

Penerapan Logika dalam Program Sederhana untuk Mendiagnosa Beberapa Penyakit

Juan Felix Parsaoran Tarigan, 13516143
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516143@std.stei.itb.ac.id

Abstract—Kesehatan merupakan hal yang penting bagi kehidupan manusia. Untuk mengobati suatu penyakit yang sudah terlanjur ada di tubuh manusia, biasanya seseorang akan pergi ke rumah sakit dan memeriksa ke dokter. Dokter akan menanyakan apa yang dirasakan, dan mungkin melakukan pemeriksaan lebih lanjut. Seringkali seseorang merasakan gejala yang kurang baik, tetapi merasa tidak perlu ke dokter. Sistem pemeriksaan gejala saat ini dapat diganti dengan sistem perangkat lunak yang dapat melakukan penalaran logika berdasarkan gejala yang dialami seseorang layaknya dokter. Makalah ini membahas tentang aplikasi logika dalam mendiagnosa beberapa penyakit menggunakan perangkat lunak(*software*) yang diprogram dengan menggunakan bahasa prolog.

Keywords—diagnosa, logika, prolog.

I. PENDAHULUAN

Jika seseorang merasakan hal yang tidak baik pada tubuhnya, maka perlu dilakukan pengecekan ke dokter untuk mengetahui apa yang salah dengan tubuhnya. Terkadang juga, seseorang merasa apa yang dirasakannya itu bukanlah hal yang perlu dikhawatirkan sehingga tidak perlu untuk ke dokter, tetapi tidak tahu bagaimana cara menyembuhkan hal yang tidak baik tersebut.

Permasalahan seperti itu membutuhkan sistem diagnosa yang mangkus dan sangkil untuk pengecekan awal suatu gejala. Dengan diketahuinya penyakit tersebut, maka diketahui pula cara untuk menanganinya. Dalam hal ini, sistem tersebut akan memanfaatkan perangkat lunak(*software*) yang diprogram dengan menggunakan bahasa prolog. Dalam makalah ini, saya menggunakan GNU Prolog untuk membuat program pendiagnosa beberapa penyakit.

Namun demikian, sistem ini tidak serta merta akan menggantikan kinerja seorang dokter. Sistem ini hanya untuk pengecekan awal suatu gejala dan juga untuk kasus-kasus kesehatan yang umum terjadi dan hanya perlu dengan diagnosa sederhana. Tentunya, kemampuan dokter sangat dibutuhkan untuk menindaklanjuti diagnosa program yang sudah tidak bisa ditangani seorang diri.

II. LANDASAN TEORI

A. Logika



Gambar 1 : Aristoteles
(sumber : sekelumitpandang.com)

Logika adalah ilmu yang membantu kita dalam berpikir dan menalar. Menalar artinya mencapai kesimpulan dari berbagai pernyataan. Logika membantu orang untuk berpikir kritis, rasional, dan koheren. Logika dimulai sejak Thales(624 SM – 548 SM), dan dilanjutkan oleh Aritoteles. Mereka membahas tentang air sebagai jiwa alam semesta dari beberapa fakta yang ada saat itu.

B. Proposisi

Logika didasarkan pada hubungan antara kalimat pernyataan (*statements*).Kalimat yang bernilai benar atau salah disebut sebagai proposisi. Proposisi yang akan ditinjau dalam penalaran logika.

Contoh proposisi:

1. Matahari terbit di barat.
2. $2 + 2 = 4$.

Contoh bukan proposisi:

1. $y = 5$.
2. $2 + x = 5$ dan $x + 5 = 6$.

Proposisi biasanya dilambangkan dengan huruf kecil p, q, r, s, \dots

Contoh:

- p : 25 adalah bilangan ganjil.
 q : ITB ada di Samosir.
 r : $2 + 5 = 7$.

