

## IDENTIFIKASI PENYAKIT PADA BUDIDAYA IKAN AIR LAUT MENGUNAKAN METODE *RIPPLE DOWN RULES* (RDR)

Agus Cahyo Nugroho<sup>1</sup>, Suyoto<sup>2</sup>, Irya Wisnubadhra<sup>3</sup>

<sup>1,2,3</sup>Program Studi Magister Teknik Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta  
Jalan Babarsari No.44, Depok, Kec. Sleman, Daerah Istimewa Yogyakarta 55281  
E-mail: [agus@staff.ukdw.ac.id](mailto:agus@staff.ukdw.ac.id)<sup>1</sup>, [suyoto@mail.uajy.ac.id](mailto:suyoto@mail.uajy.ac.id)<sup>2</sup>, [iry@mail.uajy.ac.id](mailto:iry@mail.uajy.ac.id)<sup>3</sup>

### ABSTRAK

*Seiring perkembangan teknologi, dikembangkan sebuah sistem yang mampu mengadopsi proses dan cara berpikir manusia yaitu sistem pakar yang mengandung pengetahuan tertentu sehingga setiap orang dapat menggunakannya untuk memecahkan masalah bersifat spesifik yaitu permasalahan diagnosis penyakit ikan air laut. Tujuan dari penelitian ini adalah membangun sistem pakar untuk mendiagnosa penyakit ikan air laut dalam bentuk website menggunakan pemrograman PHP dengan database MySQL. Sistem pakar untuk mendiagnosa penyakit ikan air laut menggunakan metode Ripple Down Rules (RDR) ini bertujuan menelusuri gejala yang ditampilkan dalam bentuk pertanyaan – pertanyaan agar dapat mendiagnosa jenis penyakit dengan berbasis website. Sistem pakar berbasis web mampu mengenali jenis penyakit ikan air laut setelah melakukan konsultasi dengan menjawab beberapa pertanyaan – pertanyaan yang ditampilkan oleh aplikasi sistem pakar serta dapat menyimpulkan beberapa jenis penyakit yang diderita oleh ikan air laut. Data penyakit yang dikenali menyesuaikan rules (aturan) yang dibuat untuk dapat mencocokkan gejala-gejala penyakit ikan air laut.*

*Kata Kunci: sistem pakar, penyakit ikan laut, ripple down rules, forward chaining*

### ABSTRACT

*Along with the development of technology, people developed a system that capable of adopting processes and human thinking as an expert system that contains specific knowledge so that everyone can use it to solve a specific problem, namely the diagnosis of marine fish disease problem. The purpose of this study is to develop an expert system for diagnosing diseases of marine fish in the form of websites using PHP with a MySQL database. Expert system for diagnosing diseases of marine fish is using Ripple Down Rules (RDR) method has a goal to discover symptoms that appear in the form of questions that can diagnose the disease based on website. Web-based expert system is able to recognize types of marine fish disease after consultation by answering a few questions that are displayed by the application of expert systems and can infer some types of marine fish disease. Data disease that already known adapt to rules which are made for matching the symptoms of marine fish disease.*

*Keywords: expert system, marine fish disease, ripple down rules, forward chaining*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Sistem pakar adalah sistem yang menggabungkan pengetahuan, fakta, aturan dan teknik penelusuran untuk memecahkan masalah yang secara normal memerlukan keahlian seorang pakar (Durkin, 1994).

Sistem pakar yang ada saat ini mengalami permasalahan dalam pengembangan dan pemeliharaan sistem. Masalah itu disebabkan oleh akuisisi pengetahuan dari pakar yang hanya dilakukan pada awal pembangunan sistem pakar, sehingga membuat sistem tidak dapat mengatasi setiap masalah-masalah baru yang muncul karena tidak terdapat fakta-fakta dan aturan untuk mengatasi permasalahan tersebut. Masalah lain yang terjadi adalah sistem pakar tidak dapat mengubah fakta-fakta dan aturan yang ada, karena dapat merusak basis aturan yang sudah terbentuk sebelumnya.

Penulis mencoba melakukan sebuah penelitian dengan menggunakan metode *Ripple Down Rules* (RDR) untuk menyelesaikan masalah yang terjadi. Metode *Ripple Down Rules* (RDR) merupakan suatu metode untuk akuisisi pengetahuan baru dari seorang pakar ke dalam sistem dan berdasarkan akuisisi pengetahuan tersebut sistem secara mandiri membuat aturan-aturan baru, dimana posisi aturan baru tersebut disesuaikan dengan basis aturan dari aturan-aturan yang sudah ada. Selain itu jika pakar menganggap pengetahuan yang dimasukkan ke dalam sistem tidak sesuai dengan pemikirannya, maka metode *Ripple Down Rules* (RDR) ini juga memperbolehkan pakar untuk menghapus ataupun mengganti pengetahuan yang sudah ada sehingga sistem ini dapat selalu memiliki pengetahuan yang benar dan terbaru yang dimiliki oleh pakar.

