

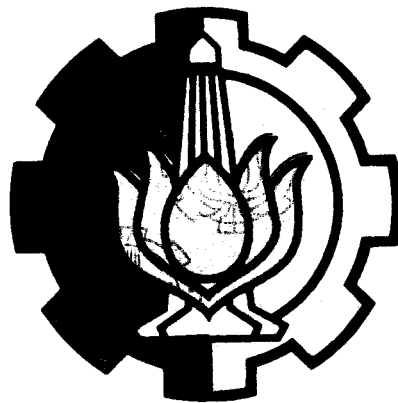
19.429/ITS/4/2004



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

TUGAS AKHIR

DESAIN DAN IMPLEMENTASI SISTEM PAKAR UNTUK PERENCANAAN PRODUKSI JAGUNG DALAM INDUSTRI PERTANIAN



R51f
006.33
Ast
d-2

2001

PERPUSTAKAAN ITS	
Tgl. Terima	16-7-2003
Tesiman Baru	H
No. Agenda Perp.	218269

Oleh :

RHEIN ASTRISANDY
NRP. 2695.100.027

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2001**

DESAIN DAN IMPLEMENTASI SISTEM PAKAR UNTUK PERENCANAAN PRODUKSI JAGUNG DALAM INDUSTRI PERTANIAN

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan

Untuk Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

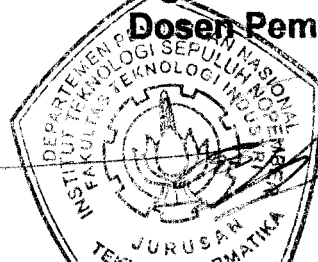
Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing



Dr. Ir. ARIE DJUNAIDY, M.Sc

NIP. 131 633 403

SURABAYA

PEBRUARI, 2001

Bismillaahirrahmaanirrahiim

“Hai orang-orang yang beriman, bertakwalah kamu dengan takwa yang sesungguhnya dan jangan sekali-kali kamu mati kecuali dalam berserah diri kepada Allah” (QS 3:102)

“ Hai orang-orang yang beriman , mintalah pertolongan dari Allah dengan kesabaran dan shalat. Sesungguhnya Allah bersama orang-orang yang sabar” (QS 1:153)

*Kupersembahkan Untuk
Ayahanda dan Ibundaku tercinta*

ABSTRAK

Perencanaan produksi dalam industri pertanian yang melibatkan pertanian rakyat, memiliki banyak faktor yang perlu dipertimbangkan. Pada umumnya aktifitas perencanaan dilakukan seorang konsultan yang biayanya sangat mahal dan hasilnya pun sangat relatif dan perlu dilakukan setidaknya setiap tahun karena memungkinkan terjadinya perubahan nilai faktor yang berpengaruh. Aktifitas perencanaan tersebut dapat diselesaikan dengan menggunakan sebuah sistem yang diasosiasikan dengan model yang dapat menunjukkan tingkat kepakaran dalam menyelesaikan suatu persoalan yang setara dengan seorang pakar yang disebut sebagai sistem pakar.

Dalam Tugas Akhir ini dirancang dan dibuat suatu sistem pakar untuk perencanaan produksi jagung dalam industri pertanian. Dalam sistem pakar ini pengguna dapat menambah dan mengedit aturan yang telah ada sehingga sistem pakar tersebut selalu *up to date*. Dengan fasilitas analisa harga, pengguna dapat mengetahui kemungkinan harga pada tahun tertentu, besarnya fluktuasi harga akibat fluktuasi rupiah serta memperoleh saran yang sesuai dengan kondisi harga yang dimasukkan oleh pengguna. Dengan fasilitas tersebut pemakai dapat mempergunakan saran terhadap kondisi harga sebagai bahan pertimbangan dalam melakukan aktifitas produksi. Dengan mekanisme *reasoning* yang digunakan pemakai dapat memperoleh informasi dan saran tentang kondisi yang dimiliki dan dapat melakukan pengecekan apakah kesimpulan yang diambil pengguna tepat. Dengan komponen penjelas *how* dan *why* pemakai dapat memperoleh informasi tentang mengapa dan bagaimana kesimpulan tersebut diperoleh.

Berdasarkan hasil ujicoba, terutama untuk kasus-kasus Perencanaan Produksi Pertanian yang melibatkan pertanian rakyat yang aturannya telah dimasukkan ke dalam sistem, sistem mampu memberikan saran sesuai dengan yang diharapkan. Sistem Pakar ini sangat berguna untuk membantu industri pertanian yang memanfaatkan pertanian rakyat dan petani itu sendiri dalam merencanakan produksi jagung. Agar diperoleh analisa harga yang lebih mendekati kenyataan sebaiknya dipergunakan metode analisa harga yang lebih sesuai terhadap kondisi data yang ada.

KATA PENGANTAR

KATA PENGANTAR

Alhamdulillah rabbi 'aalamin Segala puji bagi Allah serta sholawat dan salam bagi nabi Muhammad . Atas berkat rahmat Allah Swt telah diselesaikan tugas akhir dengan judul:

DESAIN DAN IMPLEMENTASI SISTEM PAKAR UNTUK PERENCANAAN PRODUKSI JAGUNG DALAM INDUSTRI PERTANIAN

Tugas akhir ini tidak akan mampu diselesaikan tanpa dukungan dari semua pihak baik moril maupun materiil dari berbagai pihak. Pada kesempatan ini saya menyampaikan terima kasih pada semua pihak yang telah memberikan dukungan dan dorongan.

- Bapak Dr. Ir. Arif Djunaidy, selaku pembimbing dan ketua jurusan teknik Informatika, yang selalu meluangkan waktu untuk memberikan bimbingan serta memberikan dorongan dan memberikan motivasi dalam menyelesaikan tugas akhir ini.
- Bapak Prof. Dr. Soemarno, M.S. selaku sumber dari basis pengetahuan dari sistem yang dibangun Rhein yang telah dengan sabar meluangkan waktu disela-sela kesibukan beliau untuk menjawab segala pertanyaan yang diajukan serta memberikan penjelasan selengkap mungkin. Tanpa bantuan beliau sangatlah tidak mungkin dibangun basis pengetahuan untuk sistem ini.
- Bapak Ir. Cholil Noor Ali selaku Dosen Wali.

