

# Implementasi dan Analisis Ripple-down Rule pada Diagnosa Penyakit Jantung

Adiyaka Niastya Ihsan Maulana  
S1 Teknik Informatika  
Universitas Telkom  
Bandung, Indonesia  
adiyaka.ihsan@gmail.com

Dade Nurjanah Ir., MT., Ph.D  
Universitas Telkom  
Bandung, Indonesia  
dadenurjanah@gmail.com

**Abstract**—RDR (*ripple-down rules*) adalah algoritma suatu metode pengakuisisi basis pengetahuan yang digunakan pada sistem pakar. Algoritma ini dapat menentukan hasil klasifikasi sekaligus memperbarui basis pengetahuan dari sistem pakar yang menggunakannya. Karena itu paper ini bertujuan untuk mengimplementasikan metode RDR ini pada sistem pakar untuk mendiagnosa penyakit jantung berdasarkan gejala-gejala yang ada. Hasil dari pengujian yang telah dilakukan dapat disimpulkan sistem pakar dengan algoritma RDR ini dapat memperbarui basis pengetahuannya tanpa memerlukan *Knowledge Engineer* dan hasil dari diagnosanya tergantung pada urutan masukan kasus yang diterima sistem pakar dan bagaimana pakar (*user*) menentukan hasil diagnosa pakar. Sistem pakar yang menggunakan algoritma RDR ini efektifitasnya akan meningkat seiring dengan banyaknya kasus yang diterima oleh sistem.

**Keywords**—RDR (*Ripple-down Rules*), *expert system*, *medical diagnosis*

## I. LATAR BELAKANG

### A. Latar Belakang

Sistem pakar merupakan sebuah sistem yang memiliki pengetahuan layaknya seorang atau lebih pakar yang dapat digunakan untuk bahan acuan sesuai bidang pengetahuan yang dimiliki oleh sistem tersebut. Dibandingkan dengan seorang pakar, kelebihan sistem pakar merupakan lebih rendahnya biaya sistem pakar, pengetahuannya yang tidak akan hilang layaknya seorang pakar yang sangat mungkin untuk lupa, dan juga produktifitasnya yang konsisten. [1]

Sistem pakar saat ini sudah terbukti memberikan manfaat yang besar bagi manusia dalam berbagai keperluan, seperti mempermudah atau mempercepat suatu pekerjaan [1]. Pekerjaan yang biasanya dilakukan oleh manusia kini dapat dilakukan oleh sistem tersebut, contoh kecilnya adalah berhitung. Dalam dunia kesehatan sistem pakar juga dapat digunakan untuk mempermudah atau mempercepat diagnosa suatu penyakit seperti penyakit jantung, contohnya adalah penggunaan Fuzzy Logic pada Sistem Pakar untuk diagnosa penyakit jantung [1]. Penyakit jantung adalah keadaan dimana jantung tidak berfungsi sebagaimana mestinya, sehingga dapat berakibat fatal terhadap yang mengidap penyakit jantung karena jantung merupakan organ vital yang penting untuk kehidupan manusia [7].

Penyebab dari penyakit jantung bisa bermacam-macam tergantung dari jenisnya. Seperti gagal jantung, yang merupakan salah satu masalah kesehatan yang pada jaman sekarang sedang marak, dapat disebabkan oleh konsumsi alkohol, penumpukan lemak pada dinding pembuluh darah, atau akibat konsumsi obat-obatan tertentu [8]. Berbagai macam jenis penyakit jantung ini dapat menimbulkan gejala-gejala tertentu yang dapat berbeda-beda tiap jenisnya [8].

Berdasarkan berbagai macam gejala yang ditimbulkan dari penyakit-penyakit ini dapat ditentukan penyakitnya oleh dokter. Namun terkadang dokter sendiri tidak hafal keseluruhan gejala dari berbagai macam penyakit jantung yang ada.

Oleh karena itu dengan adanya sistem pakar yang dapat membantu diagnosa penyakit jantung berdasarkan gejala yang ada diharapkan dapat membantu dokter dalam memberikan hasil diagnosa yang lebih cepat dan juga tepat. Saat ini sudah terdapat beberapa sistem pakar yang dapat digunakan untuk mendiagnosa penyakit jantung, namun sistem pakar yang sudah ada menggunakan metode Fuzzy Logic. Dari sistem pakar yang sudah ada keluarannya berupa apakah pasien sakit atau tidak dengan beberapa tingkatan dari tidak sakit hingga pada level sakit. Sistem pakar yang dirancang pada Tugas Akhir ini bertujuan untuk memberikan diagnosa penyakit apa yang diderita pasien, tidak hanya untuk menentukan apakah pasien sakit atau tidak. Sistem pakar ini juga ditujukan kepada dokter untuk meningkatkan produktifitas dokter dalam mendiagnosa pasiennya.

