

## KONSEP INFERENSI PADA MODEL PENGETAHUAN BERBASIS TERNARY GRID <sup>1</sup>

Yuliadi Erdani

Politeknik Manufaktur Negeri Bandung  
Jl. Kanayakan No. 21 Dago, Bandung – 40135, Telp (022)-250 0241  
e-mail: [yul\\_erdani@polman-bandung.ac.id](mailto:yul_erdani@polman-bandung.ac.id)

### Abstrak

Mesin inferensi dari sistem pakar bertugas melakukan pencarian aturan-aturan yang dapat digunakan untuk mengambil keputusan atas fakta-fakta masukan yang diberikan oleh pengguna. Kualitas dan kinerja mesin inferensi bertanggung jawab terhadap kualitas keputusan yang dihasilkan dan efisiensi proses inferensi yang dilakukan. Kesalahan-kesalahan dalam proses inferensi dapat berakibat fatal karena keputusan yang dihasilkan tidak sesuai dengan yang diharapkan oleh orang pakar, selain itu kesalahan-kesalahan tersebut dapat mengakibatkan sistem pakar masuk ke dalam situasi yang tidak menentu atau tidak berakhir (*endless loop/hang*).

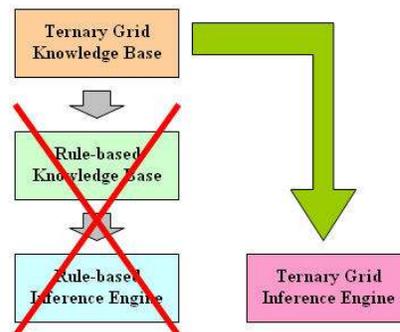
Untuk mengatasi masalah-masalah tersebut, maka telah dikembangkan konsep inferensi yang dapat mengatasi kesalahan-kesalahan pada proses inferensi dan meningkatkan efisiensi proses inferensi itu sendiri. Metoda yang digunakan adalah dengan cara mengoptimalkan aturan-aturan terlebih dahulu dan selanjutnya memainkan (*matching*) fakta-fakta masukan kepada aturan-aturan yang sudah optimal tersebut. Optimasilasi aturan dilakukan dengan cara mengeliminasi beberapa aturan yang redundan dan kritis. Konsep inferensi tersebut selanjutnya dapat diimplementasikan pada mesin inferensi (*inference engine*). Dari beberapa contoh kasus yang diberikan, terlihat bahwa proses inferensi ini dapat menghasilkan keputusan dengan benar tanpa melakukan iterasi yang lama, sehingga prosesnya efisien.

**Keyword:** *intelligent systems, expert systems, inference engine*

### 1. PENDAHULUAN

Sebagai bagian dari sistem pakar atau sistem basis pengetahuan, mesin inferensi (*inference engine*) bertugas melakukan inferensi terhadap aturan-aturan yang disimpan pada Basis Pengetahuan. Mesin inferensi merupakan otaknya sistem basis pengetahuan yang mengolah informasi dari basis pengetahuan. Cara kerja dari mesin inferensi adalah mengolahnya fakta yang diberikan oleh pengguna dan mencari keterkaitan antara fakta-fakta tersebut dengan fakta-fakta dan aturan-aturan yang disimpan pada basis pengetahuan.

Mesin inferensi memiliki peranan yang cukup besar terhadap keefektifan suatu sistem pakar, karena mesin inferensi disamping menentukan kesimpulan atau keputusan dari suatu sistem pakar, juga menentukan efisiensi proses inferensi yang berimplikasi terhadap waktu proses inferensi. Pada sistem pakar CongaXpert [Yuliadi, 2003], mesin inferensi yang dikembangkan menggunakan metoda *forward chaining* berbasis aturan.