- Para Dosen pengajar Teknik Informatika atas curahan ilmu yang telah diberikan. Semoga menjadi ilmu yang bermanfaat.
- Staf Tata Usaha dan karyawan Jurusan Teknik Informatika: Mas Yudi, Pak Muin, Mbak Irna, terima kasih atas bantuannya selama Rhein menjalani kuliah di Informatika.
- Staff dan Pejabat Lembaga Penelitian Universitas Brawijaya, atas bantuan literatur dan diskusi yang diberikan pada Rhein.
- Ayahanda tercinta yang telah memberikan kesempatan kepada Rhein untuk menjalani segala proses dalam hidup ini secara mandiri dengan segala fasilitas dan arahan yang diberikan. Penyelesaian tugas akhir ini merupakan sebagian kecil usaha Rhein untuk menunjukkan bakti kepada Ayahanda. Semoga Allah memperlancar jalan Ayahandaku tercinta.
- Ibunda tercinta dengan iringan doa dan kasih sayang yang tanpa henti mengiringi Rhein, memberikan kekuatan tersendiri. Kesabaran dan segala perhatian ibunda dalam mendidik Rhein tidak akan pernah mampu Rhein balas dengan apapun, karena memang bunda tidak pernah mengharap balasan apapun. Berkat dorongan bunda akhirnya Rhein mampu menyelesaikan tugas akhir ini. Semoga Allah melimpahkan Kasih dan Sayangnya untuk Ibundaku tercinta.
- Budheku sayang, perhatian dan kasih sayang yang diberikan kepada Rhein memberikan bagian penting. Cerita-cerita menarik yang budhe tuturkan selalu menjadi bahan refreasing yang menyegarkan bagi Rhein. Semoga Allah selalu memberikan rahmatNya pada budheku tercinta.

- Dhona, adikku tercinta terimakasih atas segala perhatian dan pertanyaanmu, terutama berkaitan dengan kapan Rhein lulus yang terus terang sangat membantu ketika semangat Rhein untuk menyelesaikan tugas akhir ini mulai turun. Semoga tercapai segala cita-citamu, dan jadi dokter yang baik ya...Partner
- Ujang, adikku tercinta terima kasih atas perhatian dan segala bantuanmu, kamu memang pendengar dan pemimpin yang baik. Percayalah apapun yang kamu putuskan itu pasti usaha terbaik yang bisa kamu lakukan. Semoga sukses dan ridla Allah selalu menyertaimu, hanya dengan menjadi dirimu sendiri itu lebih dari cukup.
- Yasmine, adikku sayang yang lucu dan cantik semoga menjadi anak yang solihah. Sayang dan cinta Rhein selalu bersamamu.
- Teguh, adikku tercinta semoga menjadi anak yang soleh dan tercapai segala cita-citamu.
- Ika, saudaraku yang baik, terimakasih atas segala bantuan yang telah diberikan kepada Rhein, terutama kesediaannya mengantar ke kampus ketika Rhein harus datang malam-malam ke kampus untuk praktikum. Semoga tercapai cita-citamu dan Allah memberikan yang terbaik untukmu.
- Tien, Tut, Nung, Arie, Wawan, Faried, Denise, Dian, Shinta, Ratih,.. beserta saudara-saudara sepupuku yang lain, segala kebersamaan diantara kita memberikan makna penting. Cintaku pada kalian semua tidak akan pernah berkurang meskipun perhatianku agak berkurang.

- Fia dan Hida kalian selalu menjadi teman dan sekaligus kakak yang baik perhatian dan dorongan kalian memberikan energi yang besar dalam penyelesaian tugas akhir ini.
- Yuni, kamu spesial kind of person , saran kamu, perhatian kamu, gaya kamu yang kocak, selalu punya cara untuk nasehatin Aku, terutama sikap kamu yang selalu bilang 'Usahakan dulu Rhen sampai kamu benar-benar tidak mampu' selalu memberikan semangat dan kesegaran buat Aku dalam memandang hidup.
- Dwi dan Mbak Nafiah, terima kasih atas pinjaman komputernya dan tumpangnya yang sangat membantu ketika ujian.
- Yuli, Elok, Dini, Eny, Mukib, Naim, Isti, selalu menyenangkan ngobrol ditelpon dengan kalian, buat Yuli, Elok dan Naim semangat dong, ayo dikerjain. Apa suporternya kurang? Buat Eny jangan sampe menyiksa diri loh, enjoy saja.
- Pak Yuli, Haris, Handa, Eko, Zakky menyenangkan berteman dengan kalian terima kasih atas segala saran yang telah diberikan, setiap saran kan berarti perbaikan.
- Seluruh teman-teman TC '95 atas kebersamaannya dalam menempuh perkuliahan selama di T.Informatika.
- Mahasiswa kedokteran Universitas Brawijaya angkatan 97, menghabiskan waktu bersama kalian adalah sangat berkualitas. Terima kasih atas bantuannya selama Rhein berada di Brawijaya.

- Mbak Nis, Mbak Us, Mbak Nurul terima kasih atas bantuan dan bimbingannya untuk lebih mengenal apa arti kehidupan, kalian telah menjadi kakak yang baik semoga diberikan ilmu yang bermanfaat.
- Angga, Ulfa, Mamiek, Ani it was fun to life with you girls. Terima kasih atas dorongannya untuk segera menyelesaikan tugas akhir ini.
- Yudie, Joko, Iqbal, Asmunin, Qoyyim, Fauzy, Fahrul, Andy, Patra, tanpa kalian sadari kalian telah memberikan dorongan agar Rhein segera lulus. Terimakasih atas dukungan dalam menyelesaikan pekerjaan kantor yang sangat membantu dalam meringankan Aku.
- Mbak Tri, terima kasih karena selalu mendorong-dorong untuk segera berangkat bimbingan, dorongan tersebut membangkitkan semangat Rhein untuk segera menyelesaikan tugas akhir.
- Agung, terima kasih atas telpon-telponnya yang selalu tanya kapan Rhein selesai kuliahnya. Dan cerita serta informasi tentang tantangan bisnis yang membuat Rhein memutuskan untuk segera menyelesaikan kuliah di TC, dan melihat sisi lain dunia.
- Yanto, kamu memang super peduli, terimakasih atas segala bantuan, nasehat, saran, omelan-omelan, kritik pedasnya, yang memberikan semangat untuk terus berjuang dalam hidup ini dan melakukan yang terbaik yang bisa dilakukan.
- Endah, temanku yang punya idealisme yang tinggi selamat kamu hampir mencapai idealismemu. Terima kasih telah menjadi teman dan tempatku

bercerita. Dan terima kasih atas teriakannya untuk segera lulus. Friends Forever.

- Mbak Yati, teman dan kakak yang pandai, darimu Rhein bisa belajar banyak hal, memahami orang lain dalam perspektif yang berbeda, dunia tidak sekedar hitam dan putih. Terima kasih atas pesannya 'jangan lupa TA-nya', Rhein tidak lupa 'kan.
- Mbak Nurul, Mbak Tiwi, Mbak Hunun, Mbak Nisa terimakasih atas perhatian dan doa yang dipanjatkan untuk Rhein.
- Dan semua pihak yang tidak bisa penulis sebutkan satu-persatu, yang telah banyak membantu terselesaikannya Tugas Akhir ini.

Semuga Allah memberikan balasan yang sebaik-baiknya. Sekali lagi terima kasih kepada semua pihak yang membantu terwujudnya tugas akhir ini.

