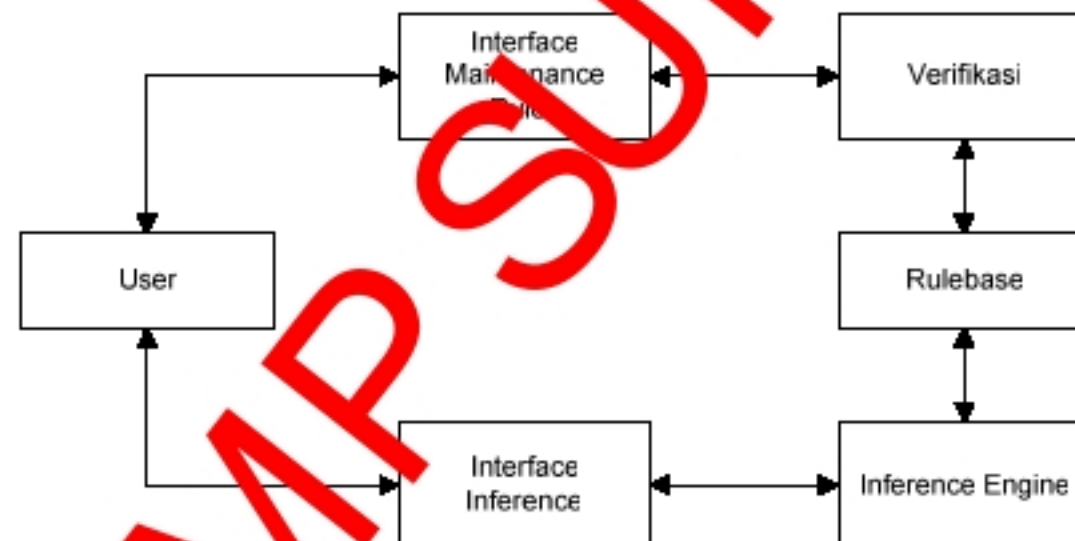


BAB III PERANCANGAN SISTEM

3.1 Perancangan Sistem.

Suatu sistem pakar tidak dapat melakukan pemeriksaan sendiri untuk suatu masalah, tetapi hanya menerima informasi tentang masalah yang dihadapi. Misalnya gejala-gejala suatu penyakit. Maka dalam membangun suatu pakar dibutuhkan pengumpulan data-data sebanyak mungkin agar kemungkinan adanya kesalahan dalam mendiagnosa dapat diperkecil. Desain arsitektur pembuatan sistem ini dapat diperlihatkan pada gambar berikut.



Gambar 3.1 Arsitektur Sistem

1. User dalam hal ini pakar adalah orang yang membuat sistem pakar, sekaligus pembuat rule-rule.

2. Interface Maintenance rules berhubungan langsung dengan user/pakar selama melakukan proses maintenance yang meliputi input, edit, maupun delete rule

3. Verifikasi merupakan proses pengecekan rule.

4. Rule-rule yang sudah diinputkan akan disimpan dalam rulebase.

5. Inference engine merupakan proses dalam mencari solusi berdasarkan tujuan konsultasi berdasarkan rule yang ada didalam rulebase
6. Interface inference merupakan bagian yang berhubungan langsung dengan user/pakar dengan menampilkan proses dan hasil dari sistem yang merupakan sistem pakar beserta treeview yang merupakan tampilan rule-rule yang ada didalam rulebase.

Hasil akhir yang diharapkan adalah terbentuknya sistem pakar yang mampu menjawab persoalan yang dihadapi user.

3.2 Perancangan Rulebase

Setelah data-data dalam bentuk rule dapat dikumpulkan, maka langkah berikutnya adalah menyusun rulebase untuk menyimpan semua rule.

Rule-rule yang dalam bentuk IF THEN adalah jenis rule yang lebih sering digunakan dalam expert system. Contoh rule dalam bentuk IF THEN adalah sebagai berikut .

RULE 1 : IF guest-age < 21

THEN alcohol-indicated = no

RULE 2 : IF guest-age >= 21

THEN alcohol-indicated = yes

RULE 3 : IF alcohol-indicated = yes AND meal = formal

THEN drink = wine

RULE 4 : IF alcohol-indicated = yes AND guest = boss

THEN drink = wine

Rulebase dalam sistem ini mendasarkan pada rule-rule yang dimasukkan oleh pengguna. Oleh sebab itu dibutuhkan media penyimpan agar pengguna tidak perlu memasukkan rule setiap kali akan menggunakan. Dengan demikian rule dapat dimasukkan secara bertahap, sehingga semakin melengkapi rulebase yang ada.

Untuk menyusun rulebase, setiap rule dipilah – pilah menjadi beberapa komponen

Misal :

```

RULE 3 : IF alcohol-indicated = yes ( CFe = 0,2 )
        AND meal = formal ( CFh = 0,4 )
        THEN drink = wine ( CFh = 0,6 )
  
