

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1. Persyaratan *Software* dan *Hardware*

Implementasi program adalah implementasi dari analisis dan desain sistem yang telah dibuat. Sebelumnya harus dipersiapkan kebutuhan-kebutuhan dari program yang akan diimplementasikan baik dari segi perangkat keras maupun perangkat lunak komputer.

4.1.1. Kebutuhan *Software*

Perangkat lunak yang digunakan dalam Tugas Akhir ini adalah :

- a. Sistem Operasi *Microsoft Windows 98* atau lebih.
- b. Database menggunakan *Microsoft Access 2000*.
- c. Bahasa Pemrograman menggunakan *Delphi 5*.

4.1.2. Kebutuhan *Hardware*

Perangkat keras yang digunakan dalam Tugas Akhir ini adalah :

- a. *Processor* minimal *Pentium III*.
- b. Memori minimal *256 Mb*.
- c. *Harddisk* berkapasitas minimal *20 Gb*.
- d. *VGA* minimal *4 Mb*.
- e. Monitor *SVGA*.
- f. *Compatible Mouse 2 Button*.
- g. *Keyboard* win *101 keys*.

4.2. Prosedur Implementasi Sistem

Setelah dilakukan perancangan sistem maka dilanjutkan tahap implementasi. Tahap ini merupakan tahapan untuk mengimplementasikan sistem yang telah dibuat sebelumnya sehingga sistem dapat dibangun dengan baik dan siap untuk diuji apakah sistem yang telah didesain atau yang telah diimplementasikan berjalan sesuai yang diharapkan sehingga dapat diketahui sejauh mana keberhasilan suatu sistem dan sejauh mana kesalahan yang ada.

Berikut ini adalah langkah-langkah dalam proses implementasi :

1. Maintenance rule.

User memasukkan rule-rule yang sudah terbentuk.

2. Verifikasi rule.

Rule yang sudah dimasukkan di verifikasi untuk dilihat apakah ada rule yang perlu dibenahi.

3. Proses Inference

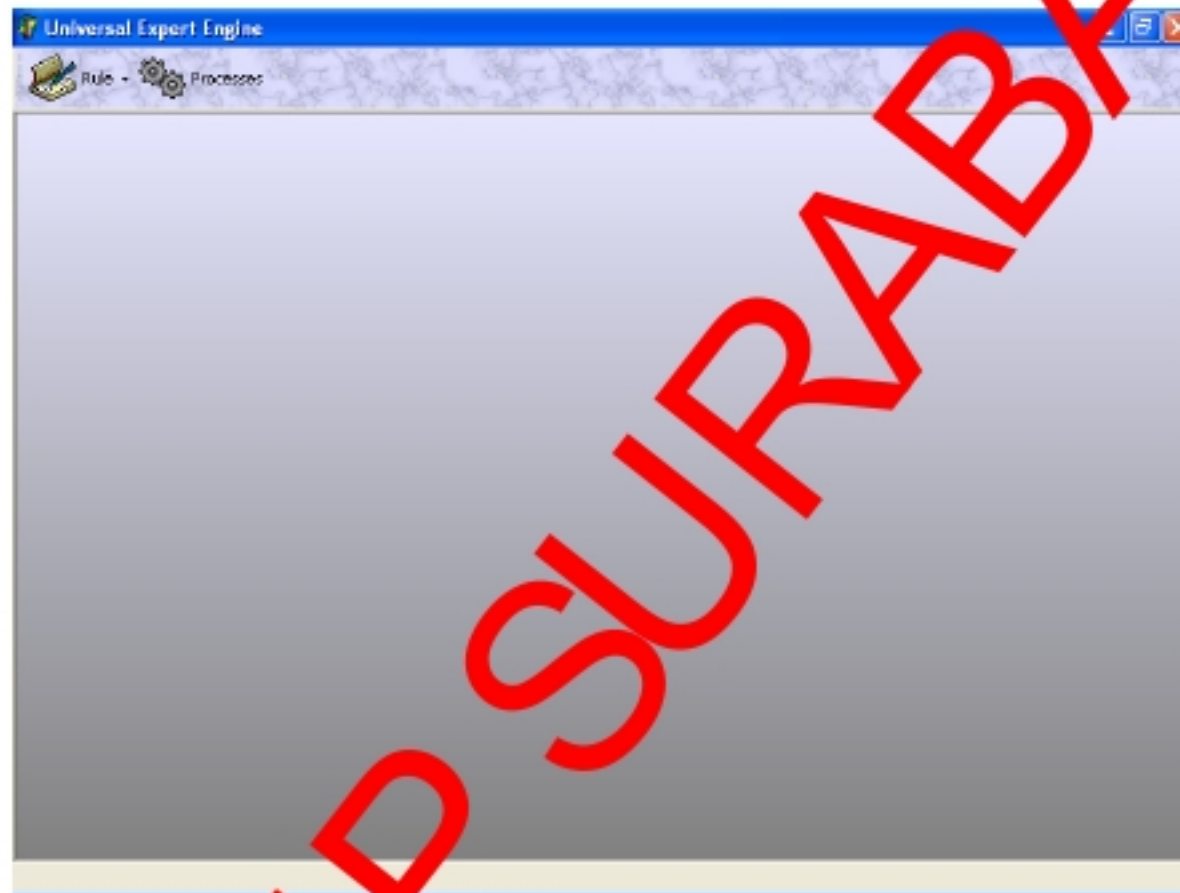
Apabila rule sudah lengkap maka rule tersebut sudah dapat diproses berdasarkan metode inference yang dipilih untuk menghasilkan hasil dari proses universal expert engine.

4. Hasil Universal Expert Engine.

Hasil *output* dari proses *Universal Expert Engine* adalah Sistem pakar yang berdasarkan kategori yang dipilih sebelumnya.

4.3 Implementasi Sistem

Sistem ini mempunyai dua buah menu utama yang terletak di baris menu utama yaitu menu *Rule* dan *Processes*. Setiap menu akan melakukan suatu proses tertentu. Gambar 4.1. merupakan tampilan awal dari perangkat lunak yang dipergunakan untuk mengakses sistem yang dibuat.



Gambar 4.1 Menu Utama

Berikut ini akan dijelaskan tentang menu utama beserta sub proses yang dimiliki oleh setiap menu tersebut.

4.3.1 Menu Rule

A Menu Rule Maintenance.

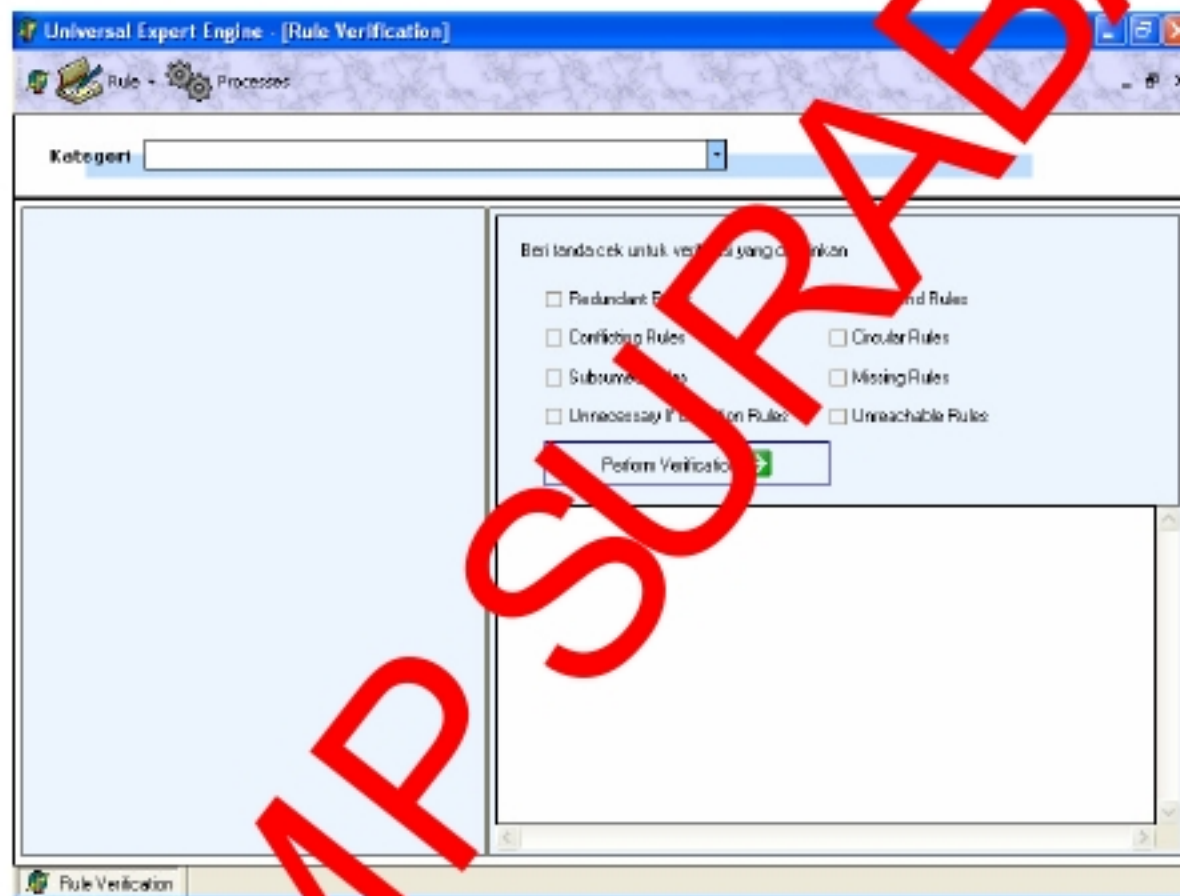
Menu ini digunakan untuk menginputkan data rule. Rule yang sudah dimasukkan akan disimpan dalam rulebase. Dari rule maintenance form, ini data rule dapat diedit maupun dihapus. Untuk menu rule maintenance form dilihat pada gambar 4.2.