Tugas akhir ini belum sampai pada tahap sempurna, maka saran dan kritik dari pembaca sangat diharapkan. Semoga karya ini dapat memberikan manfaat dan dicatat sebagai ilmu yang bermanfaat.

Amiin ya Robbal Alamiin.

Sidoarjo, 8 Februari 2001

Rhein Astrisandy

DAFTAR ISI

	Halaman
LEMBAR JUDUL	
LEMBAR PENGESAHAN	
LEMBAR DEDIKASI	
ABSTRAK	
KATA PENGANTAR	i
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat	2
1.3 Perumusan Masalah	2
1.4 Batasan Masalah	3
1.5 Metodologi Pelaksanaan Tugas Akhir	4
1.6 Sistematika Penulisan	5
BAB II Sistem Pakar	7
2.1 Definisi Sistem Pakar	7
2.2 Pentingnya sistem pakar	11

2.3	Seni Pengambilan keputusan	12
2.4	Konsep Sistem Pakar	15
2.5	Komponen dari pengetahuan dalam sistem pakar.....	19
2.6	Beberapa pertimbangan dalam representasi pengetahuan	20
2.6.1.	Model Alternatif dalam Representasi.....	21
2.7	Akuisisi Pengetahuan.....	28
2.7.1.	Akuisisi Pengetahuan dan Domain Kepakaran	29
2.7.2.	Pemilihan Domain.....	30
2.7.3.	Pemilihan Perekayasa Pengetahuan	31
2.7.4.	Pemilihan Pakar.....	32
2.7.5.	Pengembangan sebuah Pohon Keputusan	33
2.8	Inference Engine	36
BAB III Perencanaan Produksi Jagung dalam Industri Pertanian		39
3.1.	Permintaan dan Produksi	39
3.2.	Perencanaan Produksi	41
3.3.	Proses yang ditempuh dalam mengambil keputusan	46
BAB IV Desain dan Pembuatan Perangkat Lunak.....		48
4.1	Perancangan komponen-komponen sistem	48
4.1.1.	Basis Pengetahuan Sistem Perencanaan Produksi jagung..	48
4.1.2.	Mekanisme Inferensi.....	61
4.1.3.	Komponen Penjelas.....	61

4.1.4.	Antarmuka Pemakai.....	62
4.1.5.	Kompunen Akuisisi.....	70
4.2	Pihak-pihak yang terlibat.....	70
6.2.1	Tahap perancangan.....	71
6.2.2	Tahap Implementasi	71
4.3	Perancangan data	72
4.3.1	Tlist.....	72
4.3.2	Properti dan Method yang digunakan dalam TList.....	73
4.3.3	Working Memory	80
4.3.4	Basis Pengetahuan.....	81
4.4	Perancangan Proses	86
4.4.1.	Forward Chaining.....	86
4.4.2.	Backward Chaining.....	88
4.5	Implementasi Penyajian Aturan	94
4.6	Implementasi Pencarian dan Penyimpanan Data	95

BAB V Ujicoba dan Evaluasi..... 109

5.1	Kasus Pertama.....	109
5.2	Kasus Kedua (menguji Kesimpulan yang Diambil).....	116
5.3	Kasus Ketiga (analisa harga).....	119
5.4	Evaluasi Sistem	123

BAB VI PENUTUP	127
6.1 Kesimpulan	127
6.2 Kemungkinan Pengembangan Lebih Lanjut	129
DAFTAR PUSTAKA	130
LAMPIRAN A: Aturan	A.1
LAMPIRAN B: Dokumentasi Meeting	B.1
LAMPIRAN C: Petunjuk Penggunaan	C.1



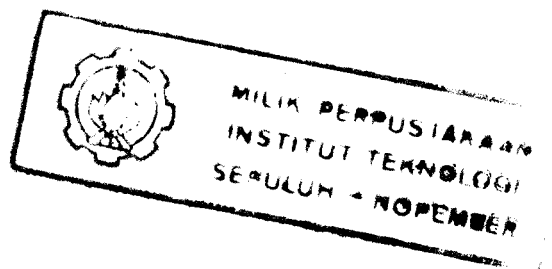
DAFTAR GAMBAR

Gambar 2.1	Generic Expert Systems	15
Gambar 2.2	OAV Networks.....	23
Gambar 2.3	Pohon Keputusan yang Mungkin	34
Gambar 2.4	Tree Lebih sederhana	35
Gambar 4.1	Pohon Keputusan Topoklimatologi	59
Gambar 4.2	Pohon Keputusan Tingkat Resiko.....	60
Gambar 4.3	Yes-No Question.....	63
Gambar 4.4	Bagan dialog Layar	64
Gambar 4.5	Menu Utama.....	65
Gambar 4.6	Melihat Semua Rule.....	65
Gambar 4.7	Menambah Atribut dan Value.....	66
Gambar 4.8	Mengedit Rule.....	66
Gambar 4.9	Menghapus Semua Rule.....	67
Gambar 4.10	Perkiraan	67
Gambar 4.11	Kondisi yang diketahui	68
Gambar 4.12	Amalisa Lokasi.....	68
Gambar 4.13	Kesimpulan	69
Gambar 4.14	ER-Diagram Basis Pengetahuan	82
Gambar 4.15	Proses Pengambilan Rule ke Working Memori	91
Gambar 4.16	Forward Chaining.....	92

Gambar 4.17	Backward Chaining.....	93
Gambar 4.18	Cara Penyimpanan Rule.....	94
Gambar 5.1	Lembar Data.....	110
Gambar 5.2	Lembar Pertanyaan.....	111
Gambar 5.3	Lembar Kesimpulan.....	112
Gambar 5.4	Combobox 'How'.....	113
Gambar 5.5	Combobox 'Why'.....	113
Gambar 5.6	Hasil How bagian pertama kasus pertama.....	114
Gambar 5.7	Hasil How bagian kedua kasus pertama.....	115
Gambar 5.8	Hasil Why kasus pertama.....	115
Gambar 5.9	Kesimpulan Benar.....	117
Gambar 5.10	How pada bagian backward.....	117
Gambar 5.11	Why pada bagian backward.....	118
Gambar 5.12	Input Data Harga.....	120
Gambar 5.13	Hasil Pengolahan Data Harga.....	121
Gambar 5.14	Saran dan Kesimpulan.....	122
Gambar 5.15	Hasil Proses Kasus A dengan 664 Klausa.....	123
Gambar 5.16	Hasil Proses Kasus A dengan 953 Klausa.....	124
Gambar 5.17	Hasil Proses inferensi Kasus B.....	124

DAFTAR TABEL

Tabel 2.1	Atribut Pesawat	33
Tabel 4.1	Jumlah penduduk, konsumsi	40
Tabel 4.2	Jumlah Produksi	40
Tabel 4.1	Atribut dan Value yang digunakan tipe Topoklimatologi	52
Tabel 4.2	Atribut dan Value yang digunakan dalam Tingkat Resiko.....	52
Tabel 4.3	Atribut dan Value yang digunakan tipe Teknologi.....	53
Tabel 4.4	Listing Atribut Topoklimatologi.....	53
Tabel 4.5	Listing Atribut Tingkat Resiko.....	54
Tabel 4.6	Listing Atribut tipe Teknologi.....	55
Tabel 4.7	Tabel Dialog	63
Tabel 5.1	Tingkat Harga Jagung Pipilan.....	119



BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Industri pertanian adalah industri yang memiliki prospek yang baik untuk dikembangkan. Dalam perencanaan produksi industri pertanian banyak hal yang perlu dipertimbangkan baik faktor permintaan pasar maupun kemampuan produksi dari industri tersebut.

