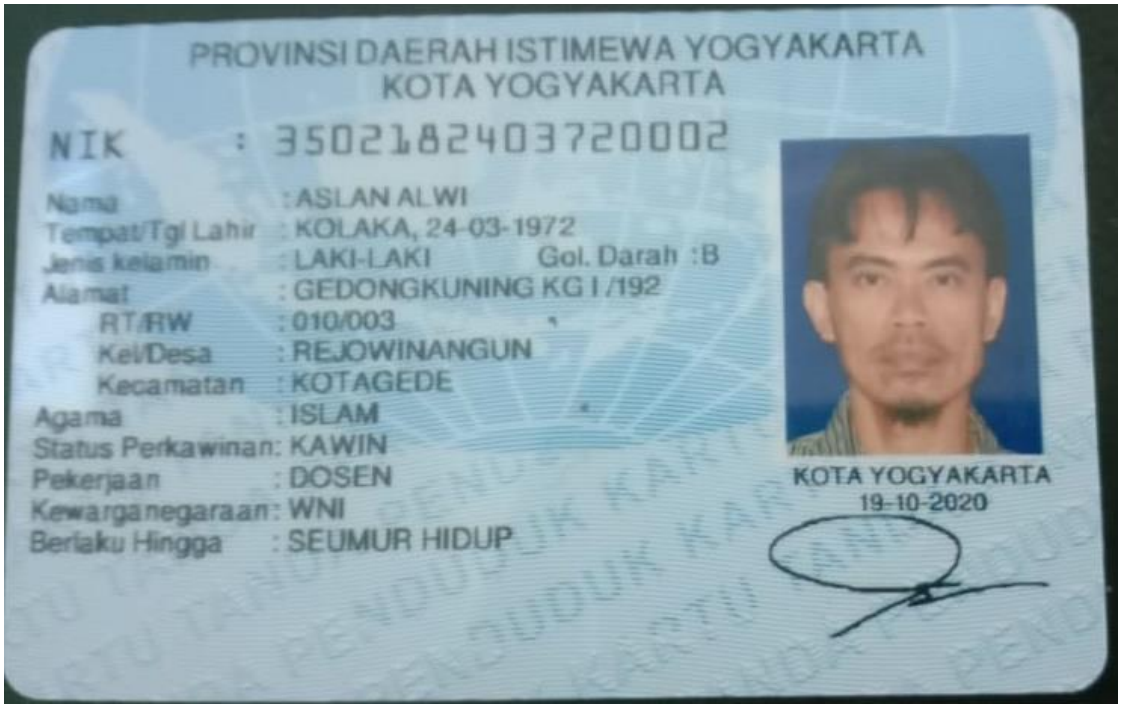
DOKUMEN HAK CIPTA

Algoritma Inferensi Forward Chaining Untuk Sistem Pakar

A. RINCIAN PENGUSUL HAK CIPTA

1. Nama : Dr. Aslan Alwi, M.Cs


PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KOTA YOGYAKARTA

NIK : 3502182403720002

Nama : ASLAN ALWI
Tempat/Tgl Lahir : KOLAKA, 24-03-1972
Jenis kelamin : LAKI-LAKI Gol. Darah : B
Alamat : GEDONGKUNING KG I /192
RT/RW : 010/003
Kel/Desa : REJOWINANGUN
Kecamatan : KOTAGEDE
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP

KOTA YOGYAKARTA
19-10-2020

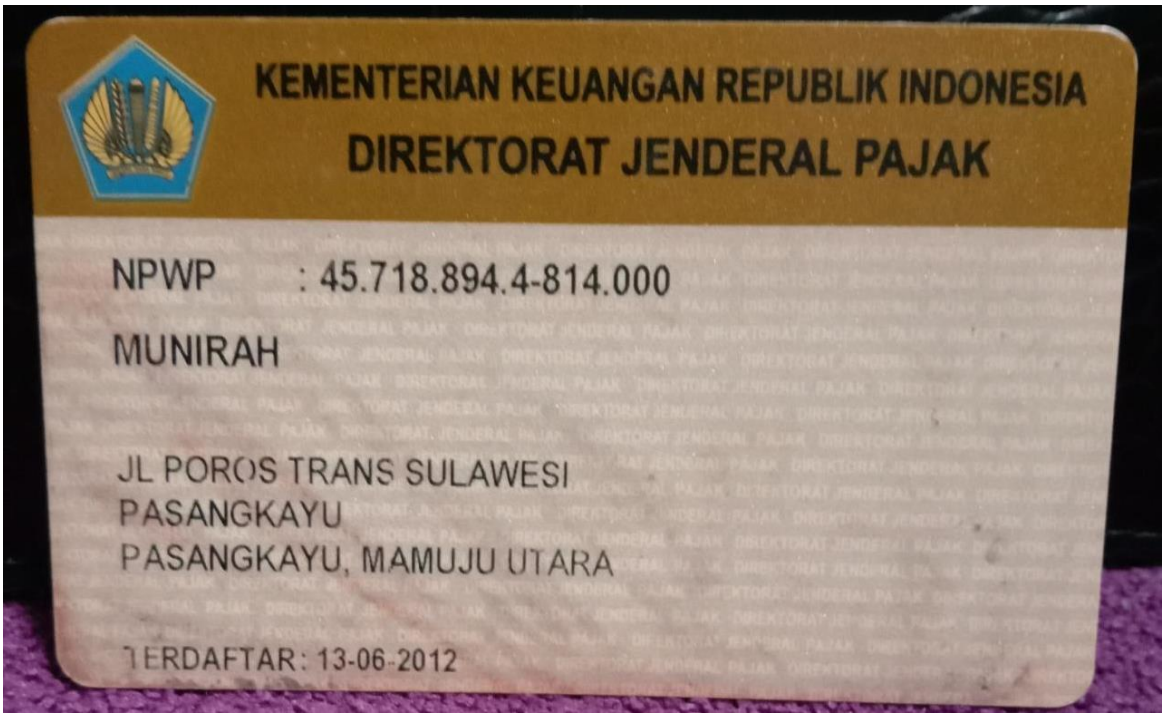

KEMENTERIAN KEUANGAN REPUBLIK INDONESIA
DIREKTORAT JENDERAL PAJAK

NPWP : 14.329.333.0-815.000
ASLAN ALWI
NIK : 3502182403720002

REPELITA NO 19 RT.0 RW.0, LATAMBAGA
LATAMBAGA LATAMBAGA
KAB. KOLAKA SULAWESI TENGGARA

KPP PRATAMA KOLAKA

2. Nama : Munirah M, S.Kom, M.T


KEMENTERIAN KEUANGAN REPUBLIK INDONESIA
DIREKTORAT JENDERAL PAJAK

NPWP : 45.718.894.4-814.000

MUNIRAH

JL POROS TRANS SULAWESI
PASANGKAYU
PASANGKAYU, MAMUJU UTARA

TERDAFTAR: 13-06-2012


PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KOTA YOGYAKARTA

NIK : 3471144711790001

Nama : MUNIRAH
Tempat/Tgl Lahir : JAKARTA
07-11-1979
Jenis kelamin : PEREMPUAN Gol. Darah : B
Alamat : GEDONGKUNING KG.I/192
RT/RW : 010/003
Kel/Desa : REJOWINANGUN
Kecamatan : KOTAGEDE
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP

KOTA YOGYAKARTA
28-01-2016

B. RINCIAN ALGORITMA INFERENSI

Bagian ini merinci algoritma yang diusulkan sebagai bagian utama dari pengusulan hak cipta. Rincian algoritma dimulai dengan sejumlah definisi terminologi yang digunakan di dalam menyatakan algoritma. Rincian itu adalah sebagai berikut:

Abstrak

Dokumen HKI ini mengusulkan sebuah mesin inferensi bagi sebuah sistem pakar atau secara umum sebuah sistem berbasis rule. Mesin inferensi dikonstruksikan menggunakan sebuah mesin turing, dimana nantinya diharapkan bahwa langkah-langkah di dalam mesin turing itu dapat dijadikan sebagai sebuah algoritma untuk mesin inferensi pada sebuah sistem pakar atau sebuah sistem berbasis rule. Rancangan mesin turing untuk berfungsi sebagai sebuah sistem inferensi terdiri dari 4 pita dan bekerja sebagaimana layaknya sebuah mesin turing universal. Bekerja sebagai sebuah mesin turing universal karena dia dapat mensimulasikan seluruh sistem pakar atau sistem berbasis rule yang anteseden dan konsekuen rule-rule nya ditulis dalam aturan konjungsi. Walau demikian, dia tidak dapat dikatakan benar-benar universal bagi sebarang sistem pakar karena masih adanya pembatasan bentuk ekspresi anteseden dan konsekuen pada rule-rule tersebut. Mesin turing yang digunakan untuk membangun mesin inferensi adalah bersifat deterministik, akan tetapi tidak menutup kemungkinan untuk memperluasnya menjadi nondeterministik sehingga dapat melakukan inferensi terhadap rule-rule yang ambigu dalam sistem pakar dan menyajikan pilihan-pilihan inferensi bagi user. Kompleksitas komputasi dari mesin turing rancangan ini adalah polinomial kuadratik, disebabkan karena pada mekanisme kerjanya, pencarian rule pada satu pita menghasilkan pencarian rule tersarang pada pita lain.

Kata Kunci

Algoritma, inferensi, sistem berbasis rule, sistem pakar, automata.

Pendahuluan

Lucas dan Gaag mengemukakan paradigma desain sistem pakar sebagai berikut:

Sistem pakar = pengetahuan + inferensi

- Akibatnya, sistem pakar biasanya terdiri dari dua komponen penting berikut: Sebuah basis pengetahuan menangkap pengetahuan domain-spesifik, dan
- Sebuah mesin inferensi yang terdiri dari algoritma untuk memanipulasi pengetahuan diwakili dalam basis pengetahuan. (Lucas & Gaag, 1991)

Memilih sebuah rule dari sejumlah rule untuk dieksekusi adalah persoalan resolusi konflik. Resolusi konflik dapat dibedakan antar satu dengan yang lain dalam beberapa cara, salah satunya adalah berdasarkan prioritas. Tiap-tiap rule diberi nilai prioritas (McDermott and Charles Lanny, 1976).

Beberapa jenis resolusi konflik yang lain (Pakiarajah and Crowther and Hartnett, 2000) adalah:

- *First in First Serve* – rule yang diterapkan adalah yang pertama cocok.
- *Last in First Serve* – rule yang diterapkan adalah yang terakhir cocok.
- *Specificity* - rule yang diterapkan adalah rule yang paling banyak faktanya cocok, dalam hal berarti lebih spesifik.
- *Recency* – rule yang diterapkan adalah rule yang paling sesuai dengan fakta yang diperoleh.

Pada Dokumen HKI ini, kami tidak memfokuskan diri pada bagaimana cara memecahkan masalah resolusi konflik ini, tetapi lebih kepada solusi umum tentang bagaimana membangun sebuah mesin inferensi. Dokumen HKI ini berusaha membangun sebuah mesin inferensi yang berupa mesin turing yang di dalamya dapat disusun berbagai langkah-langkah mesin yang mungkin, sedemikian rupa sehingga masalah resolusi konflik juga dapat ditulis di dalamnya semata sebagai langkah-langkah mesin turing.