```

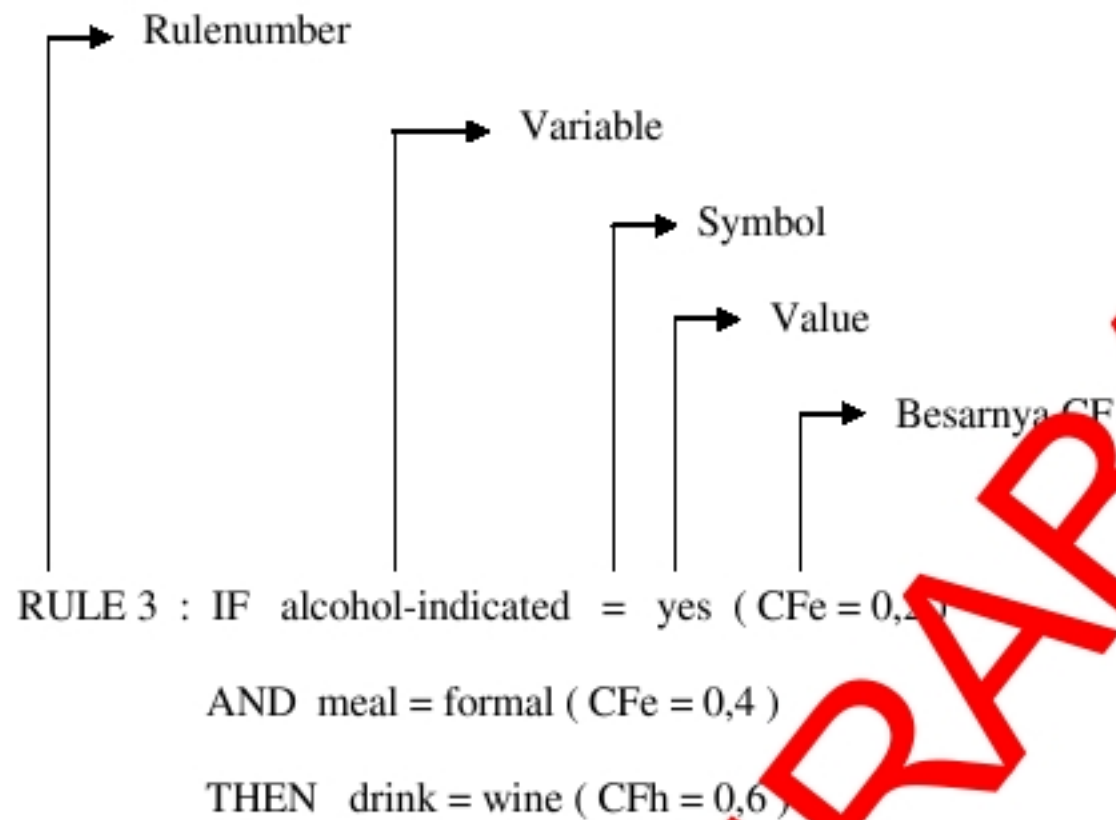
Setiap rule dibagi 2 bagian, yaitu : premis dan Konklusi

```

RULE 3 : IF alcohol-indicated = yes ( CFe = 0,2 )
        AND meal = formal ( CFh = 0,4 )
        THEN drink = wine ( CFh = 0,6 )
  
```

} Premis
 → Konklusi

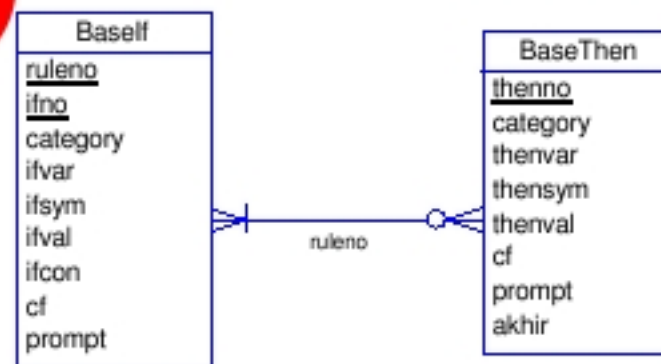
Setiap kondisi dibagi menjadi komponen-komponen kecil yaitu rule number, variable, symbol, value dan besarnya Certainty Factor (CF)



3.2.1 Conceptual Data Model

Conceptual Data Model (CDM) adalah struktur logika dari suatu database. CDM digunakan untuk representasikan, menentukan kebutuhan-kebutuhan dalam proses database.

Dari penjelasan diatas maka conceptual data model dalam sistem ini adalah sebagai berikut :



Gambar 3.2 Gambar Conceptual Data Model

3.2.2 Struktur Rulebase

Media penyimpanan yang dipilih adalah sistem database. Ini dimaksudkan untuk memudahkan pengorganisasian rule dan menyederhanakan operasi pembacaan / penulisan rule.

Data base yang digunakan pada aplikasi ini adalah Microsoft Access. Keputusan ini dibuat karena kesederhanaan pengoperasannya dan kemampuannya yang cukup untuk manage data yang kompleks sekalipun. Microsoft Access sangat mudah untuk dioperasikan, hanya membutuhkan sistem yang memiliki Microsoft Office, dimana hampir semua PC yang beroperasi dengan Microsoft Windows memilikinya.

Rulebase digunakan untuk menyimpan rule-rule yang dimasukkan oleh pakar dan disimpan sedemikian rupa sehingga memungkinkan rulebase tersebut dilakukan proses inferensi, maintenance maupun verifikasi rule.

Database yang digunakan pada sistem ini antara lain :

1. Tabel BaseIf

Fungsi Menyimpan data rule dalam kondisi If

Tabel 3.1 Tabel BaseIf

Nama Field	Type	Lebar	Keterangan
Rulen	Number	Long integer	Untuk menyimpan nomor rule
Catego	Text	50	Untuk menyimpan jenis kategori
Ifvar	Text	50	Untuk menyimpan nama variabel
Ifsym	Text	5	Untuk menyimpan simbol
Ifval	Text	50	Untuk menyimpan nama value
Ifcon	Text	5	Untuk menyimpan penghubung ('or' atau 'and')
Ifno	Number	Long integer	Untuk menyimpan urutan (1,2,3,..) bergantung pada jumlah premis if

Cf	Number	Long integer	Untuk menyimpan besarnya confidence factor premis
Prompt	Text	50	Untuk menyimpan text pertanyaan pada konsultasi