Penulis memilih studi kasus di bidang budidaya ikan laut dikarenakan pelaksanaan pembangunan sektor budidaya ikan laut pada dasarnya dapat dilakukan dengan cepat, efektif dan menguntungkan

karena memiliki berbagai kekuatan, peluang dan akses pasar yang cukup luas. Secara fisik, (Kusumastanto, 2003), menyatakan bahwa Indonesia memiliki potensi yang melimpah untuk pembangunan industri perikanan budidaya. Potensi tersebut meliputi wilayah perairan nasional seluas 3,1 juta km<sup>2</sup>, luas *Zona Ekonomi Eksklusif (ZEE)* sekitar 2.8 juta km<sup>2</sup>, panjang garis pantai mencapai 81.000 km dan memiliki jumlah pulau sebanyak 17.499 buah yang dapat digunakan untuk penguatan kapasitas produksi budidaya ikan laut. Berdasarkan data statistik, (KKP, 2009), pemanfaatan potensi budidaya laut masih berkisar 0,3% dengan 12,502,396 Ha lahan potensi yang masih dapat dikembangkan. Kenaikan rata-rata produksi budidaya ikan laut dalam kurun waktu 2009-2010 juga meningkat sekitar 20% dengan nilai produksi mencapai 10,3 Triliun (KKP, 2009). Kondisi ini menunjukkan bahwa peluang investasi dan pengembangan industri budidaya ikan laut di Indonesia cukup menjanjikan.

Salah satu hambatan utama dalam keberlanjutan produksi budidaya adalah kematian yang disebabkan oleh infeksi mikroorganisme patogen dan degradasi kualitas lingkungan. Kondisi ini berkorelasi positif dengan semakin intensifnya sistem budidaya yang dikembangkan (Cao *et al.*, 2007). Secara global, potensi kerugian ekonomi akibat wabah penyakit yang ditimbulkan oleh infeksi mikroorganisme patogen cukup signifikan dan berdampak kepada jumlah produksi, keuntungan dan keberlanjutan sistem budidaya. Kerugian ekonomi pada industri budidaya akibat wabah penyakit diperkirakan mencapai US\$ 9 miliar per tahun (Subasinghe *et al.*, 2001) dan berdampak kepada penurunan jumlah produksi ikan budidaya di seluruh dunia (Hill, 2005). Di Indonesia, (Zafran *et al.*, 1997) menyatakan bahwa infeksi oleh parasit *Benedenia*, *Neobenedenia*, *Diplectanum*, *Pseudorhabdosynochus*, *Haliotrema*, *Trichodina*, *Lepeophtheirus*, dan *Cryptocaryon irritans* telah menjadi wabah umum pada ikan Kerapu. Sementara, infeksi yang disebabkan oleh *Iridovirus* (Fris Johnny dan Des Roza, 2009) dan *Nervous Necrosis Virus (NNV)* (Sukadi, 2004) telah menjadi hambatan tersendiri bagi peningkatan jumlah produksi. Kondisi ini membuktikan bahwa masalah penyakit dalam perkembangan budidaya ikan laut memerlukan perhatian yang sangat serius.

## 1.2 Rumusan Masalah

Bagaimana mengidentifikasi penyakit pada budidaya ikan air laut menggunakan metode *Ripple Down Rules (RDR)*.

## 1.3 Manfaat Penelitian

Manfaat penelitian ini :

- Sebagai acuan bagi masyarakat pembudidaya dan pengambil kebijakan untuk mengembangkan

teknik diagnosa dan pengendalian penyakit yang tepat, efektif dan sistematis.

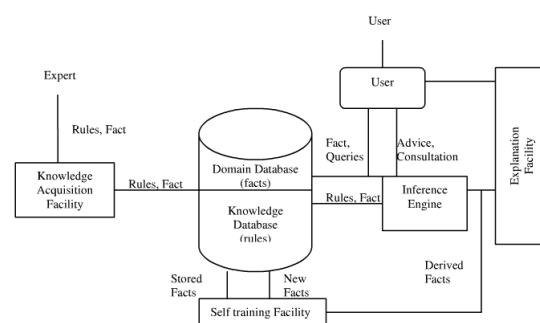
- Sebagai acuan bagi petugas perikanan baik daerah maupun pusat untuk menumbuhkan, mengembangkan dan meningkatkan kewaspadaan terhadap timbulnya penyakit pada budidaya ikan laut di Indonesia.

## 2. PEMBAHASAN

Sistem pakar adalah program komputer cerdas yang menggunakan pengetahuan dan prosedur-prosedur inferensi untuk menyelesaikan masalah-masalah yang membutuhkan seorang pakar dalam menemukan solusinya. Oleh sebab itu sistem pakar merupakan sebuah sistem yang mengemuliskan kemampuan membuat keputusan dari seorang pakar (Giarratano, 2005).

Arsitektur dalam sistem pakar memiliki variasi bentuk yang berbeda-beda, yang didasari oleh komponen-komponen yang berbeda juga. Tetapi pada umumnya komponen dasar yang pasti ada dalam sistem pakar antara lain, antarmuka pengguna, basis pengetahuan, akuisisi pengetahuan dan mekanisme inferensi (*inference engine*). Mekanisme inferensi merupakan pusat dari sistem yang mengatur jalannya sistem pakar tersebut. *Knowledge base* merupakan tempat penyimpanan pengetahuan dari seorang pakar. Antarmuka pengguna merupakan sarana berkomunikasi antara pengguna dengan sistem. Akuisisi pengetahuan digunakan untuk menyimpan pengetahuan dari seorang pakar ke dalam basis pengetahuan.