Perencanaan produksi dalam industri pertanian yang melibatkan pertanian rakyat, memiliki banyak hal yang perlu dipertimbangkan baik faktor harga pasar maupun kemampuan produksi dari pertanian rakyat dan potensi alam. Pada umumnya aktifitas perencanaan dilakukan seorang konsultan yang biayanya sangat mahal dan hasilnya pun sangat relatif. Dan hal itu perlu dilakukan setidaknya setiap tahun karena pada saat itu potensi pasar memiliki peluang terjadinya perubahan.

Kompleksitas proses yang terjadi dalam perencanaan tersebut terutama yang berkaitan dengan banyaknya faktor yang perlu dipertimbangkan maka sangat memungkinkan untuk dikembangkan suatu sistem berbasis pengetahuan untuk menyelesaikan permasalahan tersebut. Dengan menggunakan Sistem Berbasis

Aturan seorang pakar tidak perlu melakukan analisa setiap tahun tetapi hanya perlu melakukan penambahan aturan jika dianggap aturan tersebut perlu dilengkapi untuk kepentingan Perencanaan Produksi Pertanian.

1.2. TUJUAN DAN MANFAAT

Tujuan dari penelitian ini adalah merancang software dengan kemampuan:

1. Memberikan analisa terhadap harga
2. Melakukan analisa terhadap kondisi topoklimatologi, tingkat resiko dan teknologi.

Hal ini dimaksudkan untuk membantu dalam pengambilan keputusan apakah suatu daerah pertanian rakyat tertentu dapat diusahakan jagung untuk kepentingan industri pertanian dan digunakan oleh petani untuk menentukan apakah lokasi yang dimiliki sesuai, yang diolah secara akurat dengan memperhatikan potensi pertanian rakyat tersebut. Dengan menggunakan mekanisme reasoning yang tepat.

1.3. PERUMUSAN MASALAH

Dalam Pembuatan software yang memiliki kemampuan untuk melakukan perencanaan produksi pertanian, ada dua proses yang harus dilakukan:

1. Analisa, yaitu melakukan analisa terhadap harga berdasarkan data lalu dengan menggunakan regresi linear sederhana.
2. Melakukan analisa terhadap kondisi topoklimatologi, tingkat resiko dan teknologi dengan menggunakan sistem pakar berbasis aturan untuk memberikan saran dan informasi tentang kondisi yang dihadapi.

Pada tugas akhir ini, perangkat lunak yang dibuat dispesifikasikan untuk memproses data-data sebagai berikut:

1. Data harga secara berkala untuk diproses dengan regresi sederhana sehingga menghasilkan informasi harga pada tahun yang diinginkan.
2. Data kondisi ideal faktor produksi untuk dibandingkan dengan data dari faktor produksi yang dimiliki dan data hasil proses analisa harga untuk diproses melalui mekanisme reasoning .

1.4. BATASAN MASALAH

Untuk melakukan penelitian dalam pengembangan software ini saya memberikan batasan-batasan yaitu sebagai berikut:

1. Tidak memperhitungkan faktor bencana alam dalam menganalisa potensi suatu daerah kecuali yang bersifat historical (hanya sebagai informasi).

2. Analisa permintaan diasumsikan sangat besar sehingga tidak diperlukan analisa permintaan oleh sistem.
3. Tanaman pertanian yang menjadi produk dibatasi jagung
4. Daerah pertanian yang dianalisa potensinya adalah daerah pertanian yang menjadi tempat memproduksi oleh industri tersebut terutamanya pertanian rakyat.

1.5. **METODOLOGI PELAKSANAAN TUGAS AKHIR**

Metodologi untuk mengembangkan software ini adalah dengan tahap-tahap sebagai berikut:

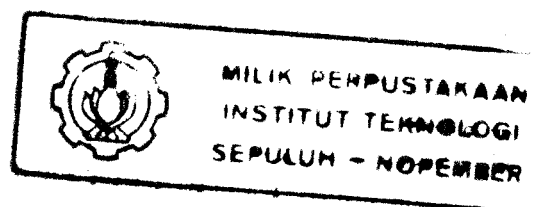
- A. Studi Literatur.** Pada tahap ini melakukan studi literatur tentang knowledge base system terutama dalam hal desain. Melakukan diskusi dengan ahli dan praktisi dibidang pertanian terutama yang berkaitan dengan syarat-syarat yang harus dipenuhi dalam mengembangkan suatu produk pertanian, dan perencanaan produksi pertanian.
- B. Melakukan Analisa Kebutuhan.** Pada tahap ini dilakukan analisa terhadap hal-hal yang harus dikembangkan dalam sistem tersebut.
- C. Melakukan Knowledge Modeling.** Pada tahap ini dilakukan pembuatan pemodelan dari mekanisme reasoning yang telah dibuat.

- D. Membangun Sistem Berbasis Pengetahuan.** Pada tahap ini dilakukan pengembangan sistem berdasarkan mekanisme reasoning yang tepat.
- E. Evaluasi dan Tes.** Pada tahap ini dilakukan uji coba untuk menyelesaikan kasus-kasus yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan bila dirasa ada yang kurang
- F. Membangun Sistem yang Sesungguhnya.** Membangun sistem dengan menggunakan metode reasoning yang telah melalui proses evaluasi dan tes.
- G. Pembuatan laporan.** Pada tahap ini dilakukan penulisan laporan, diantaranya menjelaskan tentang dasar teori yang dipergunakan, gambaran sistem, proses reasoning dan lain-lain.

1.6. SISTEMATIKA PENULISAN

Sistematika pembahasan mengenai perangkat lunak yang dibuat dalam Tugas Akhir ini disusun dalam beberapa bab, yang dijelaskan sebagai berikut:

- ↳ **Bab I** berisi Pendahuluan yang memuat latar belakang pembuatan perangkat lunak, perumusan masalah, tujuan penelitian, pembatasan masalah, metodologi penelitian, dan sistematika penulisan buku Tugas Akhir ini.