2. Tabel BaseThen

Fungsi : Menyimpan data rule dalam kondisi Then

Tabel 3.2 Tabel Base Then

Nama Field	Tipe	Lebar	Keterangan
Ruleno	Number	Long integer	Untuk menyimpan nomor rule
Category	Text	50	Untuk menyimpan jenis kategori
Thenvar	Text	50	Untuk menyimpan nama variabel
Thensym	Text	5	Untuk menyimpan simbol
Thenval	Text	50	Untuk menyimpan nama value
Thenno	Number	Long integer	Untuk menyimpan urutan (1,2,3,..) bergantung pada jumlah premis then
Cf	Number	Long integer	Untuk menyimpan besarnya confidence factor hipotesa
Prompt	Text	50	Untuk menyimpan text kesimpulan
Akhir	Yes/no		Untuk penanda rule tersebut sebagai akhir dari proses/ kesimpulan

3. Tabel Varlist

Fungsi : Menyimpan data input sementara

Tabel 3.3 Tabel Varlist

Nama Field	Tipe	Lebar	Keterangan
Variable	Text	50	Untuk menyimpan variabel hasil inference
Sign	Text	5	Sebagai penanda apakah variabel tersebut sudah diinisialisasi atau belum

Value	Text	50	Untuk menyimpan value hasil inference
-------	------	----	---------------------------------------

Gambaran penyimpanan record pada database :

Misalnya : kategori inputan adalah Pemilihan Makanan dan Minuman

RULE 3 : IF alcohol-indicated = yes (CFe = 0,2)

AND meal = formal (CFe = 0,4)

THEN drink = wine (CFh = 0,6)

Tabel 3.4 Gambaran Penyimpanan Record Tabel BaseIf

Ruleno	Category	Ifvar	Ifsym	Ifval	Ifcon	Ifno	Cfe
3	Pemilihan Makanan dan Minuman	Alcohol-indicated	=	Yes	And	1	0,2
3	Pemilihan Makanan dan Minuman	Meal	=	Formal	Final	2	0,4

Tabel 3.5 Gambaran Penyimpanan Record Tabel BaseThen

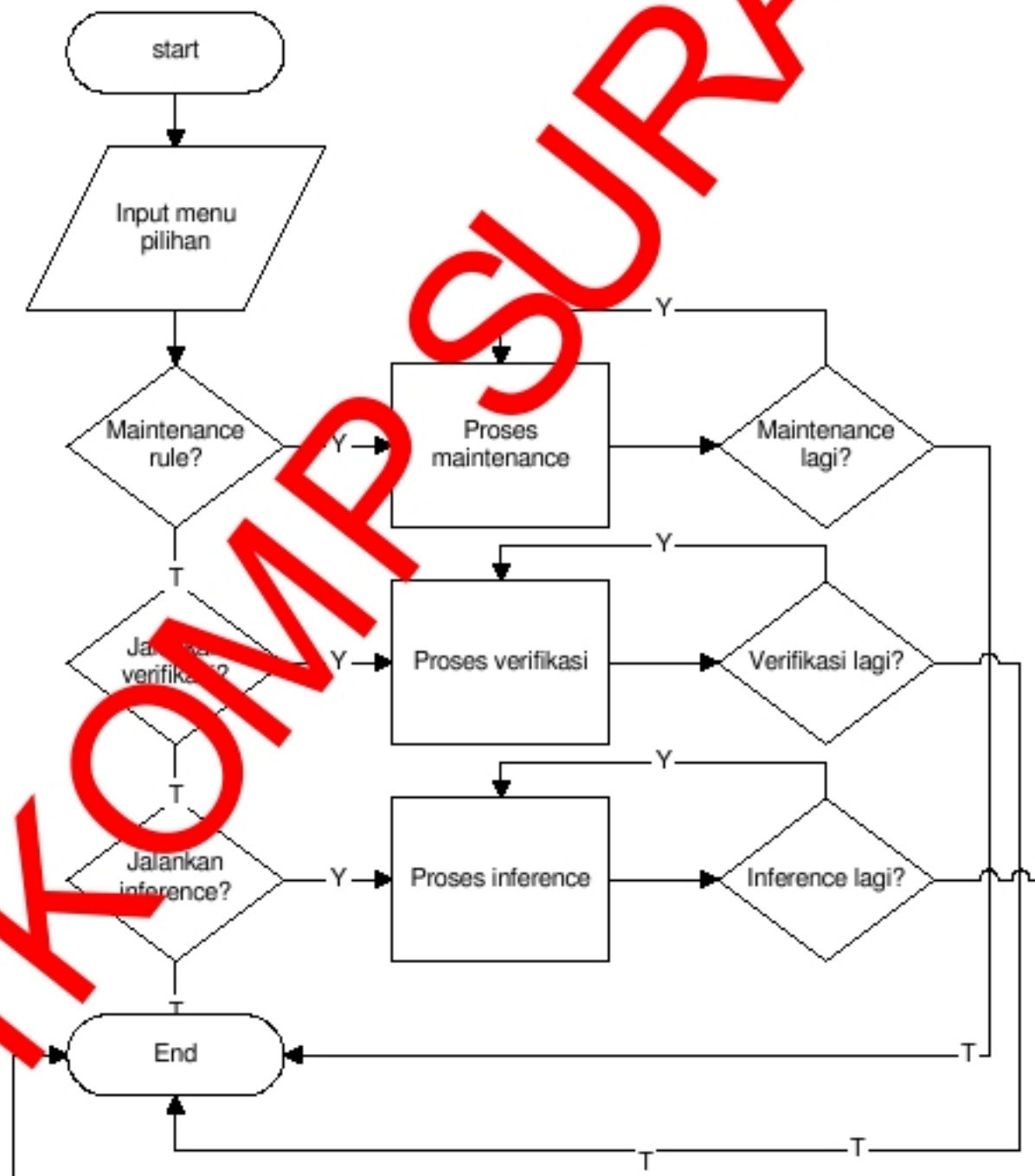
Ruleno	Category	Ifvar	Ifsym	Ifval	Ifcon	Ifno	Cfh
3	Pemilihan Makanan dan	Drink	=	Wine	Final	1	0,6

	Minuman						
--	---------	--	--	--	--	--	--

3.3 Diagram Alir Sistem

Diagram Alir (flowchart) adalah diagram yang menunjukkan alir di dalam program atau prosedur system secara logika dan menjelaskan secara rinci langkah – langkah dari proses program.