### 2.1 Arsitektur Sistem Pakar



**Gambar 1. Arsitektur sistem pakar (James Martin & Steve Osman, 1998, halaman 30)**

Penjelasan setiap komponen yang terdapat pada Gambar 1 di atas sebagai berikut :

#### 1. Akuisisi Pengetahuan (*Knowledge Acquisition Facility*)

Dahulu pengaturan basis pengetahuan merupakan hal yang sulit dan menyita banyak waktu. Setiap fakta, aturan dan hubungan harus dimasukkan ke dalam basis pengetahuan. Namun perangkat lunak yang ada sekarang memperbolehkan pakar untuk membuat dan mengubah basis

pengetahuan melalui *Knowledge Acquisition Facility (KAF)*. *Knowledge Acquisition Facility (KAF)* ini bertujuan untuk menyediakan suatu dialog antara sistem pakar dengan seorang pakar untuk memperoleh pengetahuan, fakta dan aturan yang kemudian menempatkannya dalam *knowledge base*, sehingga membuat *knowledge base* menjadi lebih mudah dalam pengaturan dan pemeliharaan.

## 2. Basis Pengetahuan (*Knowledge Base*)

Komponen ini merupakan bagian yang terdapat dalam proses inferensi yang bertujuan untuk menyimpan informasi dan aturan-aturan penyelesaian suatu pokok bahasan masalah beserta atributnya. *Knowledge Base* mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah yang dapat berasal dari pakar, jurnal, majalah, dan sumber pengetahuan lain. Basis pengetahuan disusun atas dua elemen dasar yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu. Sedangkan aturan merupakan informasi tentang cara memperoleh fakta baru dari fakta yang telah diketahui.

## 3. Mekanisme Inferensi (*Inference Engine*)

Mesin inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan dari pakar untuk menganalisa data dan menghasilkan suatu kesimpulan atau hasil akhir yang direpresentasikan melalui *user interface* kepada pengguna. Komponen ini memiliki tugas utama untuk mengevaluasi kondisi (*condition*) dan memeriksa semua kondisi dalam sebuah aturan (*rule*) telah terpenuhi. Mekanisme inferensi yang terdapat dalam penelitian ini menggunakan algoritma *forward chaining* dalam penyusunan aturannya, dimana dari gejala-gejala yang diberikan oleh user dan kemudian mendapatkan hasil atau kesimpulan berupa penyakit yang diderita ikan laut tersebut.

## 4. Fasilitas Penjelas (*Explanation Facility*)

Komponen tambahan ini akan meningkatkan kemampuan dari sistem pakar yaitu untuk menggambarkan penalaran sistem kepada pengguna (*user*). Dimana komponen ini berfungsi untuk menjelaskan kepada pengguna tentang bagaimana kesimpulan diambil oleh sistem pakar.

## 5. Antarmuka Pengguna (*User Interface*)

Komponen dalam sistem pakar yang digunakan untuk berkomunikasi antara sistem dengan pengguna (*user*). Antarmuka menerima informasi yang diberikan oleh pengguna lalu mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem lalu menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna.

## 6. Memori Kerja (*Working Memory*)

Komponen ini menyimpan sementara setiap fakta jawaban dari pengguna yang kemudian digunakan oleh mesin inferensi untuk membandingkan fakta tersebut dengan aturan yang

ada dalam basis aturan dengan menggunakan metode *forward chaining* yang nantinya menghasilkan jawaban berupa penyakit yang dicari oleh *user*.

## 7. Sistem Pembelajaran (*Self Training Facility*)

Komponen dalam sistem pakar yang berguna untuk pembelajaran sistem secara mandiri, sehingga dapat mengolah aturan (*rule*) dan fakta (*fact*) dengan benar. Komponen ini merupakan fasilitas untuk mengatur posisi dari pengetahuan yang didapat dari pakar ke dalam basis pengetahuan (*knowledge base*), dimana pengetahuan tersebut secara mandiri dikonversikan oleh sistem ke dalam bahasa yang dapat dibaca oleh komputer. Metode *Ripple Down Rules (RDR)* merupakan bagian dari *self training facility* dimana metode ini akan melakukan pembelajaran secara mandiri terhadap suatu fakta yang diberikan oleh pengguna, yang mengubahnya menjadi bentuk aturan-aturan agar dapat dilakukan proses inferensi berdasarkan aturan tersebut.

### 2.2 *Ripple Down Rules (RDR)*

*Ripple Down Rules (RDR)* pada awalnya adalah teknik akuisi pengetahuan secara umum dimana area aplikasi yang dimaksud di sini adalah kemampuan untuk menambahkan komentar klinis atau interpretasi laporan laboratorium untuk membantu dokter dalam memberi rujukan. Artinya, dokter yang memerintahkan tes darah patologi kimia, menerima tidak hanya hasil laboratorium tetapi saran dari ahli patologi terhadap interpretasi hasil pengujian, pengujian lebih lanjut yang mungkin diperlukan dan sebagainya. Banyak laporan patologi berisi semacam komentar sederhana dan seadanya; tujuan dari penggunaan *Knowledge Base System (KBS)* adalah untuk memberikan komentar jauh lebih rinci yang diberikan oleh ahli patologi di klinik pengelolaan pasien tertentu.

Keuntungan dari area ini untuk sistem pakar atau teknologi *Artificial Intelligence (AI)* lainnya adalah bahwa tidak ada permintaan atau harapan terhadap dokter untuk menerima laporan. Dokter tidak harus berinteraksi dengan sistem, atau mengubah cara operasionalnya. Dokter bisa memilih untuk mengagendakan tes diagnostik untuk pasien, melihat laporan, termasuk memberikan komentar interpretatif.

