

TUGAS BESAR 2

Pembangunan Sistem Berbasis Rule dengan menggunakan CLIPS

Diajukan untuk memenuhi nilai tugas besar 2

Mata Kuliah IF3170 Intelegensi Buatan

Dosen Pengampu: Dr. Nur Ulfa Maulidevi, S.T., M.Sc.



Dibuat Oleh:

Bryan Bernigen / 13520034

Ng Kyle / 13520040

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

1. Deskripsi Masalah

Rule Based System (RBS) merupakan Knowledge Based System berdasarkan pada inferensi menggunakan rules-rules yang ada (rules sebagai representasi basis pengetahuan). Rules didefinisikan sebagai prekondisi dan aksi (*logical implication*). Proses inferensi dilakukan dengan pengaktifan berbagai rules yang terdefinisi pada program sesuai dengan ketersediaan fakta-fakta yang ada dengan resolusi konflik untuk menentukan rules yang akan dikerjakan dari kumpulan rules pada kumpulan rules yang teraktivasi (*Conflict Set*).

Inferensi rule dalam mendapatkan kesimpulan pada RBS dilakukan dengan 2 kemungkinan cara yaitu forward chaining atau backward chaining. Pada CLIPS (C Language Integrated Production System), proses inferensi dilakukan dengan Forward Chaining, yaitu melalui berdasarkan fakta yang ada, terbentuk conflict set yang ada akan dikerjakan sebuah rules untuk menambahkan fakta baru dari hasil inferensi hingga ditemukan kesimpulan.

Diberikan sebuah Decision Tree untuk melakukan prediksi terkait kondisi seseorang yang mengidap penyakit hepatitis B yang bersumberkan dari [From logical inference to decision trees in medical diagnosis](#), akan dibuat representasi RBS dalam bahasa CLIPS.

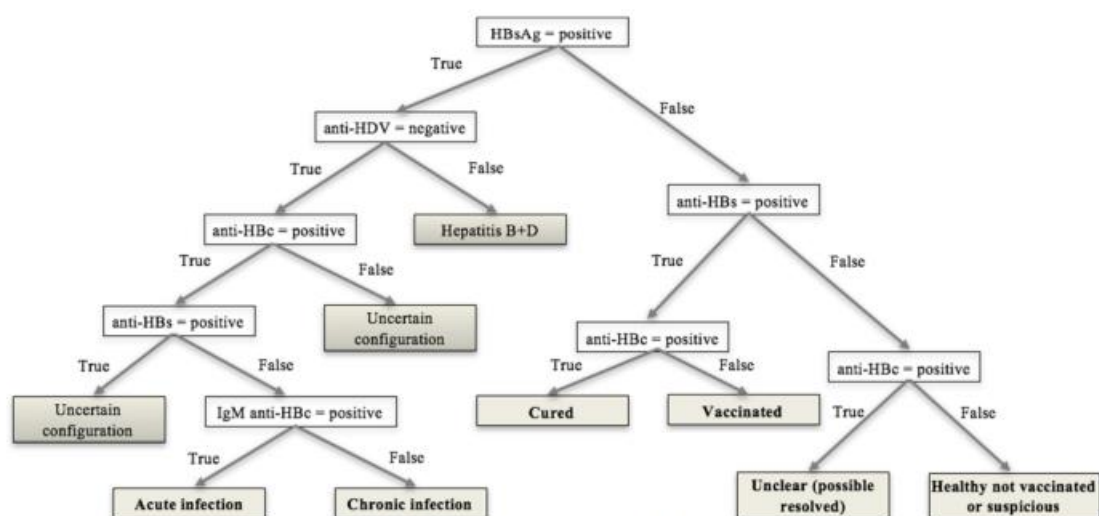


Fig. 2. The decision tree for hepatitis B predictions

2. Implementasi dan Penjelasan

Pada implementasi dibuat 17 rules, dengan 2 rules di antaranya merupakan rule output baik output ketiga goal tercapai, maupun rule ketika tidak tercapai goal akibat kesalahan input user.

