

MESIN INFERENSIA UMUM

MM IRFAN SUBAKTI & ALEXANDER L. ROMY

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Kampus ITS, Jl. Raya ITS – Sukolilo, Surabaya, Indonesia 60111

Telp. (+62-31) 593 9214, Fax (+62-31) 591 3804

E-mail: is@its-sby.edu, romyinf99@plasa.com

Website: <http://is.its-sby.edu>

ABSTRAK

Sistem Berbasis Pengetahuan yang disusun dari rule (aturan) disebut dengan Rule-Based System (Sistem Berbasis Aturan). Bagian tersulit untuk membuat basis pengetahuan adalah akuisisi pengetahuan. Sekali pengetahuan diakuisisi maka ia harus diorganisasi. Perangkat lunak yang menangani pengetahuan ini disebut basis pengetahuan. Basis pengetahuan dapat diorganisasikan dalam pelbagai konfigurasi untuk membantu inferensia (atau reasoning/penjelasan suatu alasan) yang cepat dari pengetahuan.

Terdapat pelbagai cara orang menyelesaikan masalah. “Sumber kekuatan” adalah cara untuk melihat proses penyelesaian masalah. Sumber kekuatan ini mulai dari penjelasan suatu alasan yang murni deduktif yang amat baik ditangani oleh sistem komputer, sampai dengan penjelasan suatu alasan induktif yang lebih sulit untuk dikomputerisasikan.

Teori Kepastian yang diaplikasikan dalam proses penjelasan suatu alasan didapatkan dari ini: Apa (what) yang merupakan hasil dari proses inferensia yang merupakan suatu permulaan saja. Lalu aspek lainnya adalah: Bagaimana (how) itu bisa dilakukan? Mengapa (why) itu bisa terjadi? Semua rule (aturan), asalkan mengikuti sintaks yang ditentukan akan diproses oleh mesin ini. Sehingga mesin ini disebut dengan mesin inferensia umum.

Kata kunci: basis pengetahuan, rule base, akuisisi pengetahuan, inferensia.

1. PENDAHULUAN

Dari pelbagai wilayah bidang Kecerdasan Buatan, Sistem Pakar adalah yang paling menantang. Salah satu dari sistem ini disebut dengan Sistem Berbasis Pengetahuan, dan jika basis pengetahuan tadi disusun oleh rule-rule maka ia disebut Rule-Based System.

Akuisisi pengetahuan adalah proses mengekstraksi, menyusun, dan mengorganisasi pengetahuan dari satu atau lebih sumber. Pelbagai peneliti dan praktisi telah mengidentifikasi proses ini sebagai batu sandungan yang menghambat pengembangan dari Sistem Pakar dan pelbagai sistem Kecerdasan Buatan lainnya.

Sekali pengetahuan diakuisisi maka ia harus diorganisasi. Perangkat lunak yang menangani pengetahuan ini disebut basis pengetahuan. Basis pengetahuan dapat diorganisasikan dalam pelbagai konfigurasi untuk membantu inferensia (atau reasoning/penjelasan suatu alasan) yang cepat dari pengetahuan.

Terdapat pelbagai cara orang mencari penjelasan dari alasan dan menyelesaikan masalah. Salah satu cara yang menarik dalam melihat proses penyelesaian masalah adalah bagaimana seseorang menggambarkan “sumber kekuatan”. Lenat [LEN1992] mengidentifikasi 9 jenis sumber: metode formal, heuristic reasoning, focus, divide and conquer, paralelisme, representasi, analogi, sinergi, dan serendipity. Sumber-sumber kekuatan ini mulai dari mulai dari penjelasan suatu alasan yang murni deduktif yang amat baik ditangani oleh sistem komputer, sampai dengan penjelasan suatu alasan induktif yang lebih sulit untuk dikomputerisasikan. Lenat percaya bahwa masa depan Kecerdasan Buatan terletak pada cara menangani sumber-sumber kekuatan tadi yang merupakan langkah awal untuk eksploitasi.

Proses pencarian yang dipandu oleh suatu pendekatan yang mengacu pada penerjemah aturan (rule interpreter) yang bekerja sebagai berikut:

- Dalam forward chaining, jika klausa premise sesuai dengan suatu situasi, maka proses mencoba untuk mendapatkan konklusi.
- Dalam backward chaining, jika tujuannya sekarang adalah untuk menentukan fakta dalam konklusi, maka proses mencoba untuk menentukan apakah klausa premise sesuai dengan situasi yang ada.

