

Representasi Sistem Berbasis Aturan Menggunakan Format Premis Terurut

Oleh

Putu Hendra Suputra

Jurusan Manajemen Informatika, FTK, Undiksha

Abstrak

Pengembangan sistem berbasis pengetahuan berpijak pada dua faktor utama yaitu pengkodean mesin inferensi dan representasi pengetahuan. Pengetahuan pada sistem berbasis aturan biasanya diekspresikan dengan format bahasa tertentu yang terpisah dari sintaks pemrograman untuk melakukan penalaran. Terdapat modul program tertentu yang terpisah yang disebut dengan *knowledge-base* dan *inference-engine*.

Teknik representasi aturan pada tahap analisis berpengaruh pada algoritma inferensi yang akan dilakukan. Salah satu metode representasi aturan adalah dengan memandang setiap kode premis sebagai string terurut yang dapat diakses secara sederhana serta mengurangi ketergantungan pada *database management system*.

Kata kunci : Aturan produksi, Premis, Sistem Pakar

Abstract

The development of knowledge-based system rests on two main factors such are the inference engine encoding and knowledge representation. Knowledge of the rules-based system is usually expressed with a specific language format that is separate from the syntax of programming to do reasoning. There are separate program module called the knowledge base and inferensce engine.

Technical representation of the rules at the analysis stage affects the inference algorithm to be performed. One method of the rules representation is to treat at each premise code as a string sequences that can be accessed in a simple way and reduce dependencies on database management system.

Keyword : Production Rule, Premises, Rule Base System

Dalam sistem pakar, representasi pengetahuan menyatakan suatu cara sistematis dalam mengkodekan apa yang diketahui seorang pakar tentang suatu domain (Jackson, 1999). Perlu diketahui bahwa yang dimaksud dengan pengkodean dalam representasi pengetahuan tidak sama dengan penyandian dalam ranah enkripsi. Pengkodean dalam hal ini adalah mengubah simbol-simbol unsur pokok pengetahuan

Representasi Sistem Berbasis Aturan Menggunakan.....(Putu Hendra Suputra)

(abstrak) ke dalam bentuk formal yang mampu merepresentasikan pengetahuan terkait, dengan tetap menjaga makna serpihan pengetahuan yang direpresentasikannya. Kode yang dihasilkan adalah kode yang mampu disimpan dan dibaca oleh komputer (*machine-readable*), serta dapat dengan mudah dialokasikan pada memori komputer saat digunakan

Dalam ilmu komputer, sistem berbasis aturan digunakan sebagai cara untuk menyimpan dan memanipulasi pengetahuan untuk menginterpretasikan informasi secara efektif dan mudah dipahami. Sistem berbasis aturan sering digunakan pada implementasi kajian kecerdasan buatan dan penelitian-penelitian. Penerapan sistem berbasis aturan banyak dilakukan pada sistem pakar (*domain specific expert system*) yang menggunakan aturan (*rule*) untuk melakukan deduksi menuju kesimpulan atau alternatif pilihan. Sistem berbasis aturan mampu melakukan analisis leksikal (*lexical analysis*) terhadap pengetahuan yang telah direpresentasikan.

Sistem berbasis aturan

Lexical analyzer adalah proses mengkonversi sederetan karakter (*string*) menjadi sederetan token. Program atau fungsi yang melakukan analisis leksikal disebut *lexical analyzer*, *lexer* atau *scanner*. Sebuah *lexer* memiliki komponen utama yang disebut *parser*.

Token

Token adalah seuntai karakter sebagai sebuah simbol yang membangun suatu aturan. Token dapat berbentuk beragam yang pada prinsipnya dapat digunakan dalam proses. Pada sistem berbasis aturan, setiap aturan (*rule*) tersusun dalam bentuk IF-THEN-RULE yang terdiri dari bagian anteseden (*kondisi*) dan konsekuen (*konklusi*). Fakta-fakta penyusun anteseden maupun konsekuen dapat berupa fakta tunggal maupun gabungan dari beberapa fakta sekaligus yang dihubungkan oleh konektor logika seperti AND maupun OR. Dalam representasi aturan ini, fakta-fakta diasosiasikan dengan token.