Tentu saja, kualitas komentar akan menjadi penting terutama apakah dokter memperhatikan komentar tersebut, tetapi kualitas komentar murni tergantung pada tingkat keahlian dari sistem, bukan pada isu-isu integrasi ke dalam alur kerja klinis. Menurut laporan (Buchanan, 2005) mengenai sistem pakar yang rutin digunakan untuk mencatat bahwa tiga dari empat sistem medis pertama rutin digunakan untuk menyediakan interpretasi klinis dari hasil tes diagnostik.

*Ripple Down Rules (RDR)* awalnya dikembangkan untuk menangani masalah pemeliharaan salah satu sistem pakar medis pertama.

Mereka pertama kali diuji dalam bidang kedokteran di sistem *PEIRS*. Namun, dalam studi ini ada domain seorang ahli tunggal yang sangat erat terlibat dalam pengembangan dan penggunaan sistem, sehingga selalu menjadi pertanyaan apakah teknik ini akan berguna di tangan yang berbeda. Telah dilakukan berbagai evaluasi untuk jenis masalah yang berbeda, tapi ini semua dilakukan dalam konteks penelitian.

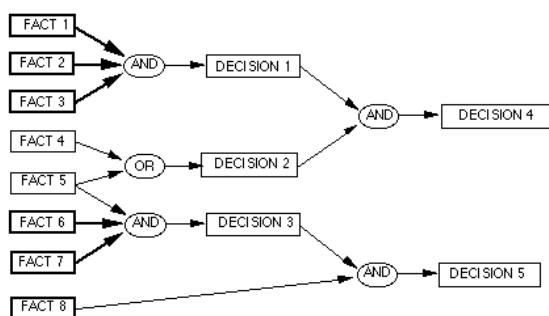
Pada perkembangannya *Ripple Down Rules (RDR)* merupakan strategi dalam mengembangkan sistem secara bertahap dimana sistem tersebut sudah digunakan. Saat sistem tidak memberikan respon yang benar maka perubahan perlu dilakukan tanpa mempengaruhi kompetensi sistem. Perubahan harus dapat dilakukan dengan mudah dan cepat serta kesulitan dalam melakukan perubahan tidak boleh meningkat meskipun sistem berkembang.

Berbagai sistem *Ripple Down Rules (RDR)* komersial telah dikembangkan untuk berbagai macam aplikasi. Riset telah membuktikan penggunaan *Ripple Down Rules (RDR)* di berbagai aplikasi seperti : pengklasifikasian masalah, konfigurasi dan pencocokan parameter, pengolahan teks, pengolahan citra, pencarian heuristik dan pencocokan algoritma genetika.

### 2.3 Forward Chaining

Metode yang digunakan dalam penelitian ini adalah penalaran maju (*forward chaining*). Penalaran maju (*forward chaining*) adalah metode inferensi yang memulai dengan menyusun fakta yang diketahui dan fakta-fakta baru dengan menggunakan aturan yang memiliki premis sesuai dengan fakta yang diketahui, dan proses ini berlanjut sampai sebuah kesimpulan tercapai atau sampai tidak ada aturan yang sesuai dengan fakta yang diketahui (Durkin, 1994).

Aturan-aturan yang ada dalam metode ini diuji satu demi satu dalam suatu urutan tertentu. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisi benar, maka aturan itu disimpan kemudian aturan berikutnya diuji. Namun sebaliknya jika kondisi salah, aturan tidak disimpan. Proses ini akan berulang sampai seluruh aturan yang ada telah teruji dengan berbagai kondisi. Hal ini bisa kita lihat di Gambar 2.



Gambar 2. Forward Chaining (Bratko ed. 4, bab 15, hal 343)

Jika ditemukan identifikasi suatu kasus tidak benar, maka pakar hanya perlu memasukkan fakta yang benar untuk diletakkan pada bagian fakta yang salah tadi, lalu sistem akan membentuk aturan yang baru berdasarkan fakta yang diberikan oleh pakar dan juga aturan tersebut akan ditandai sebagai *updating rule*. *Updating rule* adalah penambahan suatu aturan pengecualian tanpa mengubah susunan dan kualitas dari *rule-rule* yang telah ada sebelumnya. Jika suatu *rule* telah mengalami *updating rule*, maka *user* akan melihat tampilan terbaru dari kasus tersebut.

Contoh pengimplementasian metode *Ripple Down Rules (RDR)* pada kasus ini bisa kita lihat pada Gambar 3 di bawah :

Aturan awal yang terbentuk dalam sistem

1. IF perilaku berenang ikan yang tidak beraturan
2. AND beberapa ikan tenggelam ke dasar bak
3. AND kemudian mengapung lagi di permukaan
4. AND pembengkakan gelembung renang
5. AND letargik
6. AND warna tubuh terlihat lebih gelap
7. AND hilang nafsu makan
8. Then Penyakit yang diderita *Nervous Necrosis Virus (NNV)*

Gambar 3. Aturan awal dalam implementasi *Ripple Down Rules (RDR)*

Lalu pakar ingin memperbaiki gejala yang ada dalam sistem dan menggantinya menjadi gejala baru, seperti dalam kasus ini pakar ingin mengubah gejala letargik menjadi gerakan lemah.

