

Identifikasi Penyakit Pada Terumbu Karang Menggunakan *Ripple Down Rules*

Agus Cahyo Nugroho

Teknik Informatika, Universitas Ciputra

UC Town, Citraland, Made, Sambikerep, Surabaya, Jawa Timur

agus.nugroho@ciputra.ac.id

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 coral reef disease. The purpose of this study is to develop an expert system for diagnosing coral reef disease in the form of websites using PHP with a MySQL database. Expert system for diagnosing coral reef disease problem is using Ripple Down Rules (RDR) method has a goal to discover symptoms that appear in the form of questions that can diagnose the coral reef disease based on website. Web based expert system is able to recognize types of coral reef disease after consultation by answering a few questions that are displayed by the application of expert systems and can infer some types of coral reef disease. Data coral reef disease that already known adapt to rules which are made for matching the symptoms of coral reef disease.

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 pada terumbu karang. Tujuan dari penelitian ini adalah membangun sistem pakar untuk mendiagnosa penyakit pada terumbu karang dalam bentuk website menggunakan pemrograman PHP dengan database MySQL. Sistem pakar untuk mendiagnosa penyakit pada terumbu karang 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 pada terumbu karang setelah melakukan konsultasi dengan menjawab beberapa pertanyaan – pertanyaan yang ditampilkan oleh aplikasi sistem pakar serta dapat menyimpulkan beberapa jenis penyakit pada terumbu karang. Data penyakit yang dikenali menyesuaikan rules (aturan) yang dibuat untuk dapat mencocokkan gejala-gejala penyakit pada terumbu karang.

Kata Kunci— sistem pakar, penyakit terumbu karang, ripple down rules, forward chaining

I. PENDAHULUAN

A. Latar Belakang

Sistem pakar adalah sistem yang menggabungkan pengetahuan, fakta, aturan dan teknik penelusuran untuk memecahkan masalah yang secara normal memerlukan keahlian seorang pakar [1].

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 dimasukan 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.

Terumbu karang yang merupakan objek penelitian adalah hewan benthik yang hidup di dasar perairan. Hewan ini sebagian besar hidupnya berkoloni yang tersusun atas *Kalsium karbonat* ($CaCO_3$) sebagai hasil sekresi dari *Zooxanthellae*. Terumbu karang merupakan habitat berbagai biota laut untuk tumbuh dan berkembang biak dalam kehidupan yang seimbang. Sadarun [2] menyatakan bahwa sifat yang menonjol dari terumbu karang adalah keanekaragaman, jumlah spesies, dan bentuk morfologinya yang tinggi dan bervariasi. Salah satu sifat dari ekosistem terumbu karang adalah kerentanan ekosistem tersebut menerima dampak dari luar. Sebagai contoh kerusakan terumbu karang akibat faktor antropogenik, faktor alami, maupun dari penyakit karang itu sendiri dapat menurunkan persentase tutupan karang hidup di suatu perairan. Menurut Kurman [3] menunjukkan hasil bahwa tutupan karang hidup di Indonesia hanya sekitar 5% dalam keadaan bagus dengan persen cover 50-75.

Selanjutnya menurut Willis *et al.* [4] penyakit karang memberikan sumbangsih terbesar kematian karang di Perairan Australia khususnya di *Great Barrier Reef*. Lebih lanjut Kellogg *et al.* [5] mengungkapkan bahwa di pantai Florida AS juga mengindikasikan bahwa kematian karang terbesar diakibatkan oleh mikroorganisme yang menyebabkan penyakit karang. Penyakit karang adalah

gangguan terhadap kesehatan karang yang menyebabkan gangguan secara fisiologis bagi biota karang [6]. Munculnya penyakit karang dicirikan dengan adanya perubahan warna, kerusakan dari skeleton biota karang, sampai dengan kehilangan jaringannya. Munculnya penyakit tersebut merupakan interaksi antara host atau inang dalam hal ini biota karang, agent/pembawa dalam hal ini patogen, dan lingkungan. Beberapa penelitian mengungkapkan bahwa penurunan kualitas lingkungan perairan sangat berperan terhadap munculnya agent atau mikroorganisme pembawa patogen terhadap karang. Sebagai contoh kenaikan dari nitrogen terlarut yang diikuti dengan penurunan kecerahan perairan dapat menyebabkan munculnya aspergillois penyebab penyakit Black Band Disease [7]. Selain itu sedimentasi, polusi yang ditimbulkan oleh limbah domestik, sampah, sampai dengan air ballast yang masuk ke ekosistem terumbu karang berpotensi munculnya patogen penyebab penyakit karang.

Penulis memilih studi kasus mengenai penyakit pada terumbu karang karena di Indonesia kematian terumbu karang akibat hama dan penyakit masih belum menjadi perhatian khusus oleh para peneliti. Hal ini terbukti masih jarang penelitian yang mengungkapkan penyakit karang di perairan Indonesia. Beberapa penelitian yang pernah dilakukan diantaranya di Perairan Wakatobi Sulawesi Tenggara, Pulau Seribu DKI Jakarta, dan Pulau Panjang Jawa Tengah [8].

B. Rumusan Masalah

Bagaimana mengidentifikasi penyakit pada terumbu karang menggunakan metode *Ripple Down Rules (RDR)*.

