yang dimasukkan oleh satu atau lebih pakar dalam bidang pengetahuan tertentu, agar setiap orang dapat memecahkan berbagai jenis masalah tertentu [3]. Sistem pakar merupakan representasi pengetahuan seorang pakar yang banyak digunakan untuk kepentingan komersial karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam program sehingga komputer dapat memberikan keputusan dan melakukan penalaran secara cerdas [4]. Pakar adalah seseorang dengan keahlian khusus yang dapat memecahkan masalah yang tidak dapat dipecahkan oleh orang biasa [5]. Dengan menggunakan sistem pakar ini, bahkan rata-rata orang bisa memecahkan masalah masalah yang sangat kompleks yang tidak dapat diselesaikan tanpa bantuan seorang pakar. Tujuan pengembangan sistem pakar sistem pakar bertujuan untuk mempermudah seseorang dalam menggunakan satu perangkat lunak yang di adopsi dari seorang pakar tanpa menggantikan peran seorang pakar dengan biaya relatif kecil [6].

Dalam membangun suatu sistem pakar dibutuhkan suatu metode pendukung. Salah satu metode yang dapat digunakan adalah metode Dempster-Shafer yang mana akan memberikan hasil diagnosa gejala kerusakan pada *website* sesuai dengan penyebab kerusakannya berdasarkan fungsi kepercayaan (*belief function*) dan pemikiran yang masuk akal (*plausible reasoning*) [7]. Metode ini digunakan untuk mengkombinasikan potongan informasi yang terpisah untuk mengkalkulasi kemungkinan dari suatu peristiwa [8]. Teori *Dempster-Shafer* merupakan salah satu metode yang mampu mengakomodasi ketidakpastian dalam klasifikasi multispectral [9].

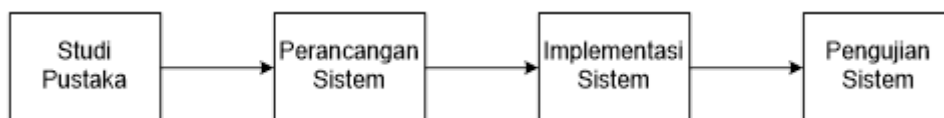
Penelitian mengenai metode *Dempster-Shafer* sebelumnya pernah dilakukan oleh Fifto Nugroho dan Alexius Ulan Bani yang berjudul Penerapan Metode *Dempster-Shafer* Pada Sistem Pakar Diagnosa Penyakit Usus Halus, dalam penelitian ini objek yang diangkat adalah membangun suatu sistem pakar dengan menggunakan metode *Dempster-Shafer* untuk mengidentifikasi dan mencari keyakinan diagnosa terhadap penyakit usus halus [10]. Penelitian menggunakan metode *Dempster-Shafer* juga telah dilakukan oleh Mohammad Hasan Basri, Ali Mahmudi, Nurlaily Vendyansyah dengan judul Perbandingan Metode *Dempster-Shafer* Dan *Certainty Factor* Untuk Diagnosis Penyakit Tanaman Terong, membandingkan metode mana yang lebih efektif untuk diagnosis penyakit tanaman terong berbasis *website* [11]. Penelitian menggunakan metode *dempster-*

*shafer* dilakukan oleh Yusra Fernando, Riduwan Napianto dan Rohmat Indra Borman untuk diagnosa penyakit psikologis gangguan control impuls berbasis *website*, memiliki 12 gejala yang dapat dipilih dan menghasilkan informasi berupa deskripsi diagnosis, penyebab dan cara penanganannya serta menghasilkan uji akurasi sebesar 85% [12]. Sementara Penelitian yang dilakukan oleh Hendra Mayatopani, Rino Subekti, Nunik Yudaningsih, Mochamad Sanwasih menggunakan metode *dempster-shafer* untuk diagnosa gangguan mental dengan mesin inferensi, memiliki 25 gejala dan 7 penyakit gangguan mental, informasi yang diberikan berupa penyakit dengan deskripsinya, penyebab dan pengobatannya serta memiliki hasil akurasi sebesar 84% [13]. Penelitian yang dilakukan oleh Nelly Astuti Hasibuan, Alwin Fau dengan melakukan kombinasi antara metode *certainty factor* dengan *dempster-shafer* untuk mendeteksi perilaku ADHD, dengan 4 gejala dan 3 penyakit metode *certainty factor* dan *dempster-shafer* memiliki kesamaan langkah untuk menentukan hasil diagnosa yaitu menentukan densitas awal kemudian kombinasi densitas [6]. Penelitian dengan metode yang sama juga dilakukan oleh Ridwan Aji Pamungkas dan Lilis Dwi Farida untuk mendeteksi dini gizi buruk pada balita, memiliki 23 gejala dan 3 jenis gangguan gizi buruk dengan tingkat akurasi sebesar 85% [14]. Penelitian lain yang dilakukan oleh Sari Iswanti dan Ratih Novia Anggraeny dengan judul Implementasi Metode *Dempster-Shafer* Pada Sistem Pakar Pendiagnosa Kerusakan Sepeda Motor, dalam penelitian ini metode *Dempster-Shafer* digunakan untuk mencari nilai keyakinan atau kepercayaan diagnosa kerusakan sepeda motor yang diperoleh berdasarkan gejala-gejala atau gangguan yang dialami oleh sepeda motor tersebut [15]. Penelitian yang dilakukan oleh Agus Setiawan, Erfan Karyadiputra, Nadiya Hijriana dan Indu Indah Purnomo, metode *dempster-shafer* digunakan untuk diagnosa terhadap penyakit bakteri dan parasite pada ikan gabus, dengan menggunakan 13 gejala dan 5 penyakit informasi yang ditampilkan berupa diagnosa penyakit, rule inferensia dan keyakinan dalam persentase [16]. Penelitian metode *dempster-shafer* yang dilakukan oleh Aminun, Bachtiar Efendi, Akmal Nasution untuk mendagnosa penyakit stroke *Transient Ischemic Attack* (TIA), dalam penelitian ini menggunakan 7 gejala dan 4 penyakit dengan hasil tingkat kepercayaan sebesar 59% diperoleh dari konsultasi berdasarkan gejala pasien yang menderita penyakit Storek Hemoragik [17]. Penelitian yang dilakukan oleh Sukmawati, Rifky Maulana, Ranti Holiyanti, Betha Nurina Sari dengan metode *dempster-shafer* untuk mendiagnosa kerusakan jaringan LAN, dengan 14 gejala jaringan, beberapa kondisi dan solusi yang diberikan serta memiliki hasil perhitungan dari contoh sebesar 64% [18].

