

KAJIAN PEMANFAATAN MODEL PENGETAHUAN UNTUK MEMBANGUN SUATU MODEL ARSITEKTUR JARINGAN SYARAF TIRUAN (JST BERBASIS-ATURAN)

Iping Supriana, Theo Zacharias

Lab. GAIB (Grafika dan Intelegensi Buatan) , Teknik Informatika
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung,
Kampus ITB, Jalan Ganesha 10 Bandung 40132 , Telp. +62 22 2508135
email: iping@informatika.org, theo@informatika.org

ABSTRAK

Jaringan syaraf tiruan (JST) banyak mendapat kritikan sebagai model kotak hitam yang walaupun dapat mengambil suatu kesimpulan dengan baik namun tidak mampu menjelaskan secara logis mengapa kesimpulan tersebut masuk akal untuk diambil. Pada makalah ini ditunjukkan bahwa pernyataan ini tidak sepenuhnya benar dengan cara memperkenalkan JST yang dapat melakukan penalaran mundur pada klausa definite logika proposisi maupun logika orde pertama. JST yang digunakan bukan merupakan JST konvensional seperti propagasi-balik (backpropagation) yang fitur utamanya adalah pembelajaran statistikal. JST yang digunakan adalah JST nonstatistik yang dibuat sedemikian rupa sehingga cocok digunakan untuk melakukan penalaran secara logis dengan kemampuan untuk memberikan argumentasi terhadap kesimpulan yang diperolehnya. JST pada makalah ini merupakan model inferensi yang massively parallel sehingga dapat digunakan untuk melakukan penalaran dengan lebih efisien jika dijalankan pada sistem komputer paralel.

Kata Kunci: jaringan syaraf tiruan, argumentasi, penalaran, inferensi, klausa definite, massively parallel.

1. PENDAHULUAN

Jaringan syaraf tiruan (selanjutnya disebut JST saja) umumnya digunakan di dunia nyata pada masalah aproksimasi fungsi (pemetaan) dari suatu pola masukan ke pola keluaran tertentu. Sebagai contoh, apa yang dilakukan JST pada masalah pembelajaran pengenalan karakter angka adalah mencari aproksimasi dari fungsi yang memetakan kesepuluh jenis pola masukan angka yang diberikan ke dalam pola tertentu yang unik yang berfungsi sebagai *id* dari angka yang bersangkutan. Jika himpunan pola masukan angka dilambangkan dengan variabel vektor \mathbf{x} dan himpunan pola keluaran *id* angka dilambangkan dengan variabel vektor \mathbf{y} , maka pembelajaran JST pada masalah pengenalan karakter angka tidak lain merupakan pencarian aproksimasi dari fungsi $f: \mathbf{x} \rightarrow \mathbf{y}$ berdasarkan nilai \mathbf{x} dan \mathbf{y} tertentu yang diberikan – sesuatu yang di dalam ilmu statistika disebut dengan regresi. Dari sudut pandang statistika, inilah apa yang sebenarnya dilakukan oleh JST: mencari aproksimasi dari suatu fungsi dengan menggunakan teknik regresi (terutama regresi nonlinier pada JST yang menggunakan fungsi aktivasi nonlinier seperti propagasi-balik). Pembahasan detail tentang bagaimana hubungan antara JST dan statistika berada di luar cakupan makalah ini dan dapat dilihat di [8].

JST jenis ini (sebut sebagai JST statistik) tentu memiliki kelebihan yang sama dengan yang dimiliki

oleh statistika, yaitu kemampuan untuk melakukan generalisasi dari data bergalau (*noisy data*). Sayangnya JST jenis ini juga memiliki kelemahan yang dimiliki oleh statistika, yaitu ketidakmampuannya dalam memberikan argumentasi yang logis terhadap kesimpulan yang diperolehnya [6, hal. 584]. Statistika mampu menyimpulkan bahwa SBY-lah yang akan memenangkan Pemilu 2004 yang lalu dengan *quick-count*-nya. Namun dengan *quick-count* yang sama, statistika tidak dapat menjelaskan secara logis mengapa presiden SBY yang menang. Demikian juga halnya dengan JST statistik. Propagasi-balik dapat memberikan *id* dari suatu pola angka yang diberikan. Namun propagasi-balik tidak mampu memberikan alasan yang logis mengapa *id* tersebut bersesuaian dengan pola angka yang diberikan. Inilah alasannya mengapa propagasi-balik biasa disebut dengan model kotak hitam (*black box*).

Pada kasus tertentu, misalnya pada contoh kasus pengenalan angka sebelumnya, fitur pemberian argumentasi terhadap suatu kesimpulan memang tidak terlalu dibutuhkan. Namun pada kasus lain, misalnya pada kasus pengambilan keputusan untuk melakukan operasi bedah jantung, keberadaan suatu argumentasi di balik pengambilan keputusan tertentu merupakan hal yang sangat esensial.

Salah satu pendekatan yang ada untuk mengatasi kelemahan ini adalah melakukan konversi dari bobot-bobot JST hasil pelatihan ke dalam representasi

simbolik. (Pada representasi simbolik seperti kalkulus predikat, pohon keputusan, dsb., pemberian argumentasi bukanlah merupakan suatu masalah.) Teknik ini disebut dengan ekstraksi aturan (*rule extraction*). Survey mengenai beberapa teknik ekstraksi aturan dapat dilihat di [1].

Pendekatan ini, walaupun mampu mengatasi kelemahan JST di atas, memiliki beberapa kelemahan berikut:

- Hasil dari ekstraksi aturan biasanya memiliki tingkat ekspresivitas yang tidak lebih tinggi dari logika proposisi. (Logika proposisi memiliki tingkat ekspresivitas yang relatif rendah karena tidak mengenal adanya variabel sehingga tidak cocok digunakan untuk merepresentasikan objek dan relasi antar objek.) Teknik ekstraksi aturan seperti BRAINNE [9] memiliki kelemahan ini.
- Salah satu keunggulan dari JST adalah sifatnya yang *massively parallel* yang menjadikannya model yang sangat efisien jika dijalankan pada sistem komputer paralel. Hasil dari ekstraksi aturan biasanya bukan merupakan model paralel sehingga tidak memiliki keunggulan ini.

Makalah ini bertujuan untuk memperkenalkan model JST nonstatistik yang dapat digunakan untuk melakukan penalaran pada logika proposisi maupun logika orde pertama (*first-order logic*, selanjutnya disebut FOL saja) dan memberikan argumentasi terhadap kesimpulan yang diperolehnya dengan tetap mempertahankan karakteristik *massively parallel* ini.

Pendekatan yang digunakan adalah kebalikan dari pendekatan ekstraksi aturan di atas, yaitu melakukan konversi dari representasi simbolik ke JST. JST yang dihasilkan merupakan model *massively parallel* yang dapat melakukan penalaran mundur dengan kemampuan untuk memberikan argumentasi. Representasi pengetahuan simbolik yang digunakan adalah logika proposisi dan FOL dalam bentuk klausa *definite*. Dasar-dasar mengenai logika proposisi dan FOL dapat dilihat di [7, hal. 194 – 310].

