Masalah dan pengetahuan dalam sistem pakar

A. Sistem Perantaian Maju (Forward Chaining Systems)

Pada sistem perantaian maju, fakta-fakta dalam dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbarui. Aturan dalam sistemmerepresentasikan aksiaksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja, sering disebut aturan kondisi-aksi. Kondisi biasanya berupa pola yang cocok dengan item yang ada di dalam memorikerja, sementara aksi biasanya berupa penambahan atau penghapusan item dalam memori kerja.

Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi (recognise-act). Mula-mula, sistem mencari semua aturan yang kondisinya terdapat di memori kerja, kemudian memilih salah satunya dan menjalankan aksi yang bersesuaian dengan aturan tersebut. Pemilihan aturan yang akan dijalankan (fire) berdasarkanstrategi tetap yang disebut strategi penyelesain konflik. Aksi tersebut menghasilkan memori kerja baru, dan siklus diulangi lagi sampai tidak ada aturanyang dapat dipicu (fire), atau goal (tujuan) yang dikehendaki sudah terpenuhi.

Sebagai contoh, lihat pada sekumpulan aturan sederhana berikut (Di sini kita memakai kata yang diawali huruf kapital untuk menyatakan suatu variabel. Pada sistem lain, mungkin dipakai cara lain, misalnya menggunakan awalan ? atau ^):

1. JIKA (mengajar X)

   DAN (mengoreksi_tugas X)

MAKA TAMBAH (terlalu_banyak_bekerja X)

2.JIKA (bulan maret)

MAKA TAMBAH (mengajar dosen)

3.JIKA (bulan maret)

MAKA TAMBAH (mengoreksi_tugas dosen)

4. JIKA (terlalu_banyak_bekerja X)

ATAU (kurang_tidur X)

MAKA TAMBAH (mood_kurang_baik X)

5. JIKA (mood_kurang_baik X)

MAKA HAPUS (bahagia X)

6. JIKA (mengajar X)

MAKA HAPUS (meneliti X)

Kita asumsikan, pada awalnya kita mempunyai memori kerja yang berisi faktaberikut:

(bulan maret) (bahagia dosen) (meneliti dosen)

Sistem Pakar mula-mula akan memeriksa semua aturan yang ada untukmengenali aturan manakah yang dapat memicu aksi, dalam hal ini aturan 2 dan 3.

Sistem kemudian memilih salah satu di antara kedua aturan tersebut dengan strategi penyelesaian konflik. Katakanlah aturan 2 yang terpilih, maka fakta (mengajar dosen) akan ditambahkan ke dalam memori kerja. Keadaan memori kerja sekarang menjadi:

(mengajar dosen)(bulan maret) (bahagia dosen) (meneliti dosen)

Sekarang siklus dimulai lagi, dan kali ini aturan 3 dan 6 yang kondisinya terpenuhi. Katakanlah aturan 3 yang terpilih dan terpicu, maka fakta (mengoreksi_tugas dosen) akan ditambahkan ke dalam memori kerja. Lantas pada siklus ketiga, aturan 1 terpicu, sehingga variabel X akan berisi (bound to) dosen, dan fakta (terlalu_banyak_bekerja dosen) ditambahkan, sehingga isi memori kerja menjadi:

(terlalu_banyak_bekerja dosen)(mengoreksi_tugas dosen) (mengajar dosen)

(bulan maret) (bahagia dosen) (meneliti dosen)

Aturan 4 dan 6 dapat diterapkan. Misalkan aturan 4 yang terpicu, sehingga fakta (mood_kurang_baik dosen) ditambahkan. Pada siklus berikutnya, aturan 5 terpilih dan dipicu, sehingga fakta (bahagia dosen) dihapus dari memori kerja. Kemudianaturan 6 akan terpicu dan fakta (meneliti dosen) dihapus pula dari memori kerja menjadi:

(mood_kurang_baik dosen) (terlalu_banyak_bekerja dosen)

(mengoreksi_tugas dosen) (mengajar dosen)

(bulan maret)

