

# APLIKASI SISTEM PAKAR BERBASIS WEB UNTUK MENDIAGNOSA AWAL PENYAKIT JANTUNG

Leo Willyanto Santoso<sup>1</sup>, Agustinus Noertjahyana<sup>2</sup>, Ivan Leonard<sup>3</sup>

<sup>1,2</sup>Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra  
Jl. Siwalankerto 121-131 Surabaya 60236  
<sup>1</sup>[leow@petra.ac.id](mailto:leow@petra.ac.id), <sup>2</sup>[agust@petra.ac.id](mailto:agust@petra.ac.id)

---

## Abstrak

Jantung merupakan organ tubuh manusia yang mempunyai peran penting dalam kehidupan manusia dan pastinya sangat berbahaya jika jantung kita mempunyai masalah mengingat bahwa banyak kematian disebabkan oleh penyakit jantung. Tapi dengan pengetahuan dan informasi yang minim, mustahil untuk dapat menjaga kesehatan jantung. Oleh karena itu dibutuhkan seorang pakar yang ahli tentang jantung dan macam-macam penyakitnya. Berdasarkan fakta diatas, maka penelitian ini dapat membantu kita untuk mendiagnosa kesehatan jantung dan mengantisipasi jika mempunyai resiko penyakit jantung dengan merancang dan mengimplementasikan. Aplikasi ini dibuat berbasis *web* dengan menggunakan PHP dan *database MySQL*. Pada sistem pakar ini diajukan beberapa pertanyaan. Setelah semua pertanyaan terjawab, maka akan tampak hasil diagnosa beserta saran yang dapat membantu mengantisipasi penyakit jantung tersebut.

**Kata kunci** : Sistem pakar, *backward chaining*, penyakit jantung

---

## 1. Pendahuluan

Jantung merupakan organ yang sangat penting bagi manusia. Jantung adalah pusat kehidupan bagi manusia. Faktor kesehatan jantung juga dipengaruhi oleh pola makanan dan pola pikir manusia tersebut. Tanda – tanda penyakit jantung pada manusia antara lain nafas berat, rasa sakit pada dada, rasa sakit pada punggung, berkeringat dingin, pingsan, gemeteran, rasa panas pada dada. Salah satu jenis penyakit jantung pada manusia adalah penyakit jantung Koroner. Penyakit ini mempunyai ciri – ciri gejala kadar kolesterol LDL yang tinggi, kadar kolesterol HDL yang rendah, tekanan darah tinggi, merokok, diabetes mellitus, kegemukan, riwayat penyakit jantung pada keluarga, kurang olah raga dan stres. Kebanyakan masyarakat awam sangat kurang memperhatikan kesehatan, terutama kesehatan jantung. Penyakit jantung dapat menyerang siapa saja dan ada juga penyakit jantung bawaan dari lahir. Biasanya masyarakat enggan memeriksakan kesehatan jantungnya karena biaya yang harus dikeluarkan sangatlah mahal dan tenaga spesialis juga masih jarang dijumpai terutama di daerah pedesaan.

Dengan perkembangan teknologi yang sangat modern, maka dikembangkan pula suatu teknologi yang mampu mengadopsi cara berfikir manusia yaitu teknologi *Artificial Intelligence* atau kecerdasan buatan. Sistem pakar adalah salah satu

bagian dari kecerdasan buatan yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang memerlukan keahlian manusia. Tujuan dari sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mempresentasikan pengetahuan manusia dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak. Sistem pakar akan memberikan solusi yang memuaskan layaknya seorang pakar. Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia pada salah satu bidang yang spesifik.

Berdasarkan hal di atas, dengan mengandalkan kemajuan di bidang teknologi dan informasi maka penelitian yang berjudul "Aplikasi *Expert System* Berbasis *Web* Untuk Mendiagnosa Awal Penyakit Jantung" menjadi sangat penting guna memberikan pencegahan awal masyarakat terhadap penyakit jantung. Proses dari sistem pakar ini adalah dengan memberikan pertanyaan tentang kondisi fisik maupun kondisi non fisik. Dari data yang didapat kemudian diproses sehingga dapat menyimpulkan penyakit jantung apa yang diderita oleh masyarakat serta bagaimana cara pencegahannya.