## II. LANDASAN TEORI

### A. Penyakit Jantung

Jantung adalah organ berotot yang memiliki empat ruang yang terdiri dari atrium kanan, atrium kiri, ventrikel kanan, dan ventrikel kiri yang terletak pada rongga dada yang dilindungi oleh tulang iga [8]. Ukuran dari jantung kurang lebih sebesar gengaman tangan kanan dan memiliki berat sekitar 250-300 gram [8]. Jantung merupakan organ yang sangat penting bagi manusia karena jantung bertugas untuk memompa darah ke berbagai tubuh [8]. Ruang pada jantung memiliki fungsi masing-masing [8]. Atrium kanan berfungsi untuk penampung darah yang kotor mengandung CO<sub>2</sub> dari seluruh tubuh yang kemudian dialirkan pada ventrikel kanan untuk di pompa ke

paru-paru [8]. Atrium kiri berfungsi untuk menampung darah yang kaya akan oksigen yang kemudian dialirkan pada vertikel kiri untuk dipompa keseluruh tubuh. Jantung memiliki sebuah siklus yang merupakan kejadian yang terjadi dalam jantung selama peredaran darah. Terdapat dua gerakan dalam siklus jantung yaitu kontraksi dan relaksasi [8].

Karena jantung merupakan organ vital bagi manusia maka jika terjadi kesalahan atau gangguan pada jantung dapat mengakibatkan hal yang fatal salah satunya adalah kematian bagi manusia [8]. Terdapat beberapa gangguan atau penyakit jantung [7]. Penyakit jantung adalah penyakit pada jantung yang terjadi akibat adanya kelainan pada pembuluh darah jantung [7]. Terdapat beberapa penyakit jantung antara lain adalah penyakit jantung koroner, penyakit pembuluh darah otak, penyakit jantung hipertensi, penyakit pembuluh darah perifer, penyakit gagal jantung, penyakit jantung rematik, penyakit jantung bawaan, penyakit kardiomiopathy, dan penyakit jantung katub [7].

Diantara berbagai penyakit jantung tersebut penyakit jantung yang paling sering ditemui adalah jantung koroner dan serangan jantung [7]. Gejala yang dapat berupa perasaan tidak enak di dada seperti terbakar, tertekan, diperas-peras, atau dicekik [7]. Rasa tersebut biasanya berlangsung lebih dari beberapa menit yang biasanya diiringi dengan gejala lain misalnya adalah sesak napas, tubuh terasa lemas, pucat, berkeringat dingin, berdebar-debar, dan perasaan cemas [7].

Terdapat beberapa faktor yang mempengaruhi terjadinya penyakit jantung antara lain adalah kolestrol, hipertensi, merokok, diabetes militus, obesitas, kurangnya aktivitas fisik, usia, jenis kelamin, dan genetik [7].

## B. Sistem Pakar

Sistem pakar adalah sebuah sistem yang mengadopsi pengetahuan manusia ke komputer untuk dapat memodelkan kemampuan menyelesaikan masalah layaknya seorang pakar di bidangnya [1]. Dengan menggunakan sistem pakar, maka orang yang awam dalam bidang tertentu dapat menyelesaikan suatu masalah di bidang tersebut dengan baik seperti seorang pakar. Selain itu sistem pakar ini juga dapat membantu seorang pakar untuk mempermudah pekerjaannya karena pemrosesan oleh komputer terhadap data yang banyak tentu akan lebih baik dan lebih cepat dibandingkan apabila diproses secara manual oleh manusia [1].

Sistem pakar ini menggunakan basis pengetahuan untuk dapat memahami, merumuskan, dan memecahkan masalah yang diterima. Basis pengetahuan ini terdiri dari 2 bagian, yaitu fakta dan aturan. Fakta ialah situasi, kondisi, dan kenyataan yang ada. Sedangkan aturan merupakan petunjuk untuk dapat menggunakan pengetahuan yang ada sesuai dengan tujuannya.[5].

Struktur dari sistem pakar ini sendiri terdiri dari :

1. Basis Pengetahuan.
2. Mesin Inferensi.

Mesin Inferensi merupakan otak dari sistem pakar ini. Mesin inferensi yang mengarahkan proses penalaran sistem terhadap suatu kondisi dengan menggunakan basis pengetahuan yang ada.