Perbedaan penelitian ini dengan penelitian sebelumnya dengan metode yang sama yaitu *dempster-shafer* adalah studi kasus penelitian ini terkait dengan kerusakan (*troubleshooting*) pada website dimana belum pernah ada yang melakukan penelitian dengan kasus ini, kemudian hal yang menjadi pembeda adalah jumlah pengujian output berdasarkan pakar menggunakan jumlah kasus lebih dari 20 kasus. Sedangkan kebaruan penelitian ini terletak pada studi kasus yang belum pernah diteliti sebelumnya sehingga untuk selanjutnya dapat dijadikan sebagai acuan untuk pengembangannya. Penelitian ini bertujuan mengembangkan sebuah sistem pakar yang dapat membantu pengguna *website* untuk dapat mengetahui kerusakan *website* yang sedang dialaminya sehingga pengguna dapat langsung memberikan tindakan lanjutan untuk mengatasi kerusakan *website* tersebut.

**2. METODE PENELITIAN**

Penelitian ini dilakukan dengan metode kualitatif dengan memiliki beberapa tahapan dalam pelaksanaan kegiatan yang tertuang pada kerangka kerja penelitian. Metode penelitian yang digunakan dalam membangun sistem pakar diagnosa dan *troubleshooting* website ini direpresentasikan pada Gambar 1.



Gambar 1. Alur Penelitian

Pada tahapan studi pustaka dilakukan pengumpulan data dari berbagai sumber seperti penelitian terdahulu dan sebuah perusahaan penyedia *web hosting* yang mana digunakan sebagai akuisisi pengetahuan dan

landasan pendukung pada penelitian dengan melakukan wawancara untuk menemukan data pakar berupa macam-macam gejala yang terjadi ketika *troubleshooting website*. Dari hasil pengumpulan data dilakukan perancangan sistem dengan menggunakan algoritma *dempster-shafer* kemudian merumuskan kebutuhan fungsional sistem, membuat daftar gejala gejala dan kerusakan pada *website*, membuat gambaran *data flow diagram* dan juga membuat rancangan basis data. Implementasi sistem dilakukan dengan membangun sistem pakar berbasis *website* dengan menggunakan PHP dan *database* MySQL berdasarkan hasil rancangan sistem. Hasil Implementasi akan dilakukan pengujian dengan menggunakan metode Blackbox dan pengujian kecocokan hasil sistem pakar dengan hasil diagnosa seorang pakar.

### 2.1. Metode Dempster-Shafer

Teori *Dempster-Shafer* pertama kali diperkenalkan oleh Dempster dan kemudian pada tahun 1976 Shafer mempublikasikan teori *Dempster* itu pada sebuah buku yang berjudul *Mathematical Theory of Evident. Dempster-Shafer Theory of Evidence*, menunjukkan suatu cara untuk memberikan bobot keyakinan sesuai fakta yang dikumpulkan [16]. Pada teori ini dapat membedakan ketidakpastian dan ketidaktahuan. Teori *Dempster-Shafer* adalah representasi, kombinasi dan propogasi ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang secara instutitif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat [9]. Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval [10]:

[Belief, Plausibility]

*Belief* (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. *Plausibility* (Pls) akan mengurangi tingkat kepastian dari *evidence*. *Plausibility* bernilai 0 sampai 1. Jika yakin akan X, maka dapat dikatakan bahwa  $Bel(X) = 1$ , sehingga rumus di atas nilai dari  $Pls(X) = 0$ . Fungsi *belief* (Bel) dinotasikan pada Persamaan (1) [19]:

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

Fungsi *plausibility* ditunjukkan pada Persamaan (2) berikut ini:

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

Dimana:

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

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

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

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

Teori *Dempster-Shafer* menyatakan adanya *frame of discrement* yang dinotasikan dengan simbol ( $\Theta$ ). *Frame of discrement* merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* yang ditunjukkan pada Persamaan (3) berikut:

$$\Theta = \theta_1, \theta_2, \dots, \theta_N \tag{3}$$

Dimana:

$\Theta$  = *frame of discrement* atau *environment*

$\theta_1, \dots, \theta_N$  = element/unsur bagian dalam *environment*

*Environment* mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban, dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan.