Sebelum masuk ke dalam pembahasan mengenai JST, pertama-tama akan dibahas pada bagian 2 mengenai klausa *definite* dan mengapa bentuk ini dipilih pada makalah ini. Selanjutnya pada bagian 3 dituliskan secara singkat apa yang dimaksud dengan JST nonstatistik. Isi utama makalah ini tentang bagaimana JST dapat digunakan untuk melakukan penalaran pada klausa *definite* logika proposisi dan FOL dituliskan pada bagian 4 dan 5. Sebagai penutup, pada bagian 6 dituliskan kesimpulan dari makalah ini dan saran pengembangannya.

2. KLAUSA DEFINITE

2.1. Kebutuhan

Masalah untuk menentukan apakah suatu basis pengetahuan dalam logika proposisi memiliki model atau tidak merupakan masalah *satisfiability* (SAT),

yaitu apakah ada interpretasi yang membuat basis pengetahuan tersebut bernilai *true*. Telah dibuktikan bahwa SAT merupakan masalah NP-Complete [4, hal. 428–434]. Berkaitan dengan ini, hampir dipastikan tidak ada aturan inferensi pada logika proposisi dengan properti *sound* dan *complete* yang dapat berjalan dalam waktu polinomial.

Ada dua pilihan yang dapat dilakukan jika diinginkan untuk mengotomatisasi aturan inferensi pada logika proposisi dengan tetap mempertahankan sifat polinomial ini:

1. Mengorbankan salah satu dari properti *sound* atau *complete*;
2. Membatasi bentuk logika proposisi yang diijinkan yang biasanya akan mengurangi tingkat ekspresivitasnya.

Pilihan pertama dapat dilakukan salah satunya dengan menggunakan aturan inferensi Modus Ponens. Modus Ponens merupakan aturan inferensi yang *sound* dan dapat berjalan dalam waktu polinomial. Namun sayangnya Modus Ponens bukanlah aturan inferensi yang *complete* pada logika proposisi [6, hal. 276 – 277].

Pilihan kedua dapat dilakukan salah satunya dengan membatasi logika proposisi ke dalam bentuk tertentu yang disebut dengan klausa *definite* dan menggunakan aturan inferensi Modus Ponens pada klausa tersebut. Selain *sound* dan dapat berjalan dalam waktu polinomial, aturan inferensi Modus Ponens juga *complete* pada klausa *definite*. Inilah alasannya mengapa bahasa pemrograman logika paling populer saat ini, yaitu Prolog, menggunakan aturan inferensi Modus Ponens dan mengambil bentuk klausa *definite* sebagai sintaks bahasanya.

Pembahasan di atas juga berlaku pada FOL. Walaupun demikian perlu dicatat bahwa masalah penentuan kebenaran suatu fakta pada FOL adalah *semi-decidable* [7, hal. 274 – 275]. Ini berarti bahwa ada algoritma yang dapat mengatakan *true* untuk semua fakta yang benar, namun tidak ada algoritma yang dapat mengatakan *false* untuk semua fakta yang tidak benar. Hal ini disebabkan karena adanya simbol fungsi pada FOL yang membuat jumlah *ground term* untuk keperluan substitusi menjadi tak hingga. Akibatnya segala bahasa pemrograman logika dengan kemampuan merepresentasikan simbol fungsi, termasuk Prolog, berpotensi untuk terjebak dalam pengulangan tak hingga (*infinite loop*).

2.2. Sintaks

Klausa *definite* merupakan disjungsi dari literal di mana ada tepat satu positif literal. Semua variabel pada klausa *definite* FOL diasumsikan terkuantifikasi universal. Berikut merupakan dua contoh kalimat logika proposisi dalam bentuk klausa *definite* (dengan P, P_1, P_2 , dan P_3 adalah literal):

$$P \rightarrow P_1 \vee \neg P_2 \vee P_3$$

Untuk memudahkan pembacaan, klausa *definite* biasa dituliskan dalam bentuk implikasi berdasarkan ekivalensi dari klausa $\neg\alpha \vee \beta$ dan kalimat $\alpha \Rightarrow \beta$ dengan α dan β merupakan sembarang kalimat logika proposisi atau FOL. Contoh klausa *definite* di atas dapat dituliskan dalam bentuk implikasi berikut:

$$P \quad \text{(disebut dengan fakta)}$$
$$P_1 \wedge P_2 \Rightarrow P_3 \quad \text{(disebut dengan aturan)}$$

Bentuk dari klausa *definite* dapat dibedakan menjadi dua macam, yaitu fakta dan aturan. Fakta terdiri dari satu literal positif pada logika proposisi atau satu predikat positif pada FOL. Aturan dibentuk oleh badan aturan dan kepala aturan yang dihubungkan dengan simbol implikasi. Badan aturan terdiri dari satu atau lebih fakta yang dihubungkan oleh operator logika konjungsi, sedangkan kepala aturan terdiri dari tepat satu fakta. Arti dari aturan adalah jika semua fakta pada bagian badan aturan benar, maka fakta pada kepala aturan juga benar.

Perlu dicatat bahwa klausa *definite* berbentuk fakta dan aturan di atas tidak mengenal simbol negasi (\neg). Batasan ini, walau akan mengurangi tingkat ekspresivitas, menjamin konsistensi dan keunikan model dari basis pengetahuan yang bersangkutan.

2.3. Aturan Inferensi

Aturan inferensi yang biasa digunakan pada klausa *definite* adalah penalaran maju (*forward-chaining*) dan penalaran mundur (*backward-chaining*). Kedua aturan inferensi ini merupakan aplikasi dari Modus Ponens sehingga keduanya dapat berjalan dalam waktu polinomial serta memiliki properti *sound* dan *complete* pada klausa *definite*. Penalaran maju biasa digunakan untuk mencari model dari suatu basis pengetahuan sedangkan penalaran mundur biasa digunakan untuk membuktikan kebenaran suatu fakta tertentu. Proses inferensi pada penalaran mundur biasa dikaitkan dengan pemberian argumentasi tentang mengapa suatu fakta benar pada kondisi tertentu. Karena fokus dari makalah ini adalah pada masalah pemberian argumentasi, proses inferensi pada JST yang dibahas pada bagian selanjutnya dibatasi hanya pada penalaran mundur.

Aturan inferensi dari penalaran maju untuk mencari model dari suatu basis pengetahuan adalah sebagai berikut:

1. Fakta pada basis pengetahuan merupakan anggota dari model.
2. Pada suatu aturan $\alpha \Rightarrow \beta$, jika semua fakta pada α adalah anggota dari model, maka fakta pada β juga merupakan anggota dari model

Pada FOL, fakta yang merupakan anggota dari model merupakan *grounded* fakta, yaitu fakta yang tidak mengandung variabel. Fakta yang mengandung variabel dapat dikonversi menjadi *grounded* fakta

dengan cara melakukan substitusi dari variabel yang ada dengan konstanta yang merupakan anggota dari tipe variabel tersebut.