## 3. Basis Data.

Basis data merupakan tempat seluruh fakta yang diperlukan untuk sistem pakar maupun hasil dari proses penarikan kesimpulan suatu permasalahan.

## 4. Antarmuka Pemakai

Merupakan perantara antara sistem dengan pengguna sistem pakar tersebut.

## C. Basis Data

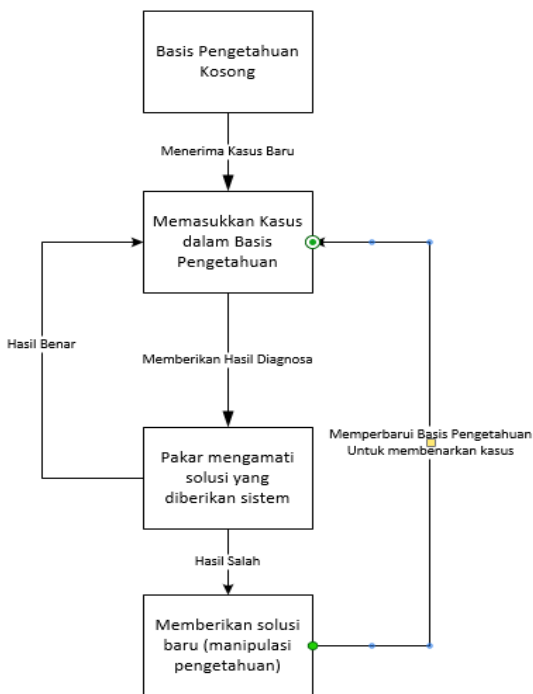
Basis Data adalah koleksi dari data-data yang terkait secara logis dan deskripsi dari data-data tersebut, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi [2]. “*Database is a self-describing collection of integrated tables*”, yang berarti database adalah sebuah koleksi data yang menggambarkan integrasi antara tabel yang satu dengan tabel yang lainnya. “*Database is a self-describing*”, disini dijelaskan bahwa struktur data saling terintegrasi dalam suatu tempat yang dikenal sebagai kamus data atau metadata [2].

Sistem pakar menggunakan basis data sebagai tempat penyimpanan basis pengetahuannya. Dengan menggunakan basis data sistem pakar dapat berfungsi sebagaimana mestinya karena setiap kali digunakan pengetahuan yang telah dipelajarinya akan selalu tetap tersimpan dan bertambah, sehingga meningkatkan efektifitas sistem pakar tersebut.

## D. Ripple-down Rules(RDR)

*Ripple-Down Rules (RDR)* merupakan sebuah metode akuisisi pengetahuan dalam pengembangan basis pengetahuan dalam suatu sistem pakar yang diperkenalkan oleh Compton dan Jansen berdasarkan sistem pakar GARVAN-ES1 (Compton dan Jansen 1988). Pendekatan pada metode RDR ini digolongkan sebagai pendekatan berdasarkan kasus (*case-based*) [4].

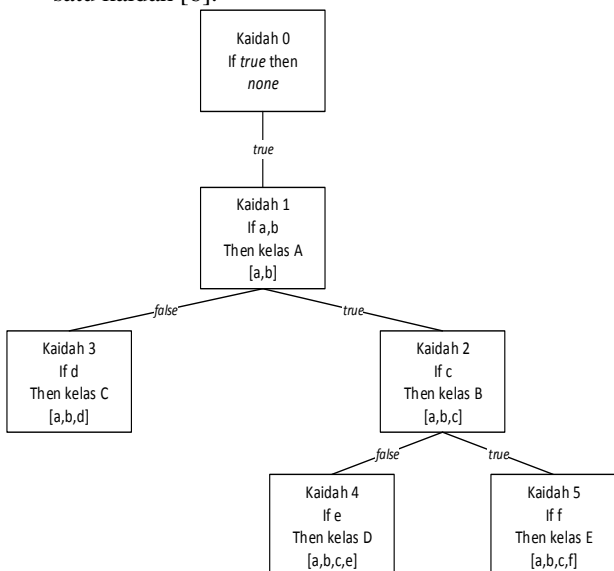
Pada RDR ini proses pembaruan pengetahuan dilakukan apabila sistem melakukan kesalahan dalam mengklasifikasikan suatu kasus baru yang belum terdapat pada basis pengetahuan pada sistem pakar tersebut [5]. Proses pembaruan pengetahuan pada sistem pakar yang menggunakan RDR pun tidak rumit karena sistem akan memperbarui basis pengetahuan begitu menerima kasus baru sehingga tidak diperlukan proses pembaruan manual terhadap sistem oleh *Knowledge Engineer* [9].