C. Manfaat Penelitian

Manfaat penelitian ini :

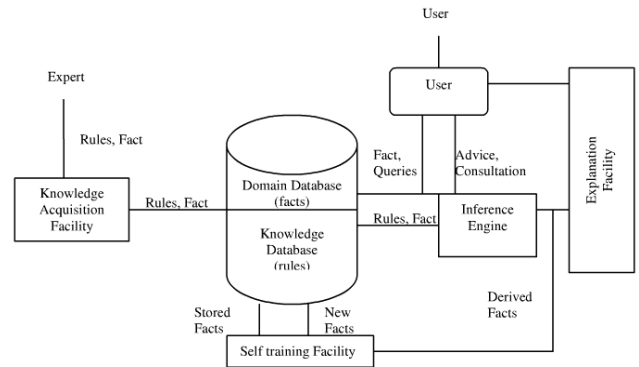
1. Sebagai acuan bagi masyarakat pembudidaya dan pengambil kebijakan untuk mengembangkan teknik diagnosa dan pengendalian penyakit pada terumbu karang yang tepat, efektif dan sistematis.
2. Sebagai acuan bagi petugas dinas kelautan baik daerah maupun pusat untuk menumbuhkan, mengembangkan dan meningkatkan kewaspadaan terhadap timbulnya penyakit pada terumbu karang di Indonesia.

II. LANDASAN TEORI

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 mengemulasikan kemampuan membuat keputusan dari seorang pakar [9]. 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.

A. Arsitektur Sistem Pakar



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

Penjelasan setiap komponen yang terdapat pada Gambar 1 diatas 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 dialami terumbu karang tersebut.

4) *Fasilitas Penjelas (Explanation Facility)* : Komponen tambahan ini akan meningkatkan kemampuan dari sistem pakar yaitu menggambarkan penalaran sistem kepada pengguna (*user*). Komponen tambahan ini juga berfungsi untuk menjelaskan kepada pengguna tentang bagaimana kesimpulan diambil oleh sistem pakar.

5) *Antarmuka Pengguna (User Interface)* : Komponen antarmuka pengguna ini digunakan untuk berkomunikasi antara sistem dengan pengguna (*user*). Komponen antarmuka pengguna berfungsi menerima informasi yang diberikan oleh pengguna lalu mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu komponen antarmuka pengguna ini juga berfungsi 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 ini berfungsi melakukan pembelajaran 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.

B. Ripple Down Rules (RDR)

Menurut penelitian yang dilakukan oleh Campton dan Peters [11] dijelaskan bahwa ketika seorang pakar mendapat pertanyaan mengenai bagaimana mencapai sebuah kesimpulan dari suatu pengetahuan, maka pakar akan

memberikan pernyataan bahwa kesimpulan yang dia berikan adalah benar. Meskipun pakar dalam memberikan kesimpulan belum diketahui secara pasti tentang kebenarannya. Hal ini mengakibatkan sistem pakar menjadi tidak *valid* dalam melakukan akuisisi pengetahuan. Oleh

sebab itu, diperlukan metode *Ripple Down Rules (RDR)* untuk menyelesaikan permasalahan tersebut.

Terdapat dua fitur utama *Ripple Down Rules (RDR)* untuk penambahan pengetahuan dalam sistem pakar [11] :

1. Jika pengetahuan yang terdapat dalam basis data tidak benar, maka sebuah aturan pembenaran yang telah dibuat digunakan untuk menggantikan aturan yang salah yang membuat aturan pembenaran tersebut hanya dapat dijalankan pada tingkatan yang sama dengan aturan yang salah. Lalu ketika aturan utama (aturan yang salah) dikerjakan, maka aturan pembenaran yang akan dipilih untuk dikerjakan daripada aturan utama tersebut.

2. Pakar hanya diperbolehkan menambah ataupun memperbaiki aturan yang berada dalam satu susunan aturan. Dimana jika memerlukan fakta dari susunan aturan yang lain maka susunan aturan baru yang dibentuk haruslah memiliki perbedaan dengan susunan aturan lain agar tidak terjadi tabrakan pengetahuan antar susunan aturan yang ada.

Pengimplementasian *Ripple Down Rules (RDR)* dalam sistem dilakukan ketika sistem tersebut sedang dijalankan. Prosesnya dimulai dengan basis data kosong yang dibangun secara bertahap dari waktu ke waktu dalam suatu kasus pemrosesan. Tugas seorang pakar disini adalah memonitor output dan secara bertahap menambahkan aturan sampai mencapai suatu kesimpulan yang dianggap benar. Sebagai sistem yang berkembang, pengetahuan diberikan oleh pakar pastinya terdapat pengetahuan yang benar, salah atau hilang. Jika salah atau hilang maka pakar akan menambahkan pengetahuan yang digunakan untuk memperbaiki pengetahuan yang salah tersebut. Lalu sistem akan mengatur ulang setiap aturan yang ditambah ataupun diubah oleh pakar dan juga memasukan aturan tersebut ke dalam basis aturan sebagai penyempurnaan dari aturan yang terbentuk dari pengetahuan yang salah. Dimana aturan ini hanya ditambahkan pada akhir dari aturan sebelumnya dan hanya dievaluasi setelah aturan sebelumnya telah dievaluasi [11].

Menurut penelitian yang dilakukan oleh Jursic dan Morzetic [12] metode *Ripple Down Rules (RDR)* digunakan untuk akuisisi pengetahuan tambahan dan pemeliharaan sistem berbasis aturan. Jika dibandingkan dengan standar jika-maka dari suatu aturan klasifikasi, *Ripple Down Rules (RDR)* memiliki bentuk yang sama yaitu IF-THEN-ELSE. Konsep IF-THEN-ELSE terjadi ketika aturan yang paling umum dibangun pertama mengalami pembenaran oleh pakar, maka suatu aturan pembenaran baru akan ditambahkan sebagai suatu proses membangun aturan baru yang mengubah aturan utama yang salah. *Ripple Down Rules (RDR)* membentuk struktur susunan aturan dimana aturan utama dijalankan saat aturan tersebut tidak mengalami pembenaran namun jika terjadi pembenaran aturan oleh pakar maka akan dilakukan perubahan bentuk susunan aturan dalam sistem pakar ini [11].