Urutan aturan yang dipicu bisa jadi sangat vital, terutama di mana aturan-aturan yang ada dapat mengakibatkan terhapusnya item dari memori kerja. Tinjau kasusberikut: andaikan terdapat tambahan aturan pada kumpulan aturan di atas, yaitu:

7 . JIKA (bahagia X)

MAKA TAMBAH (memberi_nilai_bagus X)

Jika aturan 7 ini terpicu sebelum (bahagia dosen) dihapus dari memori, maka Sistem Pakar akan berkesimpulan bahwa saya akan memberi nilai bagus( ) 1 . Namun jika aturan 5 terpicu dahulu, maka aturan 7 tidak akan dijalankan (artinya saya tidak akan memberi nilai bagus).

B. Strategi penyelesaian konflik (conflict resolution strategy)

Strategi penyelesaian konflik dilakukan untuk memilih aturan yang akan diterapkanapabila terdapat lebih dari 1 aturan yang cocok dengan fakta yang terdapat dalammemori kerja. Di antaranya adalah:

1. No duplication. Jangan memicu sebuah aturan dua kali menggunakan fakta/data yang sama, agar tidak ada fakta yang ditambahkan ke memori kerjalebih dari sekali.
2. Recency. Pilih aturan yang menggunakan fakta yang paling baru dalam memorikerja. Hal ini akan membuat sistem dapat melakukan penalaran dengan mengikuti rantai tunggal ketimbang selalu menarik kesimpulan baru menggunakan fakta lama.
3. Specificity. Picu aturan dengan fakta prakondisi yang lebih spesifik (khusus) sebelum aturan yang mengunakan prakondisi lebih umum. Contohnya: jika kita mempunyai aturan “JIKA (burung X) MAKA TAMBAH (dapat_terbang X)” dan “JIKA (burung X) DAN (pinguin X) MAKA TAMBAH (dapat_berenang X)” sertafakta bahwa tweety adalah seekor pinguin, maka lebih baik memicu aturan kedua dan menarik kesimpulan bahwa tweety dapat berenang.
4. Operation priority. Pilih aturan dengan prioritas yang lebih tinggi. Misalnya adafakta (bertemu kambing), (ternak kambing), (bertemu macan), dan (binatang_buas macan), serta dua aturan: “JIKA (bertemu X) DAN (ternak X) MAKA TAMBAH (memberi_makan X)” dan “JIKA (bertemu X) DAN (binatang_buas X) MAKA TAMBAH (melarikan_diri)”, maka kita akan memilih aturan kedua karena lebih tinggi prioritasnya.

C. Sistem Perantaian Balik (Backward Chaining Systems)

Sejauh ini kita telah melihat bagaimana sistem berbasis aturan dapat digunakan untuk menarik kesimpulan baru dari data yang ada, menambah kesimpulan ini kedalam memori kerja. Pendekatan ini berguna ketika kita mengetahui semua fakta awalnya, namun tidak dapat menebak konklusi apa yang bisa diambil. Jika kita tahu kesimpulan apa yang seharusnya, atau mempunyai beberapa hipotesis yangspesifik, maka perantaian maju di atas menjadi tidak efisien. Sebagai contoh, jika kita ingin mengetahui apakah saya dalam keadaan mempunyai mood yang baik sekarang, kemungkinan kita akan berulangkali memicu aturan-aturan dan memperbarui memori kerja untuk mengambil kesimpulan apa yang terjadi pada bulan Maret, atau apa yang terjadi jika saya mengajar, yang sebenarnya perlu terlalu kita ambil pusing. Dalam hal ini yang diperlukan adalah bagaimana dapat menarik kesimpulan yang relevan dengan tujuan atau goal.

Hal ini dapat dikerjakan dengan perantaian balik dari pernyataan goal (atauhipotesis yang menarik bagi kita). Jika diberikan sebuah goal yang hendak dibuktikan, maka mula-mula sistem akan memeriksa apakah goal tersebut cocok dengan fakta-fakta awal yang dimiliki. Jika ya, maka goal terbukti atau terpenuhi. Jika tidak, maka sistem akan mencari aturan-aturan yang konklusinya (aksinya) cocok dengan goal. Salah satu aturan tersebut akan dipilih, dan sistem kemudianakan mencoba membuktikan fakta-fakta prakondisi aturan tersebut menggunakan prosedur yang sama, yaitu dengan menset prakondisi tersebut sebagai goal baruyang harus dibuktikan.