Gambar 1 Flowchart metode Ripple-down Rules [5]

1) Representasi Basis Pengetahuan RDR

Basis pengetahuan pada metode RDR direpresentasikan dalam bentuk pohon biner yang terdiri dari *True* atau *False* untuk setiap cabangnya kecuali untuk simpul pada akar paling awal yang hanya memiliki satu cabang yang berupa *True*. Tiap simpul pada *binary tree* di RDR merepresentasikan satu kaidah [6].



Gambar 2 Struktur basis pengetahuan RDR [10]

2) Mesin Inferensi

Mesin inferensi pada RDR mengarahkan suatu kasus kepada suatu penyelesaian berdasarkan pada basis pengetahuan [4]. Suatu kasus akan diperiksa dengan menggunakan representasi pohon biner termasuk pada kaidah yang mana dengan mengecek dari akar pohon biner ke setiap cabangnya [11].

Pada akar kasus akan selalu menerima hasil *True* yang kemudian akan diteruskan ke simpul berikutnya, selanjutnya akan dicek kembali untuk tiap cabangnya ke simpul yang mana kasus tersebut menerima hasil *True* berikutnya. Solusi atau penyelesaian sebuah kasus apabila kasus tersebut sudah sampai kepada simpul yang sudah tidak memiliki cabang. Misal contoh kasus A memiliki atribut a,b,c,f maka berdasarkan Gambar 2.3 kasus A diklasifikasikan pada Kelas E karena simpul terakhir berhenti di Kaidah 5.

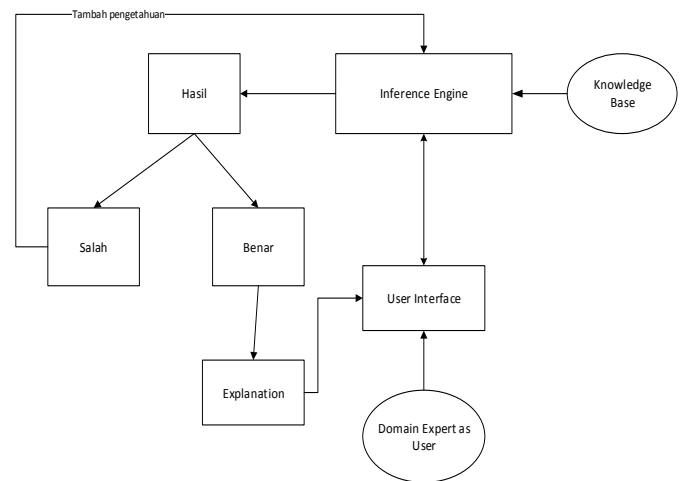
III. PERANCANGAN SISTEM

Pada paper ini akan membahas terkait tentang penggunaan fuzzy evolutionary algorithm, oleh karena itu terdapat beberapa pengertian dan pembahasan terkait dengan metode, algoritma, dan tools yang akan digunakan.

A. Gambaran Umum Sistem

Sistem pakar yang menggunakan metode *Ripple-down Rule* ini terdiri dari 3 bagian, yaitu proses memasukkan kasus atau pengetahuan baru, dan memeriksa kasus yang diterima lalu memberikan solusinya, kemudian dari solusi yang diberikan dapat ditambahkan solusi baru untuk memperbaiki apabila solusi sebelumnya yang diberikan tidak sesuai menurut pakar.

Secara umum sistem pakar diagnosa penyakit jantung dapat digambarkan seperti diagram dibawah ini.



Gambar 3 Gambaran Umum Sistem

B. Data Penyakit Jantung

Data yang diperoleh dari Pakar berupa data yang terdiri dari jenis penyakit jantung beserta gejala-gejalanya. Bentuk data yang dikumpulkan tersebut dapat dilihat pada tabel di bawah ini.

Tabel 1 Data Penyakit Jantung

NO.	Disease	Symptoms	
		Symptoms A	Symptoms B
1	Disease X	v	v
2	Disease Y	v	v
3	Disease Z	v	
4	Disease M	v	v
...	...	...	...
...	...		v
n	dst	v	v

Tabel 2 Representasi Basis Data

Representasi Basis Data <i>Binary Tree</i> .	Format Basis Data pada XML.
	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="no"?&gt; &lt;HeartDisease xmlns="aaasa"&gt; &lt;Node id="0"&gt; &lt;Disease&gt;Normal&lt;/Disease&gt;&lt;Status&gt;T&lt;/Status&gt; &lt;Node id="1"&gt; &lt;Symptomp&gt;shob, palp&lt;/Symptomp&gt; &lt;Disease&gt;Heart Attack&lt;/Disease&gt; &lt;Status&gt;T&lt;/Status&gt; &lt;Node id="2"&gt; &lt;Symptomp&gt;ahrb&lt;/Symptomp&gt; &lt;Disease&gt;Heart Attack&lt;/Disease&gt; &lt;Status&gt;F&lt;/Status&gt; &lt;/Node&gt; &lt;/Node&gt;&lt;/HeartDisease&gt;</pre>