Metode statistik standar didasarkan pada asumsi bahwa ketidakpastian adalah probabilitas dimana suatu kejadian (atau fakta) benar atau salah. Pada Teori Kepastian, ketidakpastian direpresentasikan dalam derajat kepercayaan. Teori Kepastian mengandalkan penggunaan Certainty Factor (CF, tingkat kepercayaan). CF menyatakan derajat kepercayaan dalam suatu kejadian (atau fakta atau hipotesis) didasarkan pada bukti-bukti (atau pendapat pakar). Dari pendekatan ini yang diaplikasikan dalam proses pencarian (inference atau reasoning) maka kita bisa mendapatkan: Apa (what) yang

merupakan hasil dari proses inferensia yang merupakan suatu permulaan saja. Lalu aspek lainnya adalah: Bagaimana (how) itu bisa dilakukan? Mengapa (why) itu bisa terjadi? Semua rule (aturan), asalkan mengikuti sintaks yang ditentukan akan diproses oleh mesin ini. Sehingga mesin ini disebut dengan mesin inferensia umum.

2. IMPLEMENTASI FORWARD DAN BACKWARD CHAINING

Forward dan Backward Chaining merupakan 2 proses yang berbeda. Forward Chaining adalah proses konvergen sedangkan Backward Chaining adalah proses divergen.

Forward chaining.

1. Identifikasi kondisi.
2. Variabel kondisi ditempatkan pada Conclusion Var. Queue dan nilainya dicatat pada Variable List.
3. Pencarian diarahkan untuk menemukan variabel di Base Variabel List dengan nama yang sama dengan nama variabel dalam daftar pertama antrian. Jika ketemu, rule dan clause number dari variabel disimpan ke Clause Variable Pointer, jika tak ketemu maka ke langkah 6.
4. Untuk setiap variabel yang ditemukan dalam IF clause dari rule, jika masih belum ada nilainya maka ini harus diisi dengan nilai baik dari user maupun dari rule itu sendiri.
5. Selanjutnya, pencarian diarahkan untuk mengecek jika fakta yang diinputkan oleh user sama dengan clause dari rule. Jika sama maka tambahkan ke daftar Conclusion Variable Queue dan Result Queue dengan nilai dari THEN clause dari rule, jika tak sama maka ke langkah 6.
6. Jika tak ada lagi statemen IF yang memiliki variabel yang sama dengan yang ada di urutan pertama dari Conclusion Variable Queue, maka urutan pertama tadi dihapus. Jika ada lagi yang lain, kembali ke langkah 3.
7. Jika tak ada lagi apa-apa di Conclusion Variable Queue, maka pencarian berhenti. Jika masih ada maka kembali ke langkah 3.

Backward chaining.

1. Identifikasi konklusi.
2. Ada 2 kemungkinan:
 - a. Pencarian diarahkan untuk mencari variabel di bagian THEN dari rule dalam Base Variable List dengan nama yang sama dengan nama yang diinputkan oleh user. Jika ketemu, rule number dan clause number dari variabel disimpan ke Conclusion Stack. Jika tak ketemu maka beri tahu user bahwa pencarian gagal.

- b. Pencarian diarahkan untuk mencari variabel di bagian THEN dari rule dalam Base Variable List dengan nama yang sama dengan nama yang ada dalam daftar pertama di Conclusion Stack. Jika ketemu, rule number dan clause number dari variabel disimpan di Conclusion Stack. Jika tak ada maka nama di daftar tadi dihapus.
3. Setiap variabel dalam bagian IF dari rule harus diisi dengan suatu nilai. Nilai ini dapat berasal dari input user atau dari rule itu sendiri.
4. Jika suatu variabel adalah variabel konklusi maka tambahkan ke daftar dalam Conclusion Stack dengan nilai dari rule number pada variabel tadi, lalu kembali ke langkah 3.
5. Selanjutnya, pencarian diarahkan untuk mengecek jika setiap fakta yang diinputkan user sama dengan clause dari rule. Jika sama maka tambahkan ke daftar di Result Queue dengan nilai dari THEN clause dari rule, dan juga hapus dari daftar di Conclusion Stack. Jika tak sama maka kembali ke langkah 2b.
6. Jika variabel dari hasil terakhir adalah variabel konklusi maka ikuti langkah 4.
7. Jika tak ada lagi di daftar Conclusion Stack, maka pencarian berhenti. Jika masih maka ke langkah 2b.