Pencarian model dari suatu basis pengetahuan dengan penalaran maju dimulai dari semua fakta yang ada. Kemudian berdasarkan fakta-fakta tersebut diturunkan fakta-fakta baru melalui aturan-aturan pada basis pengetahuan yang bersangkutan. Begitu seterusnya sampai dengan tidak ada fakta baru yang dapat diturunkan lagi.

Kebalikan dari penalaran maju, inferensi pada penalaran mundur dimulai dari fakta yang akan dicoba untuk dibuktikan kebenarannya (sebut sebagai fakta kueri). Kemudian dicari semua aturan dengan kepala aturan yang mengandung fakta kueri tersebut. Dari semua aturan tersebut, dicoba untuk dibuktikan kebenarannya dari fakta-fakta pada badan aturan. Begitu seterusnya sampai dengan fakta yang akan dibuktikan kebenarannya ada secara eksplisit pada basis pengetahuan yang berarti fakta kueri terbukti kebenarannya. Jika kondisi terakhir ini tidak pernah dicapai, disimpulkan bahwa fakta kueri tidak dapat dibuktikan kebenarannya.

3. JST BERBASIS ATURAN (NONSTATISTIK)

Sebelum masuk ke dalam pembahasan tentang bagaimana JST nonstatistik dapat digunakan untuk penalaran, pada bagian ini akan dibahas terlebih dahulu mengenai apa yang dimaksud dengan JST nonstatistik ini.

Sampai dengan saat ini belum ada definisi baku tentang apa itu JST. Walaupun demikian, berdasarkan beberapa definisi dari berbagai sumber [2, hal. 3; 5, hal. 12] dapat dikatakan bahwa JST merupakan sistem pemroses informasi yang memiliki kemiripan karakteristik dengan jaringan syaraf biologis. Beberapa kemiripan karakteristik yang biasa disyaratkan ada pada JST adalah sebagai berikut:

1. JST terdiri dari sejumlah besar unit pemroses informasi, disebut dengan neuron, yang dapat aktif secara paralel.
2. Setiap neuron memiliki memori masing-masing yang berukuran kecil yang biasa digunakan untuk menyimpan sebuah bilangan nyata (disebut dengan nilai aktivasi dari neuron tersebut).
3. Suatu neuron biasa berkomunikasi dengan neuron lain dengan cara mengirimkan sebuah nilai aktivasi ke neuron lain tersebut.
4. Nilai aktivasi yang dikirimkan suatu neuron merupakan hasil aplikasi dari fungsi aktivasi neuron tersebut terhadap nilai aktivasi yang dimilikinya.
5. Nilai aktivasi yang dikirimkan oleh suatu neuron terlebih dahulu dikalikan dengan bobot dari sisi yang dilaluinya sebelum sampai ke neuron tujuan.
6. Jika suatu neuron menerima sebuah nilai aktivasi dari neuron lain, maka neuron tersebut akan

menambahkan nilai aktivasi tersebut ke nilai aktivasinya.

Dengan kata lain, karakteristik utama dari JST adalah sifat *massively parallel*-nya dan penggunaan *message passing* sebagai teknik komunikasi utama antar unit pemroses informasi. Kemampuan untuk belajar dari contoh bergalau yang populer ada pada JST bukanlah merupakan karakteristik utama yang harus ada. Kenyataannya, jaringan Hopfield yang merupakan JST yang cukup populer (walau tidak sepopuler propagasi-balik) tidak memiliki teknik pembelajaran.

JST yang tidak dapat belajar ini tentu tidak terlalu memiliki kaitan dengan statistika. Untungnya, tiadanya kemampuan untuk belajar secara statistik ini diikuti dengan hilangnya karakteristik kotak hitam yang biasa dikaitkan pada JST. Arsitektur JST nonstatistik ini dapat dibuat sedemikian rupa sehingga cocok digunakan untuk melakukan penalaran secara logis dengan kemampuan untuk memberikan argumentasi terhadap kesimpulan yang diperolehnya. JST nonstatistik ini merupakan model inferensi yang *massively parallel* sehingga dapat melakukan penalaran secara efisien jika dijalankan pada sistem komputer paralel.

4. PENALARAN DENGAN JST

Pada bagian ini akan dibahas mengenai bagaimana JST dapat digunakan untuk merepresentasikan pengetahuan dalam bentuk klausa *definite* logika proposisi, melakukan penalaran mundur untuk membuktikan kebenaran suatu fakta dan memberikan argumentasi dari kesimpulan yang diperolehnya. Beberapa kemungkinan solusi untuk penanganan variabel pada FOL dibahas pada bagian selanjutnya.

4.1. Ide Dasar

Ide dasar dari proses inferensi pada JST untuk klausa *definite* logika proposisi adalah sebagai berikut:

1. Kebenaran setiap fakta direpresentasikan dengan satu neuron.
2. Setiap aturan direpresentasikan dengan JST yang arsitektur dan bobot-bobotnya diatur sedemikian rupa sehingga jika neuron-neuron yang merepresentasikan kebenaran semua fakta pada badan aturan aktif maka neuron yang merepresentasikan kebenaran suatu fakta pada kepala aturan juga aktif.

Pendekatan representasi lokal ini diambil agar makna dari keaktifan neuron-neuron tertentu pada proses inferensi mudah dipahami. Hal ini tentu memiliki peran yang penting pada proses pemberian argumentasi.

4.2. Fungsi Aktivasi

Fungsi aktivasi untuk setiap neuron adalah fungsi tangga (*step function*) berikut ini:

$$f(x) = \begin{cases} 1 & \text{jika } x \geq 1 \\ 0 & \text{jika } x < 1 \end{cases} \quad (1)$$

Berdasarkan fungsi aktivasi di atas, jika nilai aktivasi suatu neuron lebih besar atau sama dengan satu, maka neuron tersebut akan menjadi aktif dan mengirimkan nilai satu ke semua neuron tujuannya. Sebaliknya jika nilai aktivasi suatu neuron lebih kecil dari satu, maka tidak ada aktivitas apapun yang dilakukan oleh neuron tersebut.

Pada JST nonstatistik ini, setelah suatu neuron aktif, nilai aktivasi dari neuron tersebut akan menjadi minus tak hingga. Neuron ini disebut dengan neuron mati karena neuron tersebut tidak akan pernah dapat mengirimkan suatu nilai ke neuron lain. Singkatnya, setiap neuron hanya dapat aktif satu kali pada setiap proses inferensi.

5. ARSITEKTUR

5.1. Jenis-jenis Neuron

Setiap neuron pada JST untuk klausa *definite* dapat dibagi menjadi dua jenis:

Table 1 Jenis-jenis neuron

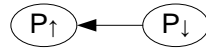
No.	Nama	Simbol	Aktif Jika ...
1.	Neuron maju	P_{\uparrow}	Pada fakta P sedang dilakukan penalaran maju
2.	Neuron mundur	P_{\downarrow}	Pada fakta P sedang dilakukan penalaran mundur

Perlu dicatat bahwa tidak ada perbedaan karakteristik dari kedua jenis neuron di atas. Pembagian jenis-jenis neuron di atas dilakukan hanya untuk membedakan fungsi (dan bukan spesifikasi) dari setiap neuron untuk mempermudah pemahaman akan proses inferensi yang sedang terjadi.