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 [13] mengenai system 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.

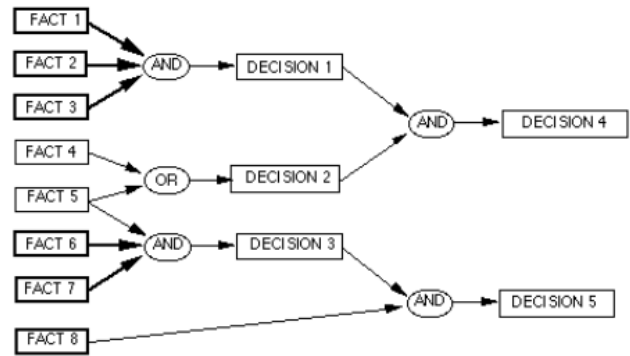
Pada perkembangannya *Ripple Down Rules (RDR)* merupakan strategi dalam mengembangkan sistem secara bertahap dimana sistem tersebut sudah digunakan. Saat sistem tidak memberikan respon yang benar maka perubahan perlu dilakukan tanpa mempengaruhi kompetensi sistem. Perubahan harus dapat dilakukan dengan 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.

C. 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 [1].

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 system

1. IF bekas luka/goresan terlihat jelas
2. AND rusaknya rangka karang
3. AND bekas gigitan berwarna putih
4. AND bekas luka ditumbuhi alga
5. AND beberapa luka bisa sembuh
6. AND ditemukan ikan predator di area survey
7. THEN penyakit adalah Fish Bites

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 bekas gigitan berwarna putih menjadi bekas gigitan berwarna coklat. 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 : P001
 Nama penyakit : Fish Bites
 Kode Gejala : G001
 Nama Gejala : bekas luka terlihat jelas
 Kode Gejala : G002
 Nama Gejala : rusaknya rangka karang
 Kode Gejala : G003
 Nama Gejala : bekas gigitan berwarna putih

- Kode Gejala : G004
 Nama Gejala : bekas luka ditumbuhi algae
 Kode Gejala : G005
 Nama Gejala : beberapa luka bisa sembuh
 Kode Gejala : G006
 Nama Gejala : ditemukan ikan predator di area survey
- Setelah itu sistem menanyakan gejala lama apa yang akan diubah di dalam sistem ini :
 Kode Penyakit : P001
 Penyakit : Fish Bites
 Kode Gejala : bekas gigitan berwarna putih
 - Lalu sistem akan meminta aturan baru yang akan ditambahkan ke dalam Sistem Pakar :
 Gejala Baru : bekas gigitan berwarna coklat
 - Setelah itu sistem akan melakukan pelacakan basis aturan yang berhubungan dengan Fish Bites dan bekas gigitan berwarna putih dan mengubah gejala lama tersebut menjadi bekas gigitan berwarna coklat.
 Gambar 4. Impementasi perubahan gejala

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

- IF bekas luka/goresan terlihat jelas
 - AND rusaknya rangka karang
 - AND bekas gigitan berwarna coklat
 - AND bekas luka ditumbuhi alga
 - AND beberapa luka bisa sembuh
 - AND ditemukan ikan predator di area survey
 - THEN penyakit adalah Fish Bites
- Gambar 5. Impementasi aturan baru

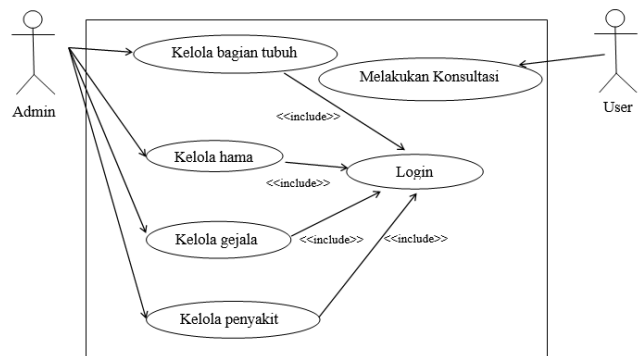
D. Bahan dan Alat

Bahan yang digunakan untuk pembuatan sistem pakar ini adalah beberapa buku yang menjelaskan tentang penyakit ikan air laut, serta pengetahuan yang diperoleh dari materi presentasi yang dipublikasikan oleh Bapak Edy Setyawan dan Ibu Estradivari pakar di bidang penyakit terumbu karang dari Terangi Foundation pada tahun 2016 [15]. Alat yang digunakan dalam penelitian ini adalah sebuah komputer dan beberapa perangkat lunak pendukung seperti : *Microsoft Word, Editplus, Bootstrap, WAMP server, PHP dan MySQL.*

E. Use Case Diagram, Struktur dan Relasi Tabel

Dalam mendukung perancangan sistem pakar ini diperlukan suatu gambaran umum interaksi antara elemen-elemen yang terlibat dalam sistem. Gambaran umum ini berguna untuk mengetahui fungsionalitas yang diharapkan dari suatu sistem. Adapun untuk mendapatkan gambaran umum mengenai interaksi antar elemen dalam sistem serta menggambarkan fungsionalitas sistem secara jelas, maka digunakan alat bantu pemodelan berupa *use case diagram*. Diagram ini terdiri atas empat komponen utama yaitu (1) *boundary*, yang menggambarkan ruang lingkup sistem; (2) *actor*, yang menggambarkan individu-individu yang berinteraksi dalam sistem sesuai dengan perannya; (3) *use case*, menggambarkan apa yang dikerjakan oleh sistem; (4)

relationship, menggambarkan relasi yang terjadi antara *actor* dan *use case*. Rancangan *use case diagram* untuk sistem pakar ini dapat dilihat pada Gambar 6.