C. Basis Data

Basis pengetahuan yang digunakan pada sistem pakar ini akan disimpan dalam format XML (*Extensible Mark-up Language*). XML digunakan karena xml merupakan basis data yang berbasis teks sehingga dapat digunakan pada berbagai platform. XML hanya dapat berlaku sebagai basis data, yang artinya untuk mengolah data pada XML tersebut dibutuhkan *parser*. *Parser* berperan dalam mengurai data pada dokumen XML sehingga XML dapat di-write atau di-read oleh sistem pakar yang dibuat pada paper ini.

Struktur basis data pada sistem pakar ini dirancang agar dapat memenuhi fungsionalitas dari sistem pakar itu sendiri. Adapun struktur dari dokumen XML yang akan digunakan adalah sebagai berikut :

```
<Node id="19">
<Symptomp>kuning, merah</Symptomp>
<Disease>apel</Disease>
<Status>F</Status> </Node>
```

Tag <Node> merupakan tag pembuka yang berisi isi dari sebuah *rule* dari sistem pakar. Didalam tag <Node> ada tag <Symptomp> yang berisi kondisi apa yang dikandung oleh *rule* tersebut, tag <Disease> yang berisi kesimpulan penyakit apa pada *rule*, dan tag <Status> yang berisi kondisi apa yang harus dipenuhi pada *rule* sebelumnya untuk dapat mengakses *rule* yang sekarang.

D. Algoritma Program

Sistem pakar pada paper ini dibangun menggunakan Bahasa pemrograman Java dengan DOM sebagai *Parser* untuk dokumen XML yang menjadi basis data dari sistem pakar ini. Sistem akan meminta user untuk memasukkan gejala-gejala yang akan diproses oleh mesin inferensi dari sistem pakar. Berdasarkan *flowchart* algoritma diatas, spesifikasi dari sistem pakar menggunakan metode *Ripple-down Rule* untuk akuisisi pengetahuan ini adalah sebagai berikut :

- Sistem akan selalu memeriksa apakah basis pengetahuan sudah berisi atau belum
- Representasi struktur basis pengetahuan ialah seperti *Binary Tree*, dengan tiap cabang/Node merupakan representasi dari *rule* dari basis pengetahuan sistem pakar ini.
- Basis pengetahuan yang dianggap belum berisi merupakan basis pengetahuan yang hanya mempunyai *rule default* yaitu, *if True then normal*.
- Mesin inferensi sistem pakar mempunyai *pointer* untuk menunjuk kepada node yang telah diperiksa atau melalui *iterasi* untuk mengetahui apakah kondisi dalam *rule* yang sedang diperiksa cocok dengan masukan .

- Tiap penambahan *rule/node* baru merupakan penambahan cabang anak dari *pointer* berupa :
  - *Except* : cabang yang ditambahkan sebagai anak *pointer* yang disebut di atas apabila *rule* sebelumnya terpenuhi kondisinya namun diagnosanya belum tepat, sehingga diagnosa yang baru ditambahkan untuk memperbaiki *rule* sebelumnya.
  - *Else* : cabang yang ditambahkan apabila *rule* sebelumnya tidak terpenuhi kondisinya sehingga menambahkan *rule* yang baru dengan diagnosa yang benar.
- Penambahan *rule/node* baru apabila user menentukan bahwa diagnosa sistem tidak tepat atau tidak sesuai pakar. Kemudian user menentukan kondisi dan hasil diagnosa yang tepat pada *rule* baru.
- Setiap setelah penambahan *rule* akan diposisikan sebagai anak pertama apabila memiliki *Status\_Node = True*, dan sebagai anak terakhir apabila memiliki *Status\_Node = False*. Hal ini dikarenakan untuk konsistensi proses *parsing* basis data XML dan struktur data basis pengetahuan dengan memposisikan cabang *true* selalu disebelah kiri dan cabang *false* disebelah kanan pada representasi struktur *Binary tree* basis pengetahuan sistem pakar ini.
- Setiap selesai pemeriksaan diagnosis atau setelah penambahan *rule* baru, sistem akan terus mengulang pertanyaan untuk memasukkan gejala yang ingin didiagnosa kecuali *user* menentukan untuk menghentikan proses tersebut.