Penalaran mundur yang sedang dilakukan pada suatu fakta P memiliki arti bahwa pada fakta P sedang dilakukan proses inferensi untuk membuktikan kebenaran dari fakta tersebut. Kebalikannya, penalaran maju yang sedang dilakukan pada fakta P memiliki arti bahwa fakta ini telah terbukti kebenarannya. Hal penting yang perlu dicatat di sini adalah bahwa semua neuron maju yang telah menjadi neuron mati (yang berarti pernah aktif) merupakan jalan yang dilalui untuk membuktikan kebenaran suatu fakta. Oleh karena itu, semua fakta yang direpresentasikan oleh neuron ini merupakan calon komponen dari fitur pemberian argumentasi. Lebih lanjut mengenai bagaimana kedua jenis neuron di atas digunakan pada proses penalaran mundur dituliskan pada bagian 4.4.

5.2. Arsitektur untuk Fakta

Arsitektur untuk suatu fakta klausa *definite* P ditunjukkan pada gambar 1 di bawah. Sebagai konvensi pada makalah ini, sisi tanpa label bobot diartikan sebagai sisi dengan bobot satu.

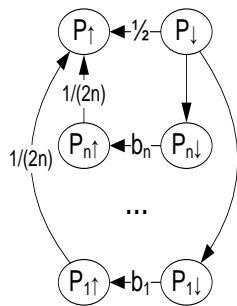


Gambar 1 Arsitektur JST untuk fakta klausa *definite*

Pada arsitektur di atas, jika akan dibuktikan bahwa fakta P benar, maka neuron mundur P akan menjadi aktif. Keaktifan neuron mundur P ini akan menyebabkan neuron maju P menerima nilai aktivasi satu sehingga neuron ini menjadi aktif. Singkatnya, arsitektur di atas memiliki makna bahwa fakta P benar.

5.3. Arsitektur untuk Aturan

Arsitektur untuk aturan $P_1 \wedge P_2 \wedge \dots P_n \Rightarrow P$ ($n \in \text{Asli}$ dan $n > 0$) ditunjukkan pada gambar di bawah ini:



Gambar 2 Arsitektur JST untuk aturan klausa *definite*

Bobot b_i ($i = 1..n$) pada sisi dari neuron mundur P_i ke neuron maju P_i untuk suatu basis pengetahuan BP tertentu adalah sebagai berikut:

$$b_i = \begin{cases} 1 & \text{jika } P_i \text{ merupakan fakta pada BP} \\ 1/2 & \text{jika } P_i \text{ bukan merupakan fakta pada BP} \end{cases} \quad (2)$$

Makna dari arsitektur JST di atas adalah bahwa jika akan dibuktikan bahwa fakta P benar, maka perlu dibuktikan bahwa semua fakta P_i juga benar. Dengan kata lain jika satu atau lebih fakta P_i tidak dapat dibuktikan benar, maka fakta P juga tidak dapat dibuktikan benar.

Perlu dicatat bahwa arsitektur JST di atas hanya benar jika aturan $P_1 \wedge P_2 \wedge \dots P_n \Rightarrow P$ ($n > 1$) merupakan satu-satunya aturan, dengan jumlah fakta pada badan aturan lebih dari satu, yang mendeduksi P. Jika ada dua atau lebih aturan seperti ini, maka kebenaran fakta pada satu aturan dapat bercampur-aduk dengan kebenaran fakta dari aturan lain sehingga dapat mendeduksi suatu fakta yang

seharusnya tidak terdeduksi. Sebagai contoh, dengan arsitektur di atas untuk aturan $P_1 \wedge P_2 \Rightarrow P$ dan $P_3 \wedge P_4 \Rightarrow P$, maka P akan terdeduksi jika P_1 dan P_3 benar walaupun P_2 dan P_4 tidak dapat dibuktikan benar.

Untuk mengatasi hal ini, jika ada lebih dari satu aturan yang mendeduksi P berikut ($n \in \text{Asli}$, $n > 1$, $n_i \in \text{Asli}$, dan $n_i > 1$ [$i = 1..n$]):

$$P_{1,1} \wedge P_{1,2} \wedge \dots P_{1,n_1} \Rightarrow P$$

$$P_{2,1} \wedge P_{2,2} \wedge \dots P_{2,n_2} \Rightarrow P$$

...

$$P_{n,1} \wedge P_{n,2} \wedge \dots P_{n,n_n} \Rightarrow P,$$

maka konversikan setiap aturan di atas masing-masing menjadi dua aturan seperti di bawah ini:

$$P_{1,1} \wedge P_{1,2} \wedge \dots P_{1,n_1} \Rightarrow P_1 \text{ dan } P_1 \Rightarrow P$$

$$P_{2,1} \wedge P_{2,2} \wedge \dots P_{2,n_2} \Rightarrow P_2 \text{ dan } P_2 \Rightarrow P$$

...

$$P_{n,1} \wedge P_{n,2} \wedge \dots P_{n,n_n} \Rightarrow P_n \text{ dan } P_n \Rightarrow P,$$

dengan P_i ($i = 1..n$) merupakan fakta baru yang dibuat untuk keperluan konversi ini. Dari hasil konversi di atas, dapat dipastikan bahwa setiap aturan $\alpha \Rightarrow P$ untuk sembarang fakta P dan sembarang konjungsi fakta α di mana terdapat lebih dari satu fakta pada α merupakan satu-satunya aturan yang mendeduksi P.

5.4. Proses Inferensi

Urutan langkah-langkah proses penalaran mundur pada JST untuk membuktikan kebenaran suatu fakta (sebut sebagai fakta kueri) menurut suatu basis pengetahuan dalam bentuk klausa *definite* logika proposisi tertentu adalah sebagai berikut:

1. Lakukan konversi pada aturan (jika diperlukan) agar tidak ada suatu fakta yang terdeduksi oleh lebih dari satu aturan yang memiliki lebih dari satu fakta pada badan aturan.
2. Bangun JST untuk basis pengetahuan yang bersangkutan dengan cara seperti yang telah dituliskan pada bagian 4.3.
3. Inisialisasi nilai aktivasi awal sebagai berikut:
 - a. Neuron mundur dari fakta kueri diinisialisasi dengan nilai satu.
 - b. Neuron lainnya diinisialisasi dengan nilai nol.
4. Aktifkan JST sampai dengan satu dari dua kondisi berikut terpenuhi:
 - a. Neuron maju dari fakta kueri aktif.
 - b. Tidak ada neuron baru yang aktif.