Gambar 6. Use Case Diagram

Adapun struktur tabel dan relasi antar tabel yang digunakan dalam perancangan sistem pakar seperti Tabel 1 sampai Tabel 8 :

Tabel 1. Tabel aturan_penyakit.sql

| Nama Kolom | Tipe Data | Keterangan |
|---------------|--------------|---|
| id_rule | int(5) | identitas dari aturan gejala yang dibuat |
| aturan_gejala | varchar(500) | aturan dari gejala suatu penyakit |
| kode_hama | varchar(5) | kode hama suatu penyakit |
| kode_jenis | varchar(5) | jenis hama suatu penyakit |
| kode_penyakit | varchar(5) | kode dari penyakit tertentu |
| urutan | int(6) | nomor urut dari suatu aturan gejala |
| urutan2 | int(6) | nomor urut prioritas kedua dari suatu aturan gejala |
| sama | int(6) | menandakan jumlah gejala yang sama dari aturan gejala |

Tabel 2. Tabel bagian.sql

| Nama Kolom | Tipe Data | Keterangan |
|---------------|--------------|-----------------------------------|
| kode_bagian | varchar(5) | kode bagian yang mengalami gejala |
| bagian_gejala | varchar(5) | nama bagian yang mengalami gejala |
| pertanyaan | varchar(200) | daftar pertanyaan yang di ajukan |

Tabel 3. Tabel gejala.sql

| Nama Kolom | Tipe Data | Keterangan |
|-------------|--------------|--|
| kode_gejala | varchar(5) | kode gejala dari suatu penyakit |
| kode_bagian | varchar(5) | kode bagian yang mengalami gejala |
| gejala | varchar(200) | gejala yang di alami oleh terumbu karang |

Tabel 4. Tabel hama.sql

| Nama Kolom | Tipe Data | Keterangan |
|------------|-------------|-------------------------------------|
| kode_jenis | varchar(5) | kode jenis hama dari suatu penyakit |
| jenis_hama | varchar(50) | jenis hama dari suatu penyakit |

Tabel 5. Tabel hitung_urutan.sql

| Nama Kolom | Tipe Data | Keterangan |
|---------------|-----------|---|
| urutan | int(5) | nomor urut dari aturan gejala |
| jumlah_urutan | int(5) | jumlah urutan dengan nomor urut tertentu dari aturan gejala |

Tabel 6. Tabel nama_hama.sql

| Nama Kolom | Tipe Data | Keterangan |
|------------|-------------|-------------------------------|
| kode_hama | varchar(5) | kode hama dari suatu penyakit |
| nama_hama | varchar(70) | nama hama dari suatu penyakit |

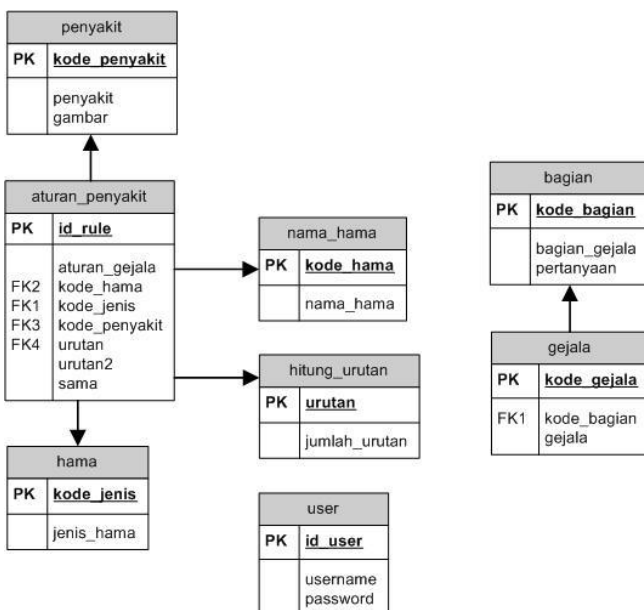
Tabel 7. Tabel penyakit.sql

| Nama Kolom | Tipe Data | Keterangan |
|---------------|--------------|---|
| kode_penyakit | varchar(5) | kode dari masing-masing penyakit |
| penyakit | varchar(100) | nama dari masing-masing penyakit |
| gambar | varchar(50) | contoh gambar dari terumbu karang yang terkena penyakit |

Tabel 8. Tabel user.sql

| Nama Kolom | Tipe Data | Keterangan |
|------------|-------------|--|
| id_user | int(3) | id dari user yang bisa login ke sistem |
| username | varchar(50) | username dari user yang bisa login ke sistem |
| password | varchar(50) | password dari user yang bisa login ke sistem |