Dalam pembuatan system ini dapat digambarkan dalam diagram alir berikut :



Gambar 3.3 Diagram Alir Sistem

Algoritma Sistem :

1. Pada tampilan menu utama, langkah 2 untuk membuat sistem pakar baru dan langkah 6 untuk menjalankan proses inference.
2. Proses maintenance yang terdiri dari input, edit maupun hapus rule
3. Simpan rule-rule dalam rulebase
4. Proses verifikasi dilakukan setelah semua rule lengkap
5. Apabila ada rule yang tidak masuk dalam verifikasi akan di edit maka kembali ke langkah 2. Apabila rule yang berhasil masuk dalam proses verifikasi langsung langkah 6.
6. Pada tampilan menu utama dipilih proses untuk menjalankan inference engine
7. Pilih kategori rule yang akan dijalankan dan metode inference engine yang digunakan
8. Tampil hasil sistem pakar sesuai kategori yang dipilih

3.4 Perancangan Inference Engine

Tugas dari inference engine adalah melakukan proses penalaran dengan menggunakan rulebase untuk menentukan jawaban yang tepat dengan metode pengambilan keputusan yang telah digunakan. Pada sistem ini menggunakan 2 metode penelusuran yaitu Forward Chaining dan Backward Chaining yang penggunaannya tergantung kebutuhan user yang diharapkan dari sistem pakar yang dihasilkan nanti.

3.4.1 Forward Chaining

Forward Chaining adalah metode penelusuran yang dimulai dari data untuk mencari kesimpulan. Metode ini biasanya digunakan dalam persoalan analisa, desain, diagnosa dan konstruksi. Secara umum, forward chaining memiliki ciri :

- * Diawali dengan kumpulan premis
- * Menghasilkan suatu kesimpulan

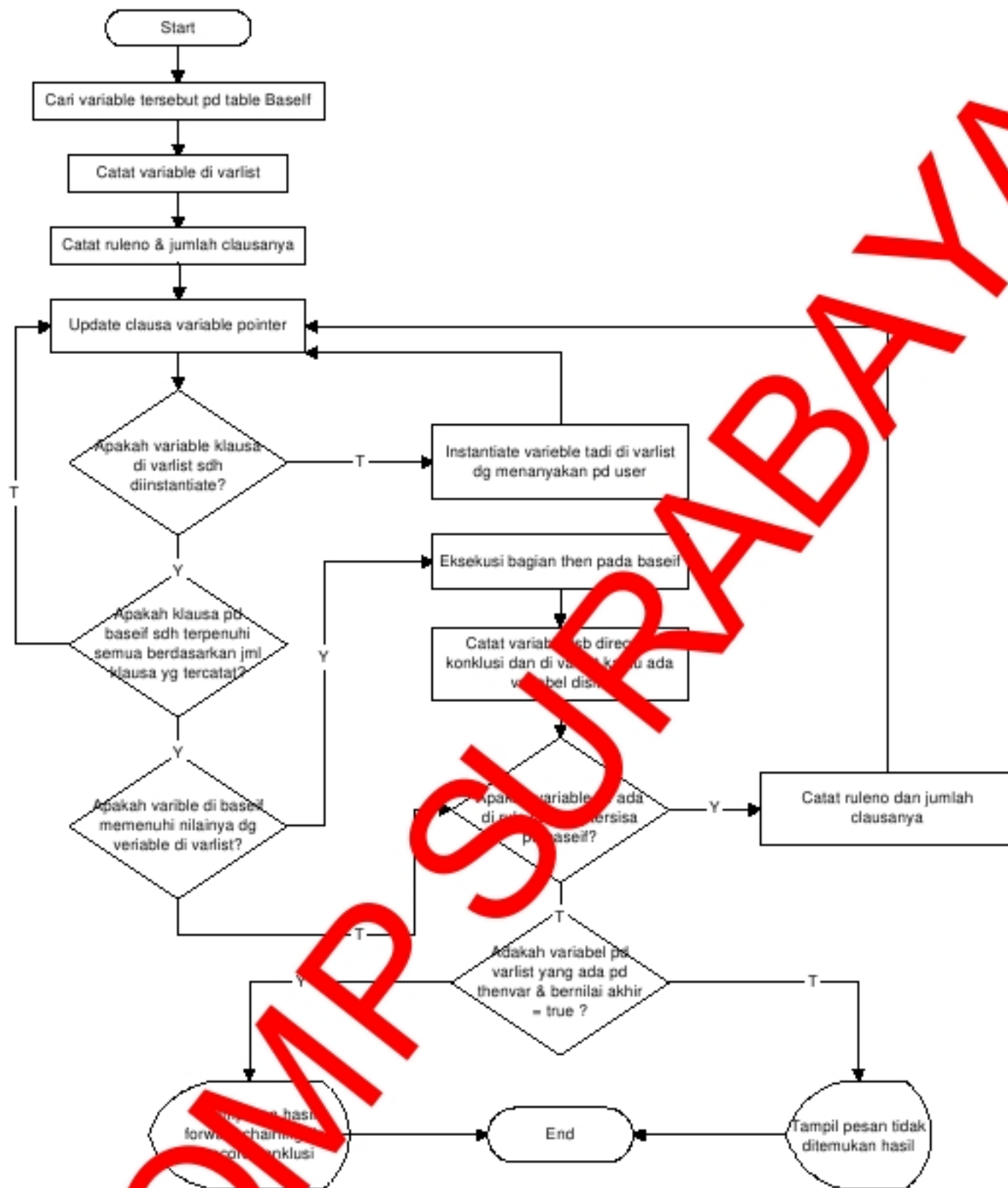
Algoritma forward chaining dalam system ini adalah sebagai berikut :