Ada pelbagai cara untuk mengembangkan kepresisian dan menjadikannya lebih bernilai ([BAU1990]).

Certainty Factors (CF)

Ini digunakan untuk menyelesaikan masalah prioritas. Pada dasarnya, setiap clause dalam setiap rule harus memiliki CF seperti contoh di bawah.

Pembuat rule bebas menentukan nilai CF, disarankan dengan nilai yang masuk akal. Kalkulasi final CF digunakan untuk menentukan apakah rule tersebut yang dicek pertama kalai baik dalam proses Forward maupun Backward Chaining. Perhitungan akhir ini bisa berbeda, tergantung pada jenis relasi setiap clause (AND, OR, atau yang lain).

Pertama, aturan CF untuk AND:

```
Rule 10 IF Clause A AND CF=75
           Clause B      CF=80
           THEN Clause C  CF=85;
Rule 10 CF=(Min(75,80,85)*85)/100=64
```

Kedua, aturan CF untuk OR:

```
IF dollar=rise      CF 80
   OR rupiah=fall   CF 60
THEN crisis=rise    CF 90;
Maka CF=80+60-(80*60)/100=140-48=92
```

CF (rule) = $92 \times 90 = 8280$, lalu dibagi dengan $100 = 8280/100 = 82,8$, setelah dibulatkan didapat 83.

Ketiga, aturan CF untuk AND/OR:

```
Rule A IF activity=letters
      OR activity=resume
      AND cost=low
      THEN word_processor=D CF 90;
```

Terlihat bahwa terdapat kondisi OR dan AND. Untuk mendapatkan CF-nya, maka rule ini dapat dipecah menjadi (CF yang dicetak tebal adalah diinputkan):

```
Rule A1: IF activity=letter CF 90
        AND cost=low CF 80
        THEN word_processor=D CF 90 OR
Rule A2: IF activity=resume CF 70
        AND cost=low CF 80
        THEN word_processor=D CF 90;
```

CF kondisi pada Rule A1 = $\min(90,80)=80$
 CF untuk Rule A1 = $80 \times 90 = 7200/100 = 72$
 CF kondisi pada A2 = $\min(70,80)=70$
 CF untuk Rule A2 = $70 \times 90 = 6300/100 = 63$
 CF untuk kedua rule didapat dengan aturan CF untuk OR sehingga didapat:
 $72 + 63 - (72 \times 63) = 135 - 4536/100 = 135 - 45 = 90$.

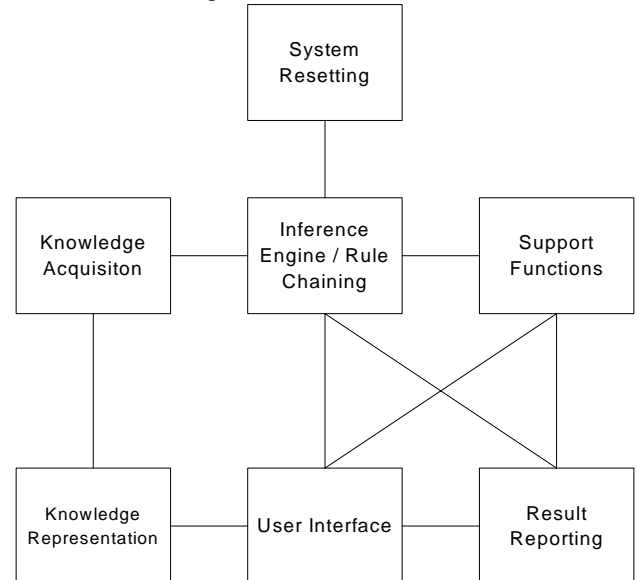
Rule attributes

Atribut yang lain dapat ditambahkan dalam rule, misalnya:

- **ELSE**
 Atribut ini digunakan sebagai alternatif lain, disaat tak ada kondisi yang memenuhi suatu masalah. Biasanya clause yang ditempatkan setelah ELSE adalah clause yang berlawanan nilainya dengan yang setelah IF. Contoh:
 IF day = true then sun = shine
 ELSE night = true then moon = shine
- **BECAUSE**
 Atribut ini ditempatkan setelah semua atribut yang lain dalam rule. Merupakan penjelasan dari suatu rule. Misal:
 IF day = true then sun = shine
 ELSE night = true then moon = shine
 BECAUSE "the world turns around forever"
- **WHENEVER**
 Atribut ini ditempatkan sebelum semua atribut yang lain dalam rule. Jika ini ditempatkan dalam rule, artinya rule yang berkaitan harus dicek setiap kali pencarian dilakukan saat proses Forward/Backward Chaining. Contoh:
 WHENEVER day = true
 IF sun = shine then rain = false