Input yang valid hanyalah positive dan negative. Kondisi awal dari program tidak terdapat fakta pada working memory (tidak ada def facts). Dan hanya terdapat 2 buah rules yang dijalankan yaitu isHSaBG yang tidak memiliki prekondisi dan rule goal-fail dengan salience -1. Karena default salience adalah 0, maka pada awal program akan selalu dijalankan rule isHSaBG.

Setiap langkah angka diminta masukan dari user, lalu bilamana masukan user valid (positive/negative) akan menghasilkan fakta baru yang mengakibatkan rule lain teraktivasi. Namun, jika masukan tidak valid, maka akan mengakibatkan hanya rule goal-fail yang aktif (dengan salience -1) yang selalu ada selama tidak terdapat goal pada working memory.

Referensi Angka:

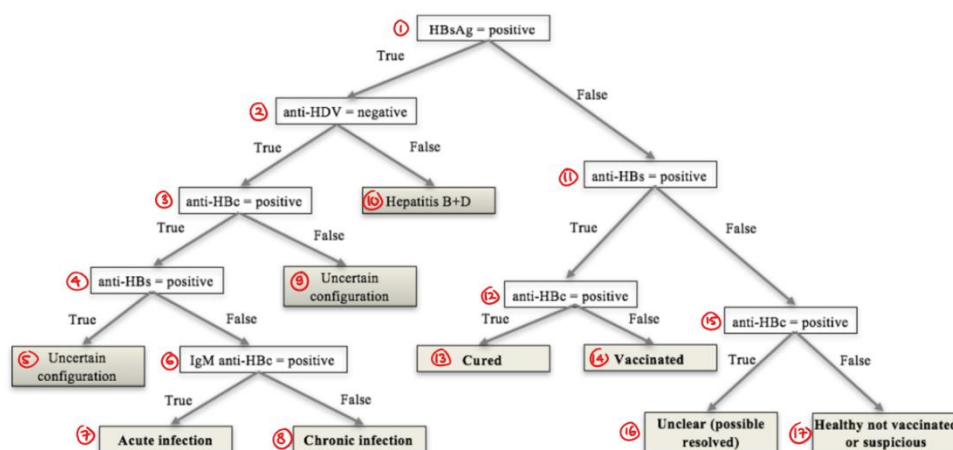


Fig. 2. The decision tree for hepatitis B predictions

Berikut merupakan rincian dari rules pada program:

No	Rules	Penjelasan
1	(defrule goal-found (goal ?g) => (printout t "Hasil prediksi = " ?g crlf));	Rule output ketika ditemukan goal (leaf pada decision tree).
2	(defrule goal-fail (declare (salience -1)) (not (goal ?g)) => (printout t "Invalid Input, Please Try Again" crlf));	Rule output ketika terjadi kesalahan input sehingga menimbulkan tidak ada rule baru yang teraktivasi (prekondisi tidak sesuai).
3	(defrule isHBsAG => (printout t "HBsAg? (positive/negative)")	Rule menentukan HBsAG. (Node 1)