Dokumen HKI ini dilatarbelakangi oleh usaha untuk membangun sebuah rule base system yang dapat mengeksekusi atau mensimulasikan sebuah automata, yaitu generalized finite state automata (gFSA) untuk keperluan membangun interpreter bagi sebuah sistem augmented reality. Tetapi, rule base system yang direncanakan untuk dibuat tersebut dapat mewakili juga sebarang sistem pakar. Terutama pada cara melakukan inferensi. Karena itu Dokumen HKI ini mengusulkan sebuah algoritma untuk inferensi bagi sebarang sistem pakar.

Algoritma ini dibuat dengan menggunakan mesin turing yang mensimulasikan seluruh rule-rule dari sistem pakar, sebagai mesin yang melakukan inferensi di dalam sistem pakar. Dalam cara pandang ini, semua rule yang ada dalam sistem pakar diletakkan sebagai sebuah barisan enkoding di dalam pita-pita mesin turing.

Input bagi mesin turing ini adalah setiap fakta yang diberikan oleh user atau setiap fakta yang dihasilkan oleh mesin turing itu sendiri ketika dia selesai satu langkah inferensi (simulasi rule).

Mesin turing ini mendasarkan inferensi pada jumlah fakta yang masuk sebagai simbol-simbol pada pola input, tidak berdasar pada urutan pertanyaan yang diajukan oleh sistem pakar kepada user. Mesin turing yang dipekerkenalkan pada Dokumen HKI ini bekerja lebih mirip sebuah metode inferensi forward chaining, akan tetapi, kita dapat membalik cara kerja ini sesuka kita hingga dia menjadi mirip cara kerja backward chaining, itu hanyalah masalah bagaimana kita menulis rule-rule sebagai barisan enkoding di dalam pita-pita mesin turing. Apakah kita menulisnya secara forward (if A then B), atau menulisnya secara backward (if B then A). Sehingga mesin turing ini bersifat universal terhadap cara-cara kita melakukan inferensi. Kita bahkan dapat menulis mesin turing ini secara nondeterministik (menyerupai sebuah mesin nondeterministik UTM) atau kita menulisnya secara probabilistik atau fuzzy. Sehingga ini memperluas kemampuan inferensi mesin turing.

Mesin Turing untuk Inferensi

Algoritma ini diusulkan secara formal dengan susunan assumsi dan definisi sebagai berikut :

Assumsi 1:

Rule ditulis dalam bentuk berikut: JIKA anteseden MAKA konsekuen

Assumsi 2:

Anteseden Rule-rule secara keseluruhan ditulis dalam bentuk normal "DAN". Yaitu berarti bahwa anteseden ditulis dalam kombinasi perangkai "DAN". Sedang bentuk konsekuensi rule tidak dibatasi.

Beberapa definisi dikemukakan untuk sebagai landasan dari algoritma ini diuraikan sebagai berikut:

Definisi 1:

Memori atribut true (memori at) adalah buffer untuk menyimpan seluruh atribut true.

Definisi 2:

Memori atribut false (memori af) adalah buffer untuk menyimpan seluruh atribut false.

Definisi 3:

Memori rule siap picu (memori rsp) adalah buffer untuk menyimpan seluruh rule yang belum dicoret.

Definisi 4:

Memori rule sudah coret (memori rsc) adalah buffer untuk menyimpan seluruh rule yang sudah dicoret. Dicoret karena sudah diperiksa ternyata memuat atribut false pada antesedennya.

Cara kerja metode ini dinyatakan dalam prosedur sebagai berikut:

Langkah 0:

Simpan seluruh rule dalam memori rsp.

Langkah 1:

Dari dalam memori rsp. Cari rule dengan anteseden terbanyak, lalu jadikan kandidat rule 1 pertama untuk di picu.

Langkah 2:

Cek true/false seluruh atribut dalam anteseden rule tersebut tadi. Jika ada false maka simpan atribut-atribut false dalam memori af lalu lanjut langkah 3a.
Jika seluruhnya true maka simpan seluruh atribut true dalam memori at lalu lanjut langkah 3b.