Proposisi dapat dinyatakan dalam 4 bentuk:

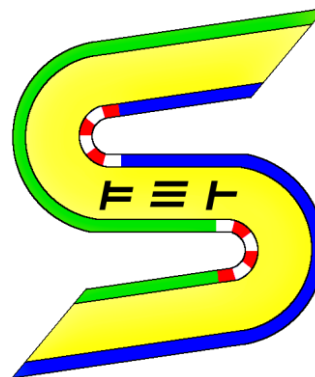
1. Proposisi atomik
Proposisi tunggal.
Contoh : saya adalah mahasiswa.
2. Proposisi majemuk
Ada 4 macam proposisi majemuk:
 1. Konjungsi: p dan q.
 2. Disjungsi: p atau q.
 3. Ingkaran: tidak p.
 4. Disjungsi Eksklusif: p atau q, tetapi tidak keduanya.
3. Implikasi
Proposisi bersyarat.
Contoh : jika hari ini hujan, saya tidak masuk kuliah.
$$p \rightarrow q$$
4. Bi-implikasi
Proposisi bersyarat dengan dua arah
Contoh : saya akan pergi kuliah jika dan hanya jika hari ini tidak hujan
$$p \Leftrightarrow q$$

Hukum-hukum proposisi:

1. Hukum identitas
 - $p \vee F \Leftrightarrow p$
 - $p \wedge T \Leftrightarrow p$
2. Hukum null
 - $p \wedge F \Leftrightarrow F$
 - $p \vee T \Leftrightarrow T$
3. Hukum negasi
 - $p \vee \sim p \Leftrightarrow T$
 - $p \wedge \sim p \Leftrightarrow F$
4. Hukum idempoten
 - $p \vee p \Leftrightarrow p$
 - $p \wedge p \Leftrightarrow p$
5. Hukum involusi
 - $\sim(\sim p) \Leftrightarrow p$
6. Hukum penyerapan
 - $p \vee (p \wedge q) \Leftrightarrow p$
 - $p \wedge (p \vee q) \Leftrightarrow p$
7. Hukum komutatif
 - $p \vee q \Leftrightarrow q \vee p$
 - $p \wedge q \Leftrightarrow q \wedge p$
8. Hukum asosiatif
 - $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$
 - $p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$
9. Hukum distributif
 - $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
 - $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
10. Hukum De Morgan
 - $\sim(p \vee q) \Leftrightarrow \sim p \wedge \sim q$

$$- \sim(p \wedge q) \Leftrightarrow \sim p \vee \sim q$$

C. Prolog (Programming in logic)



Gambar 2 : Logo GNU Prolog
(Sumber : gprolog.org)

Prolog pertama kali dikembangkan oleh Alain Colmetrouer dan P.Rousel di Universitas Marseilles Prancis pada tahun 1972. Prolog menjadi populer untuk aplikasi AI. Prolog merupakan bahasa pemrograman deklaratif yang berbeda dengan bahasa procedural pada umumnya. Dalam bahasa prolog hanya terdapat fakta-fakta dan aturan yang mengatur relasi antar fakta dan sesuatu yang dicari dalam program disebut *query*.

Dalam prolog, Pemrogram menentukan tujuan (*goal*) dan komputer akan menentukan bagaimana cara mencapai tujuan tersebut serta mencari jawabannya. Caranya dengan menggunakan "*Formal Reasoning*" yaitu membuktikan cocok tidaknya tujuan dengan data-data yang telah ada dan relasinya. Prolog memecahkan masalah seperti yang dilakukan oleh pikiran manusia.

Fakta-fakta dalam pemrograman prolog merupakan proposisi. Sedangkan *rule* merupakan aturan penalarannya. Rule dalam program dibuat sedemikian rupa, sehingga fakta-fakta dalam program dapat terelasi dengan baik. Tak jarang rule-rule dalam pemrograman prolog menggunakan rekursif.

Contoh fakta :

```

/* Nomor 1 */
/*Deklarasi Fakta*/
/* Bagian <1> */
pria(kevin).
pria(richard).
pria(rahman).
pria(turfa).
pria(taufan).
pria(faiz).
pria(rizki).
pria(adrian).
pria(cloud).
/* Bagian <2> */
wanita(rachel).
wanita(irene).
wanita(tasya).
wanita(holy).
wanita(emilia).
/* Bagian <3> */
menikah(kevin,rachel).
menikah(rachel,kevin).
menikah(richard,irene).
menikah(irene,richard).
menikah(rahman,tasya).
menikah(tasya,rahman).
menikah(taufan,holy).
menikah(holy,taufan).
menikah(faiz,emilia).
menikah(emilia,faiz).