Tujuan dari pembuatan aplikasi ini adalah membuat suatu *web* sistem pakar analisa penyakit jantung pada manusia, dimana pada *web* ini terdapat sistem yang dapat menganalisa penyakit jantung serta memberikan saran dan pencegahan yang

terbaik dan juga berisi tentang informasi penting yang berguna bagi masyarakat tentang apa itu penyakit jantung, gejala serta cara pencegahannya serta fasilitas konsultasi langsung dengan dokter jantung melewati *web*.

Permasalahan yang dihadapi dalam penelitian ini adalah:

1. Bagaimana membuat sebuah *web* sistem pakar yang menarik, tidak membosankan serta *user friendly*.
2. Bagaimana membuat sebuah *web* menjadi sebuah sistem pakar menganalisa penyakit jantung secara *online*.
3. Bagaimana menganalisa sebuah penyakit jantung pada manusia.
4. Bagaimana membuat suatu *web* informasi yang menarik dan lengkap tentang beberapa informasi tentang apa itu jantung serta beberapa penyakit jantung.
5. Bagaimana membuat *rule* baru terhadap penyakit jantung pada sistem yang ada di dalam *web*.

Manfaat dari pembuatan *web* ini adalah untuk memudahkan *user* dalam menganalisa awal penyakit jantung dan memberikan pengetahuan tentang penyakit jantung serta gejala – gejalanya disertai tindakan yang harus diambil untuk pencegahannya sebagai langkah awal dalam mengantisipasi penyakit jantung.

## 2. Sistem Pakar

### 2.1 Pengertian

Sistem pakar adalah salah satu cabang dari *Artificial Intelligence* (AI) yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar [1, 2]. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya.

Seorang pakar (*human expert*) adalah seorang individu yang memiliki kemampuan pemahaman yang superior dari suatu masalah. Misalnya: seorang dokter, penasehat keuangan, pakar mesin motor. Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. Faktor perbandingan kemampuan antara seorang pakar dengan sebuah sistem pakar dapat dilihat pada Tabel. 1.

Tabel 1. Perbandingan antara Human Expert dan Sistem Pakar

Faktor	Human Expert	Sistem Pakar
Time availability	Hari kerja	Setiap saat

Geografis	Lokal / tertentu	Dimana saja
Keamanan	Tidak tergantikan	Dapat diganti
Dapat habis	Ya	Tidak
Performansi	Variabel	Konsisten
Kecepatan	Variabel	Konsisten
Biaya	Tinggi	Terjangkau

Ada beberapa alasan mendasar mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar, di antaranya:

- Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
- Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
- Seorang pakar akan pensiun atau pergi. Seorang pakar memerlukan biaya mahal

### 2.2 Knowledge Base

*Knowledge Base* berisi informasi atau pengetahuan yang diberikan oleh pakar dan diimplementasikan ke dalam sistem komputer dengan menggunakan metode representasi tertentu [2]. Metode representasi pengetahuan adalah cara untuk menstrukturkan pengetahuan yang dimiliki oleh pakar agar mudah diolah oleh computer. Pengetahuan tersebut merupakan bahan dasar dari sebuah sistem pakar karena keahlian pakar disimpan didalamnya. Informasi atau fakta yang dimasukkan oleh user selama berkonsultasi akan dicocokkan oleh mesin pengambil keputusan dengan pengetahuan yang terdapat dalam basis pengetahuan. Basis pengetahuan itu tidak statis, maka memerlukan modifikasi dan perbaikan sejalan dengan perubahan kondisi dalam problem domain.