The screenshot shows the 'Rule Management Form' window. It features a 'Kategori' dropdown menu. The main area is divided into two sections: 'IF Clause' and 'Then Clause'. Each section contains a 'Prompt' field and a 'CF' (Confidence Factor) field. The 'Then Clause' section also includes an 'OK' button. At the bottom, there is a checkbox labeled 'Remember rule penutup pada topik ini' which is checked. A 'Category Menu' is visible in the bottom right corner, containing options: 'New Category', 'Rename Category', and 'Delete Category'. At the very bottom, there are buttons for 'New', 'Delete', and 'Save Rule'.

Gambar 4.2 Rule Maintenance Form

B Menu Rule Verification.

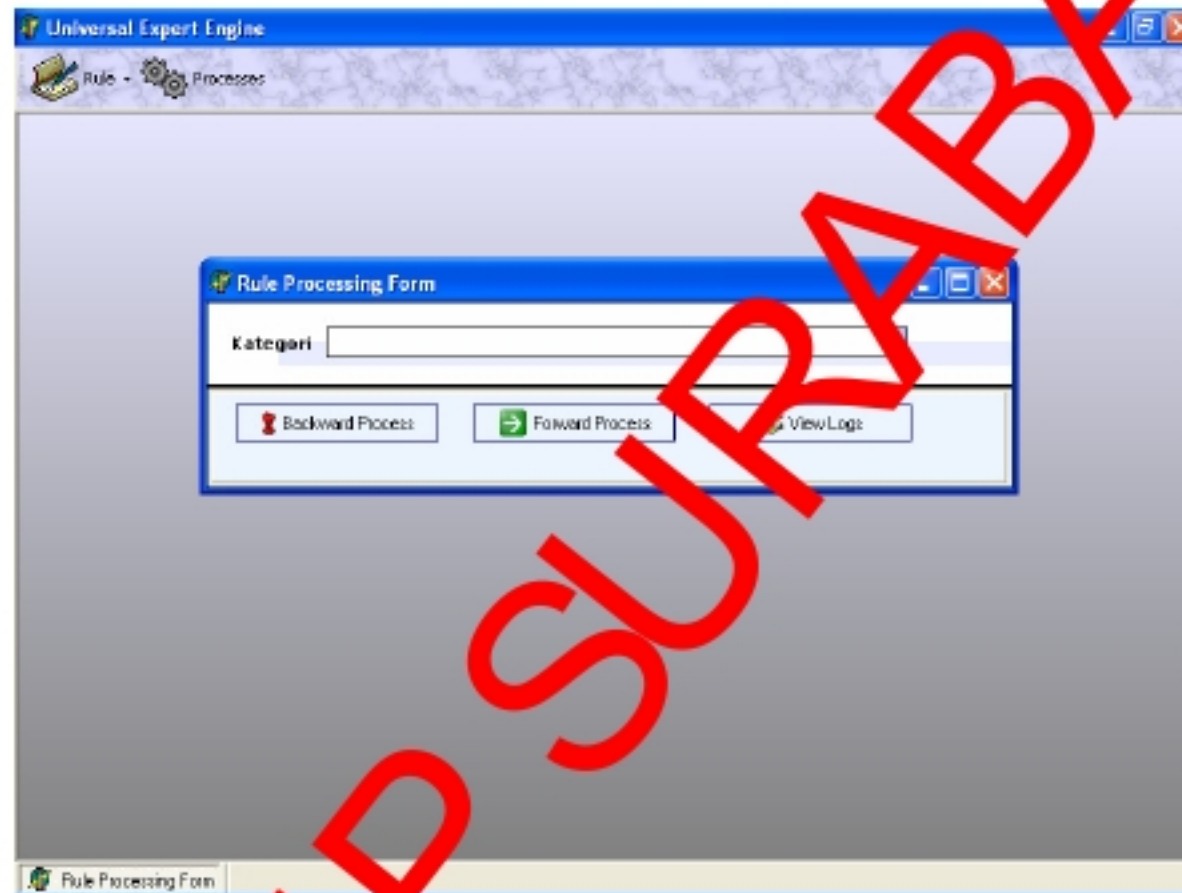
Menu ini digunakan untuk proses verifikasi rule. Dipilih jenis kategori rule yang akan di verifikasi. Rule yang dipilih dapat dilihat dari tampilan treeview. Proses verifikasi dapat dijalankan dengan memilih jenis verifikasi yang diinginkan. Untuk form menu rule verification dapat dilihat pada gambar 4.3.



Gambar 4.3 Rule Verification Form

4.3.2 Menu Processes.

Menu ini digunakan untuk melakukan proses inference. Dengan memilih kategori dan jenis inference yang diinginkan. Untuk form menu Processes bisa dilihat pada gambar 4.4



Gambar 4.4 Processes Form

4.4 Uji Coba dan Evaluasi Sistem

Uji coba validasi ini bertujuan untuk memastikan bahwa sistem yang telah dibuat dengan benar telah sesuai dengan kebutuhan atau tujuan yang diharapkan.

4.4.1 Uji Coba Verifikasi – Redundant Rule :

Dilakukan dengan memberikan rule yang mengandung redundant rule.

RULE 181 : IF alcohol indicated = yes

AND meal = formal

THEN drink = wine

RULE 182 : IF meal = formal

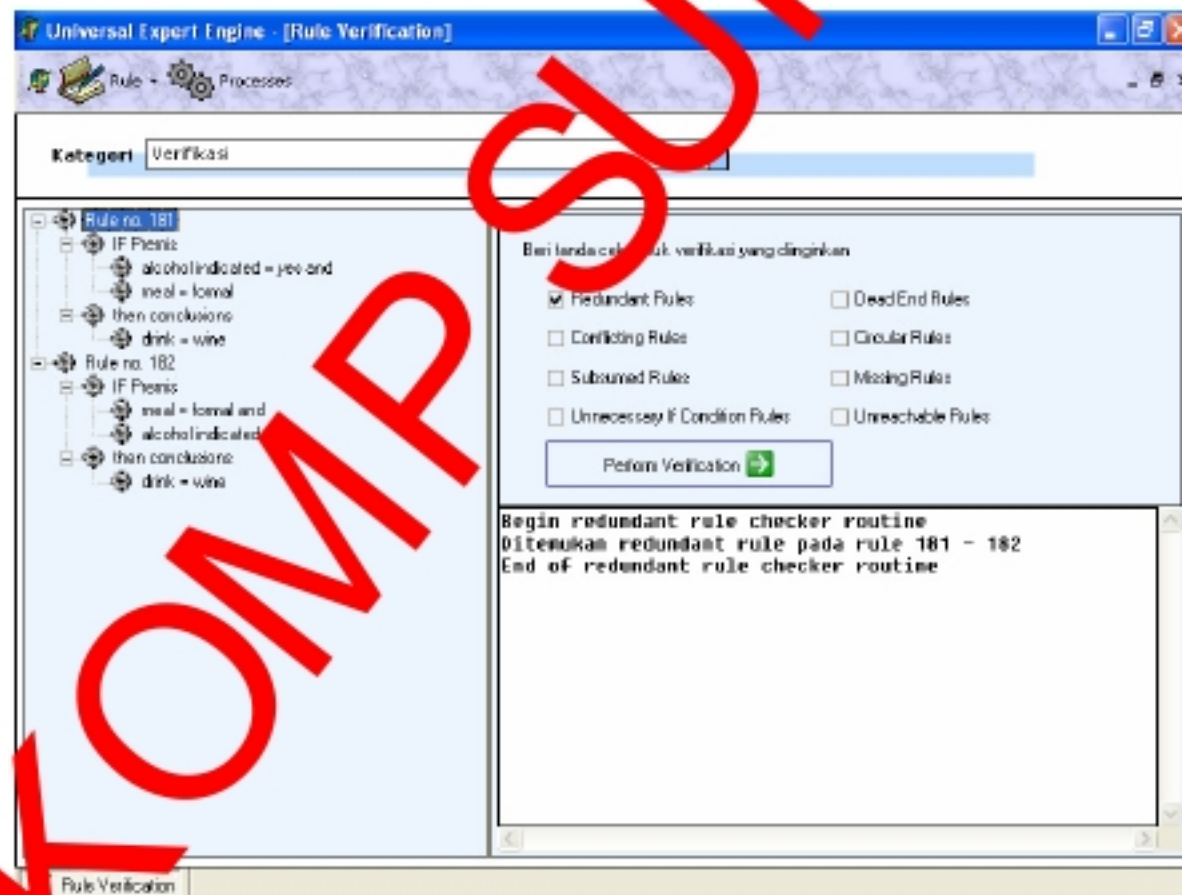
AND alcohol indicated = yes

THEN drink = wine

Dari proses verifikasi yang dilakukan, maka sistem akan menghasilkan pemberitahuan :

```
Begin redundant rule checker routine
Ditemukan redundant rule pada rule 181 - 182
End of redundant rule checker routine
```

Hasil proses verifikasi seperti yang di tunjukkan pada Gambar 4.5



Gambar 4.5 Rule Verification – Redundant Rule

4.4.2 Uji Coba Verifikasi – Conflicting Rule :

Dilakukan dengan memberikan rule yang mengandung conflicting rule.