Gambar 1. Mesin inferensi Ternary Grid

Pada KasTerGrid [Yuliadi, 2004], pengetahuan pakar hasil akuisisi tidak bisa langsung disimpulkan oleh mesin inferensi karena mesin tersebut masih bekerja pada model pengetahuan sistem produksi [Newell, 1972]. Agar pengetahuan dapat diakuisisi maka pengetahuan tersebut harus dikonversi kembali ke Pengetahuan berbasis aturan sehingga istilahnya menjadi “pengetahuan berbasis Aturan yang sudah dioptimalkan (optimized Rule-based Knowledge)”.

<sup>1</sup> Penelitian ini didanai oleh Dana Penelitian Hibah Bersaing DP2M-DIKTI, 2007-2009

Dengan dikembangkannya teknik inferensi berbasis Ternary Grid maka tahapan konversi dari Pengetahuan Ternary Grid tidak perlu diubah kembali ke Pengetahuan berbasis Aturan melainkan dapat langsung disimpulkan oleh mesin inferensi sehingga proses akuisisi pengetahuan dapat lebih sederhana dan waktu proses akuisisi lebih singkat yang implikasinya berupa peningkatan kinerja sistem pakar atau sistem basis pengetahuan [Yuliadi, 2007] (gambar 1).

Makalah ini membahas secara jelas konsep inferensi yang telah dikembangkan baik dalam bentuk narasi, gambar-gambar maupun penjelasan matematik. Keefektifan dan keefisienan proses inferensi ini ditunjukkan dengan beberapa contoh kasus. Pada makalah ini diulas pula secara ringkas dan jelas beberapa penelitian terdahulu yang mengantarkan kepada penelitian ini.

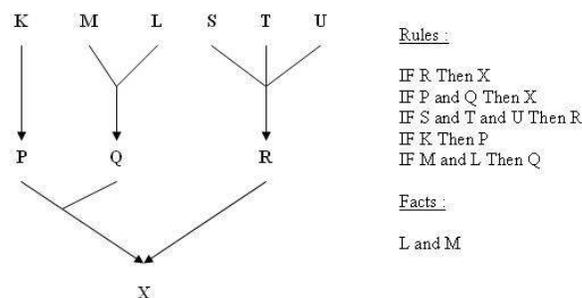
## 2. TINJAUAN PUSTAKA

Inferensi adalah proses membuat keputusan berdasarkan pengetahuan yang ada [David, 2003]. Bagi seorang pakar, inferensi berarti proses membuat keputusan atau memberikan informasi konsultasi berdasarkan keahlian yang dimilikinya. Perpaduan antara orang pakar dengan komputer menghasilkan apa yang disebut dengan sistem pakar. Sistem pakar dapat diartikan sebagai sekumpulan program komputer (perangkat lunak) yang bekerja seperti seorang pakar manusia dan berfungsi untuk menyelesaikan suatu permasalahan berdasarkan pengetahuan yang dimilikinya dan disimpan di memorinya (basis pengetahuan) [Patterson, 1990] [Tanimoto, 1990] [Williamson, 1986]. Salah satu komponen yang ada pada sistem pakar adalah mesin inferensi. Pada sistem pakar berbasis aturan, mesin inferensi bekerja mengolah fakta dan aturan untuk menentukan aturan mana yang dapat dipakai. Terdapat dua teknik yang digunakan dalam melakukan inferensi, yaitu *forward chaining* dan *backward chaining* [Lunze, 1994] [Giarratano, 2005], [Ignizio, 1991], [Jackson, 1998]. Meskipun metoda tersebut merupakan metoda baku, namun tidak menjamin akan efisiensi dan keefektifan suatu sistem pakar karena hal itu tergantung pula kepada struktur dan model pengetahuan yang diterapkan.

### *Forward Chaining*

Pada mesin inferensi dengan teknik *forward chaining*, aturan-aturan yang memungkinkan dapat digunakan dikumpulkan terlebih dahulu, kemudian dieksekusi. Aturan-aturan tersebut adalah aturan yang bagian kondisinya (fakta) sudah dikenal. Fakta-fakta tersebut bisa datang dari masukan pengguna sebagai bagian dari masalah atau fakta yang dihasilkan dari proses inferensi sebelumnya [Patterson, 1990].