```

Gambar 3: fakta-fakta tentang keluarga.

Contoh rule :

```

/* Nomor 2 */
/* Deklarasi Rules */
/* Bagian <1> */
ibu(X,Y):-anak(Y,X),wanita(X).
/* Bagian <2> */
keponakan(X,Y):-anak(X,Z),kakak(Y,Z).
keponakan(X,Y):-anak(X,Z),kakak(Z,Y).
/* Bagian <3> */
kakek(X,Y):-anak(Y,Z),anak(Z,X),pria(X).
/* Bagian <4> */
mertua(X,Y):-anak(Z,X),menikah(Y,Z).
/* Bagian <5> */
suami(X,Y):-menikah(X,Y),pria(X).
/* Bagian <6> */
anaklelaki(X,Y):-anak(X,Y),pria(X).
/* Bagian <7> */
cucu(X,Y):-anak(Z,Y),anak(X,Z).
/* Bagian <8> */
bibi(X,Y):-keponakan(Y,X),wanita(X).
/* Bagian <9> */
adikipar(X,Y):-kakak(Z,X),menikah(Y,Z).
adikipar(X,Y):-menikah(X,Z),kakak(Y,Z).
/* Bagian <10> */
anaktunggal(X,Y):-anak(X,Y),\+kakak(Z,X),\+kakak(X,Z).

```

Gambar 4: Rule tentang keluarga

Contoh program prolog sederhana:

```

1 maxlist(X):-maxlistlok(X,Y,Z),write(Z),!.
2 maxlistlok([],0,0):-!.
3 maxlistlok([A|B],Y,Z):-maxlistlok(B,C,D),A>C,Y is A,search([A|B],A,X),Z is X.
4 maxlistlok([A|B],Y,Z):-maxlistlok(B,C,D),A<C,Y is C,search([A|B],C,X),Z is X.
5 search([],Z,0):-!.
6 search([A|B],Z,X):-A =\= Z,search(B,Z,Y),X is Y+1.
7 search([A|B],Z,X):-A = Z,X is 1.

```

Gambar 5 : source code program pencarian indeks dari bilangan terbesar dalam list

```

| ?- maxlist([6,4,12,8,5,7]).
3
yes

```

Gambar 6: query untuk menghasilkan indeks bilangan terbesar dalam list (max = 12).

```

| ?- kriteria_pacar(cantik,kaya,baik,penurut).
no
| ?- kriteria_pacar(cantik,baik,pintar,perhatian).
yes

```

Gambar 7: query untuk menemukan kriteria pacar seseorang.

D. Sistem Diagnosa Penyakit berbasis Perangkat Lunak

Kemajuan teknologi saat ini mendorong setiap orang atau instansi dalam memperbaharui layanan mereka kepada pengguna jasa. Tak terkecuali dalam pelayanan kesehatan. Instansi-instansi kesehatan berlomba-lomba memberikan pelayanan yang mumpuni bagi setiap orang yang memakan jasa mereka. Hal yang perlu dipertimbangkan adalah pelayanan kesehatan saat ini masih terlalu banyak menyita materi pasien, baik waktu dan biaya. Sistem seperti ini dapat diganti dengan sistem pelayanan kesehatan dengan perangkat lunak asalkan perangkat lunak tersebut diprogram sesuai dengan cara kerja dan analisis seorang dokter dalam mendiagnosa suatu penyakit. Sistem ini tentunya dapat lebih mudah dijangkau oleh setiap orang.

E. Gejala beberapa penyakit endemik di Indonesia

Berikut ini adalah beberapa penyakit beserta gejala-gejalanya yang sering terjadi di Indonesia:

1. Demam Berdarah
 - Suhu tubuh tinggi.
 - Kehilangan nafsu makan.
 - Mual dan muntah.
 - Nyeri otot.
 - Sakit kepala
2. Tifus
 - Suhu tubuh tinggi
 - Kehilangan nafsu makan
 - Lemas
 - Nyeri otot
 - Sakit kepala
3. Anemia
 - Pucat
 - Lemas
 - Sakit kepala
 - Sesak napas
 - Rambut rontok
4. Maag
 - Perut panas
 - Cepat kenyang
 - Muntah
 - Sering sendawa
 - Perut kembung