RULE 181 : IF alcohol indicated = yes

AND guest = boss

THEN drink = wine

RULE 182 : IF alcohol indicated = yes

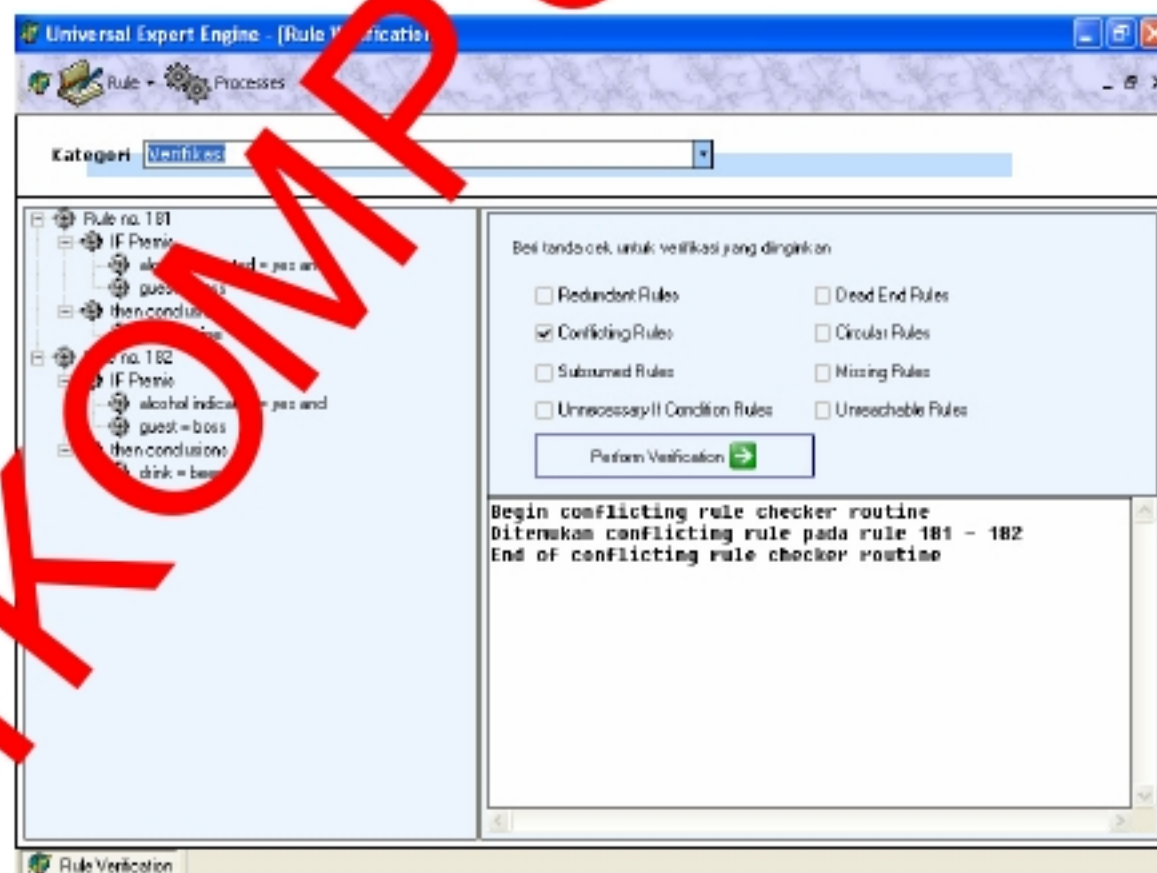
AND guest = boss

THEN drink = beer

Dari proses verifikasi yang dilakukan maka sistem akan menghasilkan pemberitahuan :

```
Begin conflicting rule checker routine
Ditemukan conflicting rule pada rule 181 - 182
End of conflicting rule checker routine
```

Hasil proses verifikasi seperti yang ditunjukkan pada Gambar 4.6



Gambar 4.6 Rule Verification – Conflicting Rule

4.4.3 Uji Coba Verifikasi – Subsumed Rule :

Dilakukan dengan memberikan rule yang mengandung subsumed rule.

RULE 181 : IF alcohol indicated = yes

AND guest = boss

THEN drink = wine

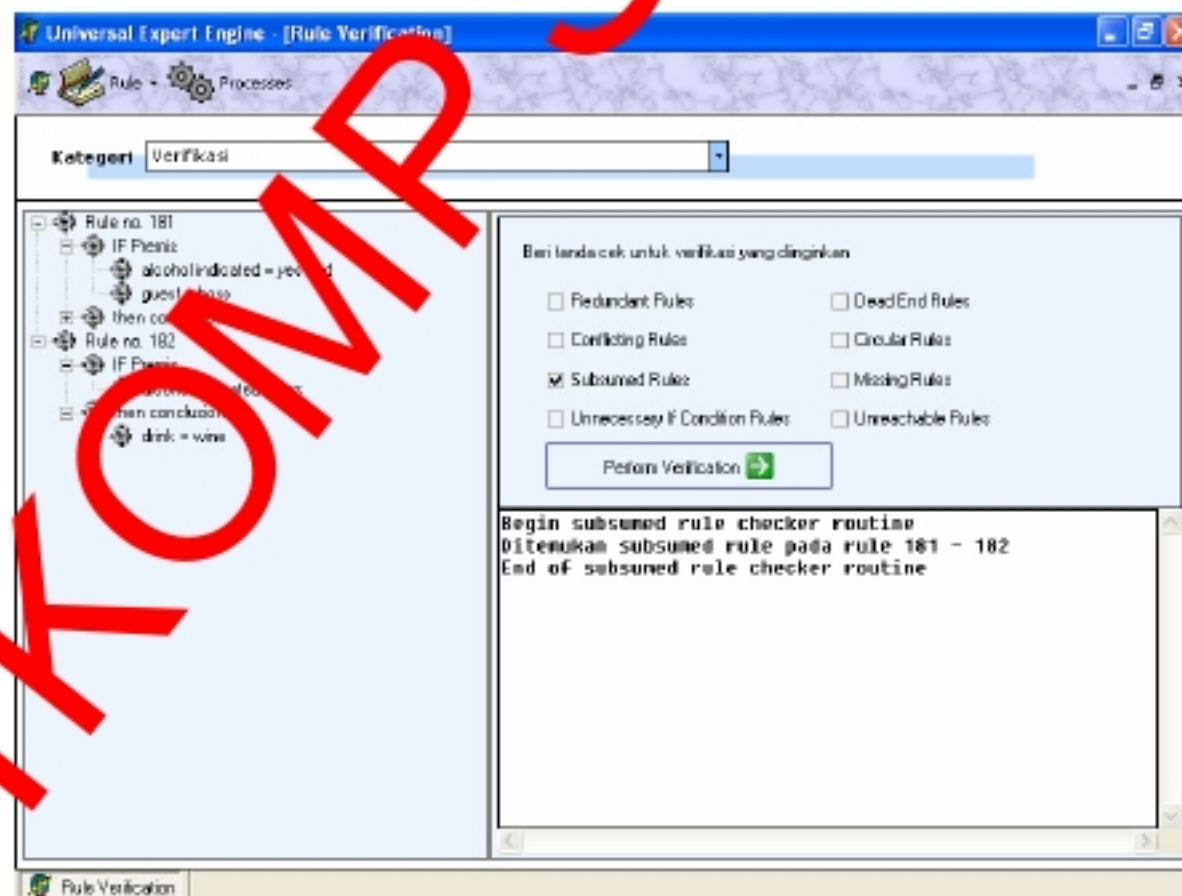
RULE 182 : IF alcohol indicated = yes

THEN drink = wine

Dari proses verifikasi yang dilakukan, maka sistem akan menghasilkan pemberitahuan :

```
Begin subsumed rule checker routine
Ditemukan subsumed rule pada rule 181 - 182
End of subsumed rule checker routine
```

Hasil proses verifikasi seperti yang ditunjukkan pada Gambar 4.7



Gambar 4.7 Rule Verification – Subsumed Rule

4.4.4 Uji Coba Verifikasi – Unnecessary If Condition Rule :

Dilakukan dengan memberikan rule yang mengandung unnecessary If condition rule.