Perhatikan bahwa pada perantaian balik, sistem tidak perlu memperbarui memorikerja, namun perlu untuk mencatat goal-goal apa saja yang dibuktikan untuk

membuktikan goal utama (hipotesis). Secara prinsip, kita dapat menggunakan aturan-aturan yang sama untuk perantaian maju dan balik. Namun, dalam prakteknya, harus sedikit dimodifikasi. Pada perantaian balik, bagian MAKA dalam aturan biasanya tidak diekspresikan sebagai suatu aksi untuk dijalankan (misalnyaTAMBAH atau HAPUS), tetapi suatu keadaan yang bernilai benar jika premisnya (bagian JIKA) bernilai benar. Jadi aturan-aturan di atas diubah menjadi:

1. JIKA (mengajar X)

   DAN (mengoreksi_tugas X)

   MAKA (terlalu_banyak_bekerja X)

2. JIKA (bulan maret)

   MAKA (mengajar dosen)

3. JIKA (bulan maret)

   MAKA (mengoreksi_tugas dosen)

4. JIKA (terlalu_banyak_bekerja X)

   ATAU (kurang_tidur X)

   MAKA (mood_kurang_baik X)

5. JIKA (mood_kurang_baik X) MAKA

   TIDAK BENAR (bahagia X)

dengan fakta awal: (bulan maret) (meneliti dosen)

Misalkan kita hendak membuktikan apakah mood saya sedang kurang baik.Mulamula kita periksa apakah goal cocok dengan fakta awal. Ternyata tidak ada fakta awal yang menyatakan demikian, sehingga langkah kedua yaitu mencari aturan mana yang mempunyai konklusi (mood_kurang_baik dosen). Dalam hal iniaturan yang cocok adalah aturan 4 dengan variabel X diisi dengan (bound to) dosen. Dengan demikian kita harus membuktikan bahwa prakondisi aturan ini, (terlalu_banyak_bekerja dosen) atau (kurang_tidur dosen), salah satunya adalah benar (karena memakai ATAU). Lalu diperiksa aturan mana yang dapat membuktikan bahwa adalah (terlalu_banyak_bekerja dosen) benar, ternyata aturan 1, sehingga prakondisinya, (mengajar X) dan (mengoreksi_tugas X), dua-duanya adalah benar (karena memakai DAN). Ternyata menurut aturan 2 dan 3, keduanya bernilai bernilai benar jika (bulan maret) adalah benar.

Karena ini sesuai dengan fakta awal, maka keduanya bernilai benar. Karena semua goal sudah terpenuhi maka goal utama (hipotesis) bahwa mood saya sedang kurang baik adalah benar (terpenuhi).

Untuk mencatat goal-goal yang harus dipenuhi/dibuktikan, dapat digunakan stack (tumpukan). Setiap kali ada aturan yang konklusinya cocok dengan goal yang sedang dibuktikan, maka fakta-fakta prakondisi dari aturan tersebut ditaruh (push)ke dalam stack sebagai goal baru. Dan setiap kali goal pada tumpukan teratas terpenuhi atau dapat dibuktikan, maka goal tersebut diambil (pop) dari tumpukan.Demikian seterusnya sampai tidak ada goal lagi di dalam stack, atau dengan katalain goal utama (yang terdapat pada tumpukan terbawah) sudah terpenuhi.

D. Pemilihan Sistem Inferensi

Secara umum kita dapat memakai panduan berikut untuk menentukan apakah kitahendak memilih perantaian maju atau balik untuk Sistem Pakar yang kita bangun.Panduan tersebut tercantum dalam Tabel berikut ini.