$P(\Theta)$  = power set (kemungkinan yang menjadi jawaban) dimana  $m : P(\Theta)[0, 1]$  sehingga didapat rumus:

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

Dengan:

$P(\Theta) = \text{power set}$

$m(X) = \text{mass function } (X)$

M adalah tingkat kepercayaan dari gejala, tujuannya mengaitkan ukuran kepercayaan elemen  $\Theta$ . Jika  $\theta$  berisi  $n$  elemen, maka subset  $\Theta$  adalah  $2^n$ , dengan hipotesis :  $m\{\Theta\} = 1, 0$ .

Jika diketahui X subset dari dengan m1 sebagai fungsi densitas, Y subset dari  $\Theta$  dengan m2 sebagai fungsi densitasnya, maka kombinasi m1 dan m2 menjadi m3 dengan Persamaan (5).

$$m3(Z) = \frac{\sum_{X \cap Y = Z} m1(X).m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X).m2(Y)} \tag{5}$$

Dimana:

$m3(Z) = \text{mass function dari evidence } (Z)$

$m1(X) = \text{mass function dari evidence } (X)$

$m2(Y) = \text{mass function dari evidence } (Y)$

$M1(X)$  diperoleh dari nilai keyakinan suatu *evidence* dikalikan dengan nilai *disbelief* dari *evidence* tersebut sedangkan  $m2(Y)$  diperoleh dari nilai keyakinan suatu *evidence* dikalikan dengan nilai *disbelief* dari *evidence* tersebut.

### 2.1.1. Perhitungan Manual Dempster-Shafer

Dari perhitungan yang telah dirumuskan pada rumus Dempster-Shafer dilakukan perhitungan manual pada suatu kasus dimana kendala yang dialami adalah sebagai berikut:

G2 = *Website Blank*

G3 = *Internal Server Error*

Dengan menggunakan metode *Demster-Shafer* akan dihitung presentase komponen kerusakan yang mungkin terjadi berdasar gejala tersebut. Nilai kepercayaan dari G2 sebagai gejala dari P1 dan P3 adalah:

$$m1\{P1, P3\} = 0.9$$

$$m1\{\emptyset\} = 1 - 0.9 = 0.1$$

Nilai kepercayaan dari G3 sebagai gejala dari P1, P2, P3, P5

$$m2\{P1, P2, P3, P5\} = 0.7$$

$$m2\{\emptyset\} = 1 - 0.7 = 0.3$$

Hasil perhiungan manual untuk menemukan nilai kepercayaan dari G1 dan G2 dapat dikelompokkan dalam Tabel 1:

Tabel 1. Perhitungan Manual *Dempster-Shafer*

	m2 {P1, P2, P3, P5}	m2{ $\Theta$ }
	0.7	0.3
m1 {P1, P3}	{P1, P3}	{P1, P3}
0.9	0.63	0.27
m1{ $\Theta$ }	{P1, P2, P3, P5}	$\emptyset$
0.1	0.07	0.03

{P1, P3} diperoleh dari irisan m1 dan m2, sedangkan nilainya hasil dari perkalian nilai m1 dan m2. selanjutnya dihitung nilai m3, karena nilai *evidential conflict* belum ada maka  $k = 0$ .

$$m3\{P1, P3\} = (0.63 + 0.27)/(1 - (0)) = 0.9$$

$$m3\{P1, P2, P3, P5\} = 0.07/(1 - (0)) = 0.07$$

Dari hasil perhitungan tersebut maka dapat disimpulkan kerusakan terjadi pada *server* dan *database*.

## 2.2. Perancangan Sistem

### 2.2.1. Kebutuhan Fungsional

Kebutuhan fungsional system dipaparkan dalam Tabel 2:

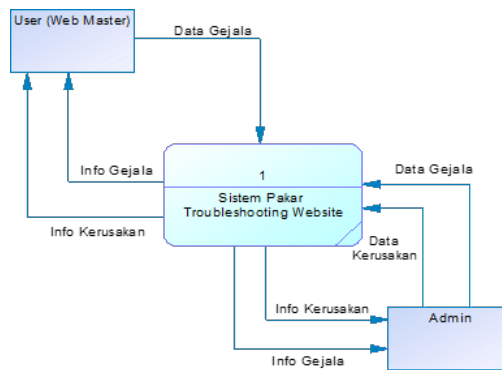
Tabel 2. Kebutuhan Fungsional

Pengguna	Kebutuhan Pengguna	Kebutuhan Pengguna
Admin	Mengelola data pakar	Fungsi mengelola data gejala, kerusakan dan rule
User (Webmaster)	Melakukan konsultasi	Fungsi menampilkan 18 pilihan gejala
	Memilih gejala yang dikeluhkan	Fungsi menampilkan gejala yang dipilih dengan checkbox
	Melihat jenis kerusakan beserta nilai persentasi	Fungsi menampilkan jenis kerusakan dan nilai persentasinya beserta gejala yang dipilih

Kebutuhan fungsional pada penelitian ini sebagai identifikasi untuk akses penggunaan dari system yang akan dibuat, setiap kategori pengguna kebutuhan dan fungsinya berbeda-beda.

