

APLIKASI SISTEM PAKAR UNTUK DIAGNOSIS AWAL GANGGUAN KESEHATAN SECARA MANDIRI MENGGUNAKAN VARIABLE-CENTERED INTELLIGENT RULE SYSTEM

Irfan Subakti, Rahmat Hidayatullah

Jurusan Teknik Informatika, Fakultas Teknologi Informasi,
Institut Teknologi Sepuluh Nopember (ITS), Surabaya, 60111, Indonesia
Email: yifana@gmail.com; rahmat_dayat@telkom.net

ABSTRAK

Masyarakat semakin memahami pentingnya makna kesehatan, utamanya jika gangguan kesehatan tersebut terjadi pada dirinya sendiri atau keluarganya. Ketika hal ini terjadi, diinginkan diketahui penyebab gejala yang dialami dan seberapa cepat dapat disembuhkan. Lebih jauh masyarakat juga semakin mandiri, banyak yang telah sadar bahwa pelbagai sakit ringan biasa, seperti pilek atau diare, tidaklah perlu memanggil ahli kesehatan atau menjalani pengobatan yang rumit. Gangguan semacam ini akan sembuh dengan sendirinya – bahkan walau tanpa mendapatkan pengobatan sekalipun. Masalah bagi mereka yang tidak terlatih secara medis adalah bahwa keluhan semacam sakit kepala atau batuk bisa jadi merupakan petunjuk awal adanya sakit yang serius. Dari sini diterbitkan buku “Dokter di Rumah Anda” yang menjelaskan diagram atau langkah-langkah yang dapat dilakukan di rumah sehingga dapat diambil langkah lebih lanjut pada gangguan yang dihadapi.

Dalam penelitian ini diimplementasikan sistem pakar untuk diagnosis awal gangguan kesehatan menggunakan Variable-Centered Intelligent Rule System (VCIRS) dari buku diatas.

Dari sistem yang telah diimplementasikan diperoleh kesimpulan bahwa secara umum sistem ini dapat berjalan dengan baik, namun demikian masih ada beberapa kekurangan karena keberhasilan diagnosis awal gangguan kesehatan sangat bergantung dengan keahlian pengguna sistem dalam membangun knowledge base (basis pengetahuan).

Kata Kunci: VCIRS, gangguan kesehatan, diagnosis kesehatan

1. PENDAHULUAN

Kesehatan merupakan hal yang begitu penting bagi manusia. Ironisnya banyak sekali penyakit-penyakit yang pada akhirnya terlambat didiagnosis sehingga mencapai tahap kronis yang membuatnya sulit untuk ditangani. Padahal setiap penyakit sebelum mencapai tahap kronis/stadium tinggi umumnya menunjukkan gejala-gejala penyakit yang telah diderita oleh pasien tetapi masih dalam tahap yang ringan misalnya sakit kepala, batuk atau nyeri pada sendi. Sayangnya karena ketidaktahuannya, mereka tidak memperhatikan hal tersebut. Mereka mengabaikan hal tersebut karena selain tidak terlalu mengganggu aktivitasnya mereka juga menganggap bahwa gangguan tersebut akan sembuh dengan sendirinya. Sampai suatu saat timbul gejala yang amat mengganggu dirinya secara fisik misalnya sakit kepala yang luar biasa atau perih pada bagian tubuh tertentu yang teramat sangat yang dapat mengganggu aktivitasnya. Kalau sampai tahap ini baru membuat mereka memeriksakan keluhannya tersebut ke dokter. Tetapi sayangnya hal tersebut seringkali sudah terlambat karena tingkat penyakitnya sudah tinggi dan sulit untuk diobati.

Dalam buku “Dokter di Rumah Anda” [Smith05] terdapat diagram-diagram/petunjuk-petunjuk secara terurut menurut penyakit yang dihadapi. Petunjuk tersebut memberikan langkah demi langkah

diagnosis gejala yang dapat dilakukan di rumah, karena diagnosis yang dilakukan masih sederhana dan tidak diperlukan bantuan dokter atau laboratorium untuk melakukannya.