### 2.3 Working Memory

Dalam *Working Memory* inilah terkandung semua fakta-fakta, baik fakta awal pada saat sistem mulai beroperasi maupun fakta-fakta yang didapat pada saat pengambilan kesimpulan dilaksanakan [2]. Basis data berada di dalam memori komputer. Terkadang basis data ini disebut basis data global karena merupakan rangkaian informasi yang luas tentang status masalah yang sudah dipecah-pecah. Basis data menerima penjelasan dari awal masalah yang akan diselesaikan sistem. Program kontrol akan mencocokkan inputan awal atau informasi dalam basis data dengan aturan-aturan dalam basis pengetahuan. Isi basis data diubah sejalan dengan perkembangan prosedur pencocokan

### 2.4 IF THEN Rules

Dari *decision table* yang telah tereduksi, setiap barisnya dapat dikonversikan menjadi *IF-THEN*

rule. Setiap baris pada *decision table* tereduksi akan membentuk satu set *rule* final.

Struktur dan *syntax* penulisan *rule* adalah sebagai berikut [4]:

- *RULE* label : Label berisi nama *rule* tersebut.
- *IF* : Sebagai penanda awal kondisi pada sebuah *rule*.
- *THEN* : Sebagai penanda awal kesimpulan pada sebuah *rule*.
- *ELSE* : Sebagai penanda awal alternatif kesimpulan pada sebuah *rule*, bersifat opsional, jadi boleh tidak ada.

Operator yang dapat digunakan pada *IF-THEN rule* adalah:

- *AND* : Semua kondisi yang dihubungkan oleh operator ini harus bernilai benar, agar kondisi keseluruhan *rule* tersebut bernilai benar. Bila ada satu kondisi yang bernilai salah, keseluruhan *rule* tersebut bernilai salah.
- *OR* : Bila semua kondisi yang dihubungkan oleh operator ini harus bernilai salah, maka kondisi keseluruhan *rule* tersebut bernilai salah. Bila ada satu kondisi atau lebih yang bernilai benar, keseluruhan *rule* tersebut bernilai benar.

### 2.5 Inference Engine

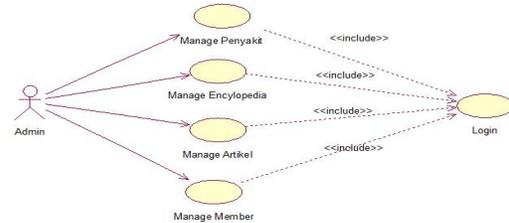
*Inference Engine* adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar [2, 4]. Mekanisme mesin ini akan menganalisa suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan terbaik. Mesin inferensi ini memulai pelacakannya dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam sistem basis data. Ada tiga tipe teknik infrensi, yaitu pelacakan ke belakang (*backward chaining*) yang memulai penalarannya dari kesimpulan hipotesa menuju fakta-fakta yang mendukung hipotesa tersebut, pelacakan ke depan (*forward chaining*) yang merupakan kebalikan dari *backward chaining*, yaitu memulai dari sekumpulan data menuju kesimpulan, dan gabungan dari *backward chaining* dan *forward chaining*.

### 2.6 Backward Chaining

Proses ini dimulai dari pencarian solusi dari kesimpulan kemudian menelusuri fakta-fakta yang ada hingga menemukan solusi yang sesuai dengan fakta-fakta yang diberikan oleh *user* [2, 4]. *Backward chaining* merupakan proses penalaran dengan pendekatan *goal\_driven*. Pendekatan *goal\_driven* memulai titik pendekatannya dari *goal* yang akan dicari nilainya kemudian bergerak untuk mencari informasi yang mendukung *goal* tersebut.