RULE 182 : IF alcohol indicated = yes

AND guest = boss

THEN drink = beer

RULE 183 : IF alcohol indicated = yes

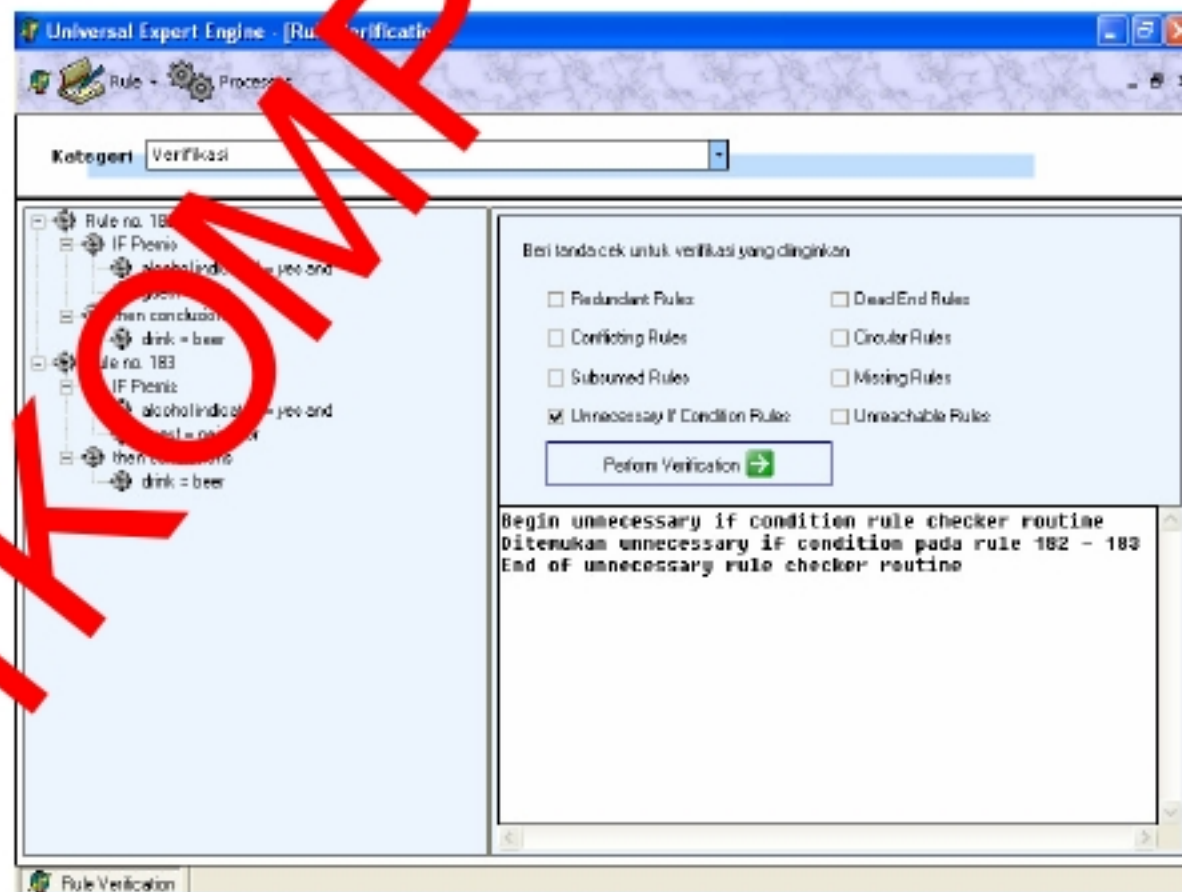
AND guest = neighbor

THEN drink = beer

Dari proses verifikasi yang dilakukan maka sistem akan menghasilkan pemberitahuan :

```
Begin unnecessary if condition rule checker routine
Ditemukan unnecessary if condition pada rule 182 – 183
End of unnecessary if condition rule checker routine
```

Hasil proses verifikasi seperti yang ditunjukkan pada Gambar 4.8



Gambar 4.8 Rule Verification – Unnecessary If Condition Rule

4.4.5 Uji Coba Verifikasi – Dead End, Missing, Unreachable Rule :

Dilakukan dengan memberikan rule yang mengandung dead end, missing dan unreachable rule.

RULE 181 : IF drink = wine

AND dinner = fish

THEN wine type = white

RULE 182 : IF alcohol indicated = yes

AND guest = boss

THEN drink = wine

RULE 183 : IF wine type = white

THEN guest go home = true

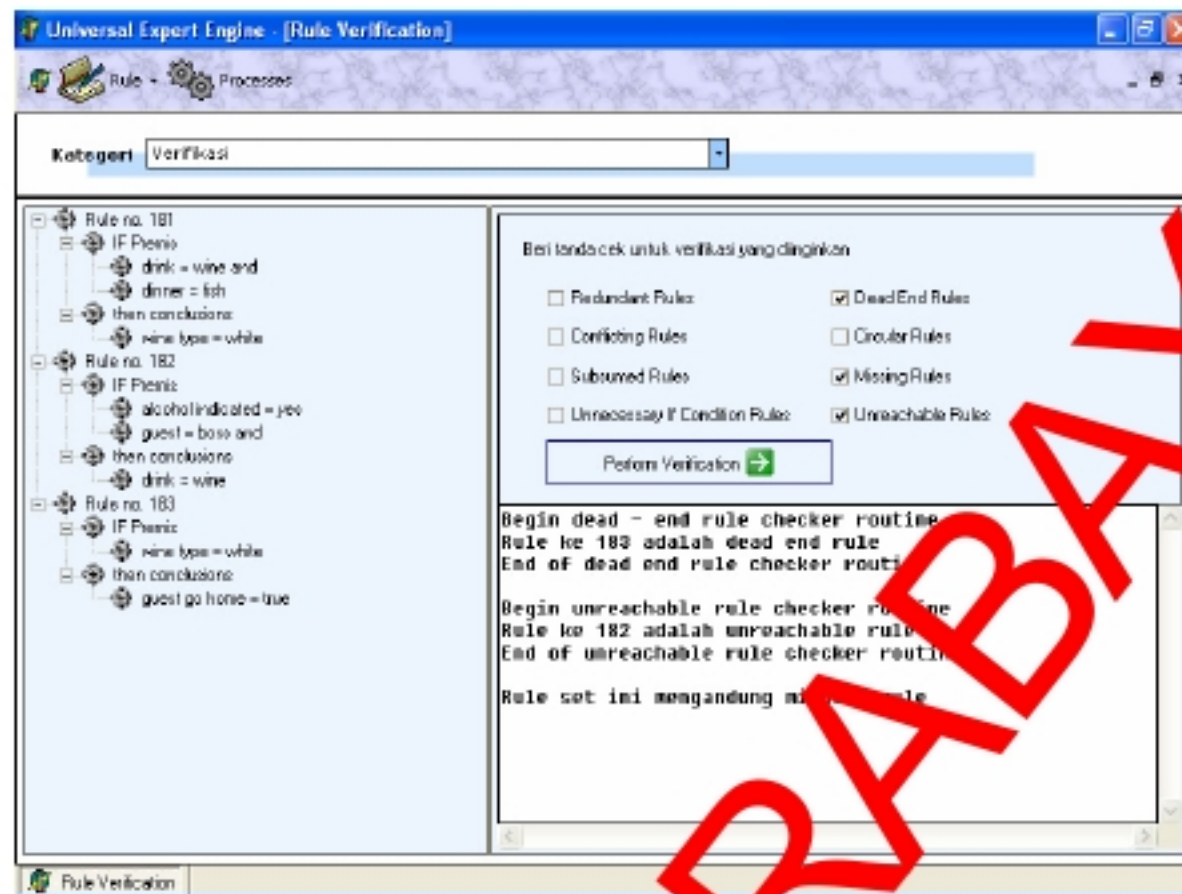
Dari proses verifikasi yang dilakukan maka sistem akan menghasilkan pemberitahuan :

```
Begin dead end rule checker routine
Rule ke 183 adalah dead end rule
End of dead end rule checker routine
```

```
Begin unreachable rule checker routine
Rule ke 182 adalah unreachable rule
End of unreachable rule checker routine
```

```
Rule set ini mengandung missing rule
```

Hasil proses verifikasi seperti yang di tunjukkan pada Gambar 4.9



Gambar 4.9 Rule Verification – Dead End, Missing dan Unreachable Rule

4.4.6 Uji Coba Verifikasi – Circular Rule :

Dilakukan dengan memberikan rule yang mengandung circular rule.

RULE 181 : IF alcohol indicated = yes

AND meal = formal

THEN drink = wine

RULE 182 : IF alcohol indicated = yes

AND guest = boss

THEN drink = beer

RULE 183 : IF guest age > 21

THEN alcohol indicated = yes

RULE 184 : IF guest age <= 21

THEN alcohol indicated = no

RULE 185 : IF alcohol indicated = yes

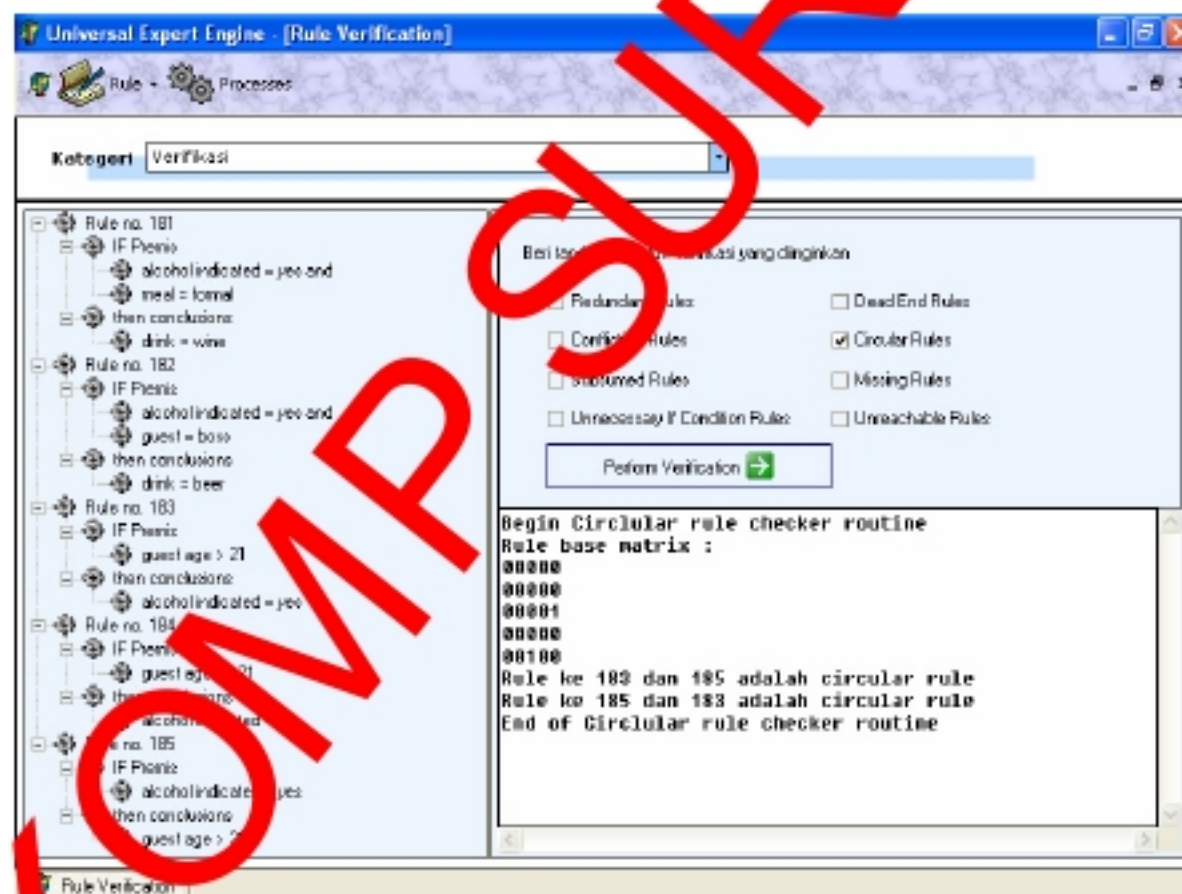
THEN guest age > 21

Dari proses verifikasi yang dilakukan, maka sistem akan menghasilkan pemberitahuan :

```

Begin circular rule checker routine
Rule base matrix
00000
00000
00001
00000
00100
Rule ke 183 dan 185 adalah circular rule
Rule ke 185 dan 183 adalah circular rule
End of circular rule checker routine
  