Fakta merupakan istilah manusia dalam mendeskripsikan suatu fenomena aktual pada kehidupan sehari-hari. Dalam upaya menterjemahkan dan memodelkan

pengetahuan manusia ke dalam bentuk yang dapat diproses oleh mesin (komputer), diperlukan suatu bentuk yang cukup fleksibel dalam merepresentasikan pengetahuan (aturan) sekaligus cukup sederhana untuk diimplementasikan ke dalam bahasa pemrograman. Penggunaan model IF-THEN dinilai efektif dengan mengasosiasikan fakta menjadi premis-premis. Secara umum, istilah premis digunakan untuk mewakili fakta-fakta dalam representasinya.

Akuisisi pengetahuan

Dalam akuisisi pengetahuan dari seorang pakar, biasanya dilakukan interview yang intensif antara sang pakar dan knowledge engineer, yang biasanya adalah seorang spesialis komputer. Fokus pekerjaan ini adalah bagaimana mengolah pengetahuan yang dimiliki pakar pada domain tertentu dituangkan ke dalam format atau special purpose language yang nantinya digunakan sebagai dasar penalaran oleh sistem berbasis aturan.

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar yaitu fakta dan aturan (*plus* parameter). Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui (*existing fact*).

Pengetahuan pada sistem berbasis aturan biasanya diekspresikan dengan format bahasa tertentu yang terpisah dari sintaks pemrograman untuk melakukan penalaran. Terdapat modul program tertentu yang terpisah yang disebut dengan *knowledge-base* dan *inference-engine*.

Representasi pengetahuan dan aturan produksi

Representasi pengetahuan adalah faktor yang memegang peranan sangat penting dalam kecerdasan buatan (AI) khususnya sistem berbasis aturan (*rule-based system*). Kunci kesuksesan sistem berbasis aturan adalah menyelesaikan permasalahan dengan pengetahuan yang dimiliki yang tercakup dalam basis pengetahuan. Basis pengetahuan ini berisikan pengetahuan-pengetahuan yang diperoleh dari mentranformasi pengetahuan manusia ke dalam bentuk yang dapat diolah oleh

Representasi Sistem Berbasis Aturan Menggunakan.....(Putu Hendra Suputra)

komputer dan pendukung sistem komputer. Pemecahan masalah pada suatu domain umumnya membutuhkan kumpulan pengetahuan terhadap objek-objek atau partikel atau *term* yang berafiliasi pada domain tersebut.

Pengetahuan harus dapat direpresentasikan secara efisien dan dapat memberi arti atau makna atau nilai yang dapat digunakan oleh sistem cerdas yang membutuhkannya. Efisiensi sangat penting karena hampir mustahil untuk merepresentasikan semua pengetahuan yang ada – bahkan pada domain yang sangat spesifik sekalipun – yang mungkin saja dibutuhkan. Begitu banyak fakta-fakta maupun fenomena yang potensial, namun sebisa mungkin diambil pengetahuan yang secara abstrak dianggap mampu merepresentasikan permasalahan dihadapi.

Pengetahuan harus memiliki makna atau arti sesuai dengan apa yang direpresentasikannya (dapat direlasikan balik ke "*real world*"). Penyusunan representasi pengetahuan bertujuan untuk memetakan fitur-fitur dalam "*real world*" ke dalam bahasa formal. Dalam proses pemetaan ke dalam bahasa formal menggunakan komputer ini, harus tetap dijaga agar apa yang direpresentasikan tetap memiliki arti yang memadai untuk dapat digunakan sebagai dasar pemecahan masalah.

Penggunaan aturan (*rule*) sebagai basis dari sistem berbasis aturan memiliki beberapa keuntungan (Giarratano dan Riley, 2005).