Langkah 3a:

Coret seluruh rule dalam memori rsp yang mengandung atribut false dalam memori af sehingga tersisa rule tanpa atribut false. Simpan seluruh rule coret dalam memori rsc. Simpan seluruh rule tersisa dalam memori rsp. Ulang Langkah 1 untuk himpunan rule dalam memori rsp.

Langkah 3b:

Jika seluruh atribut true maka picu rule tersebut. Lalu simpan seluruh atribut konsekuen dalam memori at. Coret rule tersebut dari memori rsp dan simpan ke memori rsc. Lanjut Langkah 4.

Langkah 4:

Dalam memori rsp, cari semua rule yang seluruh atribut2 antesedennya adalah subset memori at. Jika ADA maka:

Picu dan kumpulkan seluruh hasil berupa atribut-atribut konsekuen ke dalam memori at. Coret semua rule dari memori rsp yang sudah picu. Pindahkhan rule coret ke dalam memori rsc. Jika memori rsp belum kosong maka ulang Langkah 4.

Jika memori rsp kosong, maka kesimpulan adalah seluruh isi memori at. Kesimpulan akhir adalah seluruh atribut konsekuen yang dimasukkan paling terakhir.
ALGORITMA SELESAI.
Jika TIDAK ADA maka lanjut Langkah 5.

Langkah 5:

Ulang Langkah 1 pada seluruh rule dalam memori rsp yang tersisa.

C. DEMONSTRASI ALGORITMA

Untuk menunjukkan bagaimana algoritma ini bekerja, diberikan sebuah contoh yang mendemosntrasikan cara kerja algoritma.

Misalkan sebuah sistem berbasis rule memiliki sejumlah rule yang dibuat secara acak sebagai berikut:

Jika A dan B dan C maka D

Jika A dan C maka K

Jika D dan C maka L

Jika L dan K maka U

Jika U dan B maka S

Hendak dibuktikan dengan dua kombinasi kasus yang diambil acak bahwa:

1. Jika A=True, B=True, C=True, menurut penalarn logika maka penalaran harus sampai pada S=True. Apakah algoritma juga akan sampai pada S=True?
2. Jika A=True, B=False, C=True, menurut penalaran logika maka penalaran harus sampai pada K=True saja. Apakah algoritma juga akan sampai pada K=True saja, tetapi tidak yang lain?

Uji 1 penggunaan algoritma:

Misalkan A=True, B=True dan C=True. Apakah dengan menggunakan algoritma maka kita dapat sampai pada kesimpulan S = True? Uji coba dilakukan sebagai berikut:

Eksekusi rule sebagai berikut:

1. Langkah 0:

   Simpan seluruh rule dalam memori rsp.

   Hasil langkah 0:

   Memori rsp:
   Rule1: Jika A dan B dan C maka D
   Rule2: Jika A dan C maka K
   Rule3: Jika D dan C maka L
   Rule4: Jika L dan K maka U
   Rule5: Jika U dan B maka S

2. Langkah 1:

   Dari dalam memori rsp.

   Cari rule dengan anteseden terbanyak, lalu jadikan kandidat rule 1 pertama untuk di picu.

   Hasil langkah 1:

   Rule1: Jika A dan B dan C maka D

3. Langkah 2:

   Cek true/false seluruh atribut dalam anteseden rule 1 tadi. Jika ada false maka simpan atribut-atribut false dalam memori af lalu lanjut langkah 3a.

   Jika seluruhnya true maka simpan seluruh atribut true dalam memori at lalu lanjut langkah 3b.

   Hasil langkah 2:

   karena A=T, B=T dan C=T, sehingga seluruh atribut anteseden disimpan dalam memori at.

   Memori at:
   A, B, C

   Pindah langkah 3b.

4. Langkah 3b:

   Jika seluruh atribut true maka picu rule tersebut.