3. STRUKTUR PROGRAM DAN APLIKASI DATABASE

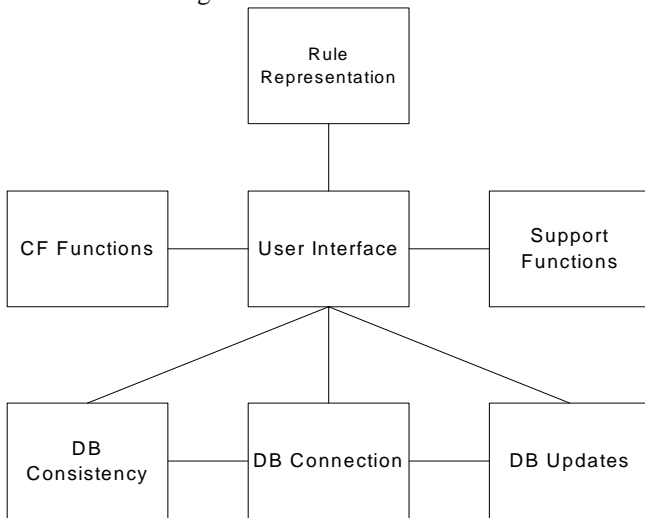
Diagram berikut menunjukkan struktur Expert System untuk pemecahan masalah dengan metode Forward dan Backward chaining.



Keterangan:

- **Knowledge Acquisiton**
 Komponen yang berisi fungsi-fungsi untuk pengambilan ilmu berupa rule-rule dari database sesuai query.
- **Knowledge Representation**
 Komponen yang berisi fungsi-fungsi untuk representasi rule-rule dalam bentuk yang mudah dipahami oleh user.
- **Inference Engine / Rule Chaining**
 Komponen yang berisi fungsi-fungsi untuk proses pemecahan masalah dengan menggunakan metode Forward atau Backward chaining.
- **User Interface**
 Komponen yang berisi fungsi-fungsi untuk interaksi user dengan aplikasi yang berupa penekanan tombol mouse dan keyboard.
- **Support Functions**
 Komponen yang berisi fungsi-fungsi untuk mendukung jalannya proses inferencing/rule chaining.
- **Result Reporting**
 Komponen yang berisi fungsi-fungsi untuk penampilan hasil proses inferencing/rule chaining, termasuk penjelasan What, Why, How.
- **System Resetting**
 Komponen yang berisi fungsi-fungsi untuk mereset sistem agar proses pencarian dapat diulangi dari awal.

Diagram berikut menunjukkan struktur Aplikasi Database untuk pemecahan masalah dengan metode Forward dan Backward chaining.



Keterangan :

- **CF Functions**
Komponen yang berisi fungsi-fungsi untuk menghitung nilai CF suatu rule.
- **DB Consistency**
Komponen yang berisi fungsi-fungsi untuk mengupdate database sesuai perubahan yang terjadi pada database rule.
- **User Interface**
Komponen yang berisi fungsi-fungsi untuk interaksi user dengan aplikasi yang berupa penekanan tombol mouse dan keyboard.
- **DB Connection**
Komponen yang berisi fungsi-fungsi untuk membuat dan memutuskan koneksi serta melakukan query.
- **Support Functions**
Komponen yang berisi fungsi-fungsi untuk mendukung jalannya mekanisme aplikasi database.
- **DB Updates**
Komponen yang berisi fungsi-fungsi untuk mengupdate database sesuai input yang diberikan oleh user.
- **Rule Representation**
Komponen yang berisi fungsi-fungsi untuk menampilkan rule-rule dari suatu ruleset dalam susunan yang mudah untuk diakses.