1. Identifikasi masukan
2. Cari variable masukan pada table baseif
3. Catat nilai variable di varlist
4. Diadakan pencatatan rule no dan jumlah klausanya
5. Update clausa variable pointer
6. Cek, variable di varlist apakah sudah di instantiate. Apabila sudah langsung ke langkah 7, apabila belum ke langkah 9
7. Cek, klausa pada baseif apakah sudah terpenuhi berdasarkan jumlah klausa yang tercatat. Apabila sudah ke langkah 8, apabila tidak kembali ke langkah 5
8. Cek, variable di baseif apakah sudah memenuhi nilainya dengan variable di varlist. Apabila sudah ke langkah 10, apabila belum ke langkah 12
9. Instantiate variable tadi di varlist dengan menanyakan pada user dan langsung kembali ke langkah 5.
10. Eksekusi bagian then
11. Catat variable tersebut direcord konklusi dan di varlist kalau ada variabelnya di situ

12. Cek, apakah variable ini masih ada di rule lain pada baseif. Jika masih, catat ruleno dan jumlah klausanya dan kembali ke langkah 5. Jika tidak, ke langkah 13
13. Cek, adakah variable yang ada pada thenvar dan bernilai akhir = true. Jika ya, maka tampilkan hasil prose forward dari record koneksi. Jika tidak, maka tampilkan pesan tidak ditemukan hasil
14. Selesai

Untuk lebih jelas proses penelusuran forward chaining dalam system ini digambarkan dalam diagram alir berikut :

STIKOMP SURABAYA



Gambar 3.4 Diagram Alir Inference Forward Chaining

3.4.2 Backward Chaining

Backward Chaining adalah metode penelusuran yang dimulai dari hasil/kesimpulan untuk mencari kondisi-kondisi yang berkaitan dengan hasil tersebut. Metode ini banyak digunakan dalam persoalan diagnosa. Secara umum, backward chaining memiliki ciri :

- * diawali dari hasil/kesimpulan
- * menelusuri kondisi-kondisi (premis) yang menyusun kesimpulan tersebut.

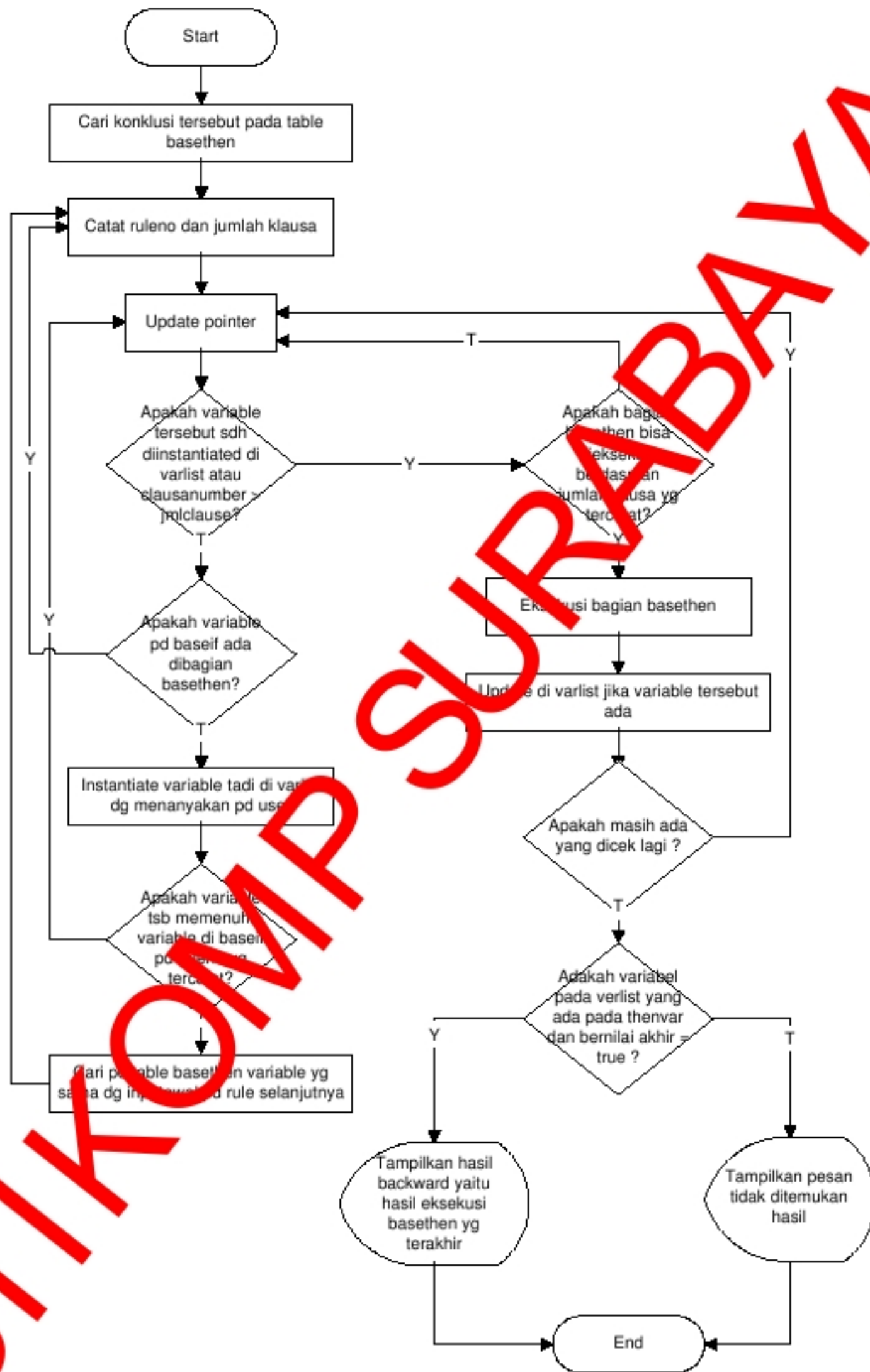
Algoritma backward chaining dalam system ini adalah sebagai berikut :