   Lalu simpan seluruh atribut konsekuen dalam memori at. Coret rule tersebut dari memori rsp dan simpan ke memori rsc. Lanjut Langkah 4.

   Hasil langkah 3b:

5. Langkah 4:

   Dalam memori rsp, cari semua rule yang seluruh atribut-atribut antesedennya adalah subset memori at. Jika ADA maka picu dan kumpulkan seluruh hasil berupa atribut-atribut konsekuen ke dalam memori at. Coret semua rule dari memori rsp yang sudah picu. Pindahkan rule coret ke dalam memori rsc. Jika memori rsp belum kosong maka ulang Langkah 4. jika memori rsp kosong, maka kesimpulan adalah seluruh isi memori at. Kesimpulan akhir adalah seluruh atribut konsekuen yang dimasukkan paling terakhir.

   ALGORITMA SELESAI. Jika TIDAK ADA maka lanjut langkah 5.

   Hasil langkah 4:

6. Perulangan ke-1 langkah 4:

   Karena memori rsp belum kosong maka langkah 4 diulangi lagi.

   Hasil langkah 4 perulangan :

7. Perulangan ke-2 langkah 4:

   Karena memori rsp belum kosong maka langkah 4 diulangi lagi.

   Hasil langkah 4 perulangan :

8. Langkah 4 terakhir:

   Karena memori rsp telah kosong, telah tercoret semua maka algoritma selesai dieksekusi.

Hasil algoritma :

Memori at:
A, B, C, D, K, L, U, S

Yaitu bahwa S=True, sesuai yang diharapkan.

Uji 2 penggunaan algoritma:

Misalkan A=True, B=False dan C=True. Dengan penalaran sederhana, mestilah penalaran hanya sampai kepada kesimpulan K=True. Tetapi harus juga dibuktikan bahwa algoritma harus sampai kepada kesimpulan K=True.

Eksekusi rule sebagai berikut:

1. Langkah 0:

   Simpan seluruh rule dalam memori rsp.

   Hasil langkah 0:

   Memori rsp:
   Rule1: Jika A dan B dan C maka D
   Rule2: Jika A dan C maka K
   Rule3: Jika D dan C maka L
   Rule4: Jika L dan K maka U
   Rule5: Jika U dan B maka S

2. Langkah 1:

   Dari dalam memori rsp.
   Cari rule dengan anteseden terbanyak, lalu jadikan kandidat rule 1 pertama ut di picu.

   Hasil langkah 1:

   Rule1: Jika A dan B dan C maka D

3. Langkah 2:

   Cek true/false seluruh atribut dalam anteseden rule 1 tadi. Jika ada false maka simpan atribut-atribut false ke dalam memori af lalu lanjut langkah 3a. Jika seluruhnya true maka simpan seluruh atribut true dalam memori at lalu lanjut langkah 3b.

4. Hasil langkah 2:

   Karena A=T, B=F dan C=T, maka simpan B ke dalam memori af.

   Memori af:
   B

Pindah langkah 3a.

5. Langkah 3a:

   Coret seluruh rule dalam memori rsp yang mengandung atribut false dalam memori af sehingga tersisa rule tanpa atribut false. Simpan seluruh rule coret dalam memori rsc. Simpan seluruh rule tersisa dalam memori rsp.

   Ulang Langkah 1 untuk himpunan rule dalam memori rsp.

   Hasil langkah 3a:

   Memori rsp:
   ~~Rule1: Jika A dan B dan C maka D~~
   Rule2: Jika A dan C maka K
   Rule3: Jika D dan C maka L
   Rule4: Jika L dan K maka U
   ~~Rule5: Jika U dan B maka S~~

   Memori rsc:
   Rule1: Jika A dan B dan C maka D
   Rule5: Jika U dan B maka S

   Memori af:
   B

6. Perulangan 1 langkah 1:

   Dari dalam memori rsp. Cari rule dengan anteseden terbanyak, lalu jadikan kandidat rule 1 pertama ut di picu.