Tabel 3.1. Panduan untuk memilih sistem inferensi

Perantaian Maju	Perantaian Balik
• Ada banyak hal yang hendak dibuktikan • Hanya sedikit fakta awal yang dipunyai • Ada banyak aturan berbeda yang dapat memberikan kesimpulan yang sama	• Hanya akan membuktikan fakta (hipotesis) tunggal • Terdapat banyak fakta awal • Jika terdapat banyak aturan yang memenuhi syarat untuk dipicu (fire) pada suatu siklus

E. Ketidakpastian dalam Aturan

Sejauh ini kita menggunakan nilai kebenaran tegas dalam fakta dan aturan yang dipakai, misalnya: jika terlalu banyak bekerja maka pasti mood kurang baik. Padakenyataanya, seringkali kita tidak bisa membuat aturan yang absolut untuk mengambil kesimpulan secara pasti, misalnya: jika terlalu banyak bekerja maka kemungkinan besar mood kurang baik. Untuk itu, seringkali aturan yang dipakai

memiliki nilai kepastian (certainty value). Contohnya: jika terlalu banyak bekerjamaka pasti mood kurang baik (kepastian 0,75)

Komponen – komponen sistem pakar

Diantara komponen-komponen dalam Gambar 1 di atas, basis pengetahuan dan mesin interfensi adalah modul paling kritis agar sistem pakar dapat berfungsi dengan baik. Pengetahuan harus direpresentasikan dan diatur secara tepat dalam basis pengetahuan. Mesin inferensi kemudian dapat menggunakan pengetahuan tersebut untuk menarik kesimpulan baru dari fakta dan aturan yang ada. Dalam bagian ini, struktur berbasis pengetahuan dan mesin inferensi pada sistem berbasis-aturan.

1. Representasi dan Organisasi Pengetahuan

   Pengetahuan pakar harus direpresentasikan dalam format yang dapat dipahami komputer dan diatur dengan tepat dalam basis pengetahuan sistem pakar. Terdapat beberapa cara yang berbeda untuk merepresentasikan pengetahuan manusia, antara lain aturan produksi, jaringan semantik, dan pernyataan logika.

   Dalam sistem berbasis aturan, pengetahuan dalam basis pengetahuan direpresentasikan dalam aturan JIKA MAKA yang menggabungkan kondisi dan kesimpulan untuk menangani situasi tertentu.

   Bagian JIKA mengindikasikan kondisi aturan tersebut diaktifkan, dan bagian MAKA menunjukkan tindakan atau kesimpulan jika semua kondisi JIKA dipenuhi.

   Keuntungan menggunakan aturan produksi adalah aturan tersebut mudah dipahami dan aturan baru dapat ditambahkan dengan mudah ke dalam basis pengetahuan tanpa memengaruhi aturan yang telah ada. Ketidakapastian yang

dihubungkan dengan tiap aturan dapat ditambahkan untuk meningkatkan keakuratannya.

Tugas utama pengembangan sistem pakar adalah memperoleh pengetahuan darimanusia dan mengubahnya menjadi aturan produksi yang dapat ditangani mesin inferensi. Mesin inferensi memilih aturan yang dapat diterapkan dari basis pengetahuan, mengintegrasikannya, dan mempertimbangkannya untukmendapatkan kesimpulan.

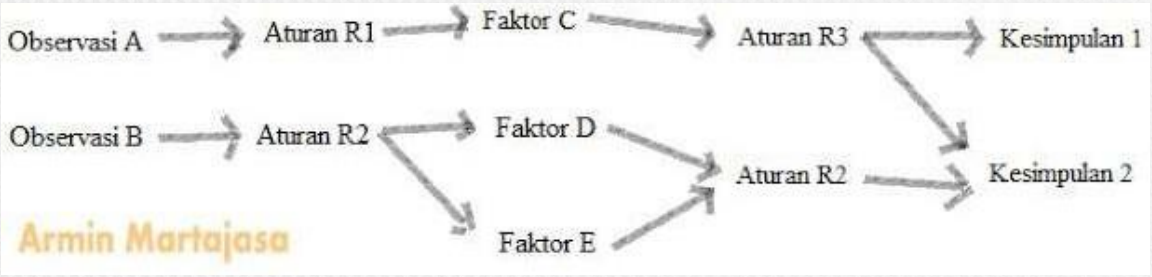
2. Mesin Inferensi

Dalam keputusan kompleks, pengetahuan pakar sering tidak dapat direprentasikan dalam aturan tunggal. Sebaliknya, aturan dapat digabungkan secara dinamis untuk mencakup barbagai kondisi. Proses penggabungan banyak aturan berdasarkan data yang tersedia, disebut inferensi. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Ada dua pendekatan populer untuk menarik kesimpulan adalah sebagai berikut:

a. Forward Chaining

Forward chaining adalah mencari bagian JIKA terlebih dahulu. Setelah semuakondisi JIKA dipenuhi, aturan dipilih untuk mendapatkan kesimpulan. Jika kesimpulan diambil dari keadaan pertama, bukan dari yang terakhir, maka ia akan digunakan sebagai fakta untuk disesuaikan dengan kondisi JIKA aturan

yang lain untuk mendapatkan kesimpulan yang lebih baik. Proses ini berlanjuthingga dicapai kesimpulan terbaik. Yang digambarkan pada Gambar 2.2.

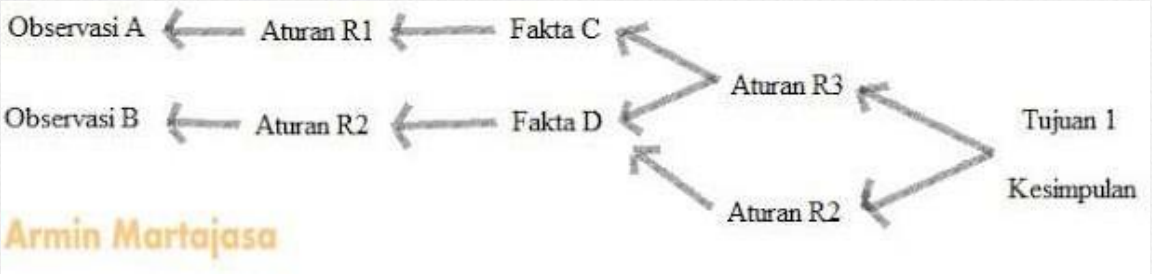


Gambar 2.2. Forward Chaining

b. Backward Chaining

Backward Chaining adalah kebalikan dari Forward Chaining. Pendekatan ini mulai dari kesimpulan dan hipotesis bahwa kesimpulan adalah benar. Mesin inferensi kemudian mengidentifikasi kondisi JIKA yang diperlukan untuk membuat kesimpulan benar dan mencari fakta untuk menguji apakah kondisi JIKA adalah benar. Jika semua kondisi JIKA adalah benar, maka aturan dipilihdari kesimpulan yang dicapai. Jika beberapa kondisi salah, maka aturandibuang dan aturan berikutnya digunakan sebagai hipotesis kedua. Jika tidak ada fakta yang membuktikan bahwa semua kondisi JIKA adalah benar atau salah, maka mesin inferensi terus mencari aturan yang kesimpulannya sesuai dengan kondisi JIKA yang tidak diputuskan untuk bergerak satu langkah ke depan memeriksa kondisi tersebut. Serupa pula, proses chaining ini berlanjut hingga suatu set aturan didapat untuk mencapai kesimpulan atau untuk

membuktikan tidak dapat mencapai kesimpulan. Yang digambarkan padaGambar 2.3.



Gambar 2.3. Backward Chaining

Kedua metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu:

a. Depth-first search, melakukan penelusuran kaidah secara mendalam darisimpul akar bergerak menurun ke tingkat dalam yang berurutan.
b. Breadth-first search, bergerak dari simpul akar, simpul yang ada pada setiaptingkat diuji sebelum pindah ke tingkat selanjutnya.
c. Best-first search, bekerja berdasarkan kombinasi kedua metode sebelumnya.