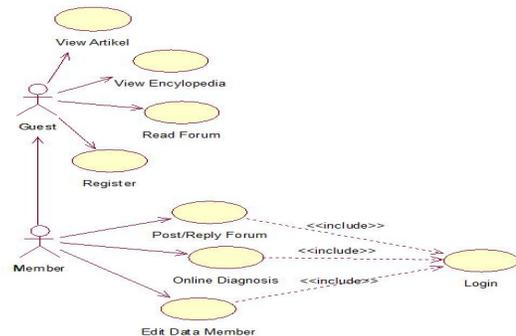
## 3. Desain dan Implementasi Sistem

*Use Case Diagram* untuk *website admin* pada sistem pakar untuk pemeriksaan awal gejala penyakit jantung dapat dilihat pada Gambar 1.



Gambar 1. *Use Case Diagram Website Admin*

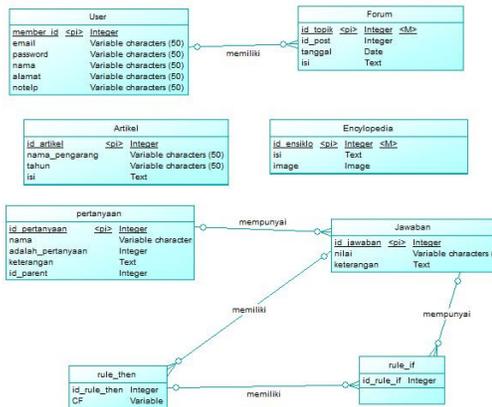
Pada Gambar 1. dapat dilihat bahwa *admin* dapat menambah dan mengubah data yang ada dalam *website*. Desain *use case diagram* untuk *website Member* dan *Guest* dapat dilihat pada Gambar 2.



Gambar 2. *Use Case Diagram Member dan Guest*

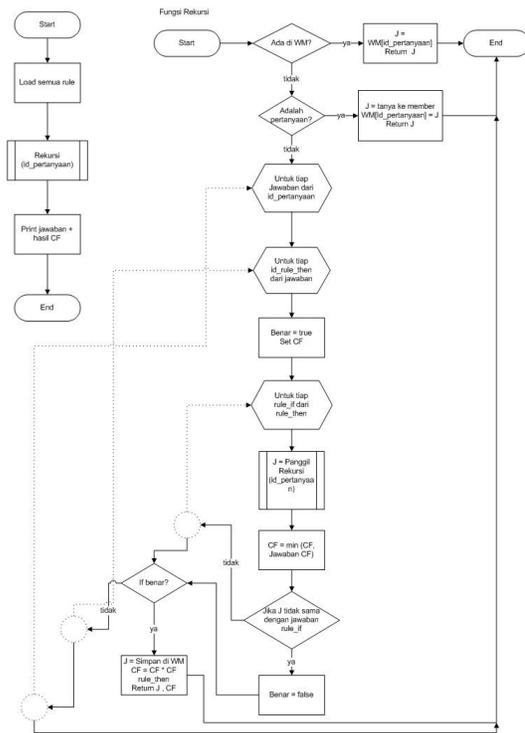
Pada Gambar 2 dapat dilihat bahwa didalam *website* terdapat 2 hak akses yang dapat mengakses *website* ini yaitu *Member* dan *Guest*. Dimana *member* dapat melakukan fasilitas berdiskusi dengan *member* lain melalui fitur Forum, selain itu *member* juga bisa menggunakan fitur *online diagnosis* yaitu dapat mengecek kesehatan jantungnya serta dapat mengedit data diri dari *member* tersebut. Sedangkan untuk *guest* hanya bisa melihat artikel, encyclopedia, membaca forum dan juga dapat melakukan registrasi untuk mendaftar menjadi *member*.

Desain dari ERD pada sistem ini dapat dilihat pada Gambar 3. Suatu *entity* dapat berelasi dengan *entity* lainnya melalui hubungan asosiasi. Asosiasi harus dispesifikasikan dalam jumlah objek yang saling berkaitan atau disebut juga dengan multiplisitas. Contoh-contoh multiplisitas adalah *one to one*, *one to many*, *many to one*, dan *many to many*. Pada ERD sistem pakar ini ada 6 relasi yang menghubungkan *entity* dengan *entity* yang lain. Setiap *entity* menghasilkan tabel untuk dimasukan kedalam *database*. *Conceptual ERD* harus diturunkan ke *Physical ERD*.