2.2.2. Data Flow Diagram

Data Flow Diagram merupakan aliran data dari proses yang biasa disebut sebagai sistem informasi. Diagram aliran data juga menyediakan informasi tentang input dan output dari setiap entitas dan proses itu sendiri. Pengembangan DFD biasanya menggunakan cara berjenjang. Dimulai dari diagram konteks, DFD level 1, level 2, dan seterusnya sesuai dengan kompleksitas dari sistem yang akan dikembangkan [20] yang bisa dilihat pada Gambar 2.

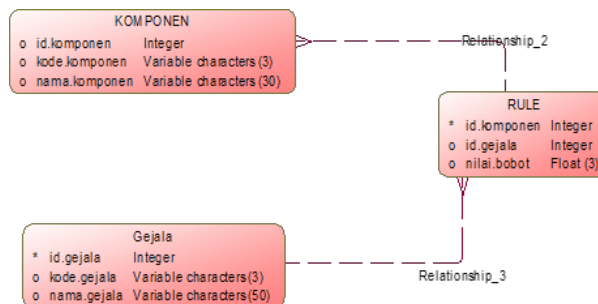


Gambar 2. Diagram Konteks Sistem (DFD)

Dalam DFD tersebut aliran data pada entitas user (*Web Master*) yang pertama adalah adanya info gejala yang akan dipilih oleh user, kemudian data gejala yang sudah dipilih tersebut menghasil info kerusakan berdasarkan proses sistem pakar. Sedangkan pada entitas admin yang pertama adalah data gejala dan data kerusakan yang diinput agar dapat tersimpan pada sistem, selain itu admin juga menerima info kerusakan dan info gejala yang dialami oleh user.

2.2.3. Perancangan Basis Data

Setelah melakukan perancangan sistem menggunakan *Data Flow Diagram*, langkah selanjutnya adalah merancang basis data berdasarkan tabel yang telah dirancang pada DFD bisa dilihat pada Gambar 3.



Gambar 3. Relasi Tabel Sistem



Dalam perancangan basis data sistem pakar untuk *troubleshoot website* ini menggunakan tiga entitas tabel yaitu Komponen, Gejala dan Rule, di mana entitas yang memiliki relasi yaitu id.komponen pada entitas komponen dan id.gejala pada entitas Gejala. Keduanya berada didalam sebuah entitas Rule dengan id.komponen menjadi *primary key*.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Analisi Kebutuhan

Sebelum menentukan hasil diagnosa kerusakan, dapat dilakukan pembuatan daftar gejala beserta kerusakannya. Proses diagnosa kerusakan pada *website* ini adalah berdasarkan gejala kerusakan yang dipilih oleh pengguna. Berikut merupakan data-data komponen kerusakan yang terjadi pada *website* berdasarkan penelitian yang dilakukan seperti pada Tabel 3:

Tabel 3. Jenis Kerusakan

Kode	Jenis Kerusakan
P1	Server
P2	Database
P3	Scripting
P4	Network
P5	Komputer Pengguna

Jenis kerusakan yang akan digunakan dalam *website* ini masing-masing diberikan kode awalan P agar mudah untuk diketahui, kerusakan tersebut diperoleh berdasarkan penelitian sebelumnya [2] yang kemudian di sesuaikan dengan pengetahuan dari pakar. Sedangkan data-data gejala yang ditampilkan apabila terjadi kerusakan pada *website* berdasarkan penelitian yang dilakukan seperti pada Tabel 4:

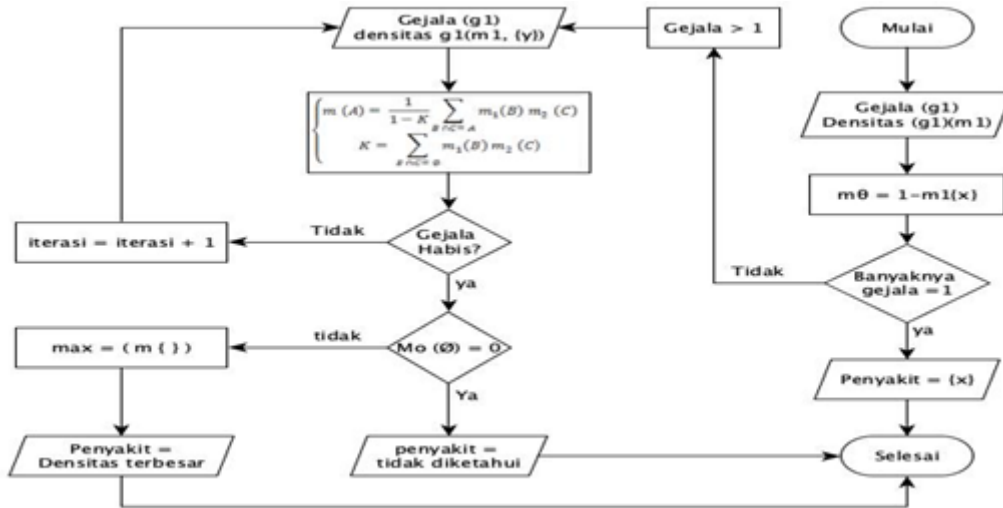
Tabel 4. Pengetahuan Gejala *Troubleshooting Website*