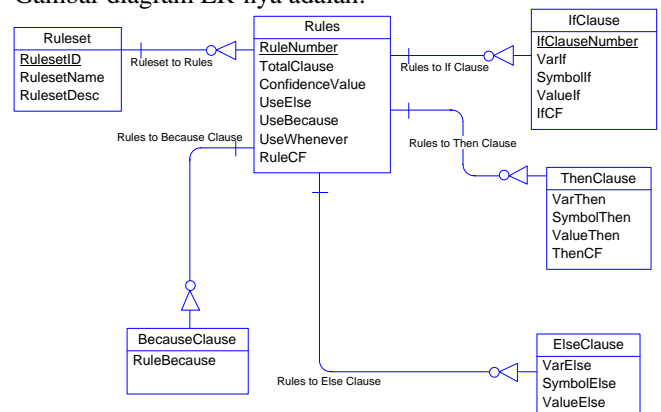
4. PEMODELAN DATA

Database yang dipakai adalah database Microsoft Access 2000 dengan nama file *RuleDB.mdb*. Tabel-tabel pada database tersebut adalah :

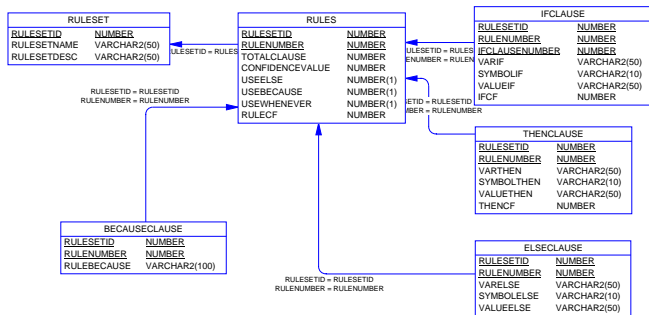
Tabel	Field	Type Data	Key (PK/FK)
Ruleset	RulesetID	Autonumber	PK
	RulesetName	Text(50)	
	RulesetDesc	Text(100)	
Rules	RulesetID	Number	PK & FK
	RuleNumber	Number	PK
	TotalClause	Number	
	ConfidenceValue	Number	
	UseElse	Boolean	
	UseBecause	Boolean	
	UseWhenever	Boolean	
	RuleCF	Number	
IfClause	RulesetID	Number	PK & FK
	RuleNumber	Number	PK & FK
	IfClauseNumber	Number	PK
	VarIf	Text(50)	
	SymbolIf	Text(10)	
	ValueIf	Text(50)	
	IfCF	Number	
ThenClause	RulesetID	Number	PK & FK
	RuleNumber	Number	PK & FK
	VarThen	Text(50)	
	SymbolThen	Text(10)	
	ValueThen	Text(50)	
	ThenCF	Number	
ElseClause	RulesetID	Number	PK & FK
	RuleNumber	Number	PK & FK
	VarElse	Text(50)	
	SymbolElse	Text(10)	
	ValueElse	Text(50)	
BecauseClause	RulesetID	Number	PK & FK
	RuleNumber	Number	PK & FK
	RuleBecause	Text(100)	

5. DIAGRAM ENTITY RELATIONAL (ER) DAN MODEL FISIK

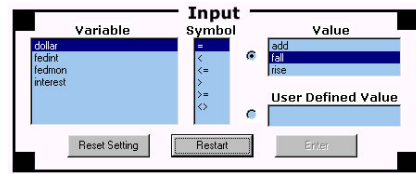
Gambar diagram ER-nya adalah:



Sedangkan gambar model fisiknya adalah:



melakukan restart selama fase “Interaksi User” pada pemodelan proses.



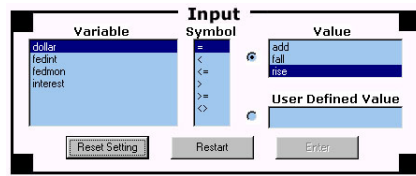
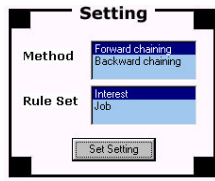
4. Reset proses

User menekan tombol “Reset Setting” sebagai tanda bahwa seluruh proses dimulai lagi dari awal, dan user diberi pilihan metode dan ruleset lagi. User dapat melakukan reset selama fase “Interaksi User” pada pemodelan proses.

6. DETIL INPUT

1. Input method dan ruleset

User memilih metode inferencing yang ingin dipakai yaitu forward atau backward chaining, dan juga memilih ruleset (kumpulan rule yang saling berhubungan) yang ingin dipakai. Pada gambar dibawah, user memilih forward chaining dan ruleset interest.