#### IV. HASIL PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan terkait dengan perancangan, eksperimen dan analisi hasil. Untuk lebih jelasnya akan di paparkan di bawah ini.

##### A. Tujuan Pengujian

Pengujian sistem ini dilakukan untuk dapat mengetahui kemampuan dan kelemahan sistem yang sudah dibangun. Pengujian yang dilakukan menurut skenario sebagai berikut :

1. Pelatihan terhadap sistem yang dalam keadaan kosong atau belum ada pengetahuan sama sekali. Pelatihan ini dilakukan untuk mengetahui bagaimana sistem menerima kasus baru saat dalam keadaan kosong.
2. Pengujian terhadap sistem yang sudah ada pengetahuan yang tepat sesuai pakar didalamnya dan diuji menggunakan kasus yang sudah benar. Pengujian ini dilakukan untuk mengetahui apakah sistem dapat memberikan solusi yang tepat apabila

diberikan kasus yang sesuai dengan pengetahuan yang ada didalamnya.

3. Pengujian terhadap sistem yang sudah terdapat pengetahuan yang belum tepat/lengkap didalamnya dan diuji menggunakan kasus yang sudah benar menurut pakar. Pengujian ini dilakukan untuk mengetahui kemampuan sistem mengenali kasus yang diberikan dan kemampuan untuk memperbaiki kasus yang belum tepat menurut pakar menjadi kasus yang tepat pakar.

##### B. Pelatihan Sistem

Sistem pakar yang masih kosong membutuhkan pelatihan sebelum dapat memenuhi fungsinya sebagai sistem yang memberikan hasil diagnosa yang seperti pakar. Dalam pelatihan ini, sistem akan diberikan kondisi atau *rule* yang sesuai dengan diagnosa pakar untuk dipelajari oleh sistem. Tabel berikut adalah langkah sistem dalam tahap pelatihan.

Tabel 3 Langkah Pelatihan

Langkah	Algoritma
1	Sistem menerima masukan berupa sebagian atau seluruh gejala penyakit jantung
2	Sistem akan memberikan diagnosa berdasarkan pengetahuannya
2a.	Apabila basis pengetahuan masih kosong maka diagnosa sistem adalah "normal"
3	Sistem akan menanyakan apakah hasil diagnosanya benar atau salah
4	apabila salah maka sistem akan meminta user memasukkan diagnosa yang benar
5	apabila benar maka sistem akan memeriksa kondisi berhenti

Pelatihan sistem tahap pertama dilakukan dengan memasukkan hanya sebagian dari gejala-gejala berbagai jenis penyakit jantung kemudian memberi masukan diagnosa yang tepat berdasarkan menurut pakar. Hal ini dilakukan agar dalam pengujian lebih lanjut nanti sistem dapat diuji coba dengan perbaikan basis pengetahuan yang sudah ada. Berikut ini adalah tabel hasil pelatihan sistem tahap pertama.

Tabel 4 Hasil Pelatihan

Case	Masukan Gejala	Diagnosa Sistem	Diagnosa Pakar
1	shob, palp, weak	normal	arrhythmi as

2	shob,palp,w eak,fash	arrhythmias	coronary artery disease
3	anx, full, dizzi, naus	normal	heart attack
4	shob,palp,di zzi,disch	normal	atrial fibrilatio n
5	shob, palp, weak, dizzi, disch	arrhythmias	heart valve disease
6	quwe,disch, shob,palp,w eak,dizzi	heart valve disease	heart valve disease + heart attack
7	swan,swle,s wab,shob,p alp,weak,diz zi	arrhythmias	heart failure
8	shob,palp,w eak,dizzi,na us,abhrp,ab hir,quwe	heart failure	congenit al heart defects
9	limate,cyan o,fasbr,pofe d,poweg,rec clng	atrial fibrillation	congenit al heart defects in infants and children
10	abhrp, abhir, faint, quwe, fatig, cougws, disch	congenital heart defects in infants and children	Heart Muscle Disease
11	shcpa, logfe, fash	Heart Muscle Disease	Pericardi rtis

Pada kolom Diagnosa Pakar, merupakan diagnosa yang seharusnya menurut pakar yang kemudian dimasukkan sebagai perbaikan diagnosa awal sistem.

### C. Pengaruh Urutan Masukan Kasus

Dalam tahap ini akan diuji pengaruh urutan masukan kasus berdasarkan gejala. Tahap uji ini akan dibagi menjadi dua bagian, yaitu : Bagian pertama yang menguji sistem pakar dengan masukan kasus yang memiliki banyak gejala. Pada bagian ini jumlah gejala masukan tiap kasusnya jumlahnya dari

gejala yang banyak secara berurutan hingga kasus yang gejala masukannya sedikit.