Dari langkah-langkah di atas, jelas bahwa neuron mundur dari fakta kueri merupakan neuron yang pertama kali aktif. Selanjutnya penalaran mundur akan mencari lintasan untuk mengaktifkan neuron mundur dari fakta pada basis pengetahuan. Jika berhasil, maka dapat disimpulkan bahwa fakta kueri terbukti benar. Untuk keperluan pemberian argumentasi, proses inferensi terus dilanjutkan dengan melakukan penalaran maju guna mencari

lintasan dari fakta pada basis pengetahuan ke fakta kueri. Fakta-fakta dan aturan-aturan yang membawa kepada kebenaran fakta kueri inilah yang merupakan argumentasi mengapa fakta kueri ini benar. Secara garis besar proses inferensi di atas melakukan pencarian pertama melebar (*breadth first search*) secara paralel.

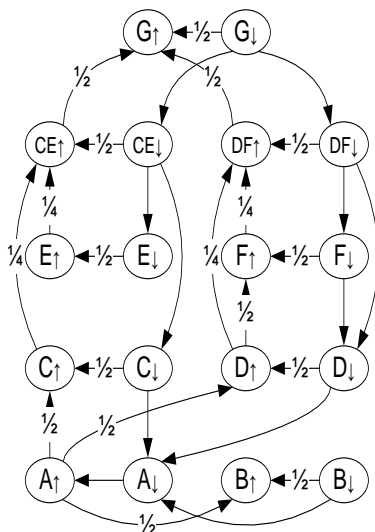
Berikut akan diberikan contoh bagaimana JST dapat melakukan inferensi dan memberikan argumentasi untuk membuktikan kebenaran dari fakta G pada basis pengetahuan di bawah ini:

$$\{A \Rightarrow B, A \Rightarrow C, A \Rightarrow D, C \wedge E \Rightarrow G, D \Rightarrow F, D \wedge F \Rightarrow G\}$$

Pada basis pengetahuan di atas, terdapat dua aturan, dengan jumlah fakta pada badan aturan lebih dari satu, yang mendeduksi G, yaitu aturan $C \wedge E \Rightarrow G$ dan $D \wedge F \Rightarrow G$. Pertama-tama kedua aturan ini dikonversi menjadi empat aturan berikut:

$$\{C \wedge E \Rightarrow CE, CE \Rightarrow G, D \wedge F \Rightarrow DF, DF \Rightarrow G\}$$

JST untuk basis pengetahuan hasil konversi di atas ditunjukkan pada gambar di bawah ini:



Gambar 3 Contoh JST untuk penalaran mundur pada klausa *definite*

Langkah berikutnya adalah menginisialisasi nilai aktivasi dari neuron mundur G dengan nilai satu dan neuron lainnya dengan nilai nol. Selanjutnya neuron mundur G ini menjadi aktif dan mengaktifkan neuron-neuron lainnya sampai dengan neuron maju G menjadi aktif. Keaktifan dari neuron maju G ini mengakhiri proses inferensi dengan kesimpulan bahwa fakta G terbukti benar.

Pada akhir proses inferensi, neuron-neuron maju yang pernah aktif adalah neuron maju A, C, D, F, DF dan G. Semua neuron ini (kecuali neuron C) merepresentasikan fakta-fakta yang benar yang membawa kepada kebenaran fakta G dan digunakan sebagai komponen utama dari pemberian

argumentasi. Argumentasi dari kesimpulan bahwa fakta G terbukti benar adalah sebagai berikut:

- Fakta G benar karena fakta D dan F benar (neuron maju D dan F pernah aktif) dan aturan $D \wedge F \Rightarrow G$ ada pada basis pengetahuan.
- Fakta D benar karena fakta A benar (neuron maju A pernah aktif) dan aturan $A \Rightarrow D$ ada pada basis pengetahuan.
- Fakta F benar karena fakta D benar (neuron maju D pernah aktif) dan aturan $D \Rightarrow F$ ada pada basis pengetahuan.
- Fakta A benar karena fakta ini ada pada basis pengetahuan.

Begitulah salah satu cara JST dapat melakukan penalaran mundur dan memberikan argumentasi terhadap kesimpulan yang diperolehnya. Bagaimana model yang dituliskan pada bagian ini dapat dikembangkan sehingga mencakup juga klausa *definite* FOL dituliskan pada bagian selanjutnya berikut ini.

6. PENANGANAN VARIABEL

Kelebihan FOL dibandingkan logika proposisi adalah kemampuannya untuk merepresentasikan objek dan relasi antar objek. FOL dapat digunakan untuk mengekspresikan fakta tentang beberapa atau semua objek secara fleksibel. Hal ini dimungkinkan dengan adanya variabel pada FOL yang dapat digunakan untuk merepresentasikan sembarang objek.

Pendekatan untuk penanganan variabel pada JST nonstatistik dapat dibedakan menjadi dua, yaitu proposionalisasi dan unifikasi. Pada makalah ini hanya dituliskan prinsip dasar bagaimana JST nonstatistik dapat digunakan untuk merepresentasikan kalimat-kalimat klausa *definite* FOL.

6.1. Proposionalisasi

Proposionalisasi merupakan teknik untuk mengkonversi kalimat-kalimat FOL ke dalam kalimat-kalimat logika proposisi. Pada klausa *definite* yang mengasumsikan semua variabel terkuantifikasi universal, prinsip dasar dari proposionalisasi adalah menduplikasi fakta dan aturan yang mengandung variabel dengan semua kemungkinan instansiasi dari fakta dan aturan tersebut. Instansiasi suatu fakta dan aturan didapat dengan melakukan substitusi dari variabel pada fakta dan aturan tersebut dengan konstanta yang valid. Jika tidak dibatasi, maka konstanta yang valid adalah semua konstanta yang muncul pada basis pengetahuan yang bersangkutan.

Sebagai contoh, basis pengetahuan dalam bentuk klausa *definite* FOL di bawah ini:

$$\{ \text{Burung}(\text{Merpati}), \text{Burung}(\text{Rajawali}), \text{Binatang}(\text{Anjing}), \text{Binatang}(\text{Kucing}), \text{Burung}(x) \Rightarrow \text{Binatang}(x) \},$$

dapat dikonversi ke basis pengetahuan dalam bentuk logika proposisi berikut:

{ Burung(Merpati), Burung(Rajawali),
 Binatang(Anjing), Binatang(Kucing),
 Burung(Merpati) \Rightarrow Binatang (Merpati),
 Burung(Rajawali) \Rightarrow Binatang (Rajawali),
 Burung(Anjing) \Rightarrow Binatang (Anjing),
 Burung(Kucing) \Rightarrow Binatang (Kucing) }

Terlihat pada contoh ini bahwa basis pengetahuan hasil konversi berukuran jauh lebih besar dari basis pengetahuan asalnya. Dengan aturan konversi di atas, diperlukan duplikasi fakta atau aturan yang bersangkutan sebanyak n^k kali untuk setiap fakta atau aturan yang memiliki k variabel dan n kemungkinan substitusi untuk variabel tersebut.

Kelemahan ini tidak seburuk seperti apa yang tampak. Nilai k yang dibutuhkan biasanya relatif kecil. Telah ada suatu *parser* yang dapat memproses sejumlah besar variasi kalimat bahasa Inggris hanya dengan menggunakan fakta dan aturan yang mengandung maksimal sembilan variabel [3; 11, hal. 3]. Sedangkan nilai n dapat dibatasi dengan mendefinisikan tipe variabel dari suatu predikat. Sebagai contoh, variabel x pada predikat Burung dapat didefinisikan bertipe $T_{Burung} = \{Merpati, Rajawali\}$. Dengan demikian substitusi $\{x / Anjing\}$ dan $\{x / Kucing\}$ bukan lagi merupakan substitusi yang valid pada predikat Burung.