1. Identifikasi masukan
2. Cari konklusi masukan pada tabel basethen
3. Diadakan pencatatan ruleno dan jumlah klausanya
4. Update pointer
5. Cek apakah variable tersebut diinstantiate atau tidak. Jika jumlahnya > jumlah klausanya. Apabila belum langsung ke langkah 6, apabila sudah ke langkah 10.
6. Cek variable pada baseif apakah ada dibagian basethen. Apabila tidak langsung ke langkah 7, apabila ada kembali ke langkah 3
7. Instantiate variable tadi dengan merivakan pada user
8. Setelah user memasukkan nilai variable, cek apakah variable tersebut memenuhi pada baseif pada ruleno yang tercatat. Apabila tidak ke langkah 9, apabila ya kembali ke langkah 4.
9. Cari pada basethen variable yang sama pada rule selanjutnya dan kembali pada langkah 3.
10. Cek pada bagian basethen, apakah bisa dieksekusi berdasarkan jumlah klausa yang tercatat. Apabila bisa langsung ke langkah 11, apabila tidak kembali pada langkah 4.
11. Eksekusi bagian basethen
12. Update di varlist jika variable tersebut ada

13. Cek, apakah masih ada yang perlu di cek. Apabila ada kembali ke langkah 4, apabila tidak ada ke langkah 14.
14. Cek, apakah variable pada varlist yang ada pada thenvar dan bernilai akhir = true. Jika ya, tampilkan hasil backward. Jika tidak, tampilkan pesan tidak ditemukan hasil.
15. Selesai

Untuk lebih jelas proses penelusuran backward chaining dalam system ini digambarkan dalam diagram alir berikut :

STIKOMMP SURABAYA



Gambar 3.5 Diagram Alir Inference Backward Chaining

3.5 Perancangan Verifikasi Rules

Verifikasi merupakan proses pengecekan rule yang bertujuan untuk menghindari kesalahan sehingga sistem dapat diimplementasikan dengan benar.

3.5.1 Redundant Rules

Seperti pada landasan teori bahwa untuk mencari rule-rule yang dianggap redundant rule, sistem akan membandingkan tiap rule satu persatu mencari kesamaan pada seluruh premis dan kesimpulan pada 2 rule yang berbeda. Apabila ada 2 rule yang persis sama premis dan kesimpulannya, maka rule tersebut dikategorikan sebagai redundant rule.

Misalnya :

RULE 3 : IF alcohol-indicated = yes

AND meal = formal

THEN drink = wine

RULE 13 : IF meal = formal

AND alcohol-indicated = yes

THEN drink = wine

Gambaran penyimpanan record pada database adalah sebagai berikut:

Tabel 3.5 Gambaran Penyimpanan Record pada table BaseIf untuk Redundant

Rule					
idno	Ifvar	Ifsym	Ifval	Ifcon	ifno
3	Alcohol-indicated	=	Yes	And	1
3	Meal	=	Formal	Final	2
13	Meal	=	Formal	And	1
13	Alcohol-indicated	=	Yes	Final	2

Tabel 3.7 Gambaran Penyimpanan Record pada Table BaseThen untuk Redundant

Rule					
Rule no	Ifvar	Ifsym	Ifval	Ifcon	ifno
3	Drink	=	Wine	Final	1
13	Drink	=	Wine	Final	1

Pada kedua rule diatas, system akan membandingkan satu sama lain pada setiap premis rule 3 dan rule 13 pada basethen dan konklusi rule 3 dan rule 13 pada basethen yang dimilikinya. Karena semua premis konklusi pada rule 3 dan rule 13 bernilai sama, maka dua rule tersebut diatas dikategorikan sebagai redundant rule.

3.5.2 Conflicting Rules

Seperti dijelaskan dalam landasan teori maka untuk mencari conflicting rules sistem membandingkan semua rule secara teliti satu dengan yang lain untuk mencari dua atau lebih rule yang memiliki premis yang sama, tetapi memiliki sebagian / seluruh kesimpulan yang berbeda. Apabila ditemukan, maka rule tersebut dikategorikan sebagai conflicting rule.

Misalnya :

RULE 4 : IF alcohol-indicated = yes

AND guest = boss

THEN drink = wine

RULE 14 : IF alcohol-indicated = yes

AND guest = boss

THEN drink = beer

Gambaran penyimpanan record pada database :

Tabel 3.8 Gambaran Penyimpanan Record pada Table BaseIf untuk Conflicting

Rule					
Ruleno	Ifvar	Ifsym	Ifval	Ifcon	Ifno
4	Alcohol-indicated	=	Yes	And	1
4	Guest	=	Boss	Final	2
14	Alcohol-indicated	=	Yes	And	1
14	Guest	=	Boss	Final	2

Tabel 3.9 Gambaran Penyimpanan Record pada Table BaseThen untuk Conflicting Rule

Ruleno	Ifvar	Ifsym	Ifval	Ifcon	Ifno
4	Drink	=	Wine	Final	1
14	Drink	=	Beer	Final	1

Pada kedua rule diatas, system akan membandingkan satu sama lain pada setiap premis rule 4 dan rule 14 pada baseif dan konklusi rule 4 dan rule 14 pada basethen yang dimilikinya. Karena semua premis konklusi pada rule 4 bernilai sama tapi konklusi pada rule 14 bernilai beda, maka dua rule tersebut diatas dikategorikan sebagai conflicting rule.