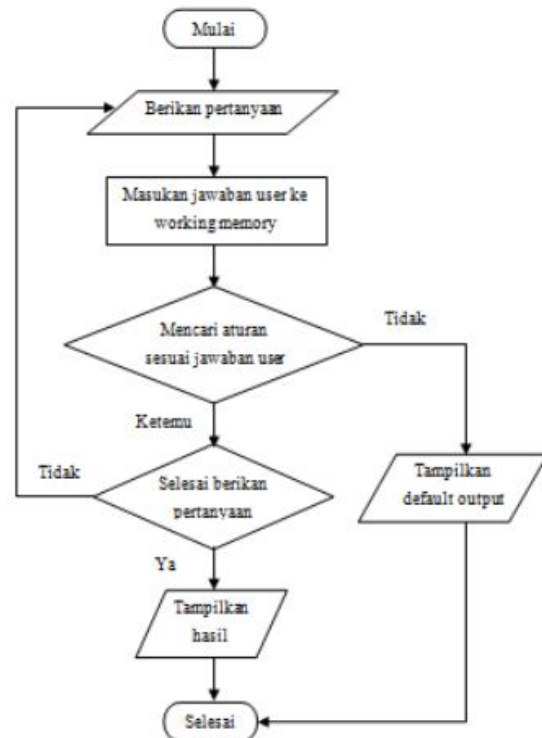
Adapun relasi antar tabel untuk pembuatan prototipe sistem pakar adalah seperti yang terlihat pada Gambar 7.



Gambar 7. Relasi Tabel Sistem Pakar

F. Diagram Alur Proses Konsultasi Pada Sistem

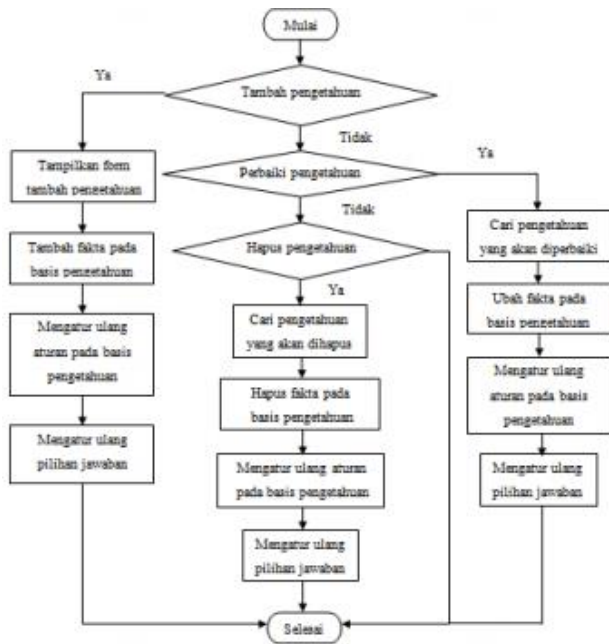
Diagram alur pada Gambar 8 menjelaskan tentang alur dari cara kerja sistem yang akan dibuat. Ketika pengguna atau pakar memilih menu konsultasi. Sistem akan mengajukan pertanyaan-pertanyaan berbeda yang akan ditampilkan satu persatu sesuai dengan basis aturan yang ada. Jawaban dari pengguna akan dimasukkan 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 8. Diagram Alur Proses Konsultasi pada Sistem

G. 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 9. Diagram Alur Ripple Down Rules (RDR)

Gambar 9 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.

H. Perancangan Basis Pengetahuan

Basis pengetahuan yang terdapat dalam sistem pakar ini menyimpan fakta yang berhubungan dengan penyakit terumbu karang 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 9, 10, 11, 12 dan 13. Selanjutnya fakta tersebut digunakan untuk membangun pohon keputusan dan menyusun basis aturan untuk mendapatkan solusi yang dibutuhkan oleh sistem.

Tabel 9. Kode dan Bagian Gejala

| Kode_B | Bagian_Gejala |
|--------|---------------|
| B001 | rangka |
| B002 | jaringan |
| B003 | koloni |
| B004 | cabang karang |
| B005 | permukaan |
| B006 | corallite |
| B007 | luka |
| B008 | zooxanthellae |
| B009 | coenesteum |
| B010 | polip |
| B011 | dasar karang |

Tabel 10. Kode dan Gejala

| G | Kode_ B | Kode_ B | Gejala |
|------|---------|---------|---|
| G001 | B001 | B001 | bekas luka terlihat jelas |
| G002 | B001 | B001 | rusaknya rangka |
| G003 | B001 | B001 | bekas gigitan berwarna putih |
| G004 | B001 | B001 | bekas luka ditumbuhi algae |
| G005 | B001 | B001 | beberapa luka bisa sembuh |
| G006 | B001 | B001 | ditemukan ikan predator di area |
| G007 | B002 | B002 | hilangnya jaringan |
| G008 | B001 | B001 | rangka berwarna putih |
| G009 | B002 | B002 | ditemukan CoT di area |
| G010 | B002 | B002 | meliputi area yg luas |
| G011 | B003 | B003 | pemangsaan di tepi koloni |
| G012 | B003 | B003 | bekas pemangsaan di koloni |
| G013 | B004 | B004 | ditemukan siput berwarna putih |
| G014 | B004 | B004 | pemagsaan dari dasar hingga ujung |
| G015 | B004 | B004 | bekas pemangsaan lebih kecil |
| G016 | B004 | B004 | bekas pemangsaan dg batas tidak teratur |
| G017 | B003 | B003 | ditemukan drupella di dasar koloni |
| G018 | B005 | B005 | ditemukan sedimen di permukaan karang |
| G019 | B002 | B002 | pertumbuhan algae di jaringan hidup |
| G020 | B002 | B002 | ditemukan band berbentuk annular/linear |
| G021 | B002 | B002 | ditemukan band berwarna hitam/coklat |
| G022 | B001 | B001 | ditemukan cyanobacteria |
| G023 | B007 | B007 | band menyebar keluar dari luka |
| G024 | B007 | B007 | ditemukan band di karang masif |
| G025 | B001 | B001 | rangka kosong ditumbuhi algae |
| G026 | B007 | B007 | batas luka discrete atau diffuse |
| G027 | B007 | B007 | batas luka tidak berpigmen |
| G028 | B002 | B002 | laju kehilangan jaringan cepat |