```

Hasil proses verifikasi seperti yang di tunjukkan pada Gambar 4.10



Gambar 4.10 Rule Verification – Circular Rule

4.4.7 Uji Coba Inference :

Kasus Pertama

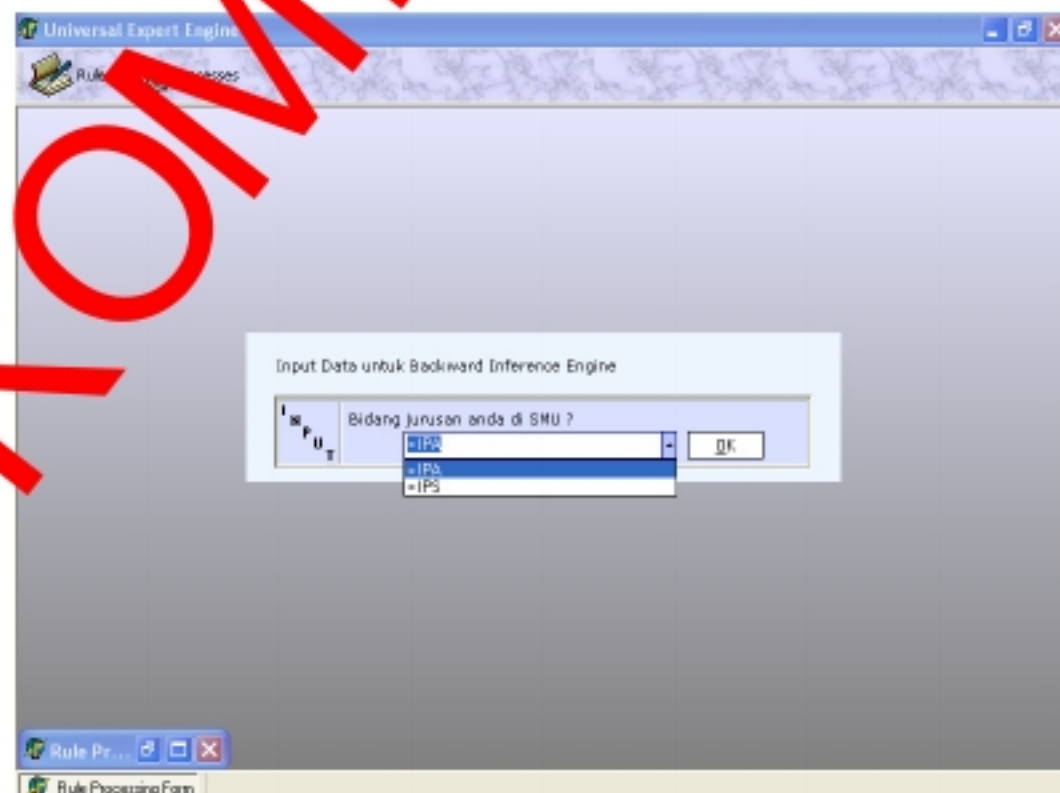
Dalam kategori rekomendasi jurusan calon mahasiswa tujuannya adalah menentukan tujuan jurusan yang dipilih sesuai dengan masukan user dalam proses konsultasinya.

Untuk contoh, apabila user memasukkan nilai / data seperti dibawah ini :

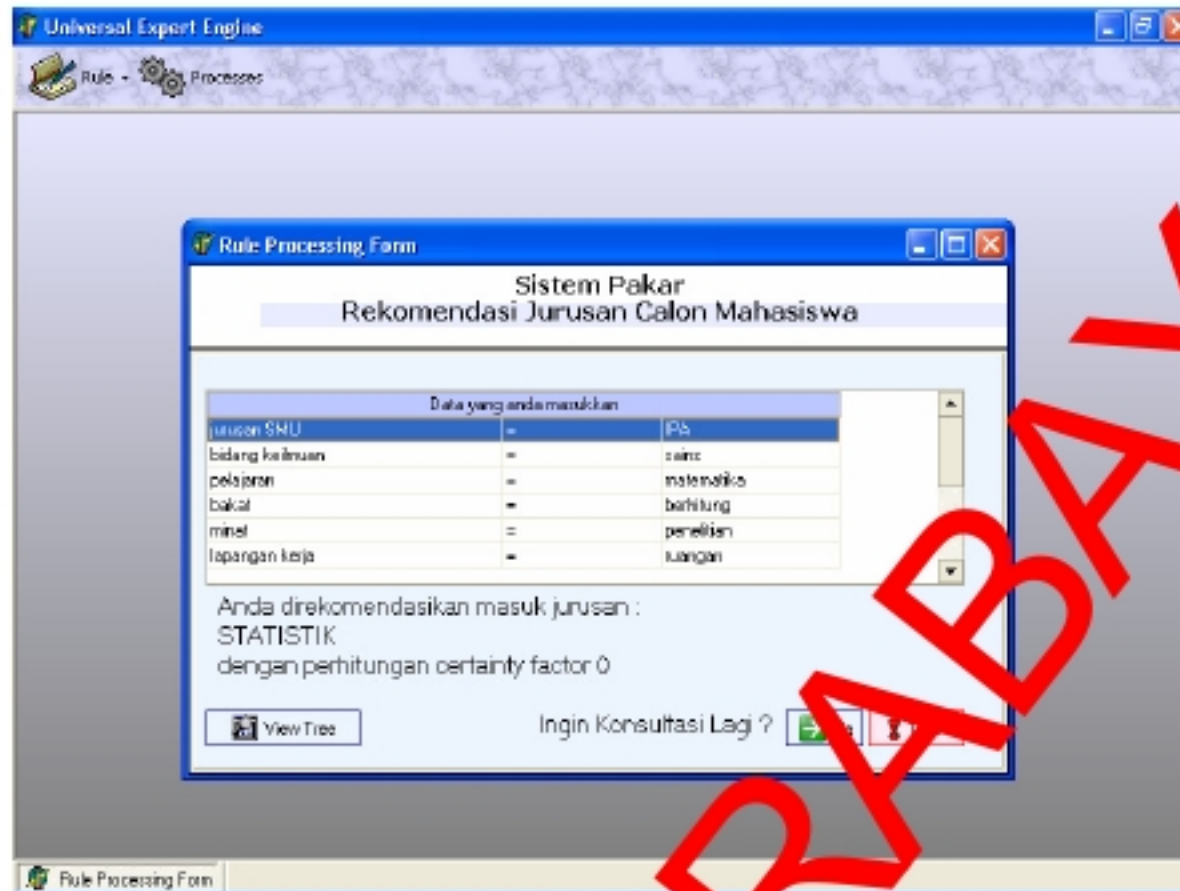
1. Jurusan SMU : IPA
2. Bidang keilmuan : sains
3. Pelajaran yang disukai : matematika
4. Bakat yang dimiliki : berhitung
5. Bidang yang diminati : penelitian
6. Jenis lapangan kerja yang diinginkan : pengan

Berdasarkan data masukan diatas, akan didapatkan hasil proses inference seperti di bawah ini :