Pada Gambar 4 berikut ini pengimplementasian perubahan gejala yang dimasukan oleh pakar :

1. Sistem akan menunjukkan semua yang berhubungan dengan penyakit yang akan diubah :

Kode Penyakit : P025  
 Nama penyakit : *Nervous Necrosis Virus (NNV)*  
 Kode gejala : G001  
 Nama gejala : perilaku berenang ikan yang tidak beraturan  
 Kode Gejala : G002  
 Nama Gejala : beberapa ikan tenggelam ke dasar bak  
 Kode gejala : G003  
 Gejala : kemudian mengapung lagi di permukaan  
 Kode gejala : ...  
 Gejala : ...

2. Setelah itu sistem menanyakan gejala lama apa yang akan diubah di dalam sistem ini :

Kode penyakit : P025  
 Penyakit : *Nervous Necrosis Virus (NNV)*  
 Kode Gejala : G005  
 Gejala lama yang akan diubah : letargik

3. Lalu sistem akan meminta aturan baru yang akan ditambahkan ke dalam sistem pakar :

Gejala Baru : gerakan lemah

4. Setelah itu sistem akan melakukan pelacakan basis aturan yang berhubungan dengan *Nervous Necrosis Virus (NNV)* dan letargik dan mengubah gejala lama tersebut menjadi gerakan lemah.

Gambar 4. Impementasi perubahan gejala

Setelah itu sistem menyusun bentuk aturan yang baru seperti pada Gambar 5 di bawah :

1. IF perilaku berenang ikan yang tidak beraturan
2. AND beberapa ikan tenggelam ke dasar bak
3. AND kemudian mengapung lagi di permukaan
4. AND pembengkakan gelembung renang
5. AND gerakan lemah
6. AND warna tubuh terlihat lebih gelap
7. AND hilang nafsu makan
8. Then Penyakit yang diderita *Nervous Necrosis Virus (NNV)*

**Gambar 5. Implementasi aturan baru**

## 2.4 Bahan dan Alat

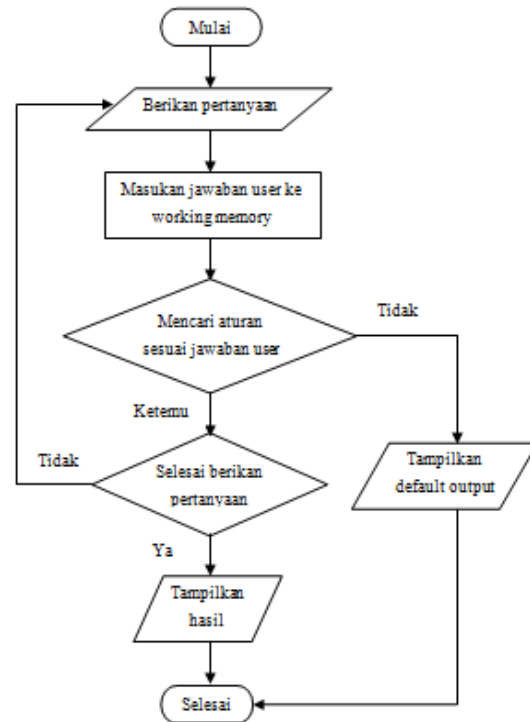
Bahan yang digunakan untuk pembuatan sistem pakar ini adalah beberapa buku yang menjelaskan tentang penyakit ikan air laut, serta pengetahuan yang diperoleh dari jurnal yang dipublikasikan oleh Bapak Romi Novriadi seorang pakar di bidang penyakit ikan air laut dari Kementerian Kelautan dan Perikanan pada tahun 2014.

Alat yang digunakan dalam penelitian ini adalah sebuah komputer dan beberapa perangkat lunak pendukung seperti : *Microsoft Word, Editplus, Bootstrap, WAMP server, PHP dan MySQL.*

## 2.5 Algoritma atau Program

### a. Diagram Alur Proses Konsultasi Pada Sistem

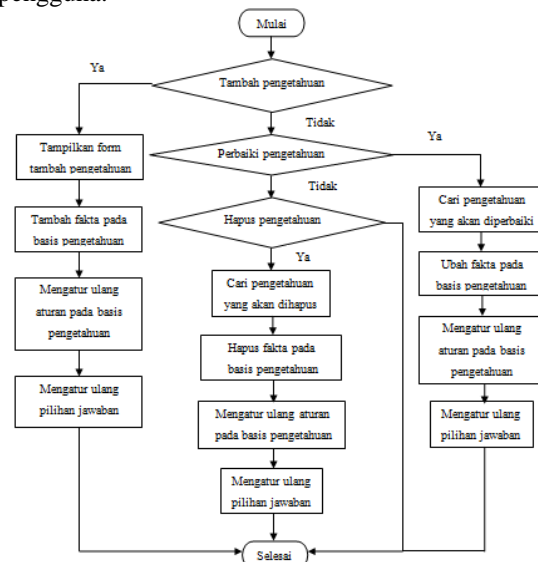
Diagram alur pada Gambar 6 menjelaskan tentang alur dari cara kerja sistem yang akan dibuat. Ketika pengguna atau pakar memilih menu konsultasi. Sistem akan mengajukan pertanyaan-pertanyaan yang berbeda yang akan ditampilkan satu persatu sesuai dengan basis aturan yang ada. Jawaban dari pengguna akan dimasukkan ke dalam *working memory* lalu sistem akan melakukan pengecekan jawaban dengan aturan yang ada dalam basis aturan. Jika aturan dengan jawaban pada *working memory* sesuai, maka akan ditampilkan pertanyaan selanjutnya sesuai dengan basis aturan yang ada. Ketika pertanyaan telah selesai diajukan kepada pengguna, maka sistem akan menampilkan hasil sesuai dengan jawaban yang dimasukkan oleh pengguna, namun jika tidak ditemukan aturan yang sesuai maka akan ditampilkan *default output* pada user.



