

Algoritma Backward Chaining pada Rule-Based Expert System

Wawan Yunanto

wawan@pcr.ac.id

<http://www.pcr.ac.id/~wawan>

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Sistem pakar sebagai salah satu aplikasi komputer, biasanya dibuat menggunakan bahasa-bahasa pemrograman yang memang didesain untuk aplikasi *Artificial Intelligence*, misalnya : LISP, EXSYS, PROLOG/Visual PROLOG. Kekurangan dari penggunaan bahasa-bahasa pemrograman di atas adalah integrasi dengan teknologi yang lain, misal : *Web-based Applications*.

Oleh Karena itu, sekarang banyak pengembang aplikasi yang menggunakan bahasa pemrograman konvensional (Delphi, .NET, Java, dan lain-lain) untuk membangun suatu sistem pakar. Kita bisa memilih bahasa mana saja, tentu disesuaikan dengan kebutuhan akan implementasi dari sistem pakar tersebut.

Walau suatu aplikasi bisa dibangun dengan bermacam-macam bahasa pemrograman, tapi ada satu yang tetap sama, yaitu algoritmanya.

Artikel ini membahas tentang algoritma mesin inferensi backward chaining untuk sistem pakar yang representasi pengetahuannya menggunakan *rule* (kaidah).

Sistem Pakar berbasis kaidah

Sistem pakar adalah suatu program komputer yang memperlihatkan derajat keahlian dalam pemecahan masalah di bidang tertentu sebanding dengan seorang pakar (Ignizio, 1991). Keahlian sistem pakar dalam memecahkan suatu masalah diperoleh dengan cara merepresentasikan pengetahuan seorang atau beberapa orang pakar dalam format tertentu dan menyimpannya dalam basis pengetahuan. Sistem pakar berbasis kaidah (*rule-based expert system*) adalah sistem pakar yang menggunakan kaidah (*rules*) untuk merepresentasikan pengetahuan di dalam basis pengetahuannya.

Mesin inferensi (*inference engine*) merupakan bagian yang bertindak sebagai pencari solusi dari suatu permasalahan berdasar pada kaidah-kaidah yang ada dalam basis pengetahuan sistem pakar. Selama proses inferensi, mesin inferensi memeriksa status dari basis pengetahuan dan memori kerja (*working memory*) untuk menentukan fakta apa saja yang diketahui dan untuk menambahkan fakta baru yang dihasilkan ke dalam memori kerja tersebut. Fakta-fakta yang merupakan hasil dari proses inferensi disimpan dalam memori kerja.

Strategi Pencarian pada Mesin Inferensi

Ada dua strategi pencarian dasar yang bisa digunakan oleh mesin inferensi dalam mencari kesimpulan untuk mendapatkan solusi bagi permasalahan yang dihadapi sistem pakar, yaitu runut maju (*forward chaining*) dan runut balik (*backward chaining*). Berikut ini penjelasan mengenai kedua strategi pencarian tersebut :

1. Runut Maju

Runut maju merupakan strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dari data-data tersebut dicari suatu kesimpulan yang menjadi solusi dari permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang premisnya sesuai dengan data-data tersebut, kemudian dari kaidah-kaidah tersebut diperoleh suatu kesimpulan. Runut maju memulai proses pencarian dengan data sehingga strategi ini disebut juga *data-driven*.

2. Runut Balik

Runut balik merupakan strategi pencarian yang arahnya kebalikan dari runut maju. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan dirunut balik jalur yang mengarah ke kesimpulan tersebut. Jika informasi-informasi atau nilai dari atribut-atribut yang mengarah ke kesimpulan tersebut sesuai dengan data yang diberikan maka kesimpulan tersebut merupakan solusi yang dicari, jika tidak sesuai maka kesimpulan tersebut bukan merupakan solusi yang dicari. Runut balik memulai proses pencarian dengan suatu tujuan sehingga strategi ini disebut juga *goal-driven*.