- ↪ **Bab II** berisi konsep yang berkaitan dengan sistem pakar itu sendiri yang menjadi landasan rancang bangun Sistem Berbasis Aturan ini.
- ↪ **Bab III** berisikan konsep dan praktis tentang perencanaan produksi jagung, apa saja faktor yang mempengaruhi aktifitas perencanaan produksi jagung. Jenis jenis orientasi dari aktifitas produksi jagung dalam industri pertanian.
- ↪ **Bab IV** berisikan perancangan sistem, yang meliputi deskripsi singkat perangkat lunak, komponen-komponen sistem, pihak-pihak yang terlibat dalam pembangunan dan penggunaan sistem, perancangan data, perancangan proses, serta implementasi dari perangkat lunak. Dalam perancangan sistem dibahas mengenai proses akuisisi pengetahuan, proses penyajian pengetahuan, dan proses inferensi. Sedangkan implementasi perangkat lunak membahas implementasi dari tiap-tiap proses penting dalam prosedur-prosedur pemrograman.
- ↪ **Bab V** mengulas tentang hasil uji coba sistem serta evaluasi terhadap hasil kerja sistem tersebut.
- ↪ **Bab VI** merupakan penutup yang berisi kesimpulan serta saran-saran untuk pengembangan lebih lanjut dalam upaya memperbaiki berbagai kelemahan yang ada untuk mencapai hasil yang lebih baik.

BAB II

SISTEM PAKAR

Dalam bab ini akan dibahas mengenai definisi sistem pakar. Manfaat sistem pakar dalam membantu menyelesaikan masalah. Komponen-komponen pendukung sistem pakar. Konsep perancangan dan pembangunan Sistem Berbasis Aturan, yaitu tahapan-tahapan apa saja yang perlu untuk dikerjakan dalam pembuatan sistem serta hal-hal lain yang harus diperhatikan .

2. 1. DEFINISI SISTEM PAKAR

Pada dasarnya tidak ada definisi tunggal untuk sistem pakar. Pada kenyataannya untuk suatu kondisi tertentu lebih cenderung untuk mendefinisikan sistem pakar sebagai suatu sistem yang secara esensial berbeda dari sistem komputer yang lain. Welbank (1983) mendefinisikan sistem pakar sebagai berikut:

"Sistem pakar adalah sebuah program yang memiliki basis pengetahuan pada sebuah domain tertentu dan menggunakan inferential reasoning yang kompleks untuk melakukan suatu pekerjaan dimana seorang pakar dapat melakukannya." [HAR-89]

Definisi yang lain mengatakan:

"Sistem pakar dapat didefinisikan sebagai sebuah program yang dapat menunjukkan tingkat kepakaran dalam menyelesaikan suatu persoalan (untuk suatu bidang tertentu dan spesifik) setara dengan seorang pakar."

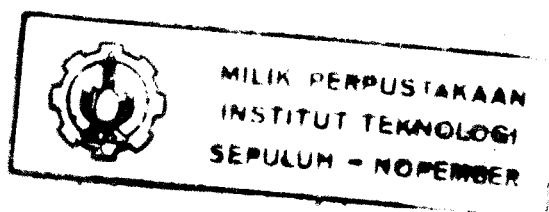
Bagaimanapun dapat dilihat bahwa ada masalah dengan definisi diatas. Secara spesifik seolah dengan jelas mengatakan bahwa sistem pakar adalah tidak lebih dari sekedar sebuah program komputer atau merupakan alternatif terbaik dari pemrograman konvensional. Dalam kenyataannya komputer dan pemrograman hanyalah merupakan alat "pendukung" dalam mengimplementasikan dan memberikan solusi terhadap suatu sistem pakar.

Sebagai konsekuensi logis kita melihat bahwa definisi diatas hanya melihat sebagian perspektif dari sistem pakar yaitu menyediakan dukungan komputasi yang dibutuhkan untuk mengimplementasikan metodologi yang ada. Bagaimana pun sistem pakar berbeda dengan hal diatas. Secara spesifik, ketika kita setuju bahwa komputer dan pemrograman komputer adalah sumber esensial untuk mendukung implementasi dan solusi dari sistem pakar sehingga berarti, adalah tidak tepat, yang membuat suatu sistem pakar berarti adalah esensi dari sistem pakar itu sendiri. Pada faktanya, seseorang dapat mengembangkan sistem pakar dan menghasilkan

suatu konklusi melalui sebuah proses manual secara langsung. Sehingga kita harus mampu mengambil faktor utama dalam membangun suatu sistem pakar, yaitu formasi dari basis pengetahuan yang berkaitan (atau lebih akurat, formasi dari model basis pengetahuan). Sehingga dapat kita lihat, peran utama dari perekayasa pengetahuan adalah untuk menyediakan dan merepresentasikan, dalam bentuk basis pengetahuan, aturan-aturan yang digunakan oleh seorang pakar dalam memberikan solusi terhadap suatu persoalan. Secara lebih tepat, perekayasa pengetahuan membangun model dari basis pengetahuan.

Peran dari perekayasa pengetahuan, seperti yang digambarkan diatas, adalah berkaitan dengan disiplin ilmu yang lain. Sebagai contoh, insinyur dan ilmuwan seringkali memfokuskan perhatian utama mereka pada pengembangan model kuantitatif. Management scientists, operation researchers, dan systems engineers, pada umumnya menggunakan model untuk meningkatkan nilai pekerjaan mereka. Sebagai contoh, seorang management scientists mungkin akan membangun sebuah model input output untuk sebuah sektor industri.¹ Dari pandangan penyederhanaan diatas, model semacam itu menyediakan layanan untuk melihat indikasi relasi antara berbagai elemen dalam sistem, dan efeknya pada output dari sistem

¹ Menurut O'Connor and Henry [1975] input-output analysis adalah berkaitan dengan menganalisa ketergantungan antara produksi dan konsumsi unit pada ekonomi modern dan menunjukkan kaitan antara sektor yang berbeda yang menyediakan barang dan jasa dari sektor lain dan perubahan persediaan produksi barang dan jasa yang terjual pada sektor lain.



apabila dilakukan perubahan pada inputnya. Data yang berkaitan dengan model ini mungkin akan diinputkan pada sebuah program komputer yang menyediakan layanan untuk menyelesaikan model input-output (yaitu sebuah program yang secara esensial mencari penyelesaian untuk sebuah deretan persamaan linear) [O'Connor dan Henry 1975]. Sejak adanya software semacam ini beserta kemampuannya, tenaga dan fikiran diusahakan untuk terlibat dalam pengembangan model input-output. Lebih jauh, hasil dari proses komputer (tidak peduli seberapa rumit programnya atau bahasa apa yang akan digunakan) secara total sangat tergantung pada keakuratan dan integritas dari model. Ketika seorang management scientists harus segera memahami bagaimana sebuah program dikembangkan untuk memperoleh hasil yang diinginkan, adalah tidak perlu (dan mungkin pada kasus tertentu akan menghabiskan waktu) bagi ilmuwan manajemen untuk mengembangkan software semacam ini sendiri atau memiliki pengenalan dengan bahasa tertentu yang digunakan oleh software. [IGN-91]

Sehingga, perlu kita cek definisi sebelumnya tentang sistem pakar untuk mencerminkan point-point yang terdapat pada diskusi diatas. Secara spesifik,

"Sebuah sistem pakar dapat didefinisikan sebagai suatu model dan prosedur yang diasosiasikan dengan model yang dapat menunjukkan tingkat

kepakaran dalam menyelesaikan suatu persoalan yang setara dengan seorang pakar, [IGN-91]

Yang dimaksud model diatas adalah representasi dari basis pengetahuan dari seorang pakar.