Mesin inferensi mencoba mengeksekusi setiap aturan tersebut. Jika suatu aturan berfungsi (dapat digunakan), maka proses inferensi dilanjutkan ke aturan lainnya. Setiap aturan yang dapat digunakan akan disimpan didalam suatu daftar.



**Gambar 2.** Proses inferensi

Pada teknik *forward chaining*, mesin inferensi membuat simpulan-simpulan secara deduksi, dimana fakta-fakta yang ada pada bagian kondisi suatu aturan diproses terlebih dahulu. Jika bagian kondisi tersebut bernilai benar (logika), maka aturan tersebut digunakan (*fired*). Selanjutnya fakta yang ada pada bagian simpulan suatu aturan dijadikan sebagai fakta baru yang sudah dikenal (fakta turunan), dan fakta-fakta ini digunakan untuk mengeksekusi aturan-aturan lainnya yang berisi fakta tersebut. Gambar 2 menjelaskan proses inferensi pada suatu pohon aturan.

Gambar 2 menunjukkan pohon aturan yang berisi 5 aturan. Pada teknik inferensi ini, sistem tidak memiliki target apapun. Target hanya akan dibuat jika ada fakta masukan yang memenuhi syarat. Jika fakta L dan M diberikan, maka mesin inferensi akan memproses fakta tersebut dan menentukan simpulannya sebagai sebuah target. Dalam hal ini Q merupakan target atas masukan fakta L dan M. Selanjutnya Q dijadikan sebagai fakta baru yang diketahui. Dari fakta Q tersebut mesin inferensi mencari target baru. Harapannya adalah target X dapat dicapai.

Namun karena ada persyaratan lain yang harus dipenuhi, yaitu fakta P, sementara fakta P tidak dikenal, maka target X tidak terpenuhi sehingga proses inferensi berhenti sampai disini.

Hal penting yang berkaitan dengan dengan teknik *forward chaining* [Patterson, 1990] [Lunze, 1994] adalah:

1. Teknik *forward chaining* adalah teknik yang dikendalikan oleh data (data driven)
2. Dari fakta masukan akan diproses semua kemungkinan simpulan secara deduktif
3. Proses inferensi akan terus berlangsung selama suatu aturan dapat digunakan, kecuali suatu aturan tidak dapat digunakan, maka proses inferensi berhenti
4. Diperlukan metoda resolusi konflik yang strategis
5. Dapat menghasilkan beberapa target yang tidak direncanakan
6. Teknik ini tidak dapat digunakan untuk melakukan pembuktian suatu target (harus dengan teknik *backward chaining*)

Secara sederhana algoritma untuk teknik *forward chaining* dapat dijelaskan sebagai berikut:

1. *Data = basis data berisi himpunan fakta masukan dari pengguna*
2. *Selama isi data tidak memenuhi kriteria suatu aturan*
  - 2.1. *Memilih aturan yang kemungkinan dapat digunakan, dimana bagian kondisinya ada pada basis data*
  - 2.2. *Data = bagian simpulan dari aturan yang dapat digunakan*

### ***Backward Chaining***

Kebalikan dengan *forward chaining*, pada *backward chaining*, hal yang pertama ditentukan adalah targetnya terlebih dahulu, yaitu fakta yang merupakan bagian simpulan dari suatu aturan. Berangkat dari target tersebut, mesin inferensi mencari aturan-aturan yang fakta bagian kondisinya memenuhi kriteria sesuai dengan himpunan fakta masukan (bergerak ke belakang). Apabila fakta-fakta bagian kondisi tersebut memenuhi kriteria, maka aturan tersebut dapat digunakan. Dari fakta-fakta baru yang didapat, selanjutnya mesin inferensi mencari lagi fakta-fakta lainnya dengan bergerak ke belakang, dari bagian simpulan ke bagian kondisi suatu aturan [Lunze, 1994] [Patterson, 1990] [James, 2008].