Cara lain untuk mengoptimasi teknik proposionalisasi ini adalah dengan melakukan *filtering* terhadap instansiasi yang mungkin dibangkitkan. Prinsip dasarnya adalah melakukan substitusi yang valid terhadap fakta-fakta pada badan aturan dengan fakta-fakta yang telah terdeduksi menurut basis pengetahuan yang bersangkutan. Predikat pada badan aturan hasil substitusi ini merupakan fakta baru yang terdeduksi. Langkah ini dilakukan terus-menerus sampai dengan tidak ada fakta baru yang terdeduksi. Selanjutnya semua aturan yang masih mengandung variabel dihapuskan dari basis pengetahuan yang bersangkutan.

Sebagai contoh perhatikan basis pengetahuan berikut:

{ Anak(P,Q), Anak(R,P), Anak(P,S),
 Anak(T,P), Anak(R,U),
 Anak(x,z) \wedge Anak(z,y) \Rightarrow Cucu(x,y) }

Hanya ada satu aturan pada basis pengetahuan di atas. Langkah-langkah untuk mengkonversi aturan tersebut menjadi beberapa aturan tanpa variabel adalah sebagai berikut:

- Pertama-tama predikat Anak(x,z) disubstitusi dengan fakta pertama, yaitu Anak(P,Q). Dari sini variabel x terikat dengan konstanta P dan variabel z terikat dengan konstanta Q.

- Selanjutnya dicari apakah ada substitusi untuk predikat Anak(Q,y). Karena tidak ada, maka dicoba melakukan substitusi untuk predikat Anak(x,z) dengan fakta berikutnya, yaitu Anak(R,P).
- Kemudian dicari apakah ada substitusi untuk predikat Anak(P,y) yang ternyata ada, yaitu Anak(P,Q) dan Anak(P,S) sehingga diperoleh dua aturan tanpa variabel berikut:

{ Anak(R,P) \wedge Anak(P,Q) \Rightarrow Cucu(R,Q),
 Anak(R,P) \wedge Anak(P,S) \Rightarrow Cucu(R,S) }

- Begitu seterusnya sampai dengan diperoleh basis pengetahuan final yang tidak mengandung variabel berikut ini:

{ Anak(P,Q), Anak(R,P), Anak(P,S),
 Anak(T,P), Anak(R,U),
 Anak(R,P) \wedge Anak(P,Q) \Rightarrow Cucu(R,Q),
 Anak(R,P) \wedge Anak(P,S) \Rightarrow Cucu(R,S),
 Anak(T,P) \wedge Anak(P,Q) \Rightarrow Cucu(T,Q),
 Anak(T,P) \wedge Anak(P,S) \Rightarrow Cucu(T,S) }

Terlihat bahwa hanya diperlukan empat duplikasi aturan dengan cara ini. Ini tentu jauh lebih efisien dibandingkan dengan menduplikasikan aturan dengan semua kombinasi substitusi lima konstanta ke tiga variabel yang mungkin, yaitu sebanyak $5^3 = 125$ duplikasi aturan.

Perlu dicatat bahwa tidak semua kalimat FOL dapat dikonversi ke dalam kalimat logika proposisi. Hal ini disebabkan adanya simbol fungsi pada FOL yang dapat mengakibatkan instansiasi dari suatu predikat berjumlah tak hingga. Keterbatasan ini tidak terlalu mengejutkan mengingat masalah penentuan kebenaran suatu fakta pada FOL adalah semi-decidable seperti yang telah disebutkan sebelumnya. Untungnya ada suatu teorema terkenal dari Herbrand [15] yang mengatakan bahwa suatu basis pengetahuan dalam FOL membawa kepada kebenaran suatu kalimat jika dan hanya jika kebenaran kalimat tersebut juga dibawakan oleh subhimpunan terbatas dari hasil konversi basis pengetahuan tersebut ke logika proposisi. Dengan kata lain, jika suatu fakta dapat dibuktikan benar menurut suatu basis pengetahuan FOL, maka fakta tersebut juga dapat dibuktikan benar menurut hasil konversi basis pengetahuan FOL tersebut ke logika proposisi.

6.2. Unifikasi

Seperti pada proposionalisasi, pada unifikasi juga dilakukan substitusi untuk menghasilkan instansiasi dari suatu fakta. Perbedaannya, unifikasi melakukan substitusi hanya jika diperlukan. Suatu variabel yang telah disubstitusi dengan suatu konstanta dikatakan terikat oleh konstanta tersebut selama proses inferensi berlangsung atau sampai dengan proses backtracking dilakukan.

Pendekatan penanganan pengikatan variabel (variable binding) pada JST dapat dibagi menjadi dua macam, yaitu propagasi signal (sign propagation) [13] dan sinkronisasi fase (phase synchronization) [12]. Karena keterbatasan tempat, hanya prinsip-prinsip dasar dari kedua pendekatan ini yang dituliskan pada makalah ini. Pembahasan detail mengenai kedua pendekatan ini dapat dilihat pada referensi-referensi yang diberikan.

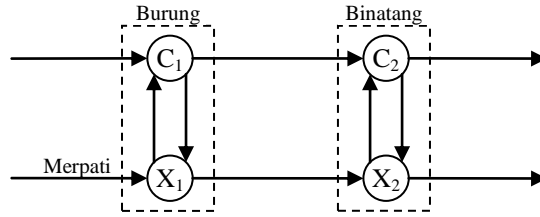
• **Propagasi Signal**

Pada pendekatan ini, setiap variabel pada suatu predikat direpresentasikan dengan satu neuron. Suatu signal unik berupa bilangan nyata positif diasosiasikan dengan sebuah konstanta. Jika suatu variabel tidak terikat, maka neuron yang merepresentasikan variabel tersebut memiliki nilai aktivasi negatif. Sebaliknya jika suatu variabel terikat dengan suatu konstanta tertentu, maka nilai aktivasi dari neuron yang merepresentasikan variabel tersebut adalah signal dari konstanta yang bersangkutan. Selanjutnya jika suatu variabel muncul pada bagian badan dan kepala suatu aturan, maka neuron yang merepresentasikan variabel pada bagian badan aturan memiliki sisi ke neuron yang merepresentasikan variabel pada bagian kepala aturan. Sisi inilah yang digunakan untuk proses unifikasi variabel pada suatu aturan dalam bentuk propagasi signal dari satu neuron ke neuron lain. Perlu dicatat bahwa proses unifikasi berupa propagasi signal ini dapat juga terjadi antar dua atau lebih variabel pada badan aturan.