| G | Kode_ B | Kode_ B | Gejala |
|------|---------|---------|---|
| G029 | B007 | B007 | luka dekat penyakit berwarna putih |
| G030 | B007 | B007 | luka bertahap berubah warna coklat |
| G031 | B007 | B007 | garis tepi di luka berbentuk garis |
| G032 | B007 | B007 | luka berbentuk focal dan multifocal |
| G033 | B007 | B007 | luka dengan garis tepi annular/linear |
| G034 | B007 | B007 | luka berbatasan dg jaringan karang sehat |
| G035 | B007 | B007 | perkembangan luka dari mm ke cm per bulan |
| G036 | B002 | B002 | pita paling depan berwarna kuning pucat |
| G037 | B002 | B002 | jaringan terinfeksi berwarna lebih gelap |
| G038 | B002 | B002 | pola luka berupa multifocal/diffuse |
| G039 | B002 | B002 | warna jaringan jadi pink/ungu/biru |
| G040 | B002 | B002 | dinding corallite menebal |
| G041 | B002 | B002 | pigmentasi berbentuk garis |
| G042 | B002 | B002 | pigmentasi berbentuk titik |
| G043 | B002 | B002 | pigmentasi berbentuk benjolan |
| G044 | B002 | B002 | pigmentasi berbentuk tidak teratur |
| G045 | B001 | B001 | respon terhadap biota pengebor |
| G046 | B001 | B001 | respon terhadap kompetisi |
| G047 | B001 | B001 | respon terhadap serangan alga |
| G048 | B002 | B002 | jaringan membengkak |
| G049 | B002 | B002 | jaringan berwarna pink hingga putih |
| G050 | B002 | B002 | ditemukan trematode parasit |
| G051 | B001 | B001 | karang masih hidup |
| G052 | B010 | B010 | polip masih terlihat |
| G053 | B002 | B002 | adanya gradien jaringan sehat dg sakit |
| G054 | B008 | B008 | hilangnya alga simbiosis pada karang |
| G055 | B008 | B008 | tekanan terhadap lingkungan (suhu) |
| G056 | B002 | B002 | area jaringan berwarna putih |
| G057 | B002 | B002 | jaringan masih ada |
| G058 | B002 | B002 | tidak ada pigmentasi jaringan |
| G059 | B002 | B002 | ditemukan garis dengan batas yg jelas |
| G060 | B002 | B002 | tingkat pemutihan karang bervariasi |
| G061 | B001 | B001 | deformasi rangka berbentuk focal/multifocal |
| G062 | B001 | B001 | ditemukan invertebrata |
| G063 | B001 | B001 | deformasi rangka tidak beraturan |
| G064 | B001 | B001 | coenosteum menebal |
| G065 | B007 | B007 | luka berbentuk circular hingga diffuse |
| G066 | B002 | B002 | berwarna lebih muda |
| G067 | B002 | B002 | tidak berwarna sama sekali |
| G068 | B002 | B002 | tidak ada jaringan antara sehat dan sakit |
| G069 | B001 | B001 | rangka utuh dan terbuka |

Tabel 11. Kode dan Nama Hama

| a | Kode_Hama | Nama_Hama |
|---|-----------|------------------------------|
| | NH001 | Ikan Kakaktua, Buntal, Betok |
| | NH002 | Bintang Laut Crown of Thorns |
| | NH003 | Siput Drupella |
| | NH004 | Algae |
| | NH005 | Cyanobacteria |
| | NH006 | Nudibranch |
| | NH007 | Flatworm |
| | NH008 | Invertebrata |
| | NH009 | Sedimen |
| | NH010 | Phormidium corallyticum |
| | NH011 | Mikroba Pathogen |
| | NH012 | Virulence |
| | NH013 | Suhu Meningkat |
| | NH014 | Tidak Diketahui |

Tabel 12. Kode dan Jenis Hama

| s | Kode_Jeni | Jenis_Hama |
|---|-----------|-----------------|
| | H001 | Ikan |
| | H002 | Siput |
| | H003 | Algae |
| | H004 | Bintang Laut |
| | H005 | Bakteri |
| | H006 | Cacing |
| | H007 | Invertebrata |
| | H008 | Sedimen |
| | H009 | Mikroba |
| | H010 | Lingkungan |
| | H011 | Tidak Diketahui |

Tabel 13. Kode dan Nama Penyakit

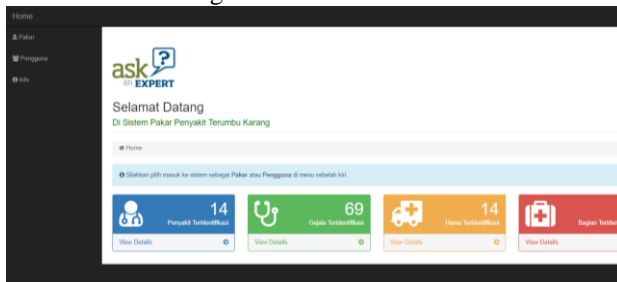
| Kode_P | Penyakit |
|--------|-----------------------------------|
| P001 | Fish Bites |
| P002 | CoT Predation |
| P003 | Drupella Predation |
| P004 | Sedimentation Damage |
| P005 | Algae Overgrowth |
| P006 | Black Band Disease |
| P007 | White Syndromes |
| P008 | Yellow Band Disease |
| P009 | Pigmentation Response |
| P010 | Trematodiasis |
| P011 | Bleaching |
| P012 | Unusual Bleaching Pattern |
| P013 | Galls |
| P014 | Growth Anomalies of Unknown Cause |