*Backward chaining* termasuk teknik inferensi yang efektif dan banyak diterapkan di berbagai aplikasi sistem pakar. *Backward chaining* disebut juga sebagai teknik inferensi yang dikendalikan oleh target (goal driven). Melihat contoh kasus yang diberikan pada gambar 2, mesin inferensi dengan *backward chaining* ini akan memulai dari target yang dimasukan oleh pengguna. Misalnya fakta R diketahui, maka mesin inferensi menjadikan fakta R tersebut sebagai target. Selanjutnya mesin inferensi memproses fakta-fakta bagian kondisi dari aturan yang ada R tersebut. Jika fakta-fakta S, T dan U diketahui, maka aturan tersebut dapat digunakan dan mesin inferensi menyimpulkan bahwa fakta R memenuhi kriteria. Demikian proses inferensi berlanjut hingga semua aturan yang memungkinkan untuk digunakan selesai diproses. Keuntungan dari *backward chaining* ini adalah dapat membuat kesimpulan-kesimpulan baru meskipun fakta-fakta yang ada tidak memenuhi kriteria, karena mesin akan memproses terus fakta-fakta yang ada ke belakang (ke bagian kondisi suatu aturan).

Meskipun *backward chaining* terbilang efektif dan banyak digunakan pada berbagai sistem pakar, namun [James, 2008] menyatakan bahwa penggunaan penggunaan teknik inferensi harus disesuaikan dengan permasalahan yang akan diselesaikan. Perbandingan antara bagian kondisi dan bagian simpulan dari aturan dapat dijadikan bahan untuk pemilihan teknik inferensi. Jika dari aturan-aturan yang ada secara rata-rata memiliki lebih banyak bagian kondisinya daripada bagian simpulannya, maka setiap hipotesis atau target akan menghasilkan banyak aktifitas untuk memproses kondisi suatu aturan, sehingga teknik inferensi dengan *forward chaining* pada situasi ini lebih sesuai. Jika situasinya sebaliknya, maka *backward chaining* lebih sesuai.

Sistem pakar yang dikembangkan oleh [Yuliadi, 2003] menggunakan teknik *forward chaining* dalam proses inferensinya. Disampin perbandingan bagian kondisi lebih banyak daripada bagian simpulannya dari aturan-aturannya yang ada, juga fungsi dari sistem pakar tersebut lebih kearah pembuatan keputusan pakar daripada melakukan validasi.

### **3. METODE PENELITIAN**

Titik berat dari penelitian ini lebih ke arah pekerjaan yang melibatkan perangkat lunak. Tidak ada perangkat keras khusus yang digunakan selain komputer (PC). Perangkat lunak yang digunakan adalah masih bahasa program Visual Basic versi 6 (VB6). Sebagai alasan kenapa VB6 masih digunakan (bukan VB.NET) dikarenakan kehandalan dan kestabilan VB6 sudah terpercay dan mampu memenuhi kebutuhan untuk

kepentingan penelitian ini. Selain itu, teknik akuisisi berbasis Ternary Grid yang sudah dikembangkan, dibuat dengan VB6 dengan hasil yang sangat memuaskan. Namun demikian penggunaan VB.NET di masa mendatang akan menjadi pilihan yang sangat memungkinkan.

Tahapan-tahapan yang dilakukan pada proses inferensi dengan Ternary Grid terdiri dari

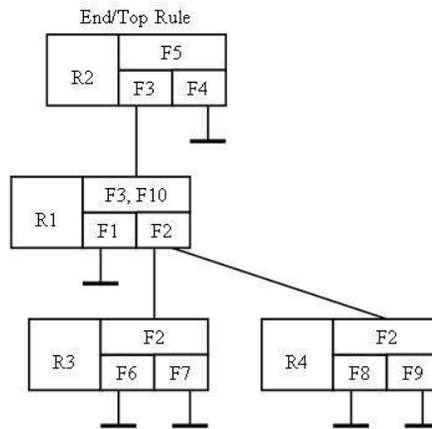
- Membuat ilustrasi pengetahuan
- Pengujian konsistensi dan kelaikan aturan
- Optimasi simpulan jamak (*multiple conclusion/fan out*)
- Inferensi secara *forward chaining*