Bagian kedua menguji sistem pakar dengan masukan kasus dari gejala yang sedikit secara berurutan hingga gejala yang banyak.

Dari pengujian dua bagian di atas akan dilihat pengaruh urutan masukan kasus terhadap bentuk struktur basis pengetahuannya.

1) Berikut ini adalah tabel tahap-tahap dan hasil pengujian sistem dengan urutan masukan kasus dari gejala yang banyak hingga masukan gejala yang sedikit tiap kasusnya

Tabel 5 Pengujian Faktor Urutan Bagian 1

Case	Gejala-gejala	Diagnosa Sistem	Diagnosa Pakar
1	abhir,abhrp,anx,cho kfe,disarm,disback,d isbebr,disch,disjaw, dizzi	normal	Heart Attack
2	abhir,abhrp,cougws, disch,dizzi,faint,fati g,palp,quwe	normal	Heart Muscle Disease
3	abhir,abhrp,cougws, disch,limate,naus,sw ab,swan,swle	normal	Heart Failure
4	cougws,disch,limate ,naus,weak,palp,sho b,quwe	normal	Congenital Heart Defects
5	disch,dizzi,palp,quw e,shob,swab,swan	normal	HVD causing heart failure
6	abhir,abhrp,disch,sh ob,palp,weak	normal	arrhythmias
7	dizzi,fash,naus,palp, shob,sweat	normal	coronary artery disease
8	cyano,fasbr,pofed,p oweg,recclng	normal	CHD in infants or Children
9	shob palp,weak,quwe,diz zi	normal	heart valve disease
10	disch,dizzi,palp,sho b	normal	atrial fibrillation
11	fash,logfe,shcpa	normal	pericarditis

Berdasarkan struktur basis pengetahuan yang dihasilkan dapat disimpulkan bahwa dengan urutan masukan kasus yang kasus-kasus di awal memiliki gejala yang banyak akan cenderung menciptakan kaidah baru dengan cabang else. Sehingga dengan masukan gejala yang semakin sedikit tiap kasusnya akan membuat struktur representasi tree basis pengetahuan ini memiliki tingkatan yang lebih banyak. Hal ini disebabkan tiap kali sistem menerima kasus dan kasus tersebut memiliki gejala yang lebih sedikit dari kasus sebelumnya maka kaidah yang sebelumnya ditambahkan tidak akan terpenuhi karena kasus yang sekarang gejala-gejalanya tidak cukup untuk memenuhi gejala kaidah yang ditambahkan hasil kasus sebelumnya.

2) Berikut ini adalah tabel tahap-tahap dan hasil pengujian sistem dengan urutan masukan kasus dari gejala sedikit hingga gejala yang banyak tiap kasusnya.

Tabel 6 Pengujian Faktor Urutan Bagian 2

Case	Gejala-gejala	Diagnosa Sistem	Diagnosa Pakar
1	fash	normal	Pericarditis
2	palp,shob	normal	Atrial Fibrillation
3	palp,quwe,shob,weak	atrial fibrillation	Heart Valve Disease
4	cyano,fasbr,limat,ped	normal	CHD in Infants and Children
5	dizzi,fash,alpal,shob,weak	pericarditis	Coronary Artery Disease
6	shob,alpal,weak,dish,dizzi,faint	atrial fibrillation	Arrhythmias
7	shob,alpal,quwe,swab,swan,swf	atrial fibrillation	HVD causing heart failure
8	shob,alpal,weak,quwe,cougws,abhrp,swle	Heart valve Disease	Congenital Heart Defects
9	abhir,abhrp,cougws,disch,dizzi,naus,alpal	normal	Heart Failure
10	abhrp,abhir,cougws,disch,dizzi,faint,fatig,naus	normal	Heart Muscle Disease
11	abhir,abhrp,anx,choke,disarm,disback,dishbebr,disch	normal	Heart Attack

Dapat disimpulkan bahwa memasukkan gejala kasus dimulai dengan gejala-gejala yang jumlahnya sedikit kemudian perlahan jumlahnya ditambah tiap pengujian kasusnya akan membuat representasi tree basis pengetahuannya lebih cenderung merata atau tidak akan selalu menggunakan cabang else untuk menambahkan kaidah baru tiap kasusnya.