Sebagai catatan bahwa tidak semua sistem pakar dikembangkan dengan cara suatu interaksi (atau setidaknya interaksi langsung) dengan seorang pakar (yang biasa disebut tahap akuisis pengetahuan). Sistem pakar dapat dan sering dikembangkan dalam suatu kondisi tidak ada seorang pakar atau tanpa campur tangan seorang pakar. Alternatif pengembangan ini disebut "*knowledge base system*". Namun karena kepakaran pada akhirnya harus ditangkap dalam "*knowledge base system*" maka istilah sistem pakar menjadi lebih sesuai untuk digunakan

2.2. PENTINGNYA SISTEM PAKAR

Metodologi dari sistem pakar akan terus mempunyai peranan yang sangat penting dalam kaitan dengan sistem pendukung keputusan sebagai akibat langsung dari semakin meningkatnya jumlah dan kompleksitas kebutuhan manusia.

Sebelum sistem pakar diterima sebagai alternatif untuk memberikan solusi terhadap berbagai persoalan yang dihadapi manusia banyak orang

memanfaatkan jasa para pakar untuk menyelesaikan persoalan yang dihadapinya. Mereka yang berhubungan dengan seorang pakar mengakui bahwa:

- kepakaran merupakan komoditas yang mahal dan langka
- tingkat kepakaran dapat sangat bervariasi
- keputusan, intuisi, dan pengalaman merupakan peranan kunci dari kepakaran.

Sehingga adanya alternatif yang lain akan langsung mengalihkan minat sistem pakar misalnya. Jika sistem semacam ini dibandingkan tingkat kepakarannya (yaitu tidak harus secara penuh bergantung pada pakar) dan lebih murah, harus secara pasti menarik dan sesuai dengan permintaan - yang merupakan salah satu alasan untuk memperoleh keuntungan pada konsep sistem pakar dalam sektor komersial. Alasan lain untuk mempertimbangkan sistem pakar adalah kompetitif.

2.3. SENI PENGAMBILAN KEPUTUSAN

Pengambilan keputusan sangatlah bervariasi mulai dari yang rutin dan sederhana sampai ke yang kompleks dan membutuhkan banyak waktu. Sebagai contoh, sebelum tidur dimalam hari kita mungkin harus mengambil sebuah keputusan pada pukul berapa weker kita akan diset agar kita bangun

tepat pada waktunya. Sangatlah jelas bahwa kita tidak perlu melakukan analisa detail untuk situasi semacam itu. Kita dapat dengan mudah memperoleh solusi dengan estimasi yang sangat sederhana tentang waktu yang dibutuhkan untuk bersiap-siap. Pada lain sisi, mempertimbangkan suatu keputusan untuk suatu perusahaan apakah perlu untuk membangun pabrik baru atau, alternatif lain yaitu memodernisasi fasilitas yang ada. Dalam realisasinya pembuatan keputusan membutuhkan keberadaan keempat faktor berikut:

1. Harus ada persoalan
2. Harus ada pembuat keputusan
3. Harus ada kebutuhan untuk menyelesaikan persoalan
4. Harus ada solusi-solusi alternatif terhadap persoalan yang timbul

Jika keempat elemen ini telah ada, maka terdapat berbagai metode sehingga kita dapat menurunkan kandidat solusi dari persoalan yang timbul untuk disajikan pada pembuat keputusan. Disiplin yang secara khusus berkaitan dengan pengembangan dan implementasi fasilitas pembuatan keputusan disebut sebagai "*decision analysis*"². Orang yang bekerja dengan

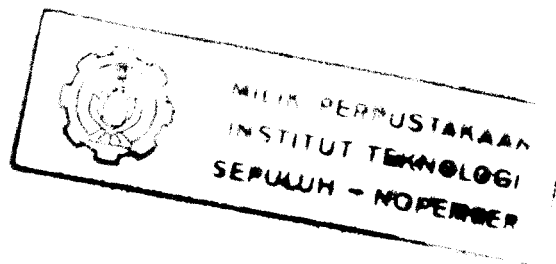
² terfokus pada beberapa bidang diantaranya management science, operations researce, dan sistem engineering adalah pada faktanya secara spesifik diarahkan untuk analisa keputusan. Bagaimanapun hal itu sangat jelas bahwa analisa keputusan adalah masalah yang secara rutin berhadapan pada banyak disiplin -ada atau tidak hal itu dalam disiplin yang dikembangkan dalam metode formal untuk mendukung pengambilan keputusan

disiplin ini dan menyajikan solusi alternatif pada pengambil keputusan disebut "*decision analysts*"

Sistem pakar dapat dipandang sebagai salah satu fasilitas (sebagai yang terpenting tentunya) untuk analisa keputusan. Sebagai kosekuensi, untuk memperoleh pemahaman yang lebih baik dan menghargai sistem pakar, adalah sangat penting untuk memahami dan menghargai analisa keputusan, elemen pendukungnya, evolusinya, dan peran dalam proses pengambilan keputusan. Lebih jauh, dengan latar belakang semacam itu, hubungan, perbedaan, dan persamaan antara topik semacam ini sebagai kecerdasan buatan, sistem pakar, pemrograman heuristik, dan sistem pendukung keputusan harus dipertimbangkan untuk dibuat lebih jelas.