Algoritma Runut Balik

Penggunaan strategi pencarian runut balik untuk membangun mesin inferensi memerlukan suatu algoritma tertentu sehingga bisa diimplementasikan menggunakan bahasa pemrograman. Salah satu algoritma runut balik yang diambil dari buku *Introduction To Expert Systems : The Development and Implementation Of Rule-Based Expert Systems* (Ignizio, 1991) adalah sebagai berikut :

1. Inisialisasi

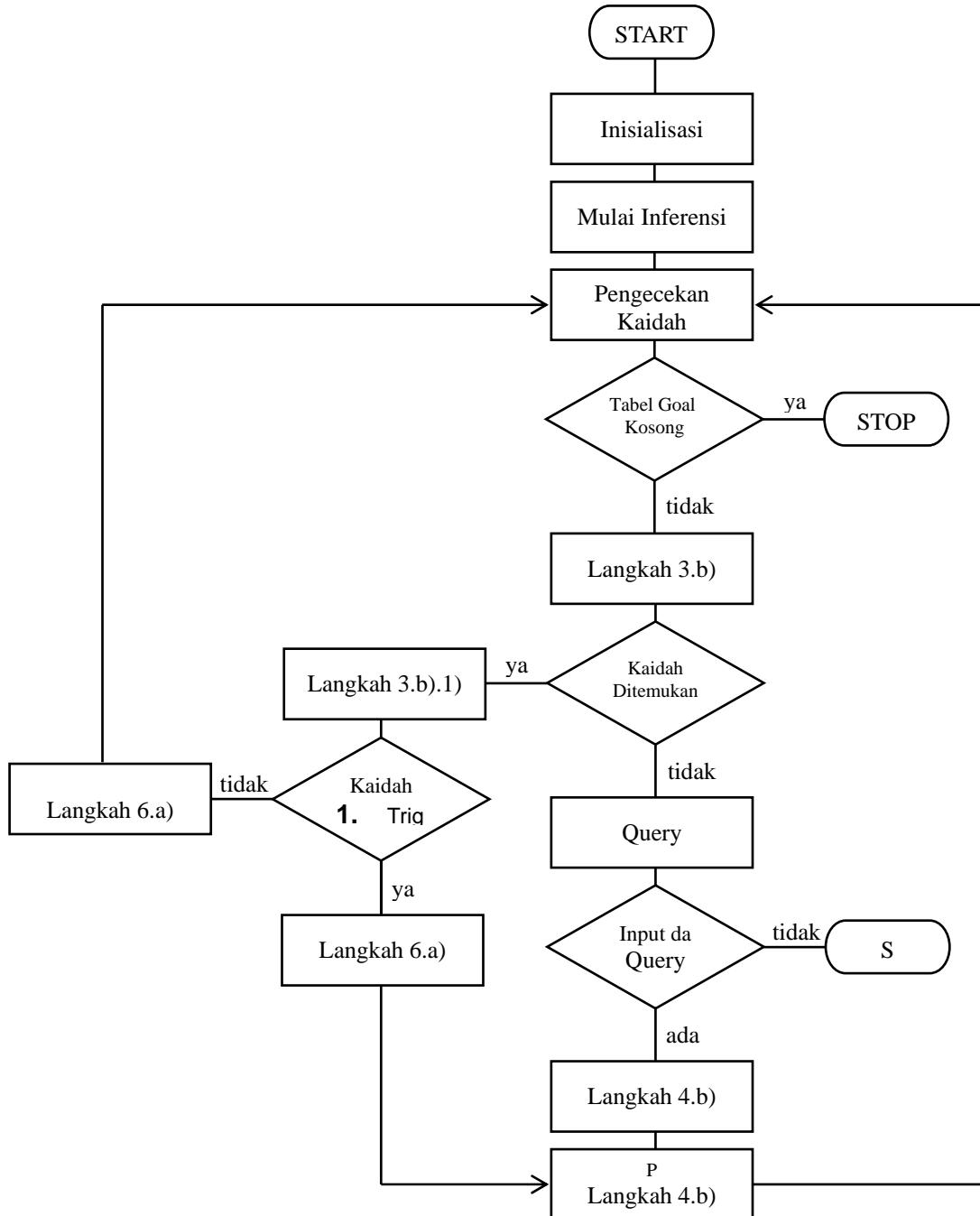
- a) Buat 3 tabel yaitu :
 - 1) Tabel Working Memory, untuk menyimpan pasangan atribut-nilai hasil dari proses inferensi.
 - 2) Tabel Goal, untuk menyimpan atribut yang nilainya sedang dicari.
 - 3) Tabel Rule/Premise Status, untuk menyimpan nomor kaidah, status dari kaidah yang bersesuaian, pasangan atribut-nilai dari klausa premis kaidah-kaidah dalam basis pengetahuan, nomor klausa premis dalam kaidah yang bersesuaian, dan status dari klausa premis tersebut.
- b) Semua klausa premis dalam tabel Rule/Premise Status diberi status *free*.
- c) Semua kaidah dalam tabel Rule/Premise Status diberi status *active*.