#### D. Analisis Hasil

Berdasarkan hasil pengujian, dapat dilihat bahwa sistem pakar yang menggunakan metode *Ripple-down rules* (RDR) ini akan selalu memberikan diagnosa yang tepat apabila kondisi masukan ke sistem sesuai dengan kondisi yang terdapat di basis pengetahuan sistem. Apabila ada diagnosa yang tidak tepat maka dapat dipastikan bahwa penyebabnya adalah data pelatihan sistem yang tidak tepat atau sistem belum dilatih dengan jumlah yang cukup sehingga kondisi-kondisi yang terdapat dalam *rules* pada basis pengetahuan sistem belum lengkap.

## V. KESIMPULAN DAN SARAN

Dari hasil penelitian dan pengujian yang telah dilakukan dapat disimpulkan bahwa metode RDR dapat membuat suatu sistem pakar lebih fleksibel dengan kemampuan memperbarui basis pengetahuannya tanpa memerlukan adanya *Knowledge Engineer*. Hasil diagnosanya pun sangat akurat dengan catatan, sistem mendapat pelatihan dengan data dan jumlah yang cukup. Setelah melakukan berbagai tahap pengujian dapat juga disimpulkan bahwa algoritma RDR yang digunakan pada tugas akhir ini :

1. User dapat menggunakan sistem pakar dengan algoritma RDR dengan mudah tanpa perlu mengetahui mengenai Knowledge Engineering. Karena algoritma RDR sudah secara otomatis mengatasi urutan dan struktur basis pengetahuan sistem pakar.

2. Struktur basis pengetahuan yang dimasukkan menggunakan algoritma RDR akan bergantung kepada urutan masukan kasus.

3. Hasil diagnosa sistem pakar akan bergantung kepada pakar yang menggunakan sistem pakar ini. Karenanya sistem pakar ini bertujuan untuk membantu pakar bukan untuk menggantikan pakar.

Dengan adanya penelitian dan pengujian diharapkan akan ada pengembangan dan penerapan algoritma *Ripple-down Rules* untuk dapat lebih memaksimalkan fungsionalitasnya pada sistem pakar. Sehingga mungkin untuk menggabungkan algoritma RDR dengan metode lain.

## VI. REFERENCES

- [1] Anonymous. n.d. *Sistem Pakar*. Accessed April 10, 2014. <http://3onoikom.wordpress.com/materi-kuliah/sistem-pakar/>.
- [2] Begg., Thomas M. Connolly and Carolyn E. 2002. *Database Systems : A Practical Approach to Design, Implementation, and Management, Third Edition*. . Massachusetts: Addison-Wesley.
- [3] Compton, P, and R Jansen. n.d. "A philosophical basis for knowledge acquisition." Sydney.
- [4] Compton, P, G Edwards, B Kang, L Lazarus, R Malor, T Menzies, P Preston, A Srinivasan, and S Sammut. n.d. "Ripple down rules : Possibilities and limitations." Sydney.
- [5] Compton, P, L Peters, G Edwards, and G Lavers T. n.d. "Experience with Ripple-Down Rules." Sydney.
- [6] Farhan, Alief. 2012. *Sistem Pakar Untuk Diagnosa Penyakit Gigi Menggunakan Induct/Ripple Down Rules (Induct/RDR)*. Bandung: Institut Teknologi Telkom.
- [7] Islamee, Ayu Ummu. 2008. *FAKTOR-FAKTOR RISIKO PENYAKIT JANTUNG*. Jakarta Pusat: Universitas Indonesia.
- [8] Majid, Abdul. 2007. *PENYAKIT JANTUNG KORONER : PATOFISIOLOGI, PENCEGAHAN, DAN PENGOBATAN TERKINI*. Medan: Universitas Sumatera Utara.

- [9] Marini, Febianti Roris. 2010. *SISTEM PAKAR MENGGUNAKAN METODE RIPPLE DOWN RULE*. Bandung: Institut Teknologi Telkom.
- [10] Ridloh, Rochman. 2008. *IMPLEMENTASI RIPPLE DOWN RULE LEARNER PADA DATA PELELANGAN*. Bandung: Institut Teknologi Telkom.
2008. *Ripple-down rules*. November. [http://en.wikipedia.org/wiki/Ripple-down\\_rules](http://en.wikipedia.org/wiki/Ripple-down_rules).
- [11] Wada, Takuya, Tadashi Horiuchi, Hiroshi Motoda, and Takashi Washio. n.d. *Characterization of Default Knowledge in Ripple Down Rules Method*. Osaka: Institute of Scientific and Industrial Research, Osaka University.
- [12] WebMD Organization. 2005-2015. *Heart Disease Symptoms & Signs of Heart Problems*. Accessed 11 24, 2014. <http://www.webmd.com/heart-disease/guide/heart-disease-symptoms>.