- a) *Modular nature*. Hal ini memudahkan pengembangan sistem secara bertahap, karena penambahan aturan (*rule*) tidak mengganggu struktur aturan yang sudah ada.
- b) *Explanation Facilities*. Membangun fasilitas penjelasan dengan menggunakan *rule* menjadi lebih mudah karena bagian anteseden dari *rule* secara eksplisit mengaktifasi *rule*.
- c) *Similarity to the human cognitive process*. *Rule* merupakan cara yang alami dalam memodelkan bagaimana manusia memecahkan masalah. Representasi sederhana *rule* dalam bentuk IF-THEN memudahkan penjelasan dalam memahami struktur pengetahuan yang ingin diperoleh.

Dengan keuntungan tersebut, pendekatan sistem berbasis aturan banyak dikembangkan dalam sistem pakar.

Post Production System

Production system kali pertama digunakan pada logika simbolik oleh Post (Giarratano dan Riley, 2005). Post menunjukkan bahwa setiap sistem matematika maupun logika dapat ditulis menjadi tipe *production rule* tertentu

Ide dasar dari Post adalah bahwa setiap sistem matematika atau logika merupakan sekumpulan aturan yang menspesifikasikan bagaimana mengubah sederetan simbol menjadi simbol-simbol yang lain. Sehingga dengan memberikan masukan sederetan input, anteseden, aturan produksi dapat memproduksi deretan keluaran baru, yaitu konsekuen. Ide ini cocok pada program dan sistem pakar yang inisial simbolnya berupa *input* dan menghasilkan *output* melalui serangkaian transformasi.

Contoh sederhana, *input string* yaitu “patient has fever”, maka output stringnya bisa berupa “take an aspirin” (dengan catatan bahwa input dan output hanya berupa string secara harfiah, tanpa makna kata). Kuncinya adalah pada manipulasi *string* berdasarkan pada sintaks (struktur), bukan secara semantik (makna kata) atau pemahaman apa arti kata fever, aspirin, patient. Manusia dapat memahami arti kata-kata tersebut, tetapi pada *Post production system* hanya mentransformasi satu *string* ke *string* lainnya.

Aturan produksi dari contoh ini adalah sebagai berikut.

Anteseden → Konsekuen

patient has fever → take aspirin

di mana tanda panah menunjukkan transformasi satu string menjadi string lainnya. Aturan ini dapat diinterpretasikan dalam bentuk yang lebih familiar yaitu notasi IF..THEN.

IF person has fever THEN take aspirin

notasi ini juga dapat berbentuk multi anteseden seperti:

person has fever AND fever is greater than 102 THEN see doctor

Pada contoh tersebut, AND bukan bagian dari string. AND mengindikasikan bahwa aturan memiliki beberapa anteseden. Dengan menggunakan *Post production system*, urutan dari aturan-aturan tidak mempengaruhi pengambilan keputusan.

Representasi Sistem Berbasis Aturan Menggunakan.....(Putu Hendra Suputra)

Meskipun aturan yang menggunakan *Post production system* sangat berguna dan menjadi dasar dalam sistem pakar, namun dalam implementasi programnya memiliki kekurangan dalam *control strategy* dalam penentuan penggunaan rule. Jika koleksi aturan berukuran sangat besar, maka akan membutuhkan waktu lebih banyak karena dalam *Post production* mengizinkan penggunaan aturan sepanjang antesedennya terpenuhi, yang mengakibatkan butuh banyak waktu untuk mencari dan mencocokkan setiap rule yang ada.

Terdapat dua istilah dalam representasi pengetahuan pada sistem ini, yaitu kunci dan aturan. Keduanya akan dijelaskan definisi dan hubungannya sebagai penyusun pengetahuan atas domain.