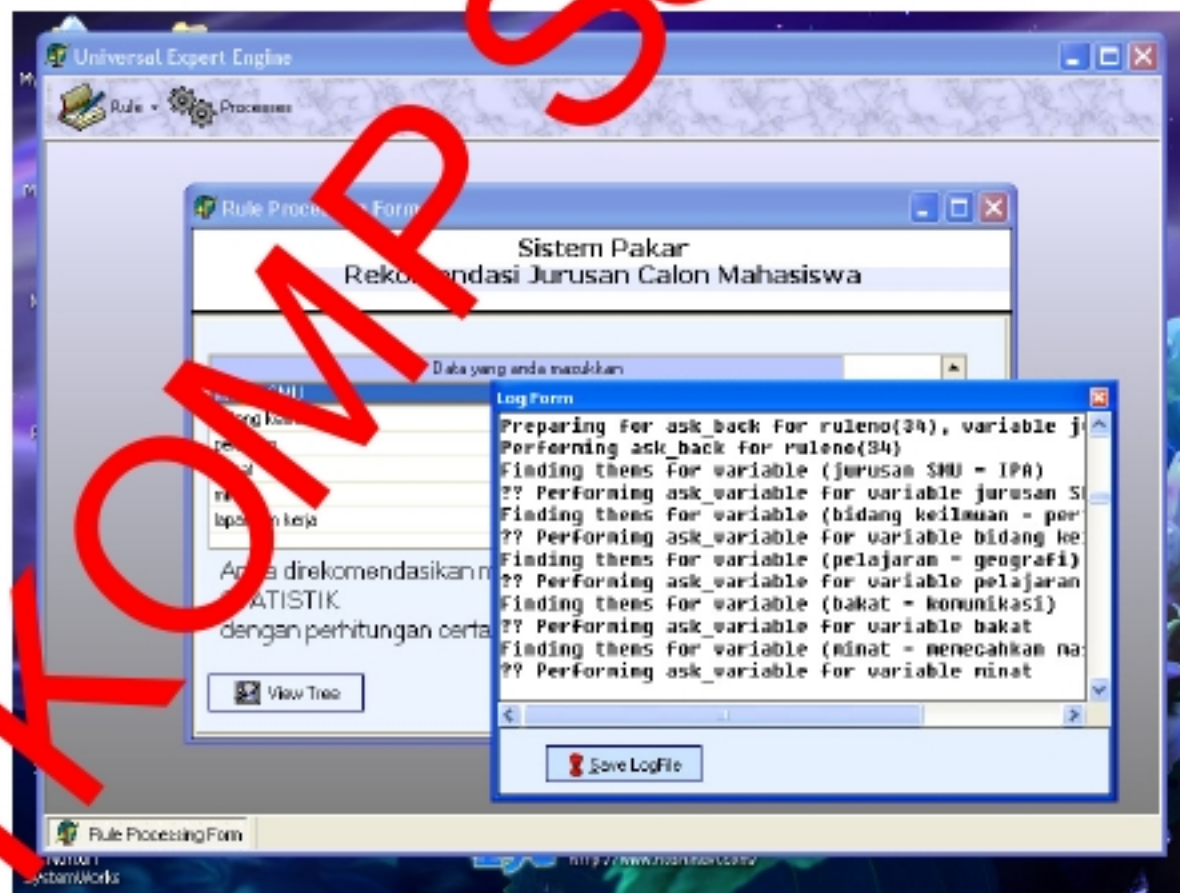
- Backward Inference Engine



Gambar 4.11 Implementasi untuk Backward Process

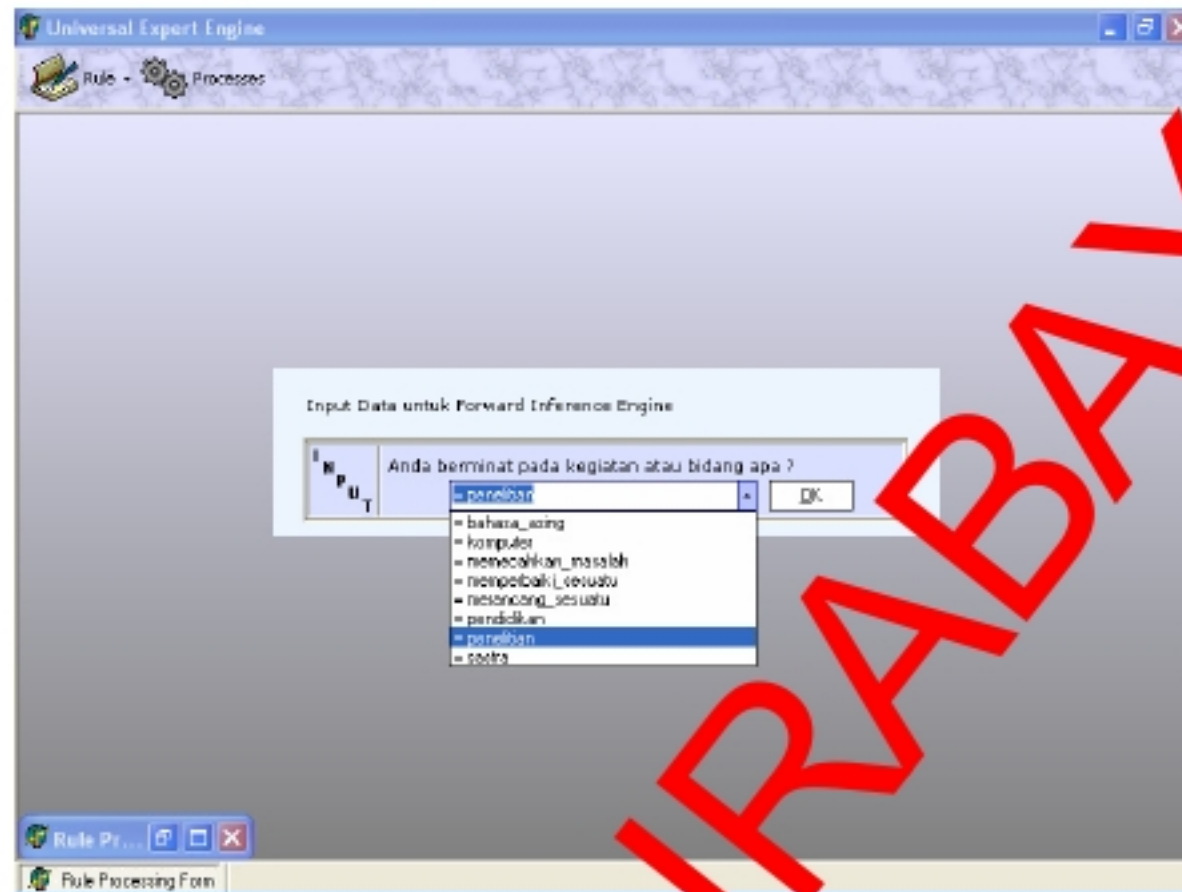


Gambar 4.12 Hasil Implementasi untuk Backward Process

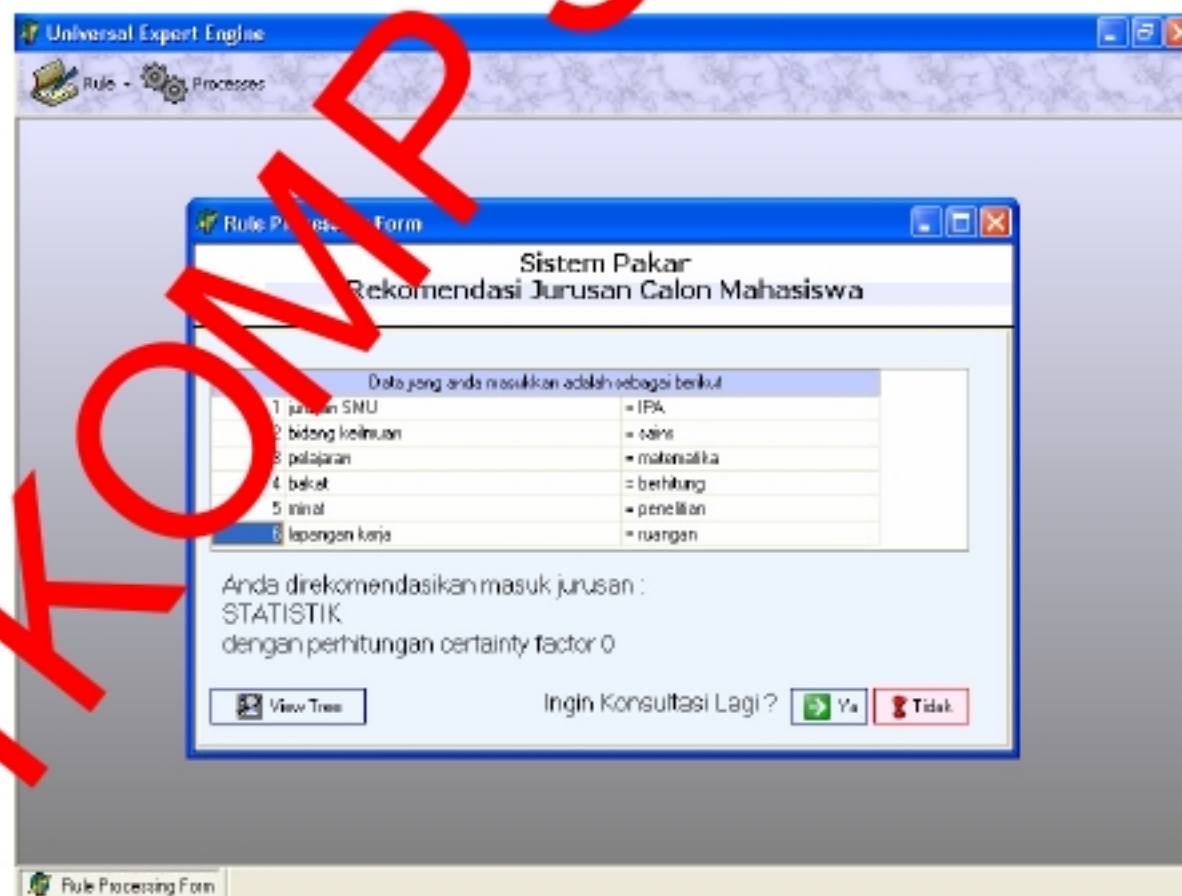


Gambar 4.13 Tampilan Logfile untuk Backward Process

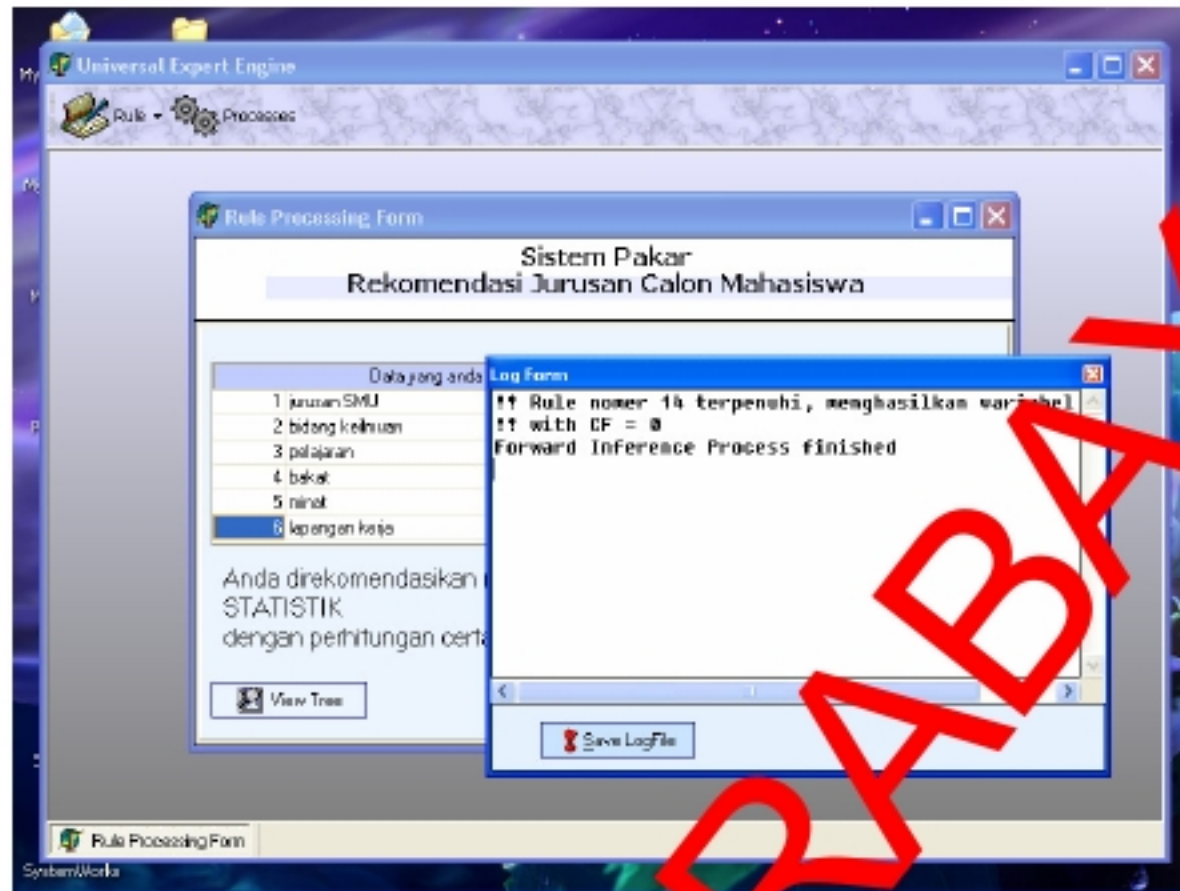
- Forward Inference Engine



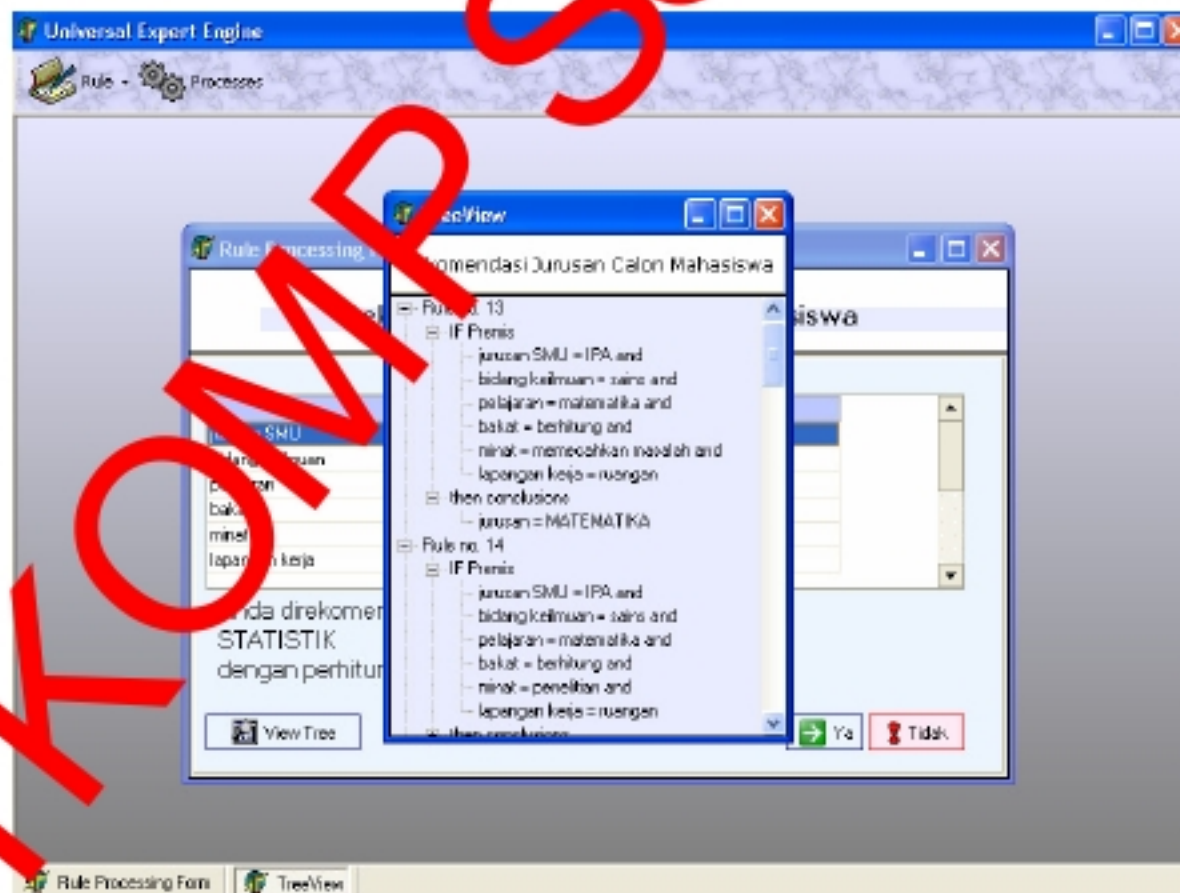