Gambar 3. Conceptual ERD

- Kemudian *user* akan diberikan beberapa pertanyaan yang diberikan oleh sistem, pertanyaan berupa gejala-gejala yang mungkin dialami oleh *user*.
- Setelah *user* menjawab semua pertanyaan maka sistem akan memproses hasil jawaban dari *user*.
- Teori yang dipakai adalah *Backward Chaining* yaitu memproses jawaban *user* berdasarkan pendekatan *goal* dan juga menggunakan teori *Certainty Factor* yaitu sistem akan memproses nilai dari masing-masing gejala berdasarkan jawaban yang diinputkan oleh *user*. Nilai CF dari setiap penyakit adalah total nilai dari masing-masing gejala dari setiap penyakit.
- Kemudian sistem akan menampilkan perkiraan resiko yang dialami oleh *user* serta menampilkan saran untuk mencegah resiko tersebut

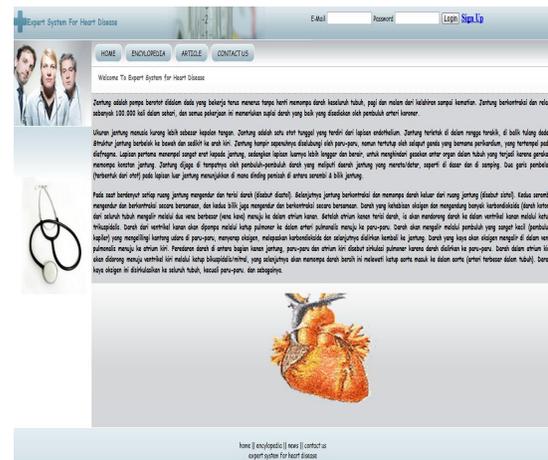


Gambar 4. Flowchart Sistem Pakar

Dari Gambar 4 dapat dilihat bahwa ada beberapa peraturan yang harus diperhatikan oleh *user*:

- *User* harus mendaftarkan diri di menu *register* lalu *user* mengisi *form* yang sudah disediakan dan harus mengisi semua kolom yang ada dengan benar.
- Setelah selesai mengisi *form register*, *user* akan mendapatkan *member\_id* dan juga *password*. *User* harus mengingat *member\_id* dan *password*.
- Setelah mendapatkan *member\_id* dan *password* maka *user* dapat melakukan *online diagnosis*.

Pada halaman aplikasi ini ditampilkan menu utama yang ada didalam *web* dan juga menampilkan informasi tentang apa itu jantung dan bagaimana jantung bekerja didalam tubuh kita. Halaman utama dapat dilihat pada Gambar 5..



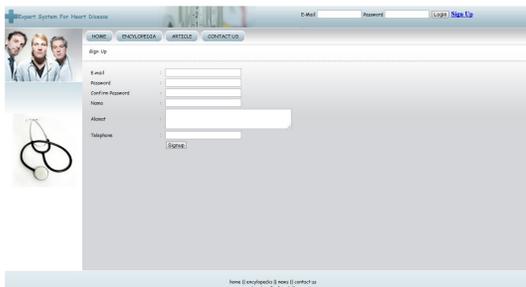
Gambar 5. Halaman Utama

*User* tidak perlu melakukan proses *login* jika tujuannya hanya melihat informasi tentang penyakit jantung di *encyclopedia page* ataupun melihat berita-berita tentang jantung di *article page*. Pada Gambar 6. menunjukkan *user* masuk ke bagian *encyclopedia page*.



Gambar 6. Halaman Ensiklopedia

Pada bagian halaman ensiklopedia ini terdapat bermacam-macam jenis penyakit jantung yang bisa dibaca baik oleh *user* ataupun oleh *member*. Terdapat juga fitur *search* yang membantu *user* untuk mencari informasi yang dibutuhkan.