Kunci

Kunci yang dimaksud di sini adalah kunci identifikasi yang menjadi bakal calon premis-premis yang mendukung terbangkitnya suatu aturan. Kunci disimpan sebagai sebuah *string* yang diberi nama sesuai premis yang diwakilkannya. Dalam pengetahuan yang sama tidak diperkenankan terdapat kunci yang sama. Premis yang berbeda harus direpresentasikan dengan kunci yang berbeda pula. Dengan kata lain, kunci merupakan daftar calon premis. Sedangkan premis adalah kunci yang disimpan di dalam *working memori* sebagai akibat dari penambahan fakta oleh pengguna maupun hasil eksekusi dari suatu aturan dalam proses penalaran.

Sebelum sistem siap untuk melakukan penalaran, kunci dan aturan terlebih dahulu dimuat ke dalam memori. Kunci disimpan dalam sebuah tabel kunci identifikasi, sedangkan premis disimpan dalam *Working Memory* yang memiliki struktur data serupa agar mudah dalam transformasi dan penggalan informasi. Daftar kunci identifikasi dan *working memory* direpresentasikan dengan struktur data seperti pada Tabel 1.

Tabel 1 Struktur data kunci identifikasi

Field	Keterangan
ID	Identitas kunci
Nama	Nama kunci

Atribut	Sebuah atribut yang bisa digunakan untuk mengelompokkan kunci
Keterangan	Berupa teks yang berfungsi untuk menyimpan informasi tentang kunci
Path	Direktori dan nama gambar yang terkait dengan kunci

Sistem yang tidak menggunakan sistem manajemen basis data khusus, mesin inferensi harus dilengkapi fungsi untuk menemukan nama premis, ataupun keberadaannya. Hal yang sama juga berlaku untuk daftar aturan dan agenda yang dijelaskan pada bagian berbeda bab ini.

Penyimpanan kunci dalam *file* mengikuti format yang telah ditentukan seperti pada Gambar 1.

[
<ID-kunci>
<nama kunci>
<atribut>
<keterangan line-1>
..
<keterangan line-N>
]
<image path>

Gambar 1 Aturan penyimpanan kunci dalam berkas (file)

Pembacaan kunci baru akan dilakukan jika sistem membaca adanya karakter pembuka “[“ (tanpa tanda kutip), dan berhenti membaca keterangan kunci setelah menemukan karakter penutup “]” (tanpa tanda kutip). Penjelasan masing-masing bagian adalah sebagai berikut.

- <ID-kunci> : Kode unik yang merepresentasikan kunci (premis) dalam aturan
- <nama kunci> : Nama kunci
- <atribut> : Atribut yang melekat pada kunci, berupa nilai integer yang maknanya ditentukan oleh pengembang sistem
- <keterangan line-N> : Berupa baris-baris string yang menyimpan penjelasan, keterangan, dan label lain yang melekat pada kunci
- <image path> : Alamat gambar jika diperlukan

Proses pembacaan oleh sistem dilakukan sampai akhir dari file (*End of File*).

Aturan

Aturan juga disimpan pada *file* teks yang dimuat ke dalam memori sebelum dapat digunakan untuk penalaran. Aturan dibuat mengikuti *Post production system*. Aturan menggunakan format IF-THEN yang kondisinya merupakan sebuah premis atau gabungan premis yang dihubungkan dengan operator AND. Jika ada aturan yang menggunakan operator OR, knowledge engineer harus memecahnya menjadi beberapa aturan terpisah sedemikian sehingga tidak ada operator OR di dalamnya. Konklusi dari aturan berupa premis tunggal. Setiap premis baik penyusun kondisi maupun kondisi memiliki tanda yang menentukan apakah premis berupa negasi atau tidak.

Secara teknis, sebuah aturan akan direpresentasikan dengan sebuah record yang berisikan bagan-bagian atau *field* seperti pada Tabel 2. Aturan seperti halnya kunci, juga disimpan pada berkas (*file*) dengan mengikuti aturan yang telah ditentukan. Penyimpanan aturan dalam *file* menggunakan format seperti Gambar 2.