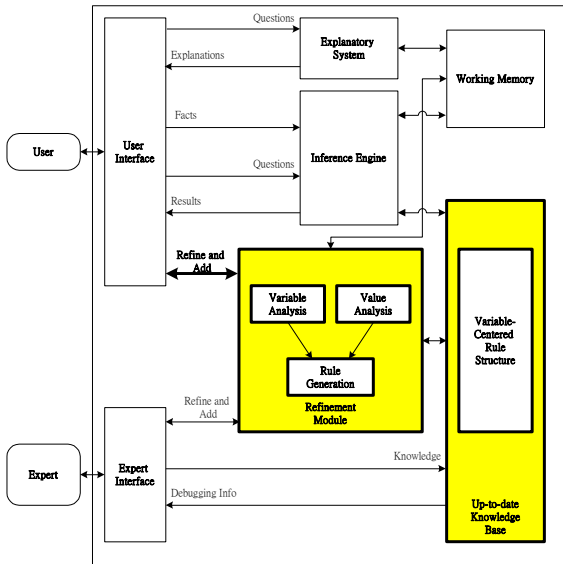
Langkah-langkah yang ada pada buku tersebut dapat disusun ke dalam suatu sistem pakar. Metode yang digunakan untuk memasukkan case-case (kasus-kasus) yang ada ke dalam knowledge base (KB - basis pengetahuan) tersebut dapat digunakan oleh VCIRS. Ini, karena VCIRS memiliki kelebihan dalam hal knowledge building (pembangunan pengetahuan) sekaligus mempunyai kemampuan yang baik dalam proses inferensianya.

2. VARIABLE-CENTERED INTELLIGENT RULE SYSTEM [Sub05b]

Metode dari VCIRS dapat digambarkan pada gambar 1, yang merupakan teknik persilangan dari Rule-Based System (RBS) dan Ripple Down Rules (RDR). VCIRS mempunyai struktur yang mengorganisasikan Rule Base (RB) sehingga pembangunan pengetahuan yang mudah, inferensia pengetahuan yang berdayaguna dan peningkatan evolusional sistem dapat didapatkan pada waktu yang sama. Sedangkan arsitektur dari VCIRS itu sendiri dapat dilihat pada gambar 2.



Gambar 1. Metode VCIRS



Gambar 2. Arsitektur VCIRS

Pertama, pembangunan pengetahuan dimudahkan dengan langkah-langkah yang sederhana dalam proses pembangunan pengetahuannya. Pengguna tidak perlu mempertimbangkan mengenai struktur KB dan dapat *update* KB secara langsung. VCIRS mengizinkan pengguna untuk memperbaiki atau menambahkan *node/rule* ke dalam KB yang telah ada. Seperti dalam RDR, perbaikan rule adalah pembuatan sebuah rule pengecualian (*exception rule*) untuk membuat benar pengklasifikasian yang salah, dimana penambahan itu mengacu pada penambahan rule baru pada level puncak *tree* KB. Sistem memandu pengguna selama proses pembangunan pengetahuan.

Inferensia pengetahuan dipertajam oleh pengetahuan (yaitu, hasil dari analisis variabel dan nilai) dari urutan derajat kepentingan (*important degree*) dan tingkat penggunaan (*usage rate*) dari *case/kasus* pada KB. Mekanisme inferensia RBS dibawa kembali dalam VCIRS, sehingga pengguna mendapatkan lebih banyak jawaban dan penjelasan dari inferensia.

Kinerja sistem ditingkatkan oleh struktur RB yang mendukung analisis variabel (*Variable Analysis*) dan analisis nilai (*Value Analysis*) untuk pembangkitan *rule*. Pembangkitan *rule* meningkatkan hasil dari inferensia dalam hal cakupan pengetahuan. Lebih jauh, analisis nilai juga memandu pengguna selama pembangunan dan inferensia pengetahuan. Bersamaan dengan pembangkitan *rule*, kemampuan ini dapat

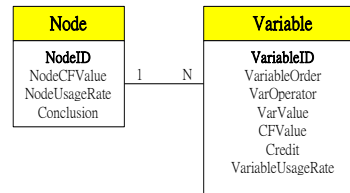
meningkatkan kinerja sistem dalam hal inferensia pengetahuan.