   Hasil perulangan langkah 1:

   Rule2: Jika A dan C maka K

9. Perulangan langkah 2:

   Cek true/false seluruh atribut dalam anteseden rule 1 tadi. Jika ada false maka simpan atribut-atribut false ke dalam memori af lalu lanjut langkah 3a. Jika seluruhnya true maka simpan seluruh atribut true dalam memori at lalu lanjut langkah 3b.

   Hasil perulangan langkah 2:

   Karena A=T, C=T, maka simpan A dan C ke dalam memori af.

   Memori at:
   A,C.

   Pindah langkah 3b.

7. Langkah 3b:

   Jika seluruh atribut true maka picu rule tersebut. Lalu simpan seluruh atribut konsekuen dalam memori at. Coret rule tersebut dari memori rsp dan simpan ke memori rsc. Lanjut Langkah 4.

   Hasil langkah 3b:

8. Langkah 4:

   Dalam memori rsp, cari semua rule yang seluruh atribut-atribut antesedennya adalah subset memori at. Jika ADA maka picu dan kumpulkan seluruh hasil berupa atribut-atribut konsekuen ke dalam memori at. Coret semua rule dari memori rsp yang sudah picu. Pindahkhan rule coret ke dalam memori rsc.

   Jika memori rsp belum kosong maka ulang Langkah 4. jika memori rsp kosong, maka kesimpulan adalah seluruh isi memori at. Kesimpulan akhir adalah seluruh atribut konsekuen yang dimasukkan paling terakhir.

   ALGORITMA SELESAI. Jika TIDAK ADA maka lanjut langkah 5.

   Hasil langkah 4:

   Tidak ada rule yang seluruh antesedennya adalah subset dari memori at.

   Lanjut ke langkah 5.

9. Langkah 5:

   Ulang langkah 1 pada seluruh rule dalam memori rsp yang tersisa tidak dicoret.

10. Perulangan 1 langkah 1:

    Dari dalam memori rsp. Cari rule dengan anteseden terbanyak, lalu jadikan kandidat rule 1 pertama ut di picu.

Hasil perulangan langkah 1:

Rule2: Jika D dan C maka L.

11. Langkah 2:

    Cek true/false seluruh atribut dalam anteseden rule 1 tadi. Jika ada false maka simpan atribut-atribut false ke dalam memori af lalu lanjut langkah 3a.

    Jika seluruhnya true maka simpan seluruh atribut true dalam memori at lalu lanjut langkah 3b.

    Hasil langkah 2:

    Karena D=F sebab B=F, tetapi C=T, maka simpan D ke dalam memori af.

    Memori af:
    B, D

    Pindah langkah 3a.

12. Langkah 3a:

    Coret seluruh rule dalam memori rsp yang mengandung atribut false dalam memori af sehingga tersisa rule tanpa atribut false. Simpan seluruh rule coret dalam memori rsc. Simpan seluruh rule tersisa dalam memori rsp.

    Ulang langkah 1 untuk himpunan rule dalam memori rsp.

    Hasil langkah 3a:

    Memori rsp:
    ~~Rule1: Jika A dan B dan C maka D~~
    ~~Rule2: Jika A dan C maka K~~
    ~~Rule3: Jika D dan C maka L~~
    ~~Rule4: Jika L dan K maka U~~
    ~~Rule5: Jika U dan B maka S~~

    Memori rsc:
    Rule1: Jika A dan B dan C maka D
    Rule5: Jika U dan B maka S
    Rule2: Jika A dan C maka K
    Rule3: Jika D dan C maka L
    Rule4: Jika L dan K maka U

    Memori af:
    B, D, L

13. Langkah 4:

    Dalam memori rsp, cari semua rule yang seluruh atribut-atribut antesedennya adalah subset memori at.Jika ADA maka picu dan kumpulkan seluruh hasil berupa atribut-atribut konsekuen ke dalam memori at. Coret semua rule dari memori rsp yang sudah picu. Pindahkhan rule coret ke dalam memori rsc. Jika memori rsp belum kosong maka ulang Langkah 4. jika memori rsp kosong, maka kesimpulan adalah seluruh isi memori at. Kesimpulan akhir adalah seluruh atribut konsekuen yang dimasukkan paling terakhir.