No	Kode	Gejala	Kerusakan					Nilai	
			P1	P2	P3	P4	P5	Belief	Plausibility
1	G1	Tidak bisa di ping	✓			✓		0.8	0.2
2	G2	Website blank	✓		✓			0.9	0.1
3	G3	Internal Server Error	✓	✓	✓		✓	0.7	0.3
4	G4	Service Unavailable	✓		✓		✓	0.8	0.2
5	G5	Connection Timeout	✓			✓		0.8	0.2
6	G6	Server Not Found		✓	✓			0.7	0.3
7	G7	Data tidak sempurna di load	✓	✓	✓	✓	✓	0.9	0.1
8	G8	Your clock is behind					✓	0.3	0.7
9	G9	Unable to connect	✓				✓	0.5	0.5
10	G10	Phishing or Malicious Content Warnings			✓			0.6	0.4
11	G11	Your connection is not private			✓	✓		0.5	0.5
12	G12	Network connection Refused	✓				✓	0.8	0.2
13	G13	the site can't be reached	✓			✓	✓	0.9	0.1
14	G14	establishing a database connection	✓	✓				0.9	0.1
15	G15	Server Forbiden	✓			✓		0.9	0.1

Daftar gejala yang digunakan adalah 15 kemudian dilakukan penghitungan dengan rumus *dempster-shafer* yang menghasilkan hubungan antara gejala dengan kerusakan yang terjadi disertai dengan nilai *belief* dan *plausibility* masing-masing. Seperti pada No 1 dengan Kode G1 dengan gejala Tidak Bisa di Ping merupakan gejala untuk kerusakan P1 dan P4 dengan *Belief* 0.8 dan *plausibility* 0.2.

#### 3.2. Flowchart Algoritma Dempster-Shafer

*Flowchart* adalah adalah bagan yang menunjukkan alir pekerjaan di dalam suatu program atau prosedur sistem secara struktural yang dijelaskan melalui logika Interface System [21]. Pada Gambar 4 merupakan *Flowchart* algoritma *Dempster-Shafer* yang menjelaskan alur yang terjadi mulai dari awal hingga tahap akhir secara sistematis.

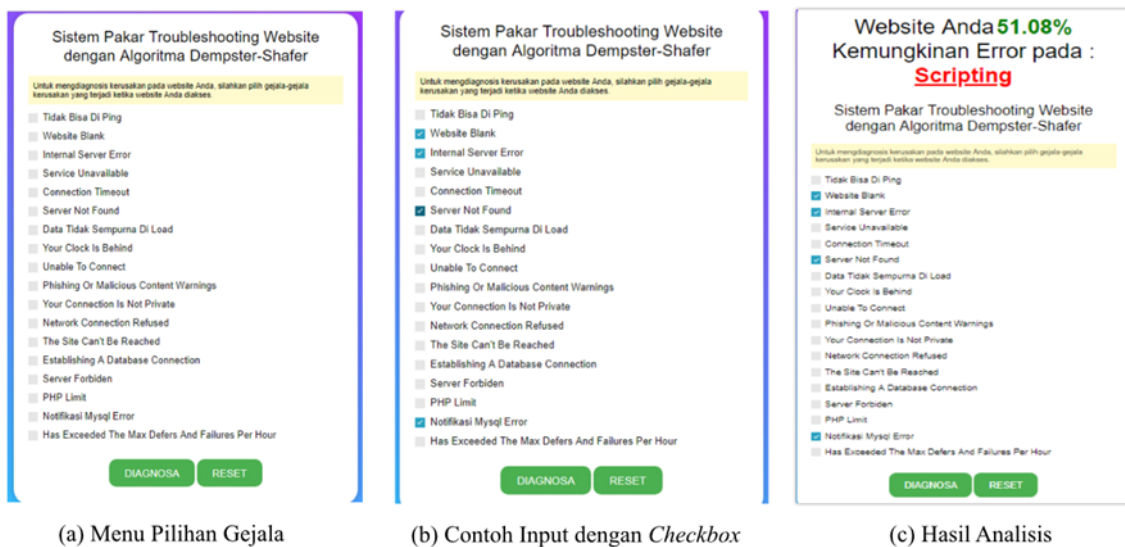


Gambar 4. Flowchart Algoritma Dempster-Shafer

Flowchart dimulai membaca basis pengetahuan, pengguna memilih gejala-gejala dengan mencentang checklist yang ditampilkan pada sistem. Selanjutnya akan dilakukan pengecekan pada inputan pengguna, apabila gejala yang dipilih hanya 1 maka hasil diagnosa kerusakan akan ditampilkan, namun apabila gejala kerusakan yang dipilih jumlahnya lebih dari 1 maka akan dilakukan perhitungan menggunakan metode Dempster-Shafer sehingga ditemukan nilai densitas kerusakan tertinggi yang mana nantinya akan menjadi kesimpulan dan presentase hasil diagnosa kerusakan pada *website*.