Variable-Centered Rule Structure digunakan untuk merepresentasikan KB dan mendukung *Refinement Module* untuk mengelola KB yang *up-to-date*. Selain itu juga mencatat kasus-kasus dan kejadiannya. Elemen fundamental dari *Variable-Centered Rule Structure* adalah variabel, yang ditempatkan/dipasang oleh pengguna. VCIRS mengelola secara cermat variabel ini mengenai nilai, struktur dan kejadiannya. Rangkaian dari variabel membentuk *node*, sedangkan rangkaian dari *node* menyusun *rule*. Maka *Variable-Centered Rule Structure* mengandung struktur *rule* dan struktur *node* yang berpusat pada variabel-variabel.

2.1 Variable-Centered Rule Structure

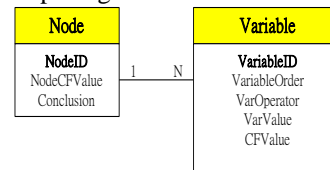
Pada bagian ini akan dijelaskan apa saja yang terlibat dalam *Variable-Centered Rule Structure*.

2.1.1 Node Structure



Gambar 3. Node Structure

Gambar 3 menjelaskan graf konseptual dari struktur *node*. Diberikan *case* baru yang disediakan oleh pengguna, dari memori kerja VCIRS memasukkannya ke dalam struktur *node* dan lalu menggunakannya struktur *rule* sebagai bagian dari KB yang *up-to-date*. Struktur *node* juga menyimpan kejadian dari variabel-variabel dan *node-node*, untuk *usage assignment* (penugasan tingkat kegunaan). Setiap *case* terdiri dari kumpulan *field* data seperti digambarkan pada gambar 4.



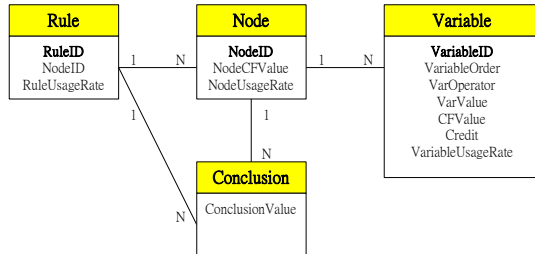
Gambar 4. Case fields

Setiap kali pengguna memasukkan *case*-nya, struktur *node* mengelola nilai dan posisinya. Informasi ini akan digunakan dalam *usage assignment*.

Dalam VCIRS pengguna dibolehkan untuk memperbaiki atau menambahkan *node* kedalam KB yang telah ada. Perbaikan *rule* adalah pembuatan dari *rule* pengecualian untuk membenarkan klasifikasi yang salah, sedangkan penambahan *rule* mengacu pada penambahan *rule* baru pada level puncak dari *tree* di KB. Sistem memandu pengguna selama proses pembangunan pengetahuan ini.

2.1.2 Rule Structure

Gambar 5 menggambarkan graf konseptual dari *Rule Structure*. Seperti telah dinyatakan sebelumnya, *case* yang dimasukkan oleh pengguna pertama kali disimpan dalam *Node Structure*; lalu ia digunakan dalam *Rule Structure*.



Gambar 5. Graf konseptual Rule Structure

2.2 Perbaikan Pengetahuan

Ada 3 tugas (*task*) dalam *Refinement Module*: analisis variabel, analisis nilai dan pembangkitan *rule*.

2.2.1 Variable Analysis

Di dalam *Variable-Centered Rule Structure* sistem mengetahui node mana yang di-*shared* (sama-sama digunakan) oleh pelbagai *rule*, dan variabel mana yang di-*shared* oleh *node*. Semakin banyak *rule* yang memakai suatu *node*, maka *node* tersebut akan semakin penting. Pertimbangan yang sama terjadi pada variabel yang di-*shared* didalam *node*.

Fakta ini menunjukkan seberapa penting suatu *node*/variabel, sebagai titik awal untuk membangkitkan *rule* baru, bersama dengan analisis nilai.

2.2.2 Value Analysis

Proses analisis nilai, yang disebut dengan *usage assignment* (pemberian nilai kegunaan), adalah untuk menentukan derajat kegunaan dari *rule/node/variabel* dalam *KB*. *Usage assignment* menggunakan informasi yang disimpan dalam *Variable-Centered Rule Structure*.