Gambar 7. Halaman Registrasi Member

Gambar 7. adalah halaman registrasi yang digunakan untuk *user* mendaftar menjadi *member* agar bisa melakukan *online diagnosis* dan juga dapat berkonsultasi dengan dokter. Setelah mendaftar, maka *user* dapat melakukan *login* yang terdapat pada halaman utama di *website* ini.

*Member* juga dapat memilih menu *online diagnosis* yang berfungsi untuk membantu *member* dalam memeriksakan kesehatan jantungnya. Pada studi khusus, telah dicoba aplikasi ini kepada penderita jantung koroner sebut bernama A, jantung koroner memiliki resiko diantara lain adalah rasa nyeri yang terjadi pada dada, dada berdebar secara mendadak, sesak nafas. Dalam hal ini *member A* telah menjawab beberapa pertanyaan yang disediakan oleh pakar, dalam pertanyaan pakar pilihan tingkat keyakinan *member A* dalam menjawab jawaban dalam pertanyaan tersebut, pilihan tingkat keyakinan tersebut dapat menunjukkan berapa tingkat keyakinan *member* terhadap jawabannya. *Online diagnosis* dapat dilihat pada Gambar 8.

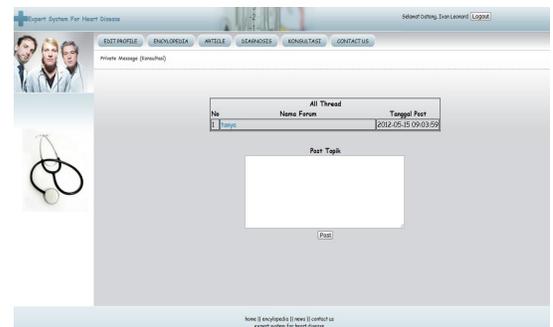


Gambar 8. Pertanyaan Pakar



Gambar 9. Hasil Diagnosa.

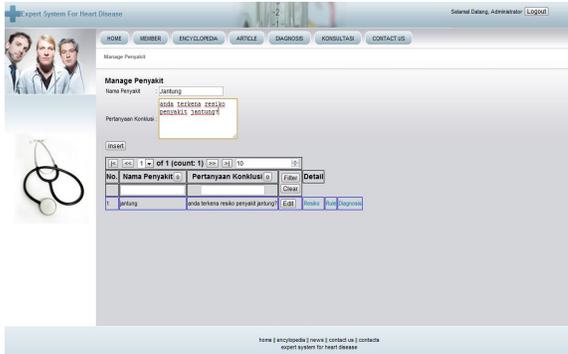
Pada Gambar 9 menunjukkan hasil dari diagnosa yang dilakukan oleh *member A* yang mendapatkan hasil bahwa A terkena penyakit jantung koroner yang beresiko ringan, terdapat juga saran dan solusi yang berguna untuk *member A*.



Gambar 10. Konsultasi

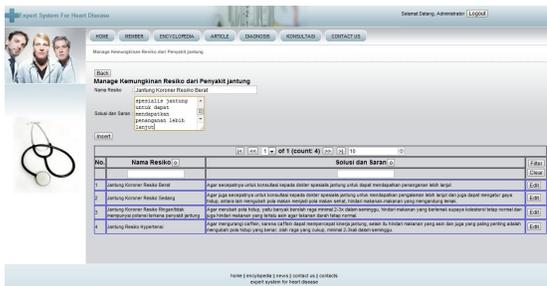
Pada Gambar 10 merupakan fitur konsultasi yang dilakukan oleh *member* untuk dapat berkonsultasi langsung dengan dokter. *Member* mengisi pertanyaan apa yang akan ditanyakan ke dokter.

Halaman berikut dikhususkan untuk *admin online diagnosis* yang dapat digunakan untuk menambah, merubah dan menghapus data yang diinginkan. Gambar 11. adalah halaman untuk menambah suatu penyakit baru. Pada halaman ini *admin* harus mengisi nama penyakit dan pertanyaan konklusi untuk mendukung penyakit tersebut, apabila *admin* ingin mengubah nama penyakit maka *admin* dapat menekan tombol *edit*.