**Gambar 6. Diagram alur proses konsultasi pada sistem**

### b. Diagram alur penerapan metode *Ripple Down Rules (RDR)*

Diagram Alur ini menjelaskan alur dari cara kerja sistem dalam penerapan metode *Ripple Down Rules (RDR)* yaitu memperbaiki dan menghapus fakta dan aturan yang ada dalam basis data dan juga pengetahuan pada pertanyaan yang diajukan kepada pengguna.



**Gambar 7. Diagram alur *Ripple Down Rules (RDR)***

Gambar 7 menjelaskan tentang kerja metode *Ripple Down Rules (RDR)* dalam sistem pakar ini. Ketika pakar telah berhasil melakukan *login* ke dalam sistem maka pakar akan diarahkan ke dalam

menu konsultasi khusus untuk pakar. Menu ini memiliki beberapa tombol yang dapat digunakan oleh pakar untuk menuju menu berikutnya. Metode *Ripple Down Rules (RDR)* terjadi ketika pakar memilih menu tambah, perbaiki ataupun menghapus pengetahuan, dimana jika pakar memilih menu tambah maka sistem akan menampilkan *form* untuk menambah pengetahuan lalu menambahkan pengetahuan tersebut ke dalam sistem. Lalu jika pakar memilih menu perbaiki ataupun hapus pengetahuan, sistem akan mencari pengetahuan lama yang ingin diubah dan mengubah pengetahuan tersebut menjadi pengetahuan baru yang dimasukan oleh pakar. Setelah penambahan, perbaiki ataupun penghapusan pengetahuan telah selesai dilakukan oleh pakar. Sistem akan membangun suatu aturan baru berdasarkan pengetahuan yang dimasukan oleh pakar lalu mengubahnya menjadi bentuk aturan-aturan baru dan memasukannya ke dalam basis aturan. Lalu sistem akan melakukan pengaturan ulang aturan-aturan yang ada dalam basis aturan.

## 2.6 Perancangan Basis Pengetahuan

Basis pengetahuan yang terdapat dalam sistem pakar ini menyimpan fakta yang berhubungan dengan penyakit ikan air laut dan gejalanya yang digunakan untuk mencari sebuah kesimpulan. Fakta tersebut didapat dari hasil wawancara dengan pakar dan sumber lain seperti buku, jurnal dan artikel. Selanjutnya diterjemahkan ke dalam bahasa komputer dan disimpan ke dalam basis data. Fakta yang ada dalam sistem pakar ini terdiri dari 5 kategori yaitu bagian gejala, gejala, nama hama penyakit, jenis hama penyakit, dan nama penyakit yang bisa kita lihat pada Tabel 1, 2, 3, 4 dan 5. Selanjutnya fakta tersebut digunakan untuk membangun pohon keputusan dan menyusun basis aturan untuk mendapatkan solusi yang dibutuhkan oleh sistem.

**Tabel 1. Kode dan bagian dari gejala**

<i>Kode</i>	<i>Bagian_gejala</i>
B001	Kepala
B002	Tubuh
B003	Sirip ekor
B004	Sirip punggung
B005	Sirip anal
B006	Insang
B007	Gelembung renang
B008	Mata
B009	Perut
B010	Limpa
B011	Saluran pencernaan
B012	Sistem saraf pusat
B013	Mukus
B014	Hati
B015	Ovarium
B016	Serabut otot

<i>Kode</i>	<i>Bagian_gejala</i>
B017	Rektum
B018	Uretra
B019	Pembuluh darah
B020	Sisik
B021	Tutup insang
B022	Usus
B023	Jantung
B024	Tingkah laku

**Tabel 2. Kode dan gejala**

<i>Kode</i>	<i>Bagian</i>	<i>Gejala</i>
G001	B001	Bintik-bintik putih di kepala.
G002	B001	Kerusakan sisik di atas kepala.
G003	B001	Luka dalam yang berwarna kemerah-merahan dan dapat masuk ke dalam sampai ke bagian tengkorak kepala ikan.
G004	B002	Warna tubuh terlihat lebih gelap.
G005	B002	Terdapat cairan keruh pada rongga tubuh.
G006	B002	Borok pada permukaan tubuh.
G007	B002	Bintik putih pada permukaan tubuh.
G008	B002	Titik putih atau keabuan pada permukaan tubuh.
G009	B002	Pendarahan pada permukaan tubuh.
G010	B002	Warna tubuh berubah menjadi pucat.
G011	B003	Sirip ekor gripis.
G012	B003	Bintik-bintik putih pada