**Ilustrasi pengetahuan**

Pengetahuan yang disimpan di dalam Basis Pengetahuan dapat diilustrasikan seperti terlihat pada gambar 3. Basis pengetahuan yang digunakan oleh Sistem Pakar pada penelitian ini memiliki format aturan produksi [Newell, 1972] yang komposisinya terdiri dari Fakta dan Aturan. Format tersebut telah dioptimalkan pada penelitian terdahulu [Yuliadi, 2005] [Yuliadi, 2007] agar didapatkan kinerja pengetahuan yang lebih baik. Format tersebut selanjutnya disebut sebagai Basis Pengetahuan Ternary Grid. Format Aturan product umumnya ditulis sebagai berikut:

$$IF <kondisi> THEN <simpulan>$$

Atau secara matematik dinyatakan sebagai berikut:  $kondisi \rightarrow simpulan$



**Gambar 3.** Ilustrasi pengetahuan

Keterangan gambar:

Rn	Yn
	Xn

Rn : Aturan ke n  
 Xn : Bagian kondisi dari Rn, Yn : Bagian simpulan dari Rn



Rantai aturan



Fakta masukan (fakta yang tidak diproduksi oleh aturan manapun)

**Pengujian konsistensi dan kelaikan (consistency and feasibility)**

Aturan yang tidak konsisten adalah aturan yang tidak memiliki makna jelas. Hal ini terjadi apabila bagian simpulan suatu aturan dijadikan lagi sebagai bagian kondisi aturan. Aturan yang tidak laik adalah aturan yang tidak akan pernah bisa diaplikasikan, sebab bagian simpulan dari aturan tidak akan pernah terpenuhi dikarenakan bagian kondisinya juga tidak akan pernah menghasilkan nilai benar (terpenuhi). Baik aturan yang tidak konsisten maupun aturan yang tidak laik memiliki struktur aturan sebagai berikut:

$$IF <...> \& <...> \& <Y> THEN <Y> + <...> + <...>$$

Dengan memperhatikan struktur Ternary Grid [Yuliadi, 2007], aturan tersebut dapat diinvestigasi dengan algoritma sebagai berikut:

$$B = \left\{ i \mid a_{ij} = 3 \right\} \text{ atau } B = \left\{ i \mid \left( \sum_j (a_{ij} \text{ div } 3) \right) > 0 \right\}$$

Yang berarti mencari aturan-aturan yang memiliki nilai 3. Aturan tersebut tidak dapat langsung dieliminasi, melainkan dikembalikan kepada pengguna untuk memperbaikinya, karena maksud atau konten suatu aturan hanya pengguna yang tahu. Mesin hanya bisa memberikan informasi bahwa susunan aturan tidak valid.

#### Optimasi simpulan jamak (multiple conclusion/fan out)

Aturan ini memiliki bagian kondisi yang sama dengan aturan lainnya namun berbeda pada bagian simpulannya. Dengan demikian kedua aturan tersebut dapat dieliminasi dan diganti dengan aturan yang baru yang mewakili keduanya. Dengan memperhatikan struktur Ternary Grid [Yuliadi, 2007], maka algoritmanya dapat dijelaskan sebagai berikut:

- o Mencari aturan-aturan yang memiliki bagian kondisi yang sama

$$Bi = \left\{ p \mid Ri1 = Rp1, i < p, p \in N \right\}$$

- o Membuat aturan baru  $Rq$  yang memiliki bagian simpulan jamak

$$Rq2 = \left\{ Ri2 \cup Rp2 \mid p \in Bi, q \in N \right\} \text{ dan } Rq1 = Ri1$$