### 3.3. Interface Sistem

Hasil yang diperoleh dari penelitian ini yaitu suatu aplikasi sistem pakar yang dapat digunakan untuk mendiagnosa kerusakan yang terjadi pada suatu *website*. Aplikasi sistem pakar ini terdiri dari Menu pilihan gejala yang digunakan untuk menampilkan daftar gejala kerusakan, kemudian pengguna diminta untuk memilih gejala yang ada untuk dilakukan diagnosis kerusakan, setelah itu klik tombol diagnosa untuk mengetahui kemungkinan kerusakan sistem yang terjadi, untuk lebih jelasnya perhatikan Gambar 5.



Gambar 5. Interface Sistem

Halaman pertama merupakan informasi gejala yang disediakan oleh sistem untuk *user*, kemudian *user* dapat melakukan checklist terhadap gejala kerusakan *website*. Pada halaman terakhir merupakan jawaban dari sistem mengenai jenis kerusakan dan nilai persentasenya dari apa yang dialami oleh user berdasarkan gejala



yang sudah dipilih.

### 3.4. Pengujian Sistem

Dalam setiap melakukan perancangan sistem aplikasi, maka tahap terakhir sebelum dipublikasikan ke masyarakat pengguna, tentunya harus dilakukan pengecekan pada sistem yang dibuat. Pengujian terhadap sistem merupakan sebuah metode untuk mengetahui apakah sistem yang dibuat berfungsi dengan baik dan benar [22]. Pengujian sistem dilakukan dengan menggunakan *Blackbox Testing* dan Pengujian output berdasarkan pakar.

#### 3.4.1. Blackbox Testing

Metode pengujian *blackbox* merupakan pengujian untuk menunjukkan kesalahan pada sistem aplikasi seperti kesalahan pada fungsi sistem aplikasi, serta menu aplikasi yang hilang [22]. Pegujian *blackbox* mendemonstrasikan fungsi dari perangkat lunak yang beroperasi, dengan mengecek apakah input sudah bisa diterima dengan baik, dan hasil output sesuai dengan apa yang diharapkan. Uji coba menggunakan metode *blackbox* ini melakukan pengecekan pada integritas informasi eksternal dimana pengujian hanya memeriksa hasil output yang dihasilkan apakah sudah sesuai dengan apa yang diharapkan. Pada Tabel 5 dilakukan pengujian terhadap fungsionalitas sistem.

Tabel 5. Pengujian dengan *Blackbox*

No	Fungsi	Test Case	Output Yang Tampil	Kesesuaian
1	Konsultasi	Pilih checkbox gejala yang dialami dan pilih tombol Diagnosa	Sistem memproses input yang dipilih sesuai rumus	Sesuai
			Menampilkan hasil berupa jenis kerusakan dan tingkat keyakinan gejala dengan persentase	Sesuai
2	Reset	Pilih Tombol "Reset"	Sistem otomatis membersihkan checkbox yang telah diisi pada saat memilih gejala, hal tersebut dilakukan jika ingin memilih gejala ulang	Sesuai

Pengujian blackbox pada sistem ini dilakukan dengan memberikan test terhadap tombol yang tersedia apakah berjalan sesuai dengan apa yang diharapkan peneliti, dari mulai proses input hingga output berupa jenis kerusakan dan persentasenya. Pengujian lainnya yaitu dilakukan dengan melakukan reset terhadap sistem yang berjalan apakah otomatis akan membersihkan proses yang dilakukan sebelumnya. Dari pengujian yang dilakukan tersebut semua tombol sudah sesuai.

#### 3.4.2. Pengujian Output Berdasarkan Pakar

Setelah melakukan *design* dan implementasi sistem, langkah yang perlu dilakukan adalah pengujian output sistem. Pengujian sistem dilakukan dengan menyediakan gejala sesuai dengan gejala yang dimasukkan ke dalam sistem. Gejala yang terdapat di dalam sistem selanjutnya dikonsultasikan dengan pakar untuk mendapatkan persentase kebenaran nilai gejala yang diterapkan ke dalam sistem, dengan hasil analisa pakar. Pada Tabel 6 merupakan hasil dari pengujian sistem berdasarkan pakar.

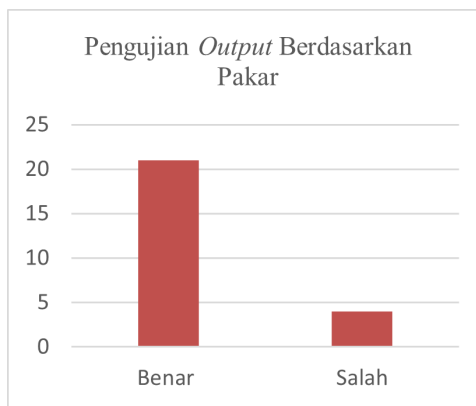
Tabel 6. Pengujian *Output* Berdasarkan Pakar

Nama Kasus	Rule Kasus	Hasil Sistem	Hasil Pakar	Keterangan
<b>K1</b>	G1^G2^G3	P1	P1	Benar
<b>K2</b>	G5^G9^G12^G13	P1	P1	Benar
<b>K3</b>	G3^G14^G17	P2	P2	Benar
<b>K4</b>	G6^G11^G7	P3	P1	Salah
<b>K5</b>	G11^G13^G16^G18	P1	-	Salah
<b>K6</b>	G11^G13^G15	P4	P4	Benar
<b>K7</b>	G9^G12	P1 & P5	P1 & P5	Benar
<b>K8</b>	G2^G4^G10^G11	P3	P3	Benar