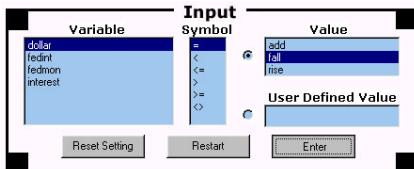
5. Menutup form

User menutup form dengan menekan tombol ‘X’ pada pojok kanan atas dari window Expert System. User dapat menutup form selama fase “Interaksi User” pada pemodelan proses.



2. Memasukkan Fakta

User memasukkan fakta yang terdiri dari 3 bagian utama yaitu variabel, simbol dan value. Nilai dari variabel dan simbol sudah ditentukan dan tidak bisa diubah secara langsung. Sedangkan untuk nilai value, user diberi kebebasan untuk memasukkan nilai dari pilihan yang tersedia pada listbox atau memasukkan teks sembarang ke dalam text box (user defined value). Pada gambar dibawah, user memasukkan fakta : dollar = fall. Input dimasukkan/diakhiri dengan penekanan tombol “Enter”.



7. DETIL OUTPUT

1. Display list method

List ini ditampilkan ketika window “Expert System” pertama kali dijalankan. Yang ditampilkan adalah daftar metode inferencing, yaitu forward dan backward chaining.



2. Display list ruleset

List ini ditampilkan ketika window “Expert System” pertama kali dijalankan. Yang ditampilkan adalah daftar ruleset yang ada pada database, misalnya interest dan job.



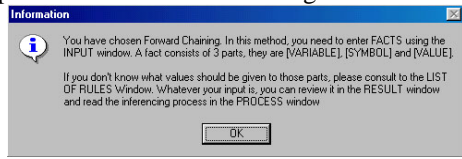
3. Restart proses

User menekan tombol “Restart” sebagai tanda bahwa proses inferencing dimulai lagi dari awal dengan menggunakan metode dan ruleset yang sama. User dapat

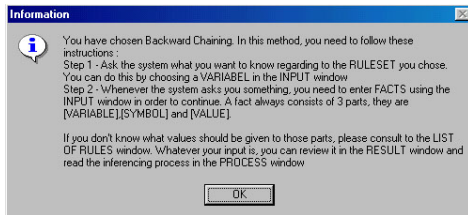
3. Display pesan awal

Pesan ini ditampilkan ketika user telah memilih metode dan ruleset yang diinginkan. Yang ditampilkan adalah

petunjuk singkat mengenai cara pemakaian aplikasi. Untuk pesan awal Forward Chaining:

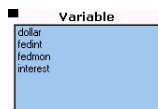


Sedangkan pesan awal Backward Chaining:



4. Display list variabel

List ini ditampilkan ketika user telah memilih metode dan ruleset yang diinginkan. Yang ditampilkan adalah variabel yang terdapat pada klausa IF (untuk forward dan backward chaining) dan klausa THEN (khusus untuk backward chaining) pada ruleset.



5. Display list symbol

List ini ditampilkan ketika user telah memilih metode dan ruleset yang diinginkan. Yang ditampilkan adalah simbol yang terdapat pada klausa IF pada ruleset.



6. Display list value

List ini ditampilkan ketika user telah memilih metode dan ruleset yang diinginkan. Yang ditampilkan adalah value yang terdapat pada klausa IF pada ruleset.



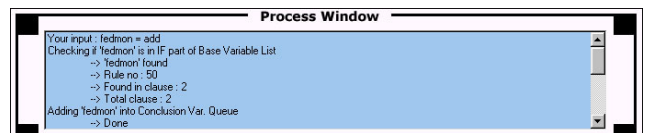
7. List rule untuk suatu ruleset

List ini ditampilkan ketika user telah memilih metode dan ruleset yang diinginkan. Yang ditampilkan adalah nama metode, ruleset, deskripsi ruleset dan list dari rule.



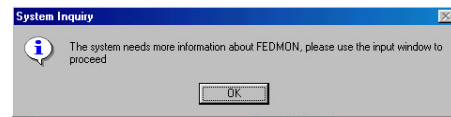
8. Laporan proses inferencing

Laporan proses inferencing merupakan display langkah demi langkah hal-hal apa saja yang dilakukan oleh sistem dalam memproses input dari user sesuai dengan metode dan ruleset yang dipilih. Laporan ini ditampilkan pada Process window.