Gambar 4.14 Implementasi untuk Forward Process



Gambar 4.15 Hasil Implementasi untuk Forward Process



Gambar 4.16 Tampilan LogFile untuk Forward Process



Gambar 4.17 Tampilan TreeView

Kasus Kedua

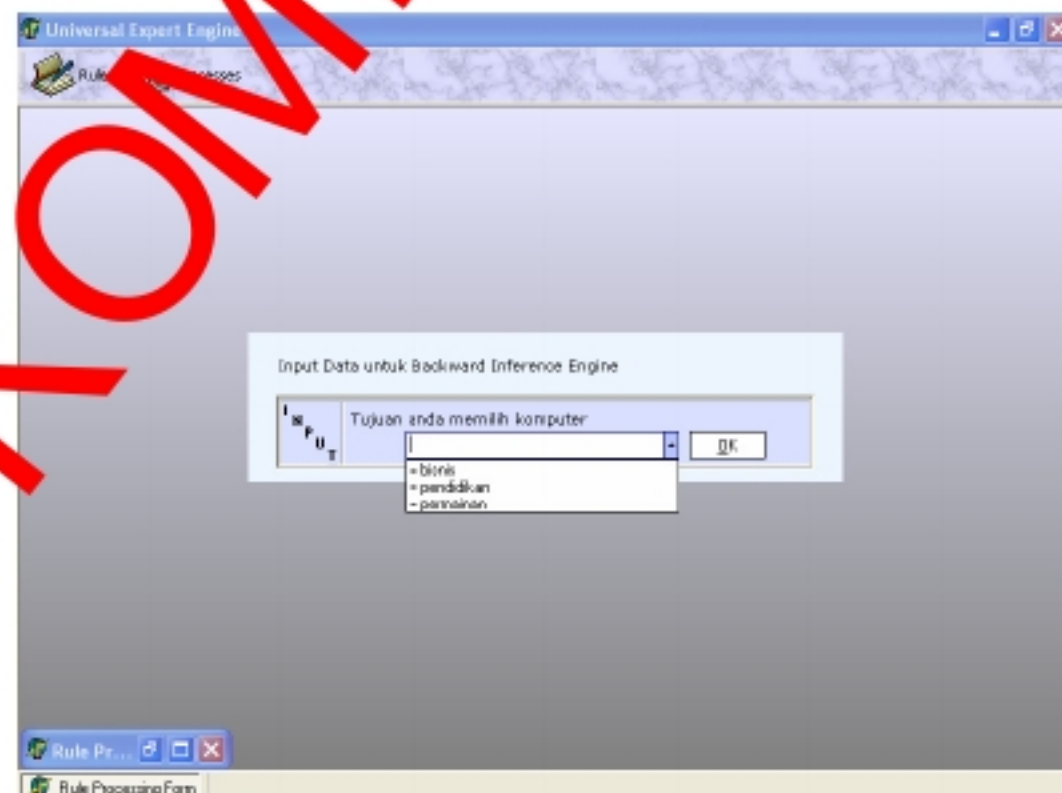
Kategori Yang dimasukkan adalah Pemilihan Komputer. Yang tujuannya menentukan jenis Komputer yang dibutuhkan sesuai dengan masukan oleh user

Apabila user memasukkan nilai / data seperti dibawah ini :