Pada saat ini pembuatan keputusan lebih banyak didasarkan pada beberapa teknik, seperti:

- Spreadsheet dan basisdata
- Analisa statistik
- Simulasi
- Berbagai metode optimasi

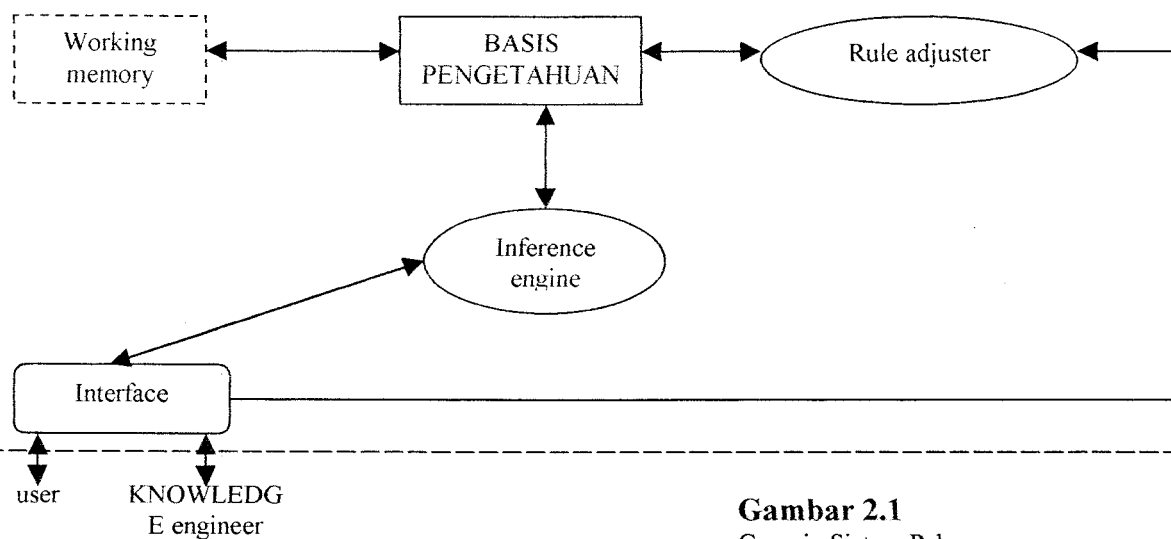


Tujuan dari analisa keputusan adalah untuk memberikan informasi yang berguna bagi pengambil keputusan untuk mendukung proses pengambilan keputusan, ketika informasi semacam ini telah dihasilkan melalui proses logika, scientific, dan proses sistematis. Kemudian terserah

pada pengambil keputusan untuk memutuskan bagaimana menginterpretasikannya dan menggunakan informasi pendukung ini dalam menghasilkan keputusan yang tepat.

2.4. KONSEP SISTEM PAKAR

Berikut ini adalah konsep dan komponen dari sebuah sistem pakar secara umum. Salah satu representasi dari sistem pakar dapat dilihat sebagai berikut:



Gambar 2.1
Generic Sistem Pakar

Semua komponen yang ada di atas garis terputus merupakan komponen yang ada dalam komputer. Sedangkan di bawah garis terputus, terdapat dua jenis pengguna dari sistem pakar: human user dan perekayasa pengetahuan.

Perekayasa pengetahuan adalah orang yang bertanggung jawab meletakkan pengetahuan kedalam basis pengetahuan sistem pakar dia melaksanakan tugas ini melalui interface dan rule adjuster.

Perekayasa pengetahuan juga sebagai interface antara seorang pakar (jika ada) dan sistem pakar. Sehingga perekayasa pengetahuan bagaimanapun harus menangkap kepakaran dari seorang pakar dan mengeskpresikan kepakaran ini dalam sebuah format yang mungkin dimasukkan kedalam basis pengetahuan dan akan digunakan oleh sistem pakar, karena itu dibutuhkan seorang perekayasa pengetahuan. Domain kepakaran akan berinteraksi secara langsung dengan sistem pakar maka akan menggantikan perekayasa pengetahuan.

Tipe individu kedua yang mengakses sistem pakar adalah **user**. Yang berarti semua orang yang akan menggunakan sistem pakar sebagai sebuah pembuat keputusan (sebagai seorang konsultan). Perekayasa pengetahuan yang sukses harus selalu ingat bahwa sistem pakar diadakan untuk kepentingan user, bukan untuk perekayasa pengetahuan atau domain kepakaran.

Interface mengatur semua input kekomputer dan kontrol dan format semua output. Interface juga mengendalikan semua komunikasi dengan perekayasa pengetahuan (atau domain kepakaran) ketika pengembangan basis pengetahuan sistem pakar.

Inference engine yang digunakan selama proses konsultasi. Selama proses konsultasi ia melakukan dua tugas utama. Pertama, memeriksa status dari basis pengetahuan dan working memory. Kedua, menyediakan pengendalian dari sesi konsultasi dengan menentukan urutan pemikiran yang harus dilakukan. Disebut juga sebagai "knowledge processor" yang bertugas menggabungkan fakta dan aturan untuk memperoleh (pemikiran) fakta yang baru.

Basis pengetahuan adalah seperti yang sering kita bahas berulang kali, adalah jantung dari sistem pakar secara tipikal berisi dua tipe pengetahuan yaitu aturan dan fakta. Fakta dalam basis pengetahuan mewakili beberapa aspek dari suatu domain spesifik dari suatu sistem pakar. Jika basis pengetahuan dibangun melalui interaksi dengan seorang pakar, aturan ini mewakili persepsi dari perekayasa pengetahuan secara heuristik yang dilakukan seorang pakar dalam mengambil keputusan. Basis pengetahuan menggunakan format basis aturan.

Working memory dapat berubah sesuai dengan persoalan yang sedang ditangani oleh sistem pakar. Isi dari memory kerja ini berupa sejumlah fakta. Namun, berbeda dengan fakta dalam basis pengetahuan fakta-fakta yang ada dalam working memori ini berupa fakta yang telah ditentukan untuk suatu persoalan tertentu yang diperoleh selama proses konsultasi. Secara lebih spesifik fakta-fakta ini diperoleh sebagai hasil dari proses pemikiran (inference process) dan disimpan dalam working memory.

Rule adjuster dalam sebagian besar sistem pakar yang ada saat ini rule adjuster hanya berfungsi sebagai editor untuk menyusun aturan ia bertugas memasukkan aturan-aturan yang dispesifikasikan oleh perekayasa pengetahuan kedalam basis pengetahuan selama fase pengembangan dari sistem pakar. Tugas lain dapat berupa pengecekan dari aturan-aturan yang diberikan terhadap konsistensi dan kelengkapan dari aturan-aturan tersebut. Dalam beberapa sistem pakar yang canggih dan ambisius, rule adjuster ini digunakan dalam upaya untuk melibatkan proses belajar. Untuk ini kita dapat mengajar sistem pakar dengan menyediakan satu set contoh dan memberikan kritik mengenai kinerja yang telah dilakukan. Jika kinerja tersebut tidak memuaskan maka rule adjuster secara otomatis melakukan revisi dari basis pengetahuan yang ada. Sebaliknya jika memuaskan rule adjuster akan memaksakan basis pengetahuan yang telah dipunyainya.