Mesin inferensi

A. Strategi penyelesaian konflik (conflict resolution strategy)

Strategi penyelesaian konflik dilakukan untuk memilih aturan yang akan diterapkanapabila terdapat lebih dari 1 aturan yang cocok dengan fakta yang terdapat dalammemori kerja. Di antaranya adalah :

1. No duplication. Jangan memicu sebuah aturan dua kali menggunakan fakta/data yang sama, agar tidak ada fakta yang ditambahkan ke memori kerjalebih dari sekali.
2. Recency. Pilih aturan yang menggunakan fakta yang paling baru dalam memorikerja. Hal ini akan membuat sistem dapat melakukan penalaran dengan mengikuti rantai tunggal ketimbang selalu menarik kesimpulan baru menggunakan fakta lama.
3. Specificity. Picu aturan dengan fakta prakondisi yang lebih spesifik (khusus) sebelum aturan yang mengunakan prakondisi lebih umum. Contohnya: jika kita mempunyai aturan “JIKA (burung X) MAKA TAMBAH (dapat_terbang X)” dan “JIKA (burung X) DAN (pinguin X) MAKA TAMBAH (dapat_berenang X)” sertafakta bahwa tweety adalah seekor pinguin, maka lebih baik memicu aturan kedua dan menarik kesimpulan bahwa tweety dapat berenang.
4. Operation priority. Pilih aturan dengan prioritas yang lebih tinggi. Misalnya adafakta (bertemu kambing), (ternak kambing), (bertemu macan), dan (binatang_buas macan), serta dua aturan: “JIKA (bertemu X) DAN (ternak X) MAKA TAMBAH (memberi_makan X)” dan “JIKA (bertemu X) DAN

(binatang_buas X) MAKA TAMBAH (melarikan_diri)", maka kita akan memilih aturan kedua karena lebih tinggi prioritasnya.

B. Sistem Perantaian Maju (Forward Chaining Systems)

Pada sistem perantaian maju, fakta-fakta dalam dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbarui. Aturan dalam sistem merepresentasikan aksi-aksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja, sering disebut aturan kondisi-aksi. Kondisi biasanya berupa pola yang cocok dengan item yang ada di dalam memori kerja, sementara aksi biasanya berupa penambahan atau penghapusan item dalammemori kerja.

Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi (recognise-act). Mula-mula, sistem mencari semua aturan yang kondisinya terdapat di memori kerja, kemudian memilih salah satunya dan menjalankan aksi yang bersesuaian dengan aturan tersebut. Pemilihan aturan yang akan dijalankan (fire) berdasarkanstrategi tetap yang disebut strategi penyelesain konflik. Aksi tersebut menghasilkanmemori kerja baru, dan siklus diulangi lagi sampai tidak ada aturan yang dapat dipicu (fire), atau goal (tujuan) yang dikehendaki sudah terpenuhi.

Sebagai contoh, lihat pada sekumpulan aturan sederhana berikut (Di sini kita memakai kata yang diawali huruf kapital untuk menyatakan suatu variabel. Pada sistem lain, mungkin dipakai cara lain, misalnya menggunakan awalan ? atau ^):

1. JIKA (mengajar X) DAN (mengoreksi_tugas X) MAKA TAMBAH(terlalu_banyak_bekerja X)
2. JIKA (bulan maret) MAKA TAMBAH (mengajar kuncoro)
3. JIKA (bulan maret) MAKA TAMBAH (mengoreksi_tugas kuncoro)
4. JIKA (terlalu_banyak_bekerja X) ATAU (kurang_tidur X) MAKA TAMBAH (mood_kurang_baik X)
5. JIKA (mood_kurang_baik X) MAKA HAPUS (bahagia X)
6. JIKA (mengajar X) MAKA HAPUS (meneliti X)

Kita asumsikan, pada awalnya kita mempunyai memori kerja yang berisi faktaberikut:

(bulan maret)

(bahagia kuncoro)

(meneliti kuncoro)

Sistem Pakar mula-mula akan memeriksa semua aturan yang ada untukmengenali aturan manakah yang dapat memicu aksi, dalam hal ini aturan 2 dan

3. Sistem kemudian memilih salah satu di antara kedua aturan tersebut denganstrategi penyelesaian konflik. Katakanlah aturan 2 yang terpilih, maka fakta (mengajar kuncoro) akan ditambahkan ke dalam memori kerja. Keadaan memori kerja sekarang menjadi:

(mengajar kuncoro)(bulan maret) (bahagia kuncoro) (meneliti kuncoro)

Sekarang siklus dimulai lagi, dan kali ini aturan 3 dan 6 yang kondisinya terpenuhi. Katakanlah aturan 3 yang terpilih dan terpicu, maka fakta (mengoreksi_tugas kuncoro) akan ditambahkan ke dalam memori kerja. Lantaspada siklus ketiga, aturan 1 terpicu, sehingga variabel X akan berisi (bound to)kuncoro, dan fakta (terlalu_banyak_bekerja kuncoro) ditambahkan, sehingga isi memori kerja menjadi:

(terlalu_banyak_bekerja kuncoro)(mengoreksi_tugas kuncoro) (mengajar kuncoro)

(bulan maret) (bahagia kuncoro)(meneliti kuncoro)

Aturan 4 dan 6 dapat diterapkan. Misalkan aturan 4 yang terpicu, sehingga fakta (mood_kurang_baik kuncoro) ditambahkan. Pada siklus berikutnya, aturan 5 terpilih dan dipicu, sehingga fakta (bahagia kuncoro) dihapus dari memori kerja. Kemudian aturan 6 akan terpicu dan fakta (meneliti kuncoro) dihapus pula dari memori kerja menjadi:

(mood_kurang_baik kuncoro) (terlalu_banyak_bekerja kuncoro)

(mengoreksi_tugas kuncoro) (mengajar kuncoro)

(bulan maret)

Urutan aturan yang dipicu bisa jadi sangat vital, terutama di mana aturan-aturanyang ada dapat mengakibatkan terhapusnya item dari memori kerja. Tinjau kasus berikut: andaikan terdapat tambahan aturan pada kumpulan aturan di atas, yaitu:

7. JIKA (bahagia X) MAKA TAMBAH (memberi_nilai_bagus X)

Jika aturan 7 ini terpicu sebelum (bahagia kuncoro) dihapus dari memori, makaSistem Pakar akan berkesimpulan bahwa saya akan memberi nilai bagus. Namun jika aturan 5 terpicu dahulu, maka aturan 7 tidak akan dijalankan (artinya saya tidak akan memberi nilai bagus).

C. Sistem Perantaian Balik (Backward Chaining Systems)

Sejauh ini kita telah melihat bagaimana sistem berbasis aturan dapat digunakan untuk menarik kesimpulan baru dari data yang ada, menambah kesimpulan ini ke dalam memori kerja. Pendekatan ini berguna ketika kita mengetahui semua fakta awalnya, namun tidak dapat menebak konklusi apa yang bisa diambil. Jika kita tahu kesimpulan apa yang seharusnya, atau mempunyai beberapa hipotesis yangspesifik, maka perantaian maju di atas menjadi tidak efisien. Sebagai contoh, jika kita ingin mengetahui apakah saya dalam keadaan mempunyai mood yang baik sekarang, kemungkinan kita akan berulangkali memicu aturan-aturan dan memperbarui memori kerja untuk mengambil kesimpulan apa yang terjadi pada bulan Maret, atau apa yang terjadi jika saya mengajar, yang sebenarnya tidak perluterlalu kita ambil pusing. Dalam hal ini yang diperlukan adalah bagaimana dapat menarik kesimpulan yang relevan dengan tujuan atau goal.

Hal ini dapat dikerjakan dengan perantaian balik dari pernyataan goal (atau hipotesis yang menarik bagi kita). Jika diberikan sebuah goal yang hendak dibuktikan, maka mula-mula sistem akan memeriksa apakah goal tersebut cocok dengan fakta-fakta awal yang dimiliki. Jika ya, maka goal terbukti atau terpenuhi. Jika tidak, maka sistem akan mencari aturan-aturan yang konklusinya (aksinya) cocok dengan goal. Salah satu aturan tersebut akan dipilih, dan sistem kemudian akan mencoba membuktikan fakta-fakta prakondisi aturan tersebut menggunakan prosedur yang sama, yaitu dengan menset prakondisi tersebut sebagai goal baru yang harus dibuktikan.