- o Eliminasi aturan-aturan lama

$$R = \left\{ Ri \cup Rp \mid p \in Bi \right\}$$

#### Inferensi secara forward chaining

Pada proses inferensi dengan teknik forward chaining ini hanya menggunakan proses iterasi. Seluruh fakta masuk dari pengguna disimpan pada himpunan fakta  $F_k$ . Selanjutnya mesin inferensi ini akan mencari semua aturan-aturan yang kemungkinan dapat digunakan dan disimpan di dalam himpunan  $R_x$ .

$$R_x = \left\{ p \mid p \rightarrow q, p \in F_k, p \in F, F_k \subset F \right\}$$

$R_x$  : sekumpulan aturan yang memungkinkan untuk dieksekusi

$F_k$  : sekumpulan fakta yang dikenal

$F$  : himpunan semua fakta yang ada pada basis pengetahuan

Dari himpunan aturan, mesin inferensi memutuskan aturan yang mana yang dapat digunakan, dan disimpan di dalam himpunan  $R_{y_n}$ .

$$R_{y_n} \subset R_x$$

Untuk menghindari duplikasi, maka dilakukan resolusi konflik dengan rumus berikut, selanjutnya aturan yang dapat digunakan disimpan didalam himpunan  $R_z$ .

$$R_z = \bigcup_n R_{y_n}$$

## 4. HASIL DAN PEMBAHASAN

Untuk mengimplementasikan metoda yang sudah dijelaskan pada bagian sebelumnya, maka beberapa eksperimen dilakukan. Sebagai bahan eksperimen, contoh kasus dalam bentuk data Ternary Grid diberikan seperti terlihat pada gambar 4.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
R1	1	1	2	0	0	0	0	0	0	0	0	0	0	0
R2	0	0	1	2	0	0	0	0	0	0	0	0	0	0
R3	0	0	0	1	1	2	0	0	0	0	0	0	0	0
R4	0	2	0	0	0	0	1	1	0	0	0	0	0	0
R5	0	2	0	0	0	0	1	1	1	0	0	0	0	0
R6	0	3	0	0	0	0	0	0	0	1	1	0	0	0
R7	0	2	0	0	0	0	0	0	0	0	0	1	1	0
R8	1	1	0	0	0	0	0	0	0	0	0	0	0	2
R9	0	0	0	1	1	2	0	0	0	0	0	0	0	0

Gambar 4. Contoh kasus, aturan dalam format Ternary Grid

Dengan merepresentasikan nilai 0 sebagai sel grid atau kotak matriks yang kosong, maka susunan matriks aturan terlihat pada gambar 5.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
R1	1	1	2											
R2			1	2										
R3				1	1	2								
R4		2					1	1						
R5		2					1	1	1					
R6		3								1	1			
R7		2										1	1	
R8	1	1												2
R9				1	1	2								

Gambar 5. Eliminasi nilai 0 pada matriks aturan

Selanjutnya aturan-aturan yang ada pada matriks tersebut dioptimalkan dengan algoritma optimasi aturan Ternary Grid. Optimalisasi meliputi peniadaan beberapa aturan dengan memperhatikan kepada beberapa sifat atau karakter aturan yang meliputi konsistensi dan kelaikan (*consistency and feasibility*) dan simpulan jamak (*multiple conclusion/fan out*).

Setelah proses optimasi aturan tersebut dilakukan, maka dihasilkan aturan-aturan yang sudah dioptimalkan seperti terlihat pada gambar 6.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
R1	1	1	2							2
R2			1	1	2					
R3		2				1	1			
R4		2						1	1	