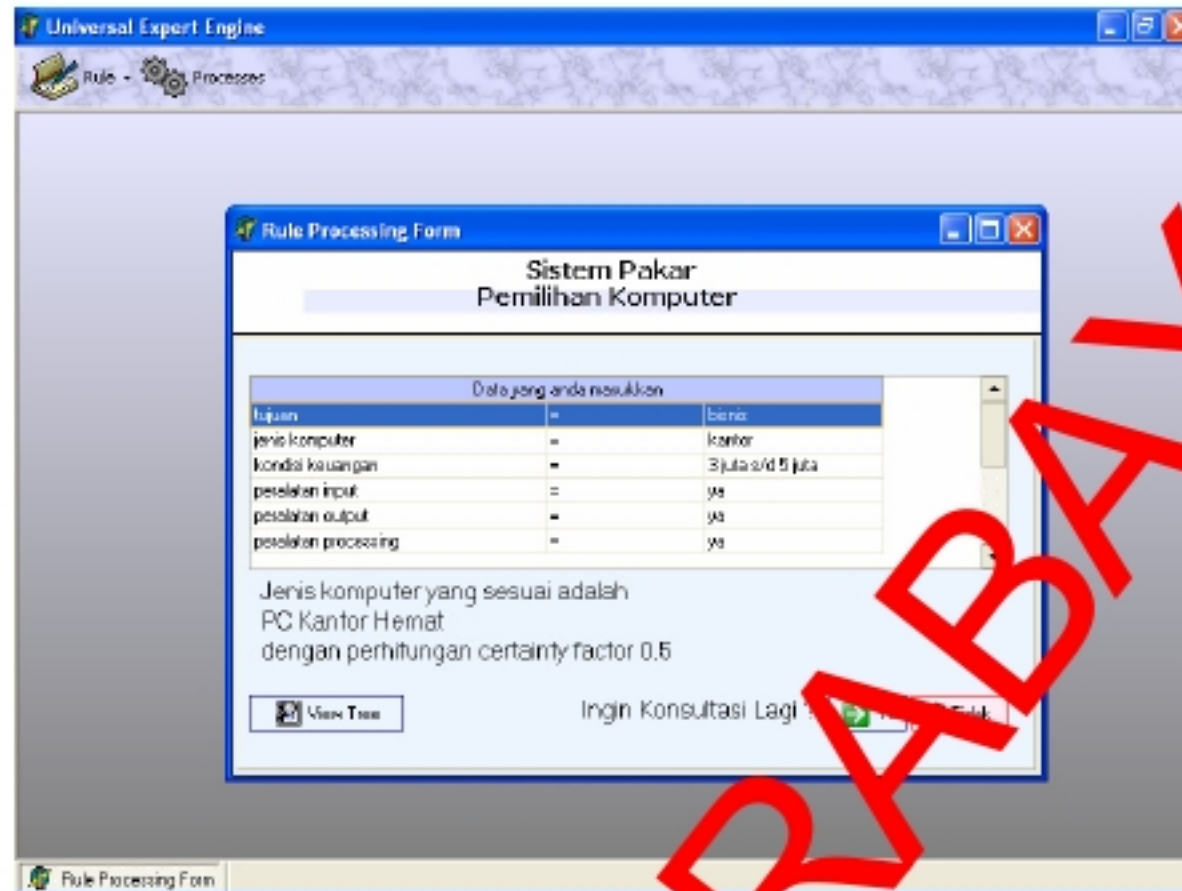
1. Tujuan pemilihan komputer : Bisnis
2. Jenis komputer : kantor
3. Kondisi keuangan : 3 juta s/d 5 juta
4. Peralatan input : ya
5. Peralatan output : ya
6. Peralatan processing : ya
7. Peralatan multimedia : ya

Berdasarkan data yang ada dalam rulebase diharapkan hasil yang muncul adalah jenis computer PC kantor hemat. Sistem akan memproses dengan hasil sebagai berikut :

Backward Inference Engine

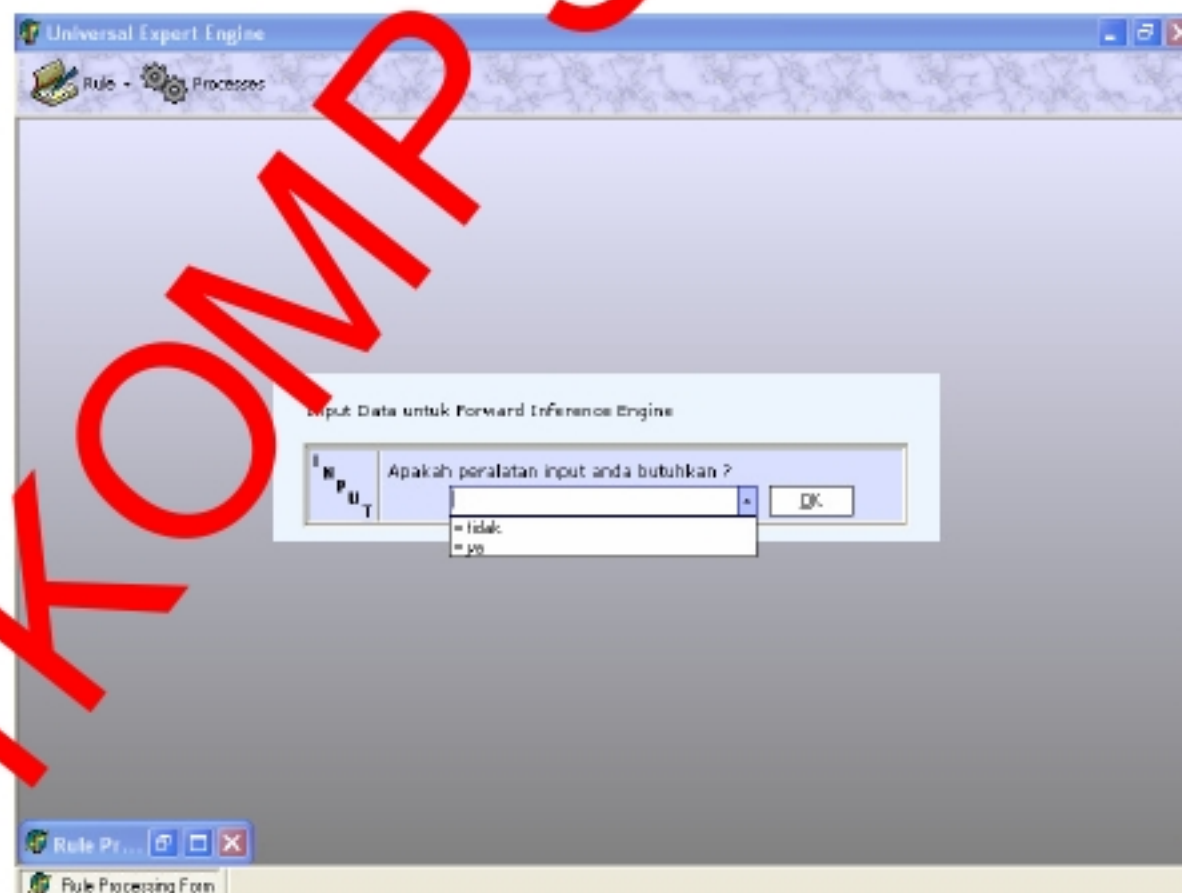


Gambar 4.18 Input Data untuk Backward

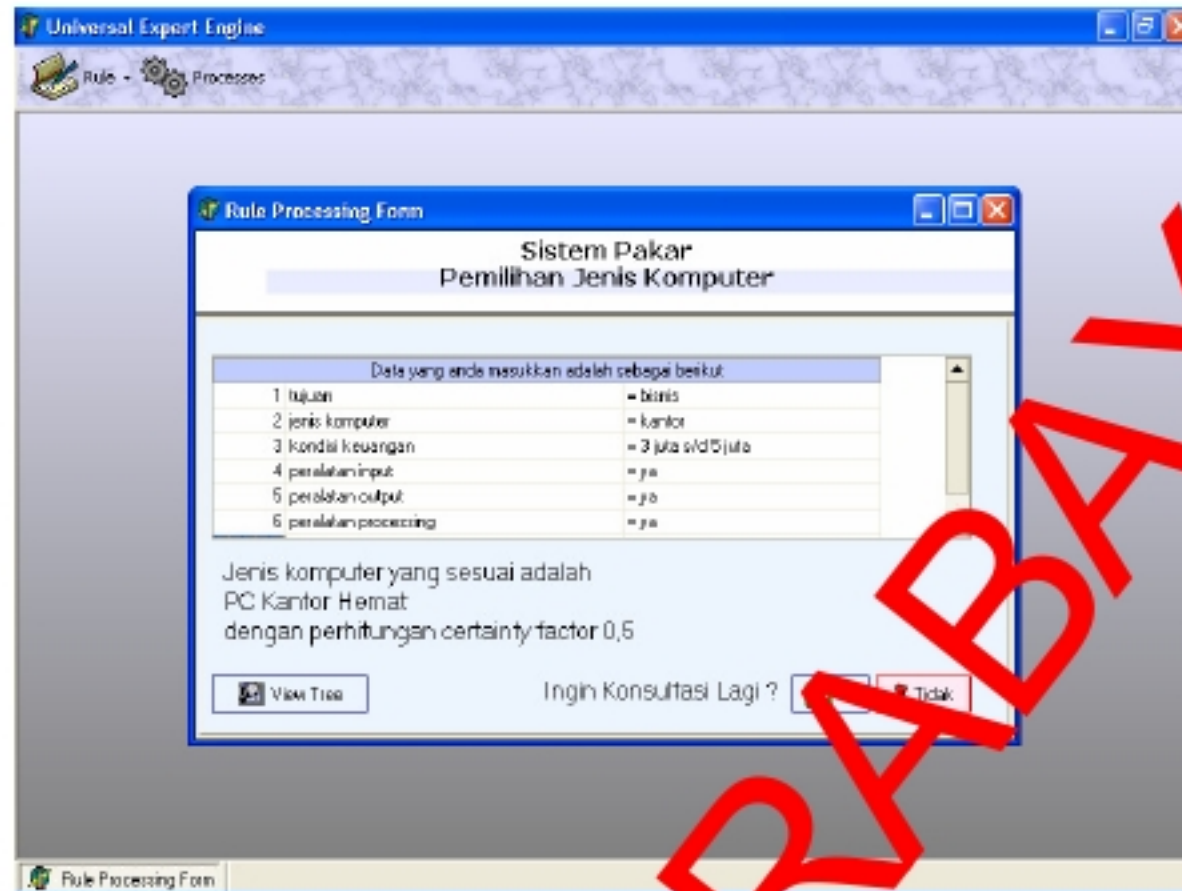


Gambar 4.19 Hasil Konsultasi untuk Backward

Forward Inference Engine :



Gambar 4.20 Input Data untuk Forward



Gambar 4.21 Hasil Konsultasi untuk Forward

Setelah melakukan testing program dengan melihat segala kelebihan maupun kekurangan dapat disimpulkan bahwa sistem ini bisa berjalan dengan baik dalam memberikan kesimpulan atau rekomendasi. Sistem ini bersifat dinamis karena suatu saat terdapat perubahan, penghapusan dan penambahan rule. Agar mengurangi adanya error dalam konsultasi akibat adanya rule yang dimasukkan, maka proses verifikasi dapat dijalankan.

Aplikasi ini sepenuhnya belum diimplementasikan secara real. Namun tidak menutup kemungkinan sistem ini dapat dipakai dengan disertai penambahan atau pengembangan yang lebih baik.