    ALGORITMA SELESAI. Jika TIDAK ADA maka lanjut langkah 5.

    Hasil langkah 4:

    Memori rsp:
    ~~Rule1: Jika A dan B dan C maka D~~
    ~~Rule2: Jika A dan C maka K~~
    ~~Rule3: Jika D dan C maka L~~
    ~~Rule4: Jika L dan K maka U~~
    ~~Rule5: Jika U dan B maka S~~

    Memori rsc:
    Rule1: Jika A dan B dan C maka D
    Rule2: Jika A dan C maka K
    Rule3: Jika D dan C maka L
    Rule4: Jika L dan K maka U
    Rule5: Jika U dan B maka S

    Memori at:
    A, C, K

14. Langkah 4 terakhir:

    Karena memori rsp telah kosong, telah tercoret semua maka algoritma selesai dieksekusi.

    Hasil algoritma :

    Memori at:
    A, C, K

    Yaitu bahwa K=True, sesuai yang diharapkan.

    Demikian algoritma ini didemonstrasikan.

D. KESIMPULAN DAN SARAN

Algoritma inferensi pada dasarnya mentabulasi frekuensi kemunculan proposisi atomik atau fakta atau gejala pada anteseden rule cukup untuk digunakan memilih rule yang mana pertama kali akan dipicu oleh

sistem pakar. Pada lingkup yang lebih luas, mungkin ada lebih dari satu rule yang dapat dipicu secara bersamaan, dikarenakan metode menghasilkan lebih dari satu rule yang dapat dipicu. Pada kasus ini semua konsekuen dari beberapa rule digabung dalam rangkaian kata "ATAU" sebagai hasil dari eksekusi rule.

Algoritma ini masih mengassumsikan bahwa anteseden tertulis dalam bentuk normal "DAN" serta bagian anteseden masih dalam bentuk "DAN" yang lebih sederhana yaitu seluruh fakta dirangkai dengan perangkai "DAN" semata, tanpa ada perangkai "ATAU" di dalamnya. Pada pengembangan yang lebih lanjut dari Dokumen HKI ini, sedang dikembangkan untuk mencakup seluruh kombinasi perangkai yang mungkin, baik pada bagian anteseden atau pada bagian konsekuen.

Keterbatasan lain yaitu bahwa algoritma ini masih terbatas pada batasan bahwa dia hnya bersifat mengeksekusi algoritma secara acyclic, tidak menggunakan rule yang sudah dieksekusi dalam satu penalaran sedang sebuah automata terkadang mencapai state sebelumnya yang telah dieksekusi atau mengulang instruksi. Akan tetapi untuk sebuah sistem pakar, algoritma ini diharapkan telah memadai sebagai algoritma inferensi untuk sebarang sistem pakar yang berbasis rule.

REFERENSI

Lucas, P. J. F., & Gaag, L. C. Van Der. (1991). *Principles of Expert Systems*. Addison-Wesley. Retrieved from https://www.cs.ru.nl/~peterl/proe.pdf

McDermott and Charles Lanny. (1976). Production system conflict resolution strategies. Retrieved from http://repository.cmu.edu/cgi/viewcontent.cgi?article=3143&context=compsci

Pakiarajah and Crowther and Hartnett. (2000). Conflict Resolution Techniques for Expert Systems Used to Classify Remotely Sensed Satellite Images. Retrieved May 13, 2016, from http://www.geocomputation.org/2000/GC025/Gc025.htm