<b>Kode</b>	<b>Bagian</b>	<b>Gejala</b>
		sirip ekor.
G013	B003	Sirip ekor mengalami erosi berat.
G014	B004	Sirip punggung gripis.
G015	B005	Sirip anal mengalami erosi berat
G016	B006	Insang mengalami kerusakan.
G017	B006	Insang mengalami pembelahan.
G018	B006	Bintik putih pada insang.
G019	B006	<i>Hyperplasia</i> sekunder dan <i>hipertropi epitel</i> pada insang.
G020	B006	Bintik putih atau abu-abu pada insang.
G021	B006	Insang bengkak dan pucat.
G022	B006	Tutup insang terbuka.
G023	B006	Produksi lendir insang berlebihan.
G024	B006	Filamen insang rusak.
G025	B006	Pendarahan ( <i>Hemoragik</i> ) pada insang.
G026	B006	Borok pada insang.
G027	B007	Pembengkakan gelembung renang.
G028	B008	Mata menonjol.
G029	B008	Mata membengkak.
G030	B008	Mata <i>exophthalmia</i> .
G031	B008	Mata menjadi buta.
G032	B008	Kerusakan syaraf retina.
G033	B009	Perut membengkak.
G034	B009	Keluar cairan kuning saat

<b>Kode</b>	<b>Bagian</b>	<b>Gejala</b>
		perut ditekan.
G035	B010	Limpa membesar.
G036	B011	Saluran pencernaan memerah.
G037	B011	Deformitas saluran pencernaan.
G038	B012	Kerusakan sistem saraf pusat.
G039	B012	Serangan pada sistem saraf pusat.
G040	B013	Produksi mukus yang berlebihan.
G041	B014	Deformitas hati.
G042	B014	Gumpalan granula pada hati.
G043	B015	Deformitas ovarium.
G044	B016	Infeksi serabut otot sehingga terisi kista.
G045	B017	Infeksi pada rektum.
G046	B018	Infeksi pada uretra.
G047	B019	Infeksi pada pembuluh darah.
G048	B020	Sisik mengalami kerontokan.
G049	B021	Tutup insang selalu terbuka.
G050	B022	Gumpalan granula pada usus.
G051	B023	Gumpalan granula pada jantung.
G052	B024	Tingkah laku berenang yg abnormal pada permukaan air.
G053	B024	Menggosokan badan.
G054	B024	Penurunan nafsu makan.
G055	B024	Berenang sendirian kemudian tenggelam.

Kode	Bagian	Gejala
G056	B024	Hilang keseimbangan.
G057	B024	Tingkat respirasi cepat.
G058	B024	Berkumpul di dekat sumber aerasi.
G059	B024	Letargik (sekarat dg gerakan lemah).
G060	B024	Kematian setelah 8-10 hari terkena virus.

**Tabel 3. Kode dan nama hama penyakit**

Kode	Nama_Hama
NH001	<i>Piscine nodavirus</i>
NH002	<i>Red Seabream Iridovirus</i>
NH003	<i>Sleepy Grouper Iridovirus</i>
NH004	<i>Ranavirus</i>
NH005	<i>Vibrio</i>
NH006	<i>Streptococcus</i>
NH007	<i>Flexibacter spp</i>
NH008	<i>Pseudomonas</i>
NH009	<i>Amyloodinium ocellatum</i>
NH010	<i>Trichodina sp</i>
NH011	<i>Cryptocaryon irritans</i>
NH012	<i>Glugea sp</i>
NH013	<i>Pleisthophora sp</i>
NH014	<i>Parasit Diplectanum</i>
NH015	<i>Parasit Benedenia</i>
NH016	<i>Haliotrema spp</i>
NH017	<i>Caligus sp</i>
NH018	<i>Zeylanicobdella arugamensis</i>
NH019	<i>Ichthyosporidium sp</i>
NH020	<i>Saprolegni sp</i>

**Tabel 4. Kode dan jenis hama penyakit**

Kode	Jenis_Hama
H001	Virus
H002	Bakteri
H003	Parasit
H004	Jamur

**Tabel 5. Kode dan nama penyakit**

Kode	Nama_Hama
P001	<i>Nervous Necrosis Virus (NNV)</i>
P002	<i>Red Seabream Iridovirus Disease (RSID)</i>
P003	<i>Sleepy Grouper Disease (SGD)</i>
P004	<i>Grouper</i>

	<i>Iridovirus Disease (GID)</i>
P005	<i>Vibriosis</i>
P006	<i>Red Boil Disease (RBD)</i>
P007	Infeksi <i>Flexibacter</i>
P008	<i>Pseudomonad hemorrhagic septicemia</i>
P009	Infeksi <i>Dinoflagellata</i>
P010	<i>Trikhodiniasis</i>
P011	<i>Cryptocaryonosis</i>
P012	Infeksi <i>Mikrosporidia Glugea</i>
P013	Infeksi <i>Mikrosporidia Pleistophora</i>
P014	Penyakit <i>Diplectanum</i>
P015	Penyakit <i>Benedenia</i>
P016	Penyakit <i>Haliotrema</i>
P017	Penyakit <i>Crustacea</i>
P018	Penyakit <i>Annelida</i>
P019	<i>Ichthyosporidiosis</i>
P020	<i>Saprolegniasis</i>

## 2.7 Implementasi

### a. Menu Konsultasi

Menu konsultasi adalah menu yang tampil ketika pakar dan pengguna menekan tombol mulai pada menu pakar atau menu pengguna seperti pada Gambar 8 di bawah. Menu ini digunakan untuk berkonsultasi dengan sistem tentang gejala-gejala yang ditemui pada ikan air laut untuk menemukan penyakit yang menyerang ikan air laut tersebut.