III. PENERAPAN LOGIKA DALAM MENDIAGNOSA BEBERAPA PENYAKIT MENGGUNAKAN PROGRAM SEDERHANA

A. Pembuatan Fakta-Fakta Gejala Penyakit

Didalam bab II, fakta-fakta dalam prolog merupakan suatu proposisi. Kita dapat membuat fakta-fakta bahwa seseorang menderita suatu penyakit berdasarkan gejala umum yang terdapat pada bab II.

```
/*Deklarasi Fakta*/
/* Bagian <1> */
demam_berdarah(suhu_tubuh_tinggi,kehilangan_nafsu_makan,mual_muntah,nyeri_otot,sakit_kepala).
tifus(suhu_tubuh_tinggi,kehilangan_nafsu_makan,lemas,nyeri_otot,sakit_kepala).
anemia(pucat,lemas,sakit_kepala,sesak_napas,rambut_rontok).
maag(perut_panas,cepat_kenyang,muntah,sering_sendawa,perut_kembung).
```

Gambar 8 : Fakta-fakta dalam program pendiagnosa penyakit.

Fakta-fakta tersebut berarti suatu penyakit memiliki beberapa atribut yang merupakan gejalanya.

B. Pembuatan Rule untuk Mendiagnosa penyakit

Rule dalam pemrograman prolog merupakan aturan penalaran. Kita dapat membuat rule yang bernama gejala. Gejala memiliki 5 atribut sesuai dengan fakta-fakta dari penyakit yang diketahui oleh program.

```
/* Deklarasi Rules */
/* Bagian <1> */
gejala(A,B,C,D,E):- demam_berdarah(A,B,C,D,E), write('\nkamu sepertinya terkena demam berdarah'),!.
gejala(A,B,C,D,E):- tifus(A,B,C,D,E), write('\nkamu sepertinya terkena tifus'),!.
gejala(A,B,C,D,E):- maag(A,B,C,D,E), write('\nkamu sepertinya terkena maag'),!.
gejala(A,B,C,D,E):- anemia(A,B,C,D,E), write('\nkamu sepertinya terkena anemia'),!.
/* Jika gejala tidak dapat dideteksi*/
gejala(A,B,C,D,E):- write('\nprogram tidak menemukan penyakit yang cocok'),!.
```

Gambar 9 : Rule dalam program pendiagnosa penyakit Rule-rule tersebut mempunyai arti, gejala(A,B,C,D,E) benar jika penyakit(A,B,C,D,E) benar dan menuliskan pesan. A,B,C,D,E merupakan gejala yang dirasakan oleh seseorang yang ingin didiagnosa.

Dengan fakta-fakta dan rule-rule tersebut, sistem akan melakukan penalaran sesuai dengan kaidah-kaidah yang berlaku dalam hukum logika.

C. Penggunaan Program untuk Mendiagnosa beberapa penyakit

Masukan dari program berupa gejala apa yang ia alami dengan format “gejala(A,B,C,D,E).” . A,B,C,D,E merupakan gejala yang dialami. Berikut beberapa hasil diagnosa dari program:

```
?- gejala(suhu_tubuh_tinggi,kehilangan_nafsu_makan,mual_muntah,nyeri_otot,sakit_kepala).
kamu sepertinya terkena demam berdarah
```

Gambar 10 : Hasil Diagnosa seseorang dengan penyakit demam berdarah

| ?- gejala(pucat,lemas,sakit_kepala,sesak_napas,rambut_rontok).
kamu sepertinya terkena anemia

Gambar 11: Hasil Diagnosa seseorang dengan penyakit anemia

| ?- gejala(sakit_punggung,mata_merah,sesak_nafas,sakit_perut,
program tidak menemukan penyakit yang cocok

Gambar 12 : Hasil Diagnosa seseorang dengan penyakit yang tidak terdeteksi program.

| ?- gejala(perut_panas,cepat_kenyang,muntah,sering_sendawa,perut_kembung).
kamu sepertinya terkena maag

Gambar 13 : Hasil Diagnosa seseorang dengan penyakit maag

| ?- gejala(suhu_tubuh_tinggi,kehilangan_nafsu_makan,lemas,nyeri_otot,sakit_kep
kamu sepertinya terkena tifus

Gambar 14 : Hasil Diagnosa seseorang dengan penyakit tifus

D. Pengembangan program lebih lanjut

Didalam makalah ini, program yang saya buat hanya dapat mendiagnosa beberapa penyakit saja. Program sederhana tersebut juga membutuhkan gejala yang tepat untuk satu penyakit, dalam arti jika seseorang tidak memiliki satu gejala dari gejala yang merupakan ciri suatu penyakit, maka program tidak menampilkan apa-apa.