Tabel 2 Struktur data aturan

Field	Keterangan
Kondisi	Kondisi berupa sebuah <i>string</i> yang merupakan gabungan ID-Kunci yang membentuk kondisi. Format gabungan ditentukan oleh sistem yang berguna untuk menentukan bagaimana program dapat menguraikan (<i>parse</i>) kondisi menjadi premis-premis penyusunnya.
Tanda kondisi	Sebuah <i>string</i> yang merupakan penyimpan tanda premis penyusun kondisi. Seperti jga field kondisi, tanda kondisi juga merupakan gabungan tanda masing masing premis kondisi.
Konklusi	Menyimpan ID kunci tertentu yang merepresentasikan konklusi dari aturan.
Tanda konklusi	Berupa karakter yang menunjukkan tanda negasi dari konklusi.
CF	Nilai certainty factor dari aturan dengan asumsi semua <i>evidence</i> (premis kondisi) memiliki CF=1

Sebagai contoh, terdapat beberapa kunci seperti berikut ini.

ID	Nama
1	Premis A
2	Premis B
3	Premis C
4	Premis D

Sistem memiliki sebuah kode format yang menentukan bagaimana kondisi dibentuk. Jika dibuat sebuah aturan :

IF Premis A AND NOT Premis B AND Premis C THEN Premis D

dengan kode format bernilai 3, artinya setiap ID kunci dibuat dalam format 3 digit, sehingga aturannya menjadi:

IF 001002003 THEN 004 , dengan tandakondisi=**101** (0=negasi) atau

kondisi: **001002003**

tandakondisi: **101**

konklusi:**004**

tandakonklusi: **1**

Penyimpanan aturan dalam berkas dilakukan dengan format seperti Gambar 2.

```
[
<kondisi >
<tanda kondisi>
<konklusi>
<tanda konklusi>
<CF>
<keterangan line-1>
..
<keterangan line-N>
]
```

Gambar 2 Aturan penyimpanan aturan (rules) dalam berkas (file)

Penjelasan masing-masing bagian adalah sebagai berikut.

<kondisi> : Kode yang merepresentasikan kondisi dari aturan, merupakan *string* gabungan dari premis-premis pendukung aturan yang mengikuti format tertentu

<tanda kondisi>: Adalah *string* gabungan yang menunjukkan tanda negasi dari premis-premis pendukung aturan

- <konklusi> : Kode yang mereferensi pada kunci yang merupakan konklusi dari aturan
- <tanda konklusi>: Penanda negasi dari premis konklusi
- <keterangan line-N> Berupa baris-baris *string* yang menyimpan penjelasan, : keterangan, dan label lain yang melekat pada aturan jika ada

Pembacaan dan penulisan kunci dan aturan harus dilakukan secara otomatis oleh sistem. Pengguna tidak melakukan pembacaan, penulisan, perubahan, maupun akses dalam bentuk apapun secara langsung terhadap berkas. Aturan penulisan dan pembacaan tersebut dituangkan di dalam kode program saat pengembangan mesin inferensi.

Daftar Pustaka

- Giarratano, J. dan Riley, G., 2005, *Expert System Principles and Programming*, PWS Publishing Company, Boston
- Hartati, S. dan Iswanti, S., 2008, *Sistem Pakar dan Pengembangannya*, Graha Ilmu, Yogyakarta
- Jackson, P., 1999, *Introduction to Expert System, Third Edition*, Addison Wesley Longman Limited, Essex
- Jain, M.B. , Amit, dan Srinivas M.B., 2008, A Web based Expert System Shell for Fault Diagnosis and Control of Power System Equipment, *Proceedings, International Conference on Condition Monitoring and Diagnosis, Beijing, China*, Beijing
- Kemp, E.A. 1996, Evaluating the Developer's Interface to a KBS Shell: Some Reflections, *Proceedings of the 6th Australian Conference on Computer-Human Interaction (OZCHI '96)*, 0-8186-7525-X/96 1996 IEEE, <http://www.computer.org>
- Land, S.M., 1992, Expert System Design Shells: A Critical Analysis, *Instructional Technology Research Online*, Penn State University, http://www.gsu.edu/research/issue_of_ex_shell_files/ , 18 Juni 2010