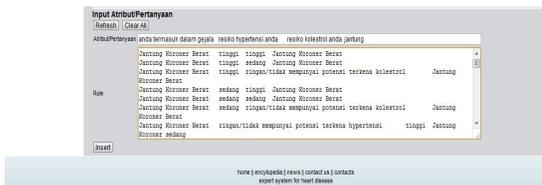
Gambar 11. Halaman *Manage Insert* Penyakit

Setelah *admin* selesai mengubah data penyakit, maka *admin* juga dapat menambah resiko dari penyakit tersebut dengan cara menekan menu Resiko di samping menu *edit*. Di dalam halaman *manage* resiko *admin* juga dapat mengubah data resiko dimana caranya sama dengan cara mengubah data penyakit. Setelah *admin* menambah resiko penyakit pada Gambar 12, *admin* dapat mengubah data resiko penyakit dengan cara menekan tombol *edit* disetiap resiko.



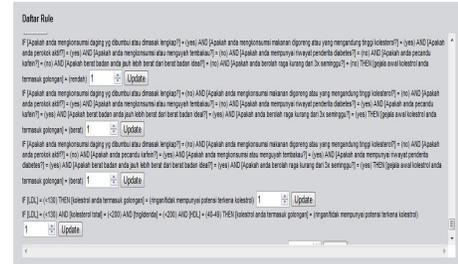
Gambar 12. Halaman *Manage Insert* Resiko Penyakit

*Admin* juga dapat menambahkan rule sistem pakar, yaitu dapat memasukkan rule penyakit dengan cara mengambil rule yang telah dibuat di *microsoft excel* kemudian *copy* rule tersebut kedalam sebuah *textbox* yang telah disediakan oleh *website* ini. Hal ini dapat dilihat pada Gambar 13.



Gambar 13. *Input Rule*

Setelah *admin* selesai menginputkan pertanyaan dan juga rule penyakit jantung maka *website* ini akan menampilkan hasil rule yang telah direduksi dengan model *dependency diagram* tentang penyakit jantung tersebut. Daftar rule yang ada dapat dilihat pada Gambar 14.



Gambar 14. Daftar Rule

#### 4. Kesimpulan dan Saran

Dari hasil perancangan dan pembuatan aplikasi *Expert System* berbasis *web* untuk mendiagnosa awal penyakit jantung dapat diambil beberapa kesimpulan sebagai berikut:

- Aplikasi sudah dapat membantu *member* untuk mendiagnosa dan berkonsultasi langsung dengan dokter.
- Dari hasil kuisioner yang dibagikan kepada beberapa *user* dan *member*, dapat ditarik kesimpulan bahwa secara keseluruhan aplikasi yang dibuat sudah cukup bagus dan mudah untuk digunakan. Hal ini terlihat dari hasil yang berasal dari penilaian aplikasi secara keseluruhan mencapai nilai 4 dari nilai yang terbaik yaitu 5.

Diharapkan aplikasi ini dapat dikembangkan lebih lanjut dengan mencakup hal sebagai berikut:

- Data *encyclopedia* dan *article* dapat diperbanyak dan diperbaharui.
- *Member* dapat mengupload foto untuk melengkapi *data profile member*.

#### Daftar Pustaka:

- [1] Durkin, J. (1994). *Expert Systems: Design and Development*. MacMillan
- [2] Giarratano, J.C. & Riley, G.D. (2004), *Expert Systems: Principles and Programming*. 4th ed. Course Technology.
- [3] McPhee, S.J., Papadakis, M., Rabow, M.W. (2010). *Current Medical Diagnosis & Treatment 2011*. United States: The McGraw-Hill, Inc
- [4] Rolston, D.W. (1989). *Principles of artificial intelligence and expert systems development*. New York. McGraw-Hill