Setiap predikat dengan k variabel dipresentasikan dengan $(k + 1)$ neuron, yaitu satu neuron untuk merepresentasikan satu variabel (disebut dengan neuron X) dan satu neuron untuk merepresentasikan satu predikat (disebut dengan neuron C). Neuron C memiliki sisi dari dan ke setiap neuron X pada suatu predikat. Sisi-sisi ini digunakan untuk memastikan bahwa neuron X aktif jika dan hanya jika neuron C aktif. Dengan kata lain suatu variabel dapat aktif jika predikat dari variabel tersebut sedang dieksplor pada proses inferensi, dan suatu predikat dapat aktif jika semua variabel pada predikat tersebut tidak terikat atau terikat dengan konstanta yang sesuai. Sebagai contoh, perhatikan aturan dalam klausa *definite* FOL berikut:

$$\text{Burung}(x) \Rightarrow \text{Binatang}(x)$$

Arsitektur dasar dari JST yang merepresentasikan aturan di atas untuk masukan $\text{Burung}(\text{Merpati})$ adalah sebagai berikut:



Gambar 4 JST propagasi signal untuk aturan $\text{Burung}(x) \Rightarrow \text{Binatang}(x)$

Pada arsitektur di atas, jika neuron C_1 aktif dan neuron X_1 menerima suatu signal bilangan nyata yang merepresentasikan konstanta Merpati , maka pertama-tama akan dilakukan pemeriksaan apakah dapat dilakukan unifikasi terhadap nilai aktivasi pada neuron X_1 dan signal yang diterimanya. Jika dapat, maka neuron X_1 akan aktif dan mengirimkan signal tersebut ke neuron X_2 .

Karakteristik ini menuntut neuron X untuk dapat berperilaku lebih dari sekedar menjumlahkan nilai masukan dan mengaplikasikan fungsi aktivasi seperti pada JST konvensional. Singkatnya, jika suatu neuron X yang merepresentasikan variabel x menerima suatu signal, maka neuron X akan mengirimkan signal tersebut ke neuron tujuannya jika salah satu dari dua kondisi berikut terpenuhi:

1. Signal yang diterima bernilai sama dengan nilai aktivasi neuron X (yang berarti variabel x terikat dengan nilai yang sama);
2. Neuron X memiliki nilai aktivasi negatif (yang berarti variabel x belum terikat).

Pada kasus terakhir, neuron X akan mengubah nilai aktivasinya menjadi bilangan nyata pada signal tersebut (yang berarti variabel x diikat dengan konstanta yang direpresentasikan oleh signal tersebut).

Kelemahan dari pendekatan ini adalah keterbatasannya dalam melakukan instansiasi banyak predikat (*multiple instantiation*) secara paralel. Pada JST, proses inferensi dituntut untuk dilakukan secara paralel yang artinya semua unifikasi yang mungkin terhadap suatu variabel harus dapat dilakukan secara simultan (tidak ada *backtracking* seperti pada proses inferensi sekuensial). Untuk dapat melakukan hal ini, perlu ada duplikasi dari neuron X untuk setiap variabel. Sayangnya jumlah duplikasi yang dibutuhkan tidak mudah untuk diperkirakan dari awal. Pendekatan ini biasanya melakukan pembatasan terhadap jumlah instansiasi yang dapat dilakukan secara simultan. Batasan ini tentu akan mengurangi kemampuan inferensi dari JST yang bersangkutan.

• **Sinkronisasi Fase**

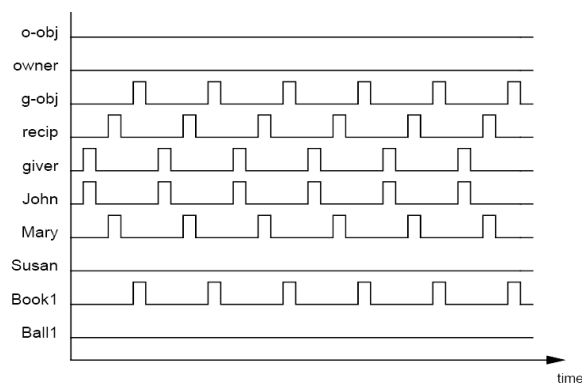
Pada pendekatan ini, setiap variabel dan konstanta direpresentasikan masing-masing dengan satu neuron. Pengikatan variabel dengan konstanta

direpresentasikan dengan neuron-neuron dari variabel dan konstanta tersebut yang aktif dengan fase yang sinkron. Untuk menghindari adanya *cross-talk*, setiap variabel dan konstanta yang sedang aktif memiliki fase keaktifan yang unik terhadap fase keaktifan variabel dan konstanta lainnya.

Sebagai contoh, perhatikan aturan dalam klausa *definite FOL* berikut:

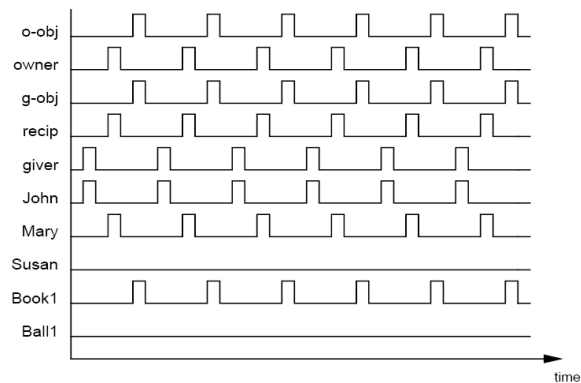
$$\text{Give}(x, y, z) \Rightarrow \text{Own}(y, z)$$

Misalkan konstanta yang valid pada basis pengetahuan adalah Ball1, Book1, Susan, Mary dan John. Jika diberikan fakta $\text{Give}(\text{John}, \text{Mary}, \text{Book1})$, maka pertama-tama neuron dari variabel x (sebut sebagai *giver*) dan konstanta John, dari variabel y (sebut sebagai *recip*) dan konstanta Mary, serta dari variabel z (sebut sebagai *g-obj*) dan konstanta Book1 akan aktif dengan fase yang sinkron seperti ditunjukkan pada gambar berikut [12, hal.14]:



Gambar 6 Pengikatan variabel dengan sinkronisasi fase (giver = John, receipt = Mary, g-obj = Book1)

Selanjutnya keaktifan variabel y dan z pada predikat Give akan menyebabkan variabel y dan z pada predikat Own juga menjadi aktif. Artinya fase keaktifan dari neuron yang merepresentasikan variabel y pada predikat Own (sebut sebagai *owner*) akan sinkron dengan neuron *recip* dan neuron John. Demikian juga halnya dengan fase keaktifan dari neuron yang merepresentasikan variabel z pada predikat Own (sebut sebagai *o-obj*) akan sinkron dengan neuron *g-obj* dan konstanta Book1. Hasil akhirnya ditunjukkan pada gambar di bawah ini [12, hal. 14]:



Gambar 5 Pengikatan variabel dengan sinkronisasi fase (giver = John, receipt = Mary, g-obj = Book1, owner = Mary, o-obj = Book1)

Pendekatan sinkronisasi fase dari dua atau lebih neuron sebagai sarana komunikasi dan informasi ini jelas tidak terdapat pada JST konvensional. Namun beberapa data pada riset neuropsikologi menunjukkan bahwa terdapat sinkronisasi aktivitas neuron pada otak yang kemungkinan memainkan peranan tertentu dalam pemrosesan informasi [10, hal. 2]. Walaupun pendapat ini masih kontroversial, bagaimanapun juga pendekatan ini terinspirasi oleh hasil riset pada otak sehingga model dari pendekatan ini masih layak untuk disebut sebagai JST.