	(assert (HBsAg(read))));	
4	(defrule isAnti-HDV (HBsAg positive) => (printout t "Anti-HDV? (positive/negative) ") (assert (Anti-HDV (read)))));	Rule menentukan isAnti-HDV. (Node 2)
5	(defrule isAnti-HBc (HBsAg positive) (Anti-HDV negative) => (printout t "Anti-HBc? (positive/negative) ") (assert (Anti-HBc (read)))));	Rule menentukan isAnti-HBc. (Node 3)
6	(defrule isAnti-HBs (HBsAg positive) (Anti-HDV negative) (Anti-HBc positive) => (printout t "Anti-HBs? (positive/negative) ") (assert (Anti-HBs (read)))));	Rule menentukan isAnti-HBs. (Node 4)
7	(defrule goal-uncertain (HBsAg positive) (Anti-HDV negative) (Anti-HBc positive) (Anti-HBs positive) => (assert (goal "Uncertain Configuration")));	Rule menentukan goal-uncertain. (Node 5)
8	(defrule IgM-Anti-HBc (HBsAg positive) (Anti-HDV negative) (Anti-HBc positive) (Anti-HBs negative) => (printout t "IgM Anti-HBc? (positive/negative) ") (assert (IgM-Anti-HBc (read)))));	Rule menentukan IgM-Anti-HBc. (Node 6)
9	(defrule goal-Acute-Infection (HBsAg positive) (Anti-HDV negative)	Rule menentukan goal-Acute-Infection. (Node 7)

	(Anti-HBc positive) (Anti-HBs negative) (IgM-Anti-HBc positive) => (assert (goal "Acute Infection")));	
10	(defrule goal-Chronic-Infection (HBsAg positive) (Anti-HDV negative) (Anti-HBc positive) (Anti-HBs negative) (IgM-Anti-HBc negative) => (assert (goal "Chronic Infection")));	Rule menentukan goal-Chronic-Infection. (Node 8)
11	(defrule goal-Uncertain-Configuration2 (HBsAg positive) (Anti-HDV negative) (Anti-HBc negative) => (assert (goal "Uncertain Configuration")));	Rule menentukan goal-Uncertain-Configuration2. (Node 9)
12	(defrule goal-HepatitisB+D (HBsAg positive) (Anti-HDV positive) => (assert (goal "Hepatitis B+D")));	Rule menentukan goal-HepatitisB+D. (Node 10)
13	(defrule Anti-HBs2 (HBsAg negative) => (printout t "Anti-HBs? (positive/negative) ") (assert (Anti-HBs (read)))));	Rule menentukan Anti-HBs2. (Node 11)
14	(defrule Anti-HBc2 (HBsAg negative) (Anti-HBs positive) => (printout t "Anti-HBc? (positive/negative) ") (assert (Anti-HBc (read)))));	Rule menentukan Anti-HBc2. (Node 12)
15	(defrule goal-Cured (HBsAg negative) (Anti-HBs positive) (Anti-HBc positive)	Rule menentukan goal-Cured. (Node 13)

	<pre>=> (assert (goal "Cured")));</pre>	
16	<pre>(defrule goal-Vaccinated (HBsAg negative) (Anti-HBs positive) (Anti-HBc negative) => (assert (goal "Vaccinated")));</pre>	Rule menentukan goal-Vaccinated. (Node 14)
17	<pre>(defrule Anti-HBc3 (HBsAg negative) (Anti-HBs negative) => (printout t "Anti-HBc? (positive/negative) ") (assert (Anti-HBc (read)))));</pre>	Rule menentukan Anti-HBc3. (Node 15)
18	<pre>(defrule goal-Unclear (HBsAg negative) (Anti-HBs negative) (Anti-HBc positive) => (assert (goal "Unclear (possible resolved)")));</pre>	Rule menentukan goal-Unclear. (Node 16)
19	<pre>(defrule goal-Healthy (HBsAg negative) (Anti-HBs negative) (Anti-HBc negative) => (assert (goal "Healthy not vaccinated or suspicious")));</pre>	Rule menentukan goal-Healthy. (Node 17)

Contoh penjalanan program:

1. State Awal

CLIPS> (reset)

CLIPS> |

Agenda

X

Reset

Run

Step

Halt Rules

Focus Stack		Salience	Rule	Basis
MAIN		0	isHSaBG	*
		-1	goal-fail	*

2. Langkah Request facts ke user

```
CLIPS> (reset)
CLIPS> (run)
HBsAg? (positive/negative) |
```

Agenda X			
<input type="button" value="Reset"/> <input type="button" value="Run"/> <input type="button" value="Step"/> <input type="button" value="Halt Rules"/>			
Focus Stack	Saliency	Rule	Basis
MAIN	0	isHSaBG	*
	-1	goal-fail	*