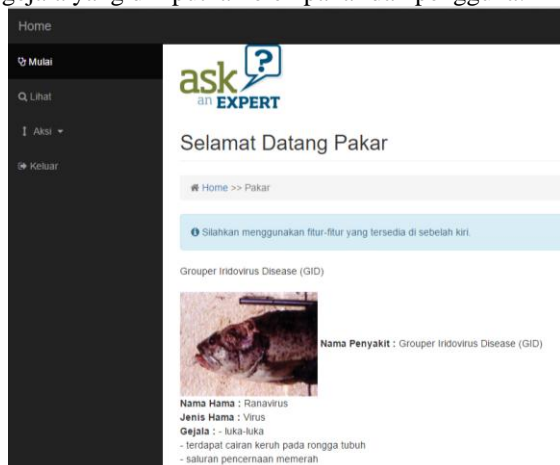
**Gambar 8. Menu konsultasi**

### b. Menu Hasil Konsultasi

Merupakan menu yang tampil setelah melakukan proses konsultasi kepada sistem seperti pada Gambar 9 di bawah. Tampilan ini berisi penjelasan



tentang penyakit, hama yang menyerang dan gejala-gejala yang diinputkan oleh pakar dan pengguna.



Gambar 9. Menu hasil konsultasi

### 3. KESIMPULAN

Kesimpulan yang dapat diambil berdasarkan hasil implementasi dan analisa sistem untuk program penerapan metode *Ripple Down Rules (RDR)* dalam sistem pakar indentifikasi penyakit ikan air laut adalah sebagai berikut:

- Sistem dapat mengajukan pertanyaan yang lebih sedikit kepada pengguna untuk menemukan suatu penyakit.
- Metode *Ripple Down Rules (RDR)* dapat digunakan untuk proses pengaturan dan pemeliharaan sistem pakar. Metode ini memiliki kemampuan untuk mengatur ulang aturan-aturan yang terdapat dalam basis aturan berdasarkan fakta-fakta yang ditambah, diubah maupun diperbaiki oleh pakar dalam sistem pakar ini.
- Sistem mampu menjaga konsistensi fakta dan aturan pada proses penambahan, perbaikan dan penghapusan didalam sistem.

### PUSTAKA

- Durkin John (1994). *Expert Systems Design and Development*. University of Akron.
- Zafran, I. Koesharyani dan K. Yuasa. (1997). *Parasit Pada Ikan Kerapu di Panti Benih dan Upaya Penanggulangannya*. Jurnal Penelitian Perikanan Indonesia. Vol. III(4):16-23.
- Martin, J., & Oxman, S (1998). *Building Expert System*. New Jersey : Prentice Hall.
- Subasinghe, R. dkk. (2001). *Aquaculture development, health and wealth*. In aquaculture in the third millennium. Technical proceedings of the conference on aquaculture in the third millennium (Subasinghe, R.P. et al., eds). pp. 167-191. Bangkok and FAO, NACA.
- Kusumastanto, T. (2003). *Pemberdayaan Sumberdaya Kelautan, Perikanan dan Perhubungan Laut dalam Abad XXI*. Pusat Kajian Sumberdaya Pesisir dan Lautan, Institut Pertanian Bogor.

Sukadi, F. (2004). *Kebijakan Pengendalian Hama dan Penyakit Ikan dalam Mendukung Akselerasi Pengembangan Perikanan Budidaya*. Dalam: Prosiding Pengendalian Penyakit pada Ikan dan Udang Berbasis Imunisasi dan Biosecurity, Unsoed Purwokerto. Hal 1 – 7.

Buchanan, Bruce G. (Winter 2005). "A (Very) Brief History of Artificial Intelligence" (PDF), *AI Magazine*, pp. 53–60, retrieved 2007-08-30.

Hill, B.J. (2005). *The Need for Effective Disease Control in International Aquaculture*. *Dev. Biol. (Basel)* (121): 3–12.

Giarratano, J. C., Riley, G.D., (2005). *Expert System Principles and Programming Fourth Edition*, Canada: Course Technology.

Cao, L., W. Wang, Y. Yang, C. Yang, Z. Yuan, S. Xiong and J. Diana. (2007). *Environmental impact of aquaculture and countermeasures to aquaculture pollution in China*. *Environmental Science in Pollution Res* 14 (7): 452 – 46.

Kementerian Kelautan dan Perikanan (KKP).

(2009). *Kelautan dan Perikanan dalam Angka*.

Pusat Data, Statistik dan Informasi KKP. Jakarta.

Fris Johnny dan Des Roza, (2009). *Kasus infeksi irido pada benih ikan kerapu pasir, *Epinephelus corralicola* di hatchery*. *Jurnal Perikanan (J. Fish . Sci.)* XI (1): 8-12.

"*Production Rules, Bratko ed. 4, chapter 15, page 343*", (Online),

<http://www.cse.unsw.edu.au/~billw/cs9414/notes/kr/rules/rules.html>, diakses 11 Januari 2016).

Romi Novriadi, (2014). *Penyakit Ikan Air Laut di Indonesia*. Kementerian Kelautan dan Perikanan. Direktorat Jenderal Perikanan Budidaya. Direktorat Kesehatan Ikan dan Lingkungan, Jakarta.