Terdapat tiga *usage degree*. Pertama, *Variable Usage Rate (VUR)* digunakan untuk mengukur tingkat kegunaan dari suatu variabel di dalam *node* yang sedang dan telah digunakan. Kedua, *Node Usage Rate (NUR)* untuk mengukur kegunaan suatu *node* pada pengeksekusian (*firing*) suatu *node*. Yang terakhir adalah *Rule Usage Rate (RUR)* yang mengukur kegunaan suatu *rule* pada pengeksekusian (*firing*) suatu *rule*. Makin besar indikator kegunaan, maka makin bergunalah nilai tersebut dan begitu pula sebaliknya.

Persamaan (1) menghitung *VUR* untuk variabel ke-*i*, (2) menghasilkan *NUR* untuk *node* ke-*j*, sedangkan (3) mendefinisikan *RUR* untuk *rule* ke-*k*.

$$VUR_i = Credit_i \times Weight_i \quad (1)$$

$$NUR_j = \frac{\sum_{i=1}^N VUR_{ij}}{N}, \quad VUR_{ij} \text{ untuk variabel ke-} i \text{ dalam node } j \quad (2)$$

$$RUR_k = \frac{\sum_{j=1}^N NUR_{jk}}{N}, \quad NUR_{jk} \text{ untuk node ke-} j \text{ dalam rule } k \quad (3)$$

Dimana:

- $Credit_i$ = kejadian dari variabel *i* dalam *Node Structure* (4)

Credit didapatkan dari *Node Structure*. Nilainya akan meningkat saat pengguna membuat *node* yang menyetujui nilai dari *case* lama.

- $Weight_i = NS_i \times CD_i$ (5)

Weight menghitung bobot (*weight*) dari variabel ke *node* yang memilikinya. Ada 2 faktor yang berkontribusi ke bobot dari sebuah variabel. Pertama adalah jumlah *node* yang berbagi (*sharing*) sebuah variabel, dan kedua adalah *CD (Closeness Degree)*, yaitu derajat kedekatan sebuah variabel pada sebuah *node*.

- NS_i = jumlah *node* yang berbagi (*sharing*) variabel *i* (6)

- $CD_i = \frac{VO_i}{TV}$ (7)

CD adalah singkatan dari *Closeness Degree*, yaitu derajat kedekatan sebuah variabel pada sebuah *node*. CD_i dalam *node* *j*, menghitung derajat kedekatan dari variabel *i* dalam *node* *j*. Makin dekat sebuah variabel pada konklusi yang dipunyai suatu *node*, makin baiklah variabel tersebut. Sehingga, *CD* mendasarkan dirinya pada urutan dari variabel dalam suatu *node* (disini, *node* adalah rangkaian dari variabel-variabel). *CD* dihitung dengan urutan variabel *VO*, dibagi dengan total variabel *TV*, yang dimiliki oleh sebuah *node*.

- VO_i = urutan dari variabel *i* dalam suatu *node* (8)

- TV = total variabel yang dimiliki suatu *node* (9)

2.3 Pembangunan Pengetahuan

Algoritma dari pembangunan pengetahuan adalah:

- I. Jika *node* yang layak ditemukan, sistem akan memberikan pengguna satu dari beberapa pilihan berikut ini.

1. Jika ia tidak setuju dengan *node* saat itu, ia dapat membuat *node* baru pada level puncak *KB*. Secara otomatis, sistem akan membuatkan *rule* baru untuk *node* baru ini. Variabel-variabel yang tak tersetujui dari *case* yang dimasukkan pengguna akan disimpan dibawah posisi *node* baru.

2. Jika ia setuju dengan satu atau lebih variabel dalam *node* saat itu, tetapi tidak menyetujui

dengan yang lainnya; ia dapat membuat node pengecualian yang mengandung variabel-variabel yang tidak disetujuinya tadi. Variabel yang tersetujui dari case yang dimasukkan pengguna akan disimpan dalam kolom *Credit* dari *Node Structure* dibawah posisi dari node lama (*parent node*), sedangkan variabel yang tak tersetujui akan disimpan dibawah posisi node baru.

3. Ia dapat memilih untuk melanjutkan penelusuran KB tanpa merubah apa pun. Disini *Variable-Centered Rule Structure* tak mendapat apa pun dari pengguna. Kerja sistem jika seperti ini serupa dengan sebuah proses inferensia dan ia membolehkan pengguna untuk menguji *KB*, yaitu mekanisme verifikasi sambil jalan (*verification-on-the-fly*).