III. IMPLEMENTASI

A. Halaman Index

Halaman index merupakan halaman pertama yang tampil ke *user* saat mengakses sistem pakar ini. Tampilannya bisa kita lihat pada Gambar 10 di bawah. Di halaman ini

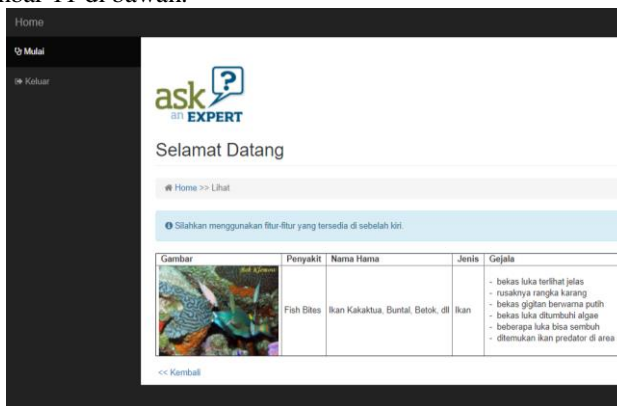
terdapat informasi rekapitulasi jumlah Penyakit Teridentifikasi, Gejala Teridentifikasi, Hama Teridentifikasi dan Bagian Teridentifikasi.



Gambar 10. Halaman Index

B. Halaman Detil Penyakit

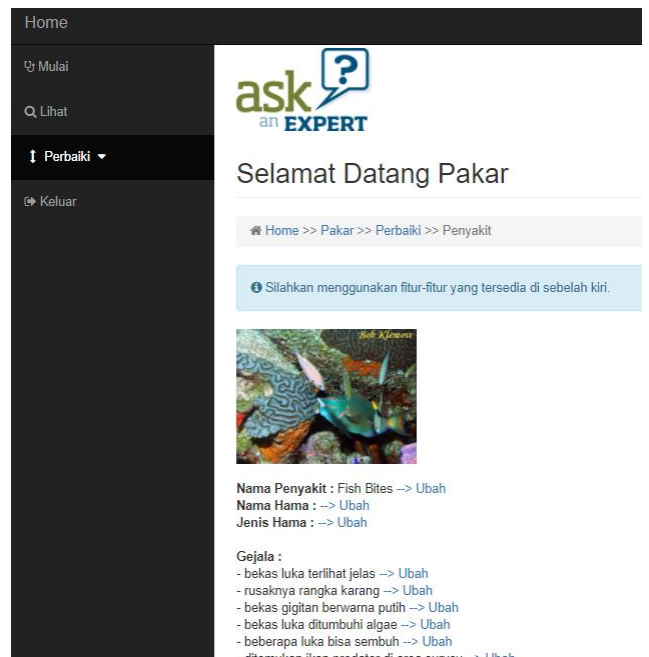
Halaman detil penyakit merupakan halaman yang berisi gambar, nama penyakit, nama hama, jenis hama dan gejala suatu penyakit. Tampilannya bisa kita lihat pada Gambar 11 di bawah.



Gambar 11. Halaman Detil Penyakit

C. Menu Perbaiki Penyakit

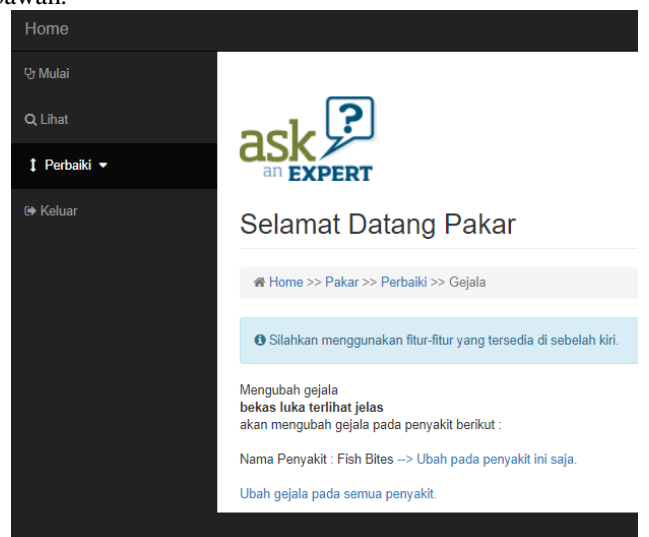
Halaman perbaiki penyakit merupakan halaman yang penting dimana kita bisa mengimplementasikan metode *Ripple Down Rules (RDR)*. Melalui menu ini kita bisa mengubah nama penyakit, nama hama, jenis hama dan gejala suatu penyakit. Tampilannya bisa kita lihat pada Gambar 12 di bawah.



Gambar 12. Menu Perbaiki Penyakit

D. Menu Perbaiki Gejala

Halaman perbaiki gejala merupakan halaman yang penting dimana kita bisa mengimplementasikan metode *Ripple Down Rules (RDR)*. Melalui menu ini kita bisa mengubah nama penyakit, nama hama, jenis hama dan gejala suatu penyakit. Tampilannya bisa kita lihat pada Gambar 13 di bawah.