Sistem Pakar "Shell" melibatkan semua komponen seperti ditunjukkan dalam gambar 2.1 kecuali basis pengetahuan. Dengan menggunakan shell ini perekayasa pengetahuan diberi kebebasan untuk mengembangkan basis pengetahuan dan memasukkan basis pengetahuan ini kedalam arsitektur sistem pakar guna membentuk satu sistem pakar yang lengkap (dalam satu domain tertentu). Dengan adanya shell maka perekayasa pengetahuan dibebaskan dari beban untuk secara berulang mengembangkan semua elemen-elemen pendukung dari suatu sistem pakar

sehingga perancang pengetahuan dapat memfokuskan dirinya pada pengembangan basis pengetahuan.

2.5. KOMPONEN DARI PENGETAHUAN DALAM SISTEM PAKAR

Kita telah biasa mendengar "pengetahuan adalah kekuatan" hal ini menjadi semakin nyata dalam sistem pakar. Pengetahuan yang disimpan dalam suatu sistem pakar terdiri dari:

- Prior Knowledge, fakta dan aturan mengenai suatu domain tertentu yang diketahui sebelum sesi konsultasi dilakukan.
- Inferred knowledge, fakta dan aturan mengenai suatu kasus tertentu yang diturunkan selama berlangsungnya proses konsultasi (dan pada saat konsultasi) dari sistem pakar. Pengetahuan ini biasanya juga disebut kesimpulan atau aturan-aturan baru yang secara umum hanya bisa diperoleh jika sistem pakar mempunyai kapabilitas untuk "belajar".

Disini lebih difokuskan pada penggunaan basis aturan untuk pengetahuan tipe pertama. Pembahasan juga lebih difokuskan pada penggunaan basis aturan untuk merepresentasikan pengetahuan kepakaran (expert knowledge).

Seperti dibahas sebelumnya bahwa pengetahuan ditempatkan baik dalam basis pengetahuan (knowledge base) dari sistem pakar dan memori kerjanya (working memory). Pengetahuan yang ditempatkan dalam basis pengetahuan adalah pengetahuan dari tipe yang pertama yaitu prior fakta dan aturan mengenai suatu domain yang spesifik. Sedang pengetahuan yang disimpan dalam memori kerja merupakan tipe pengetahuan yang kedua yaitu pengetahuan turunan (inferred knowledge) mengenai suatu persoalan yang sedang diselesaikan. Pengetahuan yang disimpan dalam memori kerja ini bersifat dinamik karena selalu berubah untuk setiap persoalan yang ditangani.

2.6. BEBERAPA PERTIMBANGAN DALAM REPRESENTASI PENGETAHUAN

Batasan pertimbangan dalam representasi pengetahuan ini adalah bahwa kita menginginkan untuk menyimpan basis pengetahuan dari suatu sistem pakar dalam memori komputer, menggunakan representasi makna. Pada kenyataannya kita tidak dengan seksama memperhatikan apakah metode yang kita pergunakan untuk menyimpan memiliki beberapa kesamaan pada cara penyimpanan pengetahuan pada otak manusia. Yang diperlukan adalah mencari satu prosedur yang cepat dan efektif untuk diaplikasikan kedalam komputer. Hal ini tidak berarti membuat usaha kita untuk mempelajari otak dan memori manusia menjadi tidak penting. Dengan

demikian perhatian utama dalam hal ini adalah bagaimana menyajikan fakta dan aturan dalam basis pengetahuan untuk:

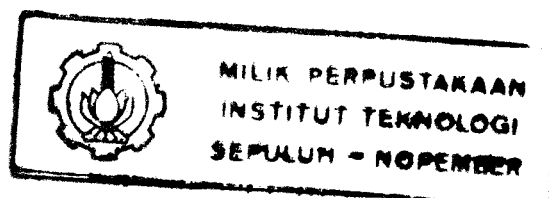
1. Menyediakan suatu format yang kompatibel dengan komputer.
2. Menjaga sedekat mungkin korespondensi antara format tersebut dengan fakta dan aturan yang sebenarnya (yaitu aturan yang dapat diterima oleh pakar dalam domain yang diinginkan).
3. Membangun suatu representasi yang dapat melakukan pelacakan dan modifikasi pengetahuan secara mudah.

Dalam kaitan dengan dua point diatas maka akan merupakan sesuatu yang sangat diperlukan untuk menggunakan format yang transparan, yaitu cara representasi yang dapat dibaca dan dimengerti secara mudah oleh manusia.

Beberapa model dari representasi pengetahuan telah dikembangkan. Seperti telah kita bicarakan, fokus utama akan difokuskan pada sistem berbasis pengetahuan untuk representasi pengetahuan.

2.6.1. MODEL ALTERNATIF DALAM REPRESENTASI PENGETAHUAN

Dalam merepresentasikan pengetahuan terdapat berbagai cara dan model untuk melakukannya. Tetapi perekayasa pengetahuan



diharapkan memilih satu diantara alternatif yang ada, yang paling sesuai dengan kondisi permasalahan yang dimiliki. Berikut ini beberapa alternatif dalam merepresentasikan pengetahuan.

2.6.1.1. AOV Triplets

Object Atribut Value triplets menyediakan cara yang menyenangkan untuk merepresentasikan fakta tertentu dalam sebuah basis pengetahuan untuk menyediakan basis dalam representasi dalam sebuah heuristik rule. Setiap AOV triplets berkaitan dengan beberapa spesifikasi entitas atau obyek.

Contoh pesawat memiliki beberapa atribut, diantaranya sebagai berikut:

- Jumlah mesin
- Tipe dari mesin
- Tipe dari desain sayap

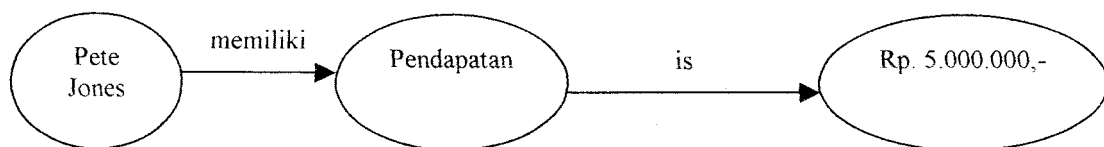
Untuk setiap atribut tersebut mempunyai beberapa value. Sebagai contoh, dalam kasus C130 pesawat kargo militer (dikenal sebagai hercules) , memiliki jumlah mesin 4, tipe mesinnya prop, dan desain sayapnya konvensional. Nilai dari AOV Triplets dapat berupa numerik atau simbolik. Fakta tersebut dapat dilist sebagai berikut:

- Jumlah mesin = 4

- Tipe dari mesin = prop
- Tipe dari desain sayap = konvensional

Dalam list tersebut diatas, object-nya sendiri tidak disebutkan. Biasanya statement diatas mewakili pasangan AV (atribut-value). Bagaimanapun yang berkaitan dengan pasangan AV adalah suatu obyek. Sehingga, beberapa pasangan AV menunjuk pada sebuah OAV triplets.

Cara lain bisa dipergunakan untuk merepresentasikan AOV triplets dengan menggunakan sebuah network representasi, seperti ditunjukkan gambar 