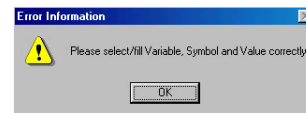
9. Informasi untuk user

Informasi ini berupa tampilan kotak pesan yang bertujuan meminta user untuk memberi informasi tambahan kepada sistem agar proses forward/backward chaining dapat dilanjutkan.

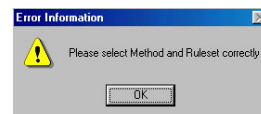


10. Pesan Kesalahan

Pesan kesalahan ditampilkan jika ada kesalahan yang dilakukan user sehingga proses tidak dapat dilanjutkan. Contoh pesan kesalahan A:

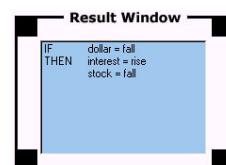


Sedangkan contoh pesan kesalahan B:



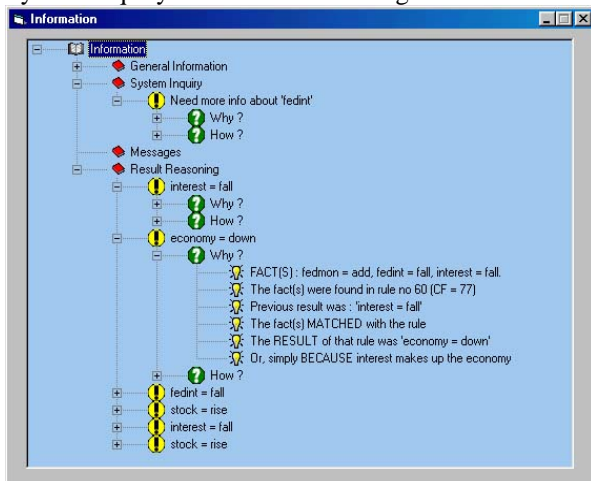
11. Kesimpulan

Kesimpulan ditampilkan ketika proses forward atau backward telah selesai. Yang ditampilkan adalah input dari user ditambah hasil pencarian sistem sesuai dengan metode dan ruleset yang dipilih.



12. Window Why dan How

Window ini menjelaskan mengapa dan bagaimana suatu keputusan diambil oleh sistem. Penjelasan meliputi System Inquiry dan Result Reasoning.



8. EVALUASI

- Metode Forward dan Backward Chaining ternyata bisa diterapkan dengan mudah.
- Akuisisi pengetahuan diperoleh dari ruleset yang terdapat dalam database. Keandalan program ini dapat dijamin selama rule-rule yang dimasukkan dalam database juga benar.
- Aplikasi Database memudahkan user untuk mengedit, mengupdate dan menghapus rule-rule.
- Adanya adanya Confidence Factor (CF) menyebabkan prioritas rule bukan berdasarkan urutannya tapi berdasarkan nilai dari CF-nya.

- Kenyamanan dalam penciptaan dan penggunaan rule bisa didapat dari penambahan atribut rule berupa ELSE, BECAUSE dan WHENEVER.

9. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil adalah:

- Penerapan metode Forward dan Backward Chaining ternyata tidak sulit, bisa dilakukan dengan mudah.
- Interaksi dengan user yang mudah menjadikannya nyaman dipakai dan dapat meminimalkan terjadinya kesalahan.
- Semua rule, asalkan sesuai dengan sintaks yang disepakati dapat diproses dengan baik (inferensi dan reasoning serta penjelasannya).

Saran untuk pengembangannya adalah:

- Untuk mempermudah pengembangan selanjutnya, bisa didesain ulang dengan menggunakan pendekatan berorientasi objek.

10. DAFTAR PUSTAKA

1. [BAU1990] Baur G.R., Pigford D.V.; *Expert Systems for Business: Concepts and Applications*, Boyd & Fraser Publishing Company, Boston-USA, 1990.
2. [IGN1991] Ignizio J.P., *Introduction to Expert Systems: The Development and Implementation of Rule-based Expert Systems*, McGraw-Hill International Editions, 1991.
3. [LEN1982] Lenat D.B., "The Ubiquity of Discovery." *Artificial Intelligence*, Vol. 19, No. 2, 1982.
4. [SUB2003] Subakti, Irfan, *Knowledge-based System: a Guideline Book*, Department of Informatics, Faculty of Information Technology, Sepuluh Nopember Institute of Technology, Surabaya, 2003.