3.5.2 Subsumed Rules

Seperti yang dijelaskan pada landasan teori bahwa suatu rule dikatakan *subsumed* dengan rule yang lain jika rule tersebut mempunyai constraints lebih pada premis yang mana mempunyai konklusi yang sama.

Misalnya :

RULE 5 : IF alcohol-indicated = yes

AND guest = neighbor

THEN drink = beer

RULE 15 : IF alcohol-indicated = yes

THEN drink = beer

Gambaran penyimpanan record pada database :

Tabel 3.10 Gambaran Penyimpanan Record pada Table BaseIf untuk Subsumed

Rule

RuleNo	Ifvar	Ifsym	Ifval	Ifcon	Ifno
5	Alcohol-indicated	=	Yes	And	1
5	Guest	=	Neighbor	Final	2
15	Alcohol-indicated	=	Yes	Final	1

Tabel 3.11 Gambaran Penyimpanan Record pada Table BaseThen untuk

Subsumed Rule

RuleNo	Ifvar	Ifsym	Ifval	Ifcon	ifno
5	Drink	=	Beer	Final	1
15	Drink	=	Beer	Final	1

Pada kedua rule diatas, system akan membandingkan satu sama lain pada setiap premis rule 5 dan rule 15 pada baseif dan konklusi rule 5 dan rule 15 pada basethen yang dimilikinya. System membandingkan kondisi pada kedua rule. Kedua rule mempunyai kondisi premis yang hampir sama meski jumlah premisnya berbeda dan mempunyai nilai konklusi yang sama. Maka kedua rule tersebut dikatakan subsumed rule.

3.5.3 Circular Rules

Seperti yang dijelaskan pada landasan teori bahwa circular rule adalah suatu keadaan dimana terjadinya proses perulangan dari suatu rule. Ini dikarenakan suatu premis dari salah satu rule merupakan konklusi dari rule yang lain atau kebalikannya.

Apabila ada dua rule yang saling berhubungan (kesimpulan dari rule yang satu dipakai pada premis rule yang lain), maka elemen yang melambangkan 2 rule tersebut diberi nilai 1. Jika tidak, maka diberi nilai 0. Setelah RBM lengkap, maka RBM ditelusuri, apabila ada elemen yang memiliki indeks sama bernilai 1, maka rule dengan indeks tersebut adalah circular rule.

Misalnya

RULE 5 : IF alcohol-indicated = yes

AND meal = formal

THEN drink = wine

RULE 4 : IF alcohol-indicated = yes

AND guest = boss

THEN drink = wine

RULE 16 :IF guest_age > 21

THEN alcohol-indicated = yes

RULE 17: IF guest_age <= 21

THEN alcohol-indicated = no

RULE 18 :IF alcohol-indicated = yes

THEN guest-age > 21

Gambaran penyimpanan record pada database :

Tabel 3.12 Gambaran Penyimpanan Record pada Table BaseIf untuk Circular

Rule					
Ruleno	Ifvar	Ifsym	Ifval	Ifcon	ifno
3	Alcohol-indicated	=	Yes	And	1
3	Meal	=	Formal	Final	2
4	Alcohol-indicated	=	Yes	And	1
4	Guest	=	Boss	Final	2
16	Guest-age	>	21	Final	1
17	Guest-age	<=	21	Final	1
18	Alcohol-indicated	=	Yes	Final	1

Tabel 3.13 Gambaran Penyimpanan Record pada Table BaseThen untuk Circular

Rule					
Ruleno	Ifvar	Ifsym	Ifval	Ifcon	ifno
3	Drink	=	Wine	Final	1
4	Drink	=	Wine	Final	1
16	Alcohol-indicated	=	Yes	Final	1

17	Alcohol-indicated	=	No	Final	1
18	Guest-age	>	21	Final	1

Dari rule – rule diatas, disusun suatu rule bit matrix (RBM) sebagai berikut :

	3	4	16	17	18
3	0	0	0	0	0
4	0	0	0	0	0
16	1	1	0	0	1
17	0	0	0	0	0
18	0	0	0	0	0

Pada RBM ini dapat dilihat, cell [16,18] dan cell[18,16] bernilai 1, sehingga rule 16 dan 18 merupakan circular rules.

3.5.4 Unnecessary or Condition Rules

Dikatakan *unnecessary rules* apabila 2 rule dengan konklusi yang sama mempunyai premis yang hampir sama. Premis dari rule-rule hampir sama kecuali satu dan tiap rule merupakan kebalikannya (kontradiksi).

Misalnya :

RULE 4 : IF alcohol-indicated = yes

AND guest = boss

THEN drink = wine

RULE 19 :IF alcohol-indicated = yes

AND guest = neighbor

THEN drink = wine

Gambaran penyimpanan record pada database :

Tabel 3.14 Gambaran Penyimpanan Record pada Table BaseIf untuk Unnecessary

If Condition Rule

Ruleno	Ifvar	Ifsym	Ifval	Ifcon	ifno
4	Alcohol-indicated	=	Yes	And	1
4	Guest	=	Neighbor	Final	2
19	Alcohol-indicated	=	Yes	And	1
19	Guest	=	neighbor	Final	2

Tabel 3.15 Gambaran Penyimpanan Record pada Table BaseThen untuk

Unnecessary If Condition Rule

Ruleno	Ifvar	Ifsym	Ifval	Ifcon	ifno
4	Drink	=	Wine	Final	1
19	Drink	=	Wine	Final	1

Pada kedua rule diatas, sistem akan membandingkan satu sama lain pada setiap premis rule 4 dan rule 19 pada baseif dan konklusi rule 4 dan rule 19 pada basethen. Sistem membandingkan kondisi pada kedua rule. Pada dua rule ini, terlihat bahwa semua konklusi pada basethen bernilai sama, sedangkan pada ifvar pada pada baseif kedua rule bernilai sama, tetapi ditemukan perbedaan pada ifval, maka rule ini digolongkan dalam unnecessary if condition rules.