II Jika tak ada satu pun node yang ditemukan, sistem akan meminta pengguna untuk membuat node baru pada level puncak dari KB dan case yang dimasukkan pengguna akan disimpan di bawah posisi node baru tadi.

2.5 Diagnosa Awal Gangguan Kesehatan secara Mandiri

Proses diagnosis ini diambil dari buku “Dokter di Rumah Anda” [Smith05] yang berisi 150 diagram yang dapat diikuti oleh orang belum pernah terlatih sekalipun untuk mengetahui seberapa bahayakah gangguan kesehatan yang dialami dan apakah tindakan selanjutnya yang harus dilakukan.

Semua diagram yang terdapat pada buku tersebut pada dasarnya menggunakan diagram “*if-then*” untuk mengkorelasikan gejala yang timbul pada penderita dengan tindakan medis yang harus dilakukan untuk mengurangi resiko fatal pada gangguan kesehatan yang dialami. Hal ini juga akan membantu pengguna mengenali kemungkinan penyebab suatu gejala, menuntut pengguna bila harus mencari bantuan medis serta memungkinkan memberi saran cara mengatasi sendiri gejala atau penyebab tersebut.

Diagram-diagram tersebut terdiri dari diagram untuk **anak-anak** dari semua tingkat usia, untuk **semua orang dewasa** (orang dewasa umum tidak membedakan usia dan jenis kelamin) serta diagram khusus untuk **pria** dan **wanita**. Semua diagram dibuku ini disusun atas studi secara intensif dan penelusuran kembali *database* kesehatan untuk mendapatkan analisis serta pembahasan yang mendalam dan komprehensif mengenai konsep diagnosis awal gangguan kesehatan secara mandiri ini oleh para dokter dan diperiksa oleh para spesialis medik.

Didalam ilmu kedokteran, penanganan setiap gangguan kesehatan terhadap berbagai kelompok penderita dibedakan. Misalnya gangguan kesehatan batuk, walaupun penyakit yang diderita sama yaitu

batuk tetapi jika penderitanya berbeda misalnya balita dan orang dewasa, tentu penanganannya berbeda. Penderita balita tentu akan mendapatkan penanganan lebih serius. Sedangkan penderita dewasa, dilihat dulu batuknya. Jika batuknya dikarenakan flu atau batuk ringan tentunya tidak perlu dibawa ke dokter, cukup istirahat dan minum obat batuk atau obat flu biasa, biasanya akan sembuh. Oleh karena alasan diatas, pengelompokan gangguan kesehatan terhadap kelompok pasien dilakukan didalam sistem ini.

Pengelompokan di atas dilakukan agar nantinya pada saat dilakukan inferensia, sistem tidak melakukan kesalahan dalam memberikan pertanyaan yang sesuai dengan keluhan dan kelompok usia pasien. Walaupun ada beberapa penyakit yang muncul di lebih dari satu kelompok usia, namun pertanyaan yang diberikan pun berbeda karena penanganannya yang berbeda pula.

Jika diperhatikan, diagram-diagram yang ada pada [Smith05] memiliki kemiripan dengan RBS, karena diagramnya menggunakan “*if-then*”. Diagram terdiri dari pertanyaan-pertanyaan yang jika suatu pertanyaan telah dijawab, maka akan menuju ke pertanyaan selanjutnya sesuai dengan jawaban yang diberikan oleh pengguna. Berdasarkan kondisi tersebut, diagram yang ada dapat di transfer kedalam komputer dalam bentuk sistem pakar berbasis aturan/rule.