Gambar 6. Aturan-aturan yang sudah dioptimalkan

Deskripsi proses inferensi dari contoh kasus yang diberikan dapat dijelaskan sebagai berikut: Jika beberapa fakta berikut F1, F4, F6 dan F7 diberikan oleh user ke sistem pakar, maka proses inferensi akan dimulai dengan menginvestigasi fakta terkecil (F1) hingga ke fakta terbesar (F7). Himpunan fakta masukan adalah:

$$F_x = \{F1, F4, F6, F7\}$$

Selanjutnya mesin inferensi akan mencari seluruh aturan-aturan pada kolom fakta F1 dimana fakta tersebut merupakan bagian kondisi dari aturan-aturan tersebut. Penjelasan dapat dilihat pada gambar 7. Pada gambar tersebut terlihat bahwa fakta F1 merupakan bagian kondisi dari aturan R1, fakta F4 merupakan bagian kondisi dari aturan R2, fakta F6 dan F7 merupakan bagian kondisi dari aturan R3.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
R1	1	1	2							2
R2			1	1	2					
R3		2				1	1			
R4		2						1	1	

Gambar 7. Inferensi fakta yang dikenal/masukan F1 s.d. F7

Hasil dari inferensi fakta tersebut didapatkan himpunan aturan yang bagian kondisinya berisi fakta-fakta yang ada pada himpunan Fx yaitu:

$$R_x = \{R1, R2, R3\}$$

Selanjutnya mesin inferensi melakukan inferensi aturan mulai dari aturan R1 hingga R3.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
R1	1	1	2							2
R2			1	1	2					
R3		2				1	1			
R4		2						1	1	

Gambar 8. Inferensi aturan (iterasi ke 1).

Pada iterasi pertama (gambar 8) terlihat bahwa aturan R1 dan R2 tidak terpakai karena ada beberapa fakta pada bagian kondisi tidak diketahui, sementara aturan R3 dapat dipakai karena semua bagian kondisinya yaitu fakta F6 dan F7 dikenal, sehingga:

$$R_{y,1} = \{R3\} \text{ dan}$$

$$F_{y,1} = \{F1, F2, F4, F6, F7\}$$

Pada iterasi ke dua, aturan R1 dapat dipakai karena fakta F2 menjadi dikenali setelah aturan R3 dipakai, sehingga:

$$R_{y,2} = \{R1, R3\} \text{ dan}$$

$$F_{y,2} = \{F1, F2, F3, F4, F6, F7\}$$

Pada iterasi ketiga, aturan R2 dapat dipakai setelah fakta F3 dikenali akibat penggunaan aturan R1, sehingga:

$$R_{y,3} = \{R1, R2, R3\} \text{ dan}$$

$$F_{y,3} = \{F1, F2, F3, F4, F5, F6, F7\}$$

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
R1	1	1	2							2
R2			1	1	2					
R3		2				1	1			
R4		2						1	1	

Gambar 9. Ilustrasi inferensi keseluruhan

Setelah iterasi ke 3 mesin inferensi berhenti bekerja karena aturan R2 merupakan aturan akhir (*end-rule/top-rule*). Gambar 9 menunjukkan ilustrasi proses inferensi keseluruhan dari kasus yang diberikan.

Untuk contoh kasus yang diberikan ini, konsep inferensi yang dikembangkan dapat berfungsi dengan efektif. Setelah semua fakta dan aturan yang dapat digunakan terkumpul, maka dilakukan resolusi konflik, sehingga tidak terjadi duplikasi baik aturan maupun fakta yang digunakan. Pada contoh tersebut tidak terdapat konflik pada aturan yang dapat digunakan.

## 5. KESIMPULAN

Proses inferensi pada Ternary Grid merupakan proses matematik untuk mengolah nilai-nilai matriks. Ditinjau dari pemrograman, matriks tersebut merupakan variabel array 2 dimensi. Situasi ini tentunya sangat menguntungkan bagi proses inferensi sistem pakar dalam membuat keputusan, karena pengetahuan yang disimpan dalam bentuk array 2 dimensi akan mudah untuk diproses secara komputasi.