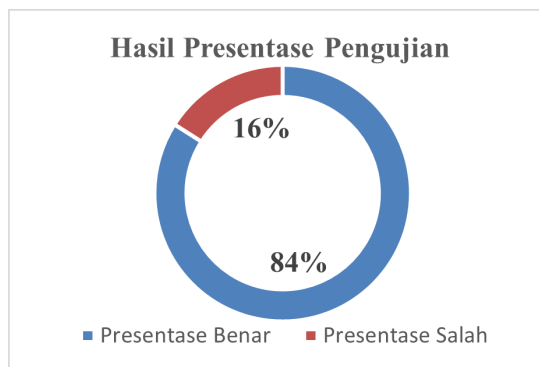
Nama Kasus	Rule Kasus	Hasil Sistem	Hasil Pakar	Keterangan
K9	G2^G5^G9	P1	P1	Benar
K10	G14^G17	P2	P2	Benar
K11	G2^G4^G10^G11	P3	P3	Benar
K12	G7^G8^G9	P1 & P5	P1 & P5	Benar
K13	G12^G15	P1	P1	Benar
K14	G16^G17	P1	P1	Benar
K15	G8^G9^G11^G12	P1 & P5	P1	Salah
K16	G11^G15	P4	P4	Benar
K17	G1^G7^G11	P4	P4	Benar
K18	G3^G6^G10^G14	P2	P2	Benar
K19	G1^G18	P1	P1	Benar
K20	G16^G17	P1	P1	Benar
K21	G1^G17	P1 & P5	P1	Salah
K22	G5^G10^G13	P1 & P5	P1 & P5	Benar
K23	G2^G10^G11^G17	P3	P3	Benar
K24	G2^G10	P3	P3	Benar
K25	G5^G6^G12	P1	P1	Benar

Pada pengujian untuk membandingkan output sistem dengan hasil dari pakar dilakukan sebanyak 25 kasus, setiap kasus memiliki rule jumlah gejala yang dipilih kemudian akan menghasilkan dua output yaitu dari sistem dan pakar, jika hasil sistem dan pakar memiliki jenis kerusakan yang sama maka keterangan kasus tersebut adalah benar.

Pada penelitian ini pengujian sistem dilakukan dengan menyediakan 25 kasus, jika merujuk penelitian sebelumnya dengan metode Dempster Shafer untuk diagnosa kerusakan jaringan LAN jumlah pengujian menggunakan satu kasus dengan hasil bernilai 64% [18]. Sementara pada penelitian ini menghasilkan presentase nilai akurat mencapai 84% dimana jumlah kasus yang memiliki keterangan benar adalah 21 sedangkan kasus dengan keterangan salah berjumlah 4. Pada Gambar 6 dan 7 hasil pengujian *output* berdasarkan pakar dan persentasenya.



Gambar 6. Grafik Pengujian *Output* Berdasarkan Pakar



Gambar 7. Grafik Persentase Pengujian

#### 4. KESIMPULAN

Dari perancangan sistem pakar menggunakan metode *dempster-shafer* ini penulis dapat mengambil beberapa kesimpulan, diantaranya adalah sistem pakar yang dirancang tersebut dapat menghasilkan diagnosa kerusakan pada suatu *website* berdasarkan kombinasi gejala terpilih. Kecepatan dalam melakukan diagnosa akan mempermudah pengguna dalam melakukan diagnosa kerusakan sisi *website* berdasarkan *rule* yang telah disediakan. Semakin banyak *rule* yang dimasukkan dalam basis pengetahuan maka akan meningkatkan akurasi dan ketepatan dalam melakukan diagnosa kerusakan. Pada pengujian dengan 25 kasus sistem pakar diagnosa dan *troubleshooting* pada *website* ini dapat menghasilkan persentase akurasi sebesar 84% terhadap analisa pakar. Saran untuk peneliti selanjutnya dapat melakukan kombinasi metode *dempster shafer* dengan metode lainnya untuk meningkatkan nilai akurasi, selain itu implementasi dapat dikembangkan dengan berbasis android sehingga memudahkan user untuk selalu mengakses sistem pakar ini.