Gambar 13. Menu Perbaiki Gejala

E. Menu Konsultasi

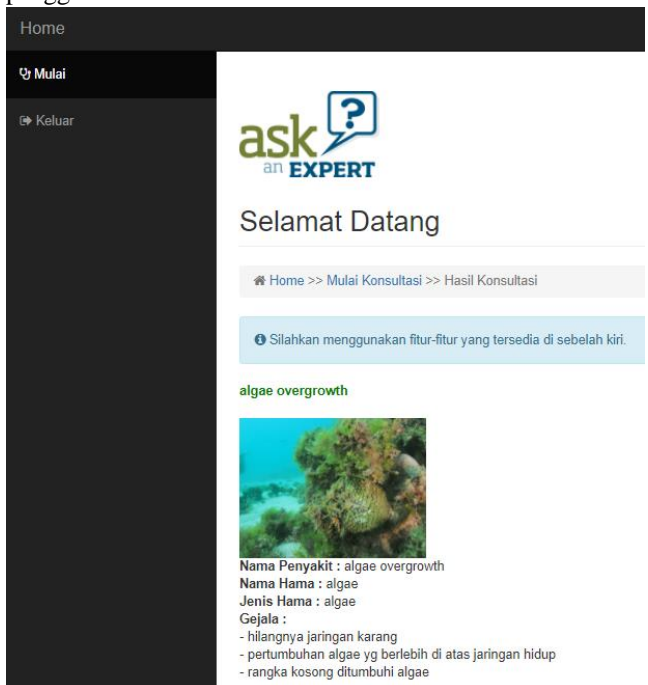
Menu konsultasi adalah menu yang tampil ketika pakar dan pengguna menekan tombol mulai pada menu pakar atau menu pengguna seperti pada Gambar 14 di bawah. Menu ini digunakan untuk berkonsultasi dengan sistem tentang gejala-gejala yang ditemui pada terumbu karang untuk menemukan penyakit yang menyerang.



Gambar 14. Menu konsultasi

F. Menu Hasil Konsultasi

Merupakan menu yang tampil setelah melakukan proses konsultasi kepada sistem seperti pada Gambar 15 di bawah. Tampilan ini berisi penjelasan tentang penyakit, hama yang menyerang dan gejala-gejala yang diinputkan oleh pakar dan pengguna.



Gambar 15. Menu hasil konsultasi

IV. KESIMPULAN

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

1. Sistem dapat mengajukan pertanyaan yang lebih sedikit kepada pengguna untuk menemukan suatu penyakit.
2. 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.

3. Sistem mampu menjaga konsistensi fakta dan aturan pada proses penambahan, perbaikan dan penghapusan didalam sistem.

DAFTAR PUSTAKA

- [1] Durkin John, Expert Systems Design and Development, University of Akron, 1994.
- [2] Sadarun, B., Pedoman Pelaksanaan Transplantasi Karang, Direktorat Konservasi dan Taman Nasional Laut dan Direktorat Jendral KP3K, DKP, Jakarta, 2006.
- [3] Kurman, W.B., White-band disease in *Acropora palmata*: implications for the structure and growth of shallow reefs, *Bull. Mar. Sci.*, 2007, 32: 639–643.
- [4] Willis, B.L., Page, C.A., Dinsdale, E.A., Coral Disease on the Great Barrier Reef., *Coral Disease and Health.*, (Rosenberg E, & Loya Y, eds)., 2004, pp 69-104., Springer-Verlag., Berlin.
- [5] Kellogg C.A., Piceno Y.M., Tom L.M., De Santis T.Z., Andersen G.L., Gray M.A. 2014. Comparing Bacterial Community Composition of Healthy and Dark Spot-Affected *Siderastrea siderea* in Florida and the Caribbean. *Plos one* Vol. 9. Issue 10. www.plosone.org. [Desember 2014]
- [6] Raymundo, L.J., Couch, C.S. and Harvell, C.D., *Coral Disease Handbook : Guidelines for Assessment, Monitoring & Management.*, Coral Reef Targeted Research and Capacity Building for Management Program, The University of Queensland, Australia, 2008.
- [7] Borger, J.L., and Steiner, S.C.C., The Spatial Temporal Dynamik of Coral Diseases in Dominica, East Indies. *Bulletin of Marine Science.*, 2005, 77(1):137- 154.
- [8] Muller E.M., Raymundo L.J., Willis B.L., Haapkyla J., Yusuf S., Wilson JR., Harvell DC., Coral Health and Disease In The Spermonde Archipelago and Wakatobi, Sulawesi, 2012: 147-159.
- [9] Giarratano, J. C., Riley, G.D., *Expert System Principles and Programming Fourth Edition*, Canada: Course Technology, 2005.
- [10] Martin, J., & Oxman, S., *Building Expert System.*, New Jersey : Prentice Hall, 1998.
- [11] Campton, P., Peters, L., Edwards, G., Lavers, T.G., *Experience with Ripple-Down Rules*, University of New South Wales, Sydney, Australia, 2005.
- [12] Jursic, M. Mozetic, I., *Learning Ripple Down Rules For Efficient Lemmatization*, Department of Knowledge Technologies, Jožef Stefan Institute, 2010.
- [13] Buchanan, Bruce G., "[A \(Very\) Brief History of Artificial Intelligence](#)" (PDF), *AI Magazine*, 2007, pp. 53–60.
- [14] Bratko, Production Rules, 4th ed, Chapter 15, 2016: 343.
- [15] Edy Setyawan, Estradivari, *Identifikasi Penyakit dan Gangguan Kesehatan pada Terumbu Karang*, Terangi Foundation, Presentation: 2016.