Perhatikan bahwa pada perantaian balik, sistem tidak perlu memperbarui memori kerja, namun perlu untuk mencatat goal-goal apa saja yang dibuktikan untuk membuktikan goal utama (hipotesis).

Secara prinsip, kita dapat menggunakan aturan-aturan yang sama untuk perantaian maju dan balik. Namun, dalam prakteknya, harus sedikit dimodifikasi. Pada perantaian balik, bagian MAKA dalam aturan biasanya tidak diekspresikan sebagai suatu aksi untuk dijalankan (misalnya TAMBAH atau HAPUS), tetapi suatukeadaan yang bernilai benar jika premisnya (bagian JIKA) bernilai benar. Jadi aturan-aturan di atas diubah menjadi:

1. JIKA (mengajar X) DAN (mengoreksi_tugas X) MAKA (terlalu_banyak_bekerjaX)
2. JIKA (bulan maret) MAKA (mengajar kuncoro)
3. JIKA (bulan maret) MAKA (mengoreksi_tugas kuncoro)
4. JIKA (terlalu_banyak_bekerja X) ATAU (kurang_tidur X) MAKA(mood_kurang_baik X)

5. JIKA (mood_kurang_baik X) MAKA TIDAK BENAR (bahagia X)

   dengan fakta awal:

   (bulan maret) (meneliti kuncoro)

Misalkan kita hendak membuktikan apakah mood sedang kurang baik. Mula-mula kita periksa apakah goal cocok dengan fakta awal. Ternyata tidak ada fakta awal yang menyatakan demikian, sehingga langkah kedua yaitu mencari aturan mana yang mempunyai konklusi (mood_kurang_baik kuncoro). Dalam hal ini aturan yangcocok adalah aturan 4 dengan variabel X diisi dengan (bound to) kuncoro. Dengan demikian kita harus membuktikan bahwa prakondisi aturan ini, (terlalu_banyak_bekerja kuncoro) atau (kurang_tidur kuncoro), salah satunya adalah benar (karena memakai ATAU). Lalu diperiksa aturan mana yang dapat membuktikan bahwa adalah (terlalu_banyak_bekerja kuncoro) benar, ternyata aturan 1, sehingga prakondisinya, (mengajar X) dan (mengoreksi_tugas X), dua- duanya adalah benar (karena memakai DAN). Ternyata menurut aturan 2 dan 3, keduanya bernilai bernilai benar jika (bulan maret) adalah benar. Karena ini sesuaidengan fakta awal, maka keduanya bernilai benar. Karena semua goal sudah terpenuhi maka goal utama (hipotesis) bahwa mood saya sedang kurang baik adalah benar (terpenuhi).

Untuk mencatat goal-goal yang harus dipenuhi/dibuktikan, dapat digunakan stack (tumpukan). Setiap kali ada aturan yang konklusinya cocok dengan goal yang sedang dibuktikan, maka fakta-fakta prakondisi dari aturan tersebut ditaruh (push)ke dalam stack sebagai goal baru. Dan setiap kali goal pada tumpukan teratas terpenuhi atau dapat dibuktikan, maka goal tersebut diambil (pop) dari tumpukan.

Demikian seterusnya sampai tidak ada goal lagi di dalam stack, atau dengan kata lain goal utama (yang terdapat pada tumpukan terbawah) sudah terpenuhi.

D. Ketidakpastian dalam Aturan

Sejauh ini kita menggunakan nilai kebenaran tegas dalam fakta dan aturan yang dipakai, misalnya: jika terlalu banyak bekerja maka pasti mood kurang baik. Pada kenyataanya, seringkali kita tidak bisa membuat aturan yang absolut untuk mengambil kesimpulan secara pasti, misalnya: jika terlalu banyak bekerja maka kemungkinan besar mood kurang baik. Untuk itu, seringkali aturan yang dipakai memiliki nilai kepastian (certainty value). Contohnya: jika terlalu banyak bekerja maka pasti mood kurang baik (kepastian 0,75).