Dari eksperimen yang sudah dilakukan, hasilnya menunjukkan bahwa pengolahan fakta dan proses inferensi aturan dapat berfungsi secara efektif. Melalui beberapa kali proses iterasi, konsep inferensi ini dapat mengeluarkan hasil yang diharapkan. Namun demikian, eksperimen akan terus dilanjutkan untuk berbagai contoh lainnya agar didapatkan hasil yang cukup signifikan. Langkah selanjutnya yaitu tim peneliti akan melakukan kaji eksperimental untuk kasus yang lebih kompleks dengan bantuan software yang dikembangkan sendiri sehingga efektivitas dan efisiensi konsep inferensi berbasis Ternary Grid yang dikembangkan ini dapat dipertanggungjawabkan.

Merujuk kepada beberapa literatur yang berkaitan dengan sistem pakar atau sistem basis pengetahuan, konsep inferensi dengan Ternary Grid ini merupakan hal baru dalam pengembangan teknik inferensi sehingga diharapkan dapat memberikan kontribusi yang signifikan bagi pengembangan sistem pakar atau sistem basis pengetahuan. Untuk e-government, konsep inferensi ini dapat digunakan oleh pemerintah atau instansi berwenang untuk membuat keputusan-keputusan secara efektif dan efisien atas kasus-kasus yang terjadi dengan merujuk kepada undang-undang atau peraturan-peraturan yang berlaku.

## 6. DAFTAR PUSTAKA

- David J.C. McKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, (2003).
- James Freeman-Hargis, Diakses di website AI Depot pada tanggal 8 Mei 2008, <http://ai-depot.com/Tutorial/RuleBased-Methods.html>
- Lunze, Prof. Dr.-Ing. Jan : Künstliche Intelligenz für Ingenieure, Band 2 : Technische Anwendungen. R. Oldenbourg Verlag, München Wien, 1995, ISBN 3-486-22306-2
- Patterson, Dan W, Artificial Intelligence and Expert System, Prentice Hall, Englewood Cliffs, New Jersey, 1990, ISBN 0-13-477100-1
- Tanimoto, S.L : KI die Grundlagen. R. Oldenbourg Verlag, München Wien, 1990, ISBN 3-486-21403-9
- Williamson, Mickey, Artificial Intelligence for Microcomputer, The Guide for Business and Decisionmakers. Brady Communications Company, New York, 1986, ISBN 0-89303-483-5
- Yuliadi Erdani, Web-Based Consultation System with Expert System, Proc. IASTED CST 2003 – International conference (International Association of Science and Technology for Development – Computer Science and Technology), May 19-21, 2003, Cancun. Mexico. ISBN – 0-88986-349-0, page 61-64

- Yuliadi Erdani, "Ternary Grid as a Potentially New Technique for Knowledge Elicitation/Acquisition", Proc. 2nd IEEE Conference on Intelligent System, Varna - Bulgaria, June 22-24, 2004. vol I: pp. 312-315. ISBN 0-7803-8278-1
- Yuliadi Erdani, "Improving the Knowledge Performance using Ternary Grid Knowledge Acquisition and Model", WSEAS Transactions on Information Science and Application, Issue 2, Volume 2, February 2005. ISSN 1790-0832
- Yuliadi Erdani, "Human Expert Knowledge Acquisition Using Ternary Grid", Jurnal Teknologi (Journal of Technology) Fakultas Teknik Universitas Indonesia, Edisi No. 3, Tahun XXI, September 2007, Jakarta – Indonesia. ISSN 0215-1685
- Yuliadi Erdani, "Mathematical Approach of Rule Optimization in Ternary Grid", Proceeding. Seminar Nasional Ilmu Komputer dan Aplikasinya (SNIKA 2007), Volume 2, No. 1, 1 November 2007, Bandung – Indonesia. ISSN 1907-882X
- Yuliadi Erdani, "Konsep Pengembangan Sistem Pakar dengan Ternary Grid", Proceeding. Seminar Nasional Ilmu Komputer dan Aplikasinya (SNIKA 2007), Volume 2, No. 1, 1 November 2007, Bandung – Indonesia. ISSN 1907-882X