3. Langkah update activated rules (agenda) berdasarkan fakta baru

```
CLIPS> (reset)
CLIPS> (run)
HBsAg? (positive/negative) positive
Anti-HDV? (positive/negative) |
```

Agenda X			
<input type="button" value="Reset"/> <input type="button" value="Run"/> <input type="button" value="Step"/> <input type="button" value="Halt Rules"/>			
Focus Stack	Saliency	Rule	Basis
MAIN	0	isAnti-HDV	f-1
	-1	goal-fail	*

4. Langkah terakhir (kesimpulan)

```
CLIPS> (reset)
CLIPS> (run 1)
HBsAg? (positive/negative) positive
CLIPS> (run 1)
Anti-HDV? (positive/negative) positive
CLIPS> |
```

Agenda X			
<input type="button" value="Reset"/> <input type="button" value="Run"/> <input type="button" value="Step"/> <input type="button" value="Halt Rules"/>			
Focus Stack	Saliency	Rule	Basis
MAIN	0	goal-HepatitisB+D	f-1,f-2
	-1	goal-fail	*

```
CLIPS> (reset)
CLIPS> (run)
HBsAg? (positive/negative) positive
Anti-HDV? (positive/negative) positive
Hasil prediksi = Hepatitis B+D
CLIPS>
```

Agenda X			
<input type="button" value="Reset"/> <input type="button" value="Run"/> <input type="button" value="Step"/> <input type="button" value="Halt Rules"/>			
Focus Stack	Saliency	Rule	Basis

5. Kasus salah masukan user
- ```
CLIPS> (reset)
CLIPS> (run 1)
HBsAg? (positive/negative) sorry
CLIPS> |
```

The image shows two screenshots of the CLIPS shell interface. The top screenshot shows the initial state where the 'goal-fail' rule is active with a salience of -1. The bottom screenshot shows the state after an invalid input, where the 'goal-fail' rule is no longer active and the focus stack is empty.

**Top Screenshot:**

Agenda X

Reset Run Step Halt Rules

| Focus Stack | Saliency | Rule      | Basis |
|-------------|----------|-----------|-------|
| MAIN        | -1       | goal-fail | *     |

```
CLIPS> (reset)
CLIPS> (run 1)
HBsAg? (positive/negative) sorry
CLIPS> (run 1)
Invalid Input, Please Try Again
CLIPS> |
```

**Bottom Screenshot:**

Agenda X

Reset Run Step Halt Rules

| Focus Stack | Saliency | Rule | Basis |
|-------------|----------|------|-------|
|-------------|----------|------|-------|

#### Analisis:

- Terlihat pada state awal terdapat rule isHSaBG dan goal-fail pada agenda, dan dilakukan rule isHSaBG (karena salience 0 isHSaBG > salience -1 goal-fail).
- Pada langkah setelah input (penambahan fakta) dari user, didapat bahwa rule yang sudah diaktifkan tidak terdapat pada agenda (refractoriness). Lalu teraktivasi rule baru berdasarkan fakta bersangkutan.
- Pada langkah yang mengakibatkan state pada leaf (pada Decision Tree), akan mengaktifkan rule goal-[xxxx]. Lalu karena memiliki salience 0, rule goal-found akan teraktivasi, lalu pada agenda akan kosong karena goal-fail akan hilang (terdapat goal sehingga prekondisi goal-fail tidak terpenuhi).
- Rule goal-fail diperlukan untuk mengantisipasi kesalahan masukan user yang mengakibatkan pada agenda hanya terdapat 1 buah rule yaitu goal-fail, sehingga akan teraktivasi goal-fail.
- Rule goal-fail perlu diberikan salience -1 agar tidak diaktifasi jika terdapat rule lain pada agenda (default salience 0).