3.5.5 Dead-End Rules

Seperti dijelaskan dalam landasan teori untuk mencari dead-end rule sistem akan mencari secara keberlanjutan suatu rule dengan mencari rule – rule yang premisnya menggunakan kesimpulan dari rule yang bersangkutan. Apabila tidak ada rule dengan premis yang menggunakan kesimpulan dari rule tersebut, dan kesimpulan tersebut bukan tujuan dari sistem, maka rule bersangkutan dikategorikan sebagai dead – end rule.

Misalnya:

1. If drink = wine and
Dinner = fish
Then wine_type = white
2. If wine_type = red and
Guest_status = drunk
Then Guest_go_home = true

Gambaran penyimpanan record pada database :

Tabel 3.16 Gambaran Penyimpanan Record pada Table BaseIf untuk Dead End

Rule	ifvar	ifsym	ifval	ifcon	ifno
1	Drink	=	Wine	And	1
1	Dinner	=	Fish	Final	2
2	Wine_type	=	Red	And	1
2	Guest_status	=	Drunk	Final	2

Tabel 3.17 Gambaran Penyimpanan Record pada Table BaseThen untuk Dead

End Rule

Rule no	Ifvar	Ifsym	Ifval	Ifcon	ifno
1	Wine_type	=	White	Final	1
2	Guest_go_home	=	True	Final	

Dari kedua rule di atas, ternyata output dari rule, wine_type = white tidak digunakan dalam rule kedua, sedangkan wine_type bukan merupakan kesimpulan akhir dari sistem. Karenanya, rule ini diplotkan sebagai dead – end rule.

3.5.6 Missing Rules

Suatu rule dikatakan Missing rule apabila telah ditemukan adanya dead end rule atau unreachable rule, maka dipastikan pada kelompok / himpunan rule tersebut memiliki missing rule.

3.5.7 Unreachable Rules

Unreachable rule merupakan kebalikan dari dead-end rule.

3.6 Perancangan User Interface

Form menu utama ini digunakan untuk memilih apakah user ingin membuat sistem pakar baru atau hanya menampilkan sistem pakar yang sudah pernah dibuat untuk dilakukan proses inference.

Rule	Processes
Rule Maintenance	
Rule Verification	
Exit	

Gambar 3.6 Tampilan Menu Utama

3.6.1 Perancangan Interface Maintenance Rules

Apabila rule maintenance pada menu utama dipilih maka interface rule maintenance akan muncul yang mana akan menghasilkan sistem pakar baru.

Tugas dari rule maintenance adalah melakukan proses pemeliharaan rule yang meliputi pengisian, perubahan maupun penghapusan rule. Untuk itu dibuat interface maintenance yang merupakan sarana yang memudahkan user dalam melakukan proses pemeliharaan rule-rule

The screenshot displays a software interface for rule maintenance. At the top, there is a label 'kategori' followed by a text input field containing the letter 'M'. Below this, the interface is divided into two main sections for rule configuration. Each section includes a 'Kategori' label and an input field, followed by an 'If Clause' section with three input fields and a vertical separator, a 'Prompt' label with an input field, and a 'Cf' label with an input field and an 'Ok' button. Below these are three empty rectangular boxes. The second section mirrors the first but with a 'Then Clause' label and input field instead of an 'If Clause'. At the bottom of the interface, there are four buttons: 'New Rule', 'Delete', 'Save Rule', and 'Categori Menu'. On the left side, there is a vertical panel labeled 'Tampilan treeview'.

Gambar 3.7 Tampilan Rule Maintenance

Apabila sub menu rule verification di pilih maka muncul tampilan pilihan dari berbagai macam proses verifikasi rule dengan memberi tanda cek pada bagian yang dipilih.

Kategori

Tampilan treeview

Beri tanda cek untuk verifikasi yang diinginkan

Redundant Rules Dead End Rules

Conflicting Rules Circular Rules

Subsumed Rules Missing Rules

Unnecessary If Condition Rules Unreachable Rules

Hasil verifikasi

Gambar 3.8 Tampilan pilihan proses verifikasi rule

3.6.2 Perancangan Interface Inference

Apabila processes dipilih dari menu utama maka interface inference akan muncul yang mana kita dapat memproses rule-rule yang kita inputkan berdasarkan kategori dan metode penelusuran yang kita pilih. Pemilihan kategori dan metode dapat dilihat dari gambar 3.9

kategori

Backward Processes Forward Processes View Logs

Gambar 3.9 Tampilan Proses – Pemilihan Metode

Setelah kategori dan metode dipilih maka hasil dari sistem yang berupa sistem pakar yang mana menampilkan input untuk konsultasi yang digunakan untuk mengisi setiap pertanyaan yang diajukan oleh sistem.

Input Data Untuk Backward Inference Engine

Pertanyaan

Ok

Gambar 3.10 Tampilan Input Konsultasi

Setelah selesai mengisi setiap pertanyaan yang diajukan maka muncul form pada gambar 3.11 dibawah ini yang merupakan hasil/kesimpulan dari konsultasi dengan menampilkan data inputan dari user, kesimpulan yang didapat melalui proses inference.

The image shows a screenshot of a software interface titled "Sistem Pakar". At the top, there is a text input field. Below it, a larger text area contains the message "Data yang anda masukkan adalah sebagai berikut". Underneath this is a field labeled "Kesimpulan". At the bottom of the interface, there are three buttons: "Treeview" on the left, and "Ingin Konsultasi lagi?" in the center, with "Ok" and "Tidak" buttons to its right.

Gambar 3.11 Tampilan Hasil Kesimpulan

Pada form diatas terdapat tombol treeview digunakan untuk menampilkan treeview yang berisi rule - rule yang digunakan sebagai acuan untuk mendapatkan kesimpulan.