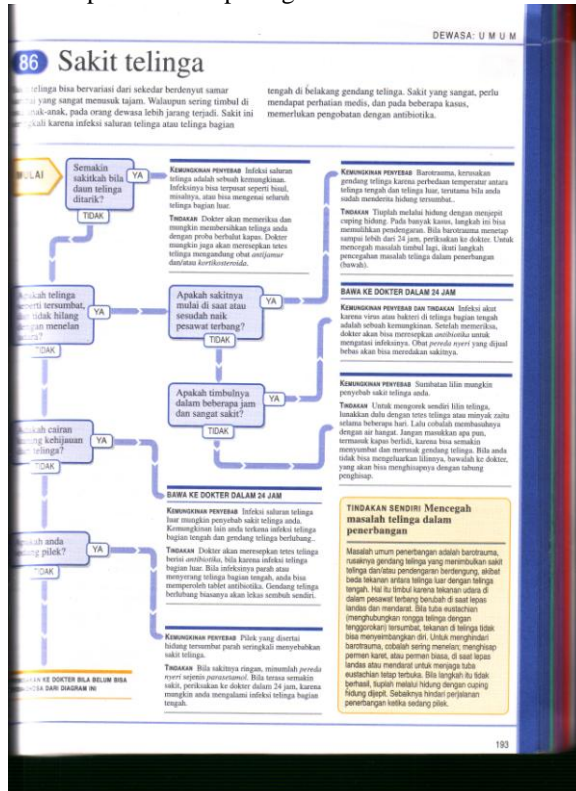
Pembangunan KB dan proses inferensia pada sistem ini menggunakan metode VCIRS [Sub05a, Sub05b, Sub06b]. Hal ini dilakukan karena VCIRS memiliki kelebihan untuk pembangunan KB dan proses inferensia, menyempurnakan metode RBS dan RDR. Proses implementasi dari diagram pada [Smith05] menjadi KB akan dijelaskan pada gambar 7 dibawah. Sebagai tambahan, implementasi ini juga didasarkan pada revisi yang dilakukan pada VCIRS dan perspektif yang ada dalam rekonstruksi *case* [Sub06b]. Sehingga implementasinya mengikuti algoritma berikut ini:

1. Setiap memasukkan case baru, bagian diagram yang berposisi sebagai **Nama Keluhan** akan dimasukkan sebagai variabel dalam *case* baru. Variabel ini nantinya berfungsi sebagai *identifier* pada saat proses inferensia. Sehingga pertanyaan yang diajukan kepada user pada saat proses inferensia (konsultasi) adalah pertanyaan seputar keluhan yang dipilih oleh user.
2. Yang berposisi sebagai **pertanyaan** akan dimasukkan kedalam variabel. Variabel ini nantinya pada saat inferensia (konsultasi) akan berfungsi sebagai pertanyaan yang diajukan kepada user yang melakukan konsultasi.
3. Bagian **kemungkinan penyebab** dan **tindakan** akan masuk ke dalam bagian konklusi. Bagian ini pada proses inferensia (konsultasi) akan

berfungsi sebagai saran atau keterangan yang diberikan kepada user

3. EVALUASI HASIL PROSES INFERENSI

Dengan menggunakan salah satu diagram yang ada pada buku [Smith05] hal 193 untuk kelompok pasien **Dewasa: Umum** dengan keluhan **Sakit Telinga**, gambar diagram secara lengkap diberikan pada gambar 8. Metode inferensia yang digunakan adalah menggunakan metode *Backward Chaining*. Dari hasil inferensia yang dilakukan didapatkan hasil seperti terlihat pada gambar 9.