#### DAFTAR PUSTAKA

- [1] A. A. Pramuditho and N. Trianto, "Pelatihan Peripheral dan Troubleshooting Computer Pada Siswa-Siswi SMKN 1 Palembang," *Jurnal Buana Informatika*, vol. 1, no. 2, p. 159, 2022.
- [2] M. Milanesio, C. Callegari, and P. Michiardi, "Network Level Perspective in Web Sessions Troubleshooting," *International Journal of Communication Systems*, vol. 32, no. 6, pp. 1–18, 2019.
- [3] B. H. Hayadi, *Sistem Pakar*. Yogyakarta: Deepublish, 2018.
- [4] R. Wahyuni, "Jurnal Aplikasi Sistem Pakar Untuk Diagnosa Penyakit Campak Dan Pencegahan Dengan Metode Forward Chaining Berbasis Web," *Jurnal Ilmu Komputer*, vol. 8 No. 2, pp. 53–59, 2019.
- [5] D. Aldo, "Sistem Pakar Diagnosis Hama Dan Penyakit Bawang Merah Menggunakan Metode Dempster Shafer," *Komputika : Jurnal Sistem Komputer*, vol. 9 No. 2, pp. 85–93, 2020.
- [6] N. A. Hasibuan and Alwin Fau, "Sistem Pakar Kombinasi Metode Certainty Factor dan Dempster Shafer," *Journal Of Information System Research (JOSH)*, vol. 3 No.2, pp. 85–90, 2022.
- [7] A. Y. Hidayat, "Sistem Pakar Diagnosa Gejala Penyakit Tuberkulosis Dengan Metode Dempster-Shafer (Studi Kasus : UPTD Puskesmas II Temon)," *KONSTELASI: Konvergensi Teknologi dan Sistem Informatika*, vol. 1 No. 2, pp. 442–453, 2021.
- [8] C. Nas, "Sistem Pakar Diagnosa Penyakit Tiroid Menggunakan Metode Dempster-Shafer," *Jurnal Teknologi dan Open Source*, vol. 2 No.1, pp. 1–14, 2019.
- [9] F. A. Mahesa and S. Sulindawaty, "Sistem Pakar Mendiagnosa Penyakit Epilepsi Menggunakan Metode Dempster Shafer," *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 4 No.6, pp. 415–424, 2021.
- [10] F. Nugroho and A. U. Bani, "Penerapan Metode Dempster Shafer Pada Sistem Pakar Diagnosa Penyakit Usus Halus," *Jurnal Media Informatika Budidarma*, vol. 6 No.1, pp. 243–250, 2022.
- [11] M. H. Basri, A. Mahmudi, and N. Vendyansyah, "Perbandingan Metode Dempster-Shafer dan Certainty Factor Untuk Diagnosis Penyakit Tanaman Terong (Studi Kasus Dusun Kejoren, Desa Gerbo, Kec. Purwodadi)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 4 No.1, pp. 230–238, 2020.
- [12] Y. Fernando, R. Napianto, and R. I. Borman, "Implementasi Algoritma Dempster-Shafer Theory Pada Sistem Pakar Diagnosa Penyakit Psikologis Gangguan Kontrol Impuls," *Insearch (Information System Research) Journal*, vol. 2 No.2, pp. 46–54, 2022.

- [13] H. Mayatopani, R. Subekti, N. Yudaningsih, and M. Sanwasih, “Pengembangan Sistem Pakar Diagnosa Gangguan Mental dengan Mesin Inferensi Menggunakan Algoritma Dempster-Shafer Theory,” *Jurnal Buana Informatika*, vol. 13 No.1, no. April, pp. 66–76, 2022.
- [14] R. A. Pamungkas and L. D. Farida, “Implementasi Dempster Shafer Untuk Deteksi Dini Gizi Buruk Pada Balita,” *Jurnal Pseudocode*, vol. 10 No.1, pp. 21–29, 2023.
- [15] S. Iswanti and R. N. Anggraeny, “Implementasi Metode Dempster-Shafer Pada Sistem Pakar Pendiagnosa Kerusakan Sepeda Motor,” *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, vol. 14 No.1, pp. 39–45, 2019.
- [16] I. I. P. Agus Setiawan, Erfan Karyadiputri, Nadiya Hijriana, “Sistem Pakar Diagnosa Penyakit Bakteri Dan Parasit Pada Ikan Gabus Menggunakan Metode Dempster Shafer,” *Technologia Jurnal Ilmiah*, vol. 13 No.1, pp. 16–22, 2022.
- [17] Aminun, B. Efendi, and A. Nasution, “Sistem Pakar Mendiagnosa Penyakit Stroke Transient Ischemic Attack ( TIA ) Dengan Menggunakan Metode Dempster Shafer Berbasis Web,” *JUTIF Jurnal Teknik Informatika*, vol. 3 No.4, pp. 1079–1087, 2022.
- [18] S. Sukmawati, R. Maulana, R. Holiyanti, and B. N. Sari, “Implementasi Metode Dempster Shafer Berbasis Web untuk Mendiagnosa Kerusakan Jaringan LAN,” *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 7 No.1, pp. 91–98, 2022.
- [19] Y. Zhang, W. Yan, G. S. Hong, J. F. H. Fuh, D. Wang, X. Lin, and D. Ye, “Data fusion analysis in the powder-bed fusion AM process monitoring by Dempster-Shafer evidence theory,” *Rapid Prototyping Journal*, vol. 28 No.5, pp. 841–854, 2022.
- [20] Y. Devi, R. Romindo, A. N. Sari, H. Tantriawan, E. E. Putri, M. A. Manuhutu, R. Turaina, D. Defiariany, N. E. Putri, T. Priyantoro, J. Jamaludin, J. Simarmata, R. Rismayani, and S. Aisa, *Analisis dan Perancangan Sistem Informasi*, Abdul Karim, Ed. Medan: Yayasan Kita Menulis, 2022.
- [21] R. A. Y. Manurung and A. D. Manuputty, “Perancangan Sistem Informasi Lembaga Kemahasiswaan Universitas Kristen Satya Wacana Salatiga,” *Jurnal SITECH : Sistem Informasi dan Teknologi*, vol. 3 No.1, pp. 9–20, 2020.
- [22] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and S. Suraya, “Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula,” *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 1 No.2, pp. 1–8, 2022.