2. Mulai inferensi
 - a) Sebutkan kesimpulan akhir (solusi dari permasalahan) yang ingin dicapai.
 - b) Letakkan atribut dari klausa kesimpulan yang merupakan solusi dari permasalahan pada puncak tabel Goal.
3. Pengecekan kaidah
 - a) Jika tabel Goal kosong maka STOP.
 - b) Jika tabel Goal tidak kosong maka cari kaidah-kaidah berstatus *active* yang atribut dari klausa kesimpulannya bersesuaian dengan atribut yang berada pada puncak tabel Goal.
 - 1) Jika hanya 1 kaidah yang ditemukan, lakukan langkah 6. Jika ada beberapa kaidah yang ditemukan, cari yang berstatus *triggered*, lakukan langkah 6. Jika tidak ada kaidah yang berstatus *triggered*, pilih salah satu kaidah lalu proses dengan langkah 6.
 - 2) Jika tidak ada kaidah yang ditemukan, lakukan langkah 4.
4. Query
Ambil salah satu data dari sekumpulan data yang diberikan oleh pemakai yang bersesuaian dengan atribut yang berada pada puncak tabel Goal.
 - a) Jika tidak ada maka STOP.
 - b) Jika ada maka pindahkan atribut yang berada pada puncak tabel Goal lalu tempatkan pada tabel Working Memory beserta nilainya, yaitu data yang diambil tadi.
5. Pembaharuan status kaidah/premis
 - a) Gunakan isi dari tabel Working Memory untuk membaharui tabel Rule/Premise Status.
 - b) Jika ada klausa premis yang berstatus *false* pada suatu kaidah maka beri status *discard* pada kaidah tersebut. Tetapi jika seluruh klausa premis pada suatu kaidah semuanya berstatus *true* maka beri status *triggered* pada kaidah tersebut.
 - c) Kembali ke langkah 3.
6. Evaluasi kaidah
 - a) Jika kaidah berstatus *triggered* maka pindahkan atribut yang berada pada puncak tabel Goal ke tabel Working Memory beserta nilainya yang terdapat dalam klausa kesimpulan pada kaidah tersebut. Ubah status kaidah tersebut menjadi *fired*. Kembali ke langkah 5.
 - b) Jika kaidah tidak berstatus *triggered* maka pilih atribut dari klausa premis pertama yang berstatus *free* lalu tempatkan pada puncak tabel Goal. Kembali ke langkah 3.

Algoritma runut balik di atas mempunyai 2 keadaan yang menyebabkan proses inferensi berhenti, yaitu :

1. Pada saat tabel Goal kosong, berarti kesimpulan yang merupakan solusi dari permasalahan sudah diperoleh.
2. Pada saat data dari pemakai yang diminta oleh sistem pakar untuk memenuhi prompt kaidah tidak ada, berarti kesimpulan yang merupakan solusi dari permasalahan tidak ditemukan.

Untuk memperjelas tentang algoritma runut balik dapat dilihat pada diagram alir berikut ini :



Algoritma ini bisa dipakai sebagai salah satu pilihan, apabila sistem pakar yang dikembangkan memang menggunakan backward chaining. Hal-hal lain yang berhubungan dengan sistem pakar, insyaAllah akan penulis coba juga untuk membuat artikelnya, sehingga kita semua bisa saling barbagi ilmu.

Referensi

Ignizio, J.P., 1991, *Introduction To Expert Systems : The Development and Implementation Of Rule-Based Expert Systems*, McGraw-Hill, Inc.

Biografi Penulis



Nama Penulis : Wawan Yunanto, S.Kom

Lulus pendidikan S1 dari Universitas Gadjah Mada, Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam.

Dosen Program Studi Teknik Komputer Politeknik Caltex Riau.