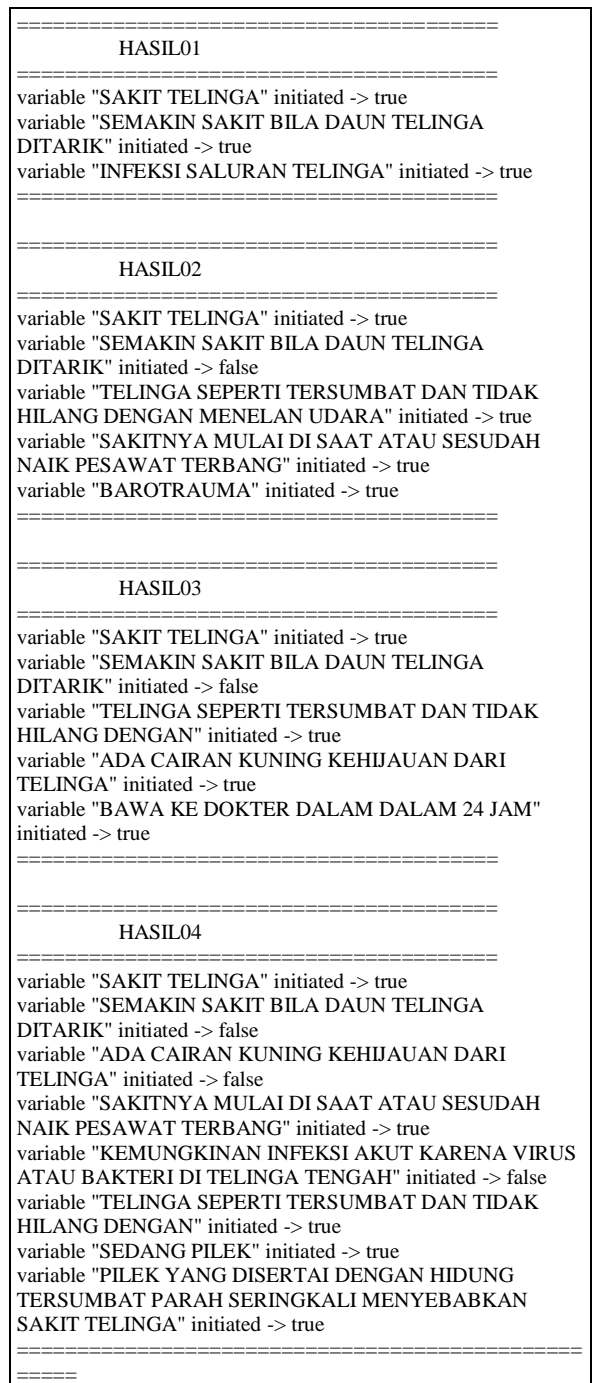
Gambar 6. Diagram dengan keluhan Sakit Telinga

Setelah membandingkan hasil beberapa proses inferensia untuk beberapa kasus pada keluhan **Sakit Telinga** diatas dengan diagram yang ada pada buku, ternyata hasil yang didapatkan/saran yang diberikan sesuai dengan diagram pada [Smith05].

Tetapi ada beberapa kasus pertanyaan yang diberikan kepada *user* tidak berurutan, hal ini dapat disebabkan kesalahan pada saat memasukkan *case* baru. Karena pada saat inferensia, variabel yang ditanyakan adalah variabel yang memiliki urutan lebih rendah/lebih duluan. Jika kita memasukkan variabel tersebut terbalik, maka hal itu akan membuat eksekusi variabel juga terbalik, sehingga membuat pertanyaan yang diberikan tidak mengikuti diagram. Jika kita memberikan jawaban yang benar sesuai dengan kondisi kesehatan/keluhan yang dialami maka didapatkan hasil yang benar.

4. KESIMPULAN

Dari sistem yang telah diimplementasikan diperoleh kesimpulan bahwa secara umum sistem ini dapat berjalan dengan baik, namun demikian masih ada beberapa kekurangan karena keberhasilan diagnosis awal gangguan kesehatan sangat bergantung dengan keahlian pengguna sistem dalam membangun KB, dalam hal ini dokter. Jika peletakan node pada saat melakukan *knowledge building* (pembangunan pengetahuan) salah, maka akan dapat merusak KB yang ada.



Gambar 7. Hasil proses inferensia

5. DAFTAR PUSTAKA

- [Sub05a] Irfan Subakti, "A variable-centered intelligent rule system," *Master Thesis*, Department of Computer Science and Information Engineering, College of Electrical and Computer Engineering, National Taiwan University of Science and Technology, Taiwan, 2005.
- [Sub05b] Irfan Subakti, "A variable-centered intelligent rule system", *Proc. of the 1st Annual International Conference: Information and Communication Technology Seminar 2005 (ICTS2005)*, Vol. 1, No. 1, 11 August 2005, Institute Technology of Sepuluh Nopember (ITS), Surabaya, Indonesia, 2005, pp. 167-174.
- [Sub06a] Irfan Subakti, *Intelegensia Mesin*, Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya, 2006.
- [Sub06b] Irfan Subakti, "Some revisions in VCIRS and cases reconstructing perspectives," *Proc. of the 2nd Annual International Conference: Information and Communication Technology Seminar 2006 (ICTS2006)*, Vol. 1, No. 1, 29 August 2006, Institute Technology of Sepuluh Nopember (ITS), Surabaya, Indonesia, 2006, pp. 233-238.
- [Smith05] Tony Smith and Sue Davidson, *Dokter di Rumah Anda*, Dian Rakyat, Jakarta, Indonesia, 2005.