Seperti pada pendekatan propagasi signal, pendekatan ini memiliki keterbatasan dalam melakukan instansiasi banyak predikat. Terdapat batasan dari jumlah instansiasi yang dapat dilakukan secara simultan pada pendekatan ini [14]. Batasan ini tentu akan mengurangi kemampuan inferensi dari JST yang bersangkutan.

7. PENUTUP

7.1. Kesimpulan

Pada makalah ini telah diperlihatkan bagaimana JST dapat melakukan penalaran mundur pada klausa *definite logika* proposisi maupun FOL dan memberikan argumentasi terhadap kesimpulan yang diperolehnya. Model JST pada makalah ini merupakan model inferensi yang *massively parallel* sehingga dapat melakukan penalaran secara efisien jika dijalankan pada sistem komputer paralel.

Dibandingkan dengan pendekatan sekuensial, proses penalaran yang dilakukan JST pada klausa *definite logika* proposisi tidak memiliki kelemahan tertentu yang signifikan. Namun sayangnya hal yang sama tidak berlaku pada klausa *definite FOL*. Sampai dengan saat ini penalaran pada JST untuk klausa *definite FOL* masih memiliki keterbatasan dalam hal jumlah variabel pada suatu fakta atau aturan, atau dalam hal jumlah instansiasi yang dapat dilakukan secara simultan. Namun tentu masih terbuka kemungkinan bahwa kelemahan ini akan dapat diatasi oleh hasil-hasil penelitian di masa mendatang.

7.2. Saran

Selain dari mengatasi keterbatasan-keterbatasan yang ada untuk penalaran pada klausa *definite* FOL di atas, hal-hal di bawah ini merupakan beberapa topik penelitian lanjutan dari makalah ini:

1. Penalaran pada JST dengan operator negasi (tidak dibatasi pada bentuk klausa *definite*) pada logika proposisi maupun FOL. Isu-isu yang berkaitan dengan hal ini adalah bagaimana mendeteksi adanya kontradiksi, penanganan model yang tidak unik dan bagaimana proses inferensinya dapat berjalan dalam waktu polinomial.
2. Penalaran nonmonotonik pada JST yang dapat melakukan penalaran pada basis pengetahuan yang mengandung kontradiksi.
3. Bagaimana teknik-teknik pembelajaran yang ada dapat digunakan bersama-sama dengan JST untuk mengakuisisi pengetahuan dalam bentuk logika formal secara otomatis.

8. DAFTAR PUSTAKA

1. **A. Darbari**, "Rule Extraction from Trained ANN: A Survey, Technical report Institut of Artificial intelligence", Dept. of Comp. Science, TU Dresden, 24 Juli 2001, <http://overcite.lcs.mit.edu/rd/1178249%2C554343%2C1%2C0.25%2CDownload/http%3AqSqqSqwww.wv.inf.tu-dresden.deqSqPublicationsqSq2000qSqvw-00-03.ps.gz>, diakses tanggal 1 Desember 2005.
2. **L. Fausett**, "Fundamental of Neural Networks", Prentice Hall, Inc., New Jersey, 1994.
3. **J. Henderson**, "Connectionist Syntactic Parsing Using Temporal Variable Binding", Journal of Psycholinguistic Research, 24 (5), 335–379, 1994, <http://citeseer.ist.psu.edu/rd/0%2C23577%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/2450/http:zSzzSzwww.dcs.ex.ac.ukzSz%7EjamiezSzjpr94.pdf/connectionist-syntactic-parsing-using.pdf>, diakses tanggal 3 Desember 2005.
4. **J. E. Hopcroft, R. Motwani, and J. D. Ullman**, "Introduction to Automata Theory, Languages, and Computation", 2nd Edition, Addison-Wesley, United States, 2001.
5. **B. Muller and J. Reinhardt**, "Neural Networks: an Introduction", Springer-Verlag Berlin Heidelberg, Jerman, 1990.
6. **S. Rusell and P. Norvig**, "Artificial Intelligence: A Modern Approach", Prentice Hall, Inc., New Jersey, 1995.
7. **S. Rusell and P. Norvig**, "Artificial Intelligence: A Modern Approach", 2nd Edition, Pearson Education, Inc., New Jersey, 2003.
8. **W. S. Sarle**, "Neural Networks and Statistical Models", Proceedings of the Nineteenth Annual SAS Users Group International Conference, Cary, NC: SAS Institute, pp 1538-1550, April, 1994, <ftp://ftp.sas.com/pub/neural/neural1.ps>, diakses tanggal 1 Desember 2005.
9. **Sestito & Dillon**, Automated Knowledge Acquisition, Prentice Hall, Australia, 1994.
10. **L. Shastri**, "Advances in Shruti – A Neurally Motivated Model of Relational Knowledge Representation and Rapid Inference using Temporal Synchrony", Applied Intelligence, 11, 79–108, 1999, http://www.icsi.berkeley.edu/%7Eshastri/psfiles/shruti_adv_98.pdf, diakses tanggal 25 Juli 2005.
11. **L. Shastri**, Temporal Synchrony, Dynamic Bindings, and SHRUTI – a representational but non-classical model of reflexive reasoning, Behavioral and Brain Sciences Vol. 19, No. 2, 331–337, 1996, <http://www.icsi.berkeley.edu/%7Eshastri/psfiles/contcom.ps.gz>, diakses tanggal 25 Juli 2005.
12. **L. Shastri and V. Ajjanagadde**, "From Simple Associations to Systematic Reasoning: A Connectionist Encoding of Rules, Variables and Dynamic Bindings using Temporal Synchrony", Behavioral and Brian Sciences, 16:3, 417–494, 1993, <http://www.icsi.berkeley.edu/%7Eshastri/psfiles/bbs.shastri.tar.gz>, diakses tanggal 25 Juli 2005.
13. **R. Sun**, "On Variable Binding in Connectionist Networks", Connection Science, Vol. 4, No. 2, 93–124, 1992, <http://www.cecs.missouri.edu/~rsun/sun.variable.ps>, diakses tanggal 8 Juli 2005.
14. **C. Wendelken and L. Shastri**, "Multiple Instantiation and Rule Mediation in SHRUTI", Connection Science, 16: 211–217, 2004, <http://www.icsi.berkeley.edu/%7Eshastri/psfiles/ConnSci04.doc>, diakses tanggal 29 Agustus 2005.
15. **M. Fitting**, "Bertrand Russels, Herbrand Theorem and The Assignment Statement", Dept. Mathematics and Computer Science, Lehman College (CUNY), Bronx, NY 10468, 1996, <http://math240.lehman.cuny.edu/fitting>, diakses tanggal 5 Desember 2005.