Program sederhana ini dapat dikembangkan lebih lanjut. Ada beberapa hal yang dapat dikembangkan:

1. Kemampuan program dalam mendeteksi penyakit

Berdasarkan kelemahan yang saya jabarkan sebelumnya, program ini dapat di-*upgrade* dengan mengkomplekskan algoritmanya. Jika gejala yang dialami oleh seseorang belum memenuhi Dia dikatakan menderita suatu penyakit, maka mungkin bisa ditampilkan kemungkinan seseorang itu menderita suatu penyakit. Misalnya, *query* dari seseorang gejala(suhu_tubuh_tinggi,mual_muntah,nyeri_otot), maka hasil dari *query* tersebut adalah 60% terkena demam berdarah.

Kemampuan program juga dapat ditingkatkan dengan menambahkan pengetahuan tentang gejala penyakit lain, seperti malaria, kolera, influenza, dll.

2. Analisis program di-*update* berdasarkan *query* pengguna yang tidak terdeteksi program

Jika *query* pengguna tidak dapat dideteksi program, maka idealnya, program menyimpan *query* tersebut dan akan dilakukan *maintenance* untuk meng-*upgrade* program, dalam hal ini, mencari tahu penyakit apa yang berhubungan dengan

query tersebut. Dengan demikian, program semakin powerful untuk mengecek penyakit yang dialami oleh seseorang.

IV. KESIMPULAN

Untuk mendiagnosa suatu penyakit yang sudah diketahui gejalanya, selain memeriksakan ke dokter, kita juga dapat mengaplikasikan hukum-hukum logika dalam menentukan jenis penyakit kita. Dalam hal ini, kita dapat memanfaatkan prolog(Programming in logic) untuk pembuatan program sederhananya. Dengan adanya program ini, pemeriksaan suatu penyakit dapat dilakukan lebih awal dan jika diperlukan tindak lanjut, dapat ditindak lanjuti dengan lebih cepat juga. Program ini juga dapat dikembangkan lebih lanjut dalam penelitian yang lebih *expert*.

V. UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa (YME) untuk setiap berkatNya dalam kehidupan saya. Saya juga ingin berterima kasih kepada kedua orang tua saya yang selalu mendukung saya selama saya menjalani pendidikan saya. Saya juga berterima kasih kepada teman-teman yang sudah membantu saya dalam menyelesaikan makalah ini. Tak ketinggalan juga saya berterima kasih kepada dosen Matematika Diskrit saya, Dra. Harlili S., M.Sc. yang sudah mengajar mata kuliah Matematika Diskrit selama satu semester ini.

REFERENSI


- [1] Rinaldi Munir. 2006. *Diktat Kuliah IF2120: Matematika Diskrit*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
- [2] <http://www.alodokter.com/demam-berdarah/gejala>, Diakses tanggal 2 Desember 2017
- [3] <https://mediskus.com/penyakit/ciri-ciri-dan-gejala-penyakit-tipes>, Diakses tanggal 2 Desember 2017
- [4] <http://hariansehat.com/ciri-gejala-awal-penyakit-anemia/>, Diakses tanggal 2 Desember 2017
- [5] <http://www.alodokter.com/sakit-maag/gejala>, Diakses tanggal 2 Desember 2017
- [6] <http://sattoksugianto.blogspot.co.id/2014/03/pengantar-prolog.html>, Diakses tanggal 2 Desember 2017

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2017

Ttd (scan atau foto ttd)

A handwritten signature in black ink on a light-colored background. The signature is stylized and appears to consist of a large, circular initial followed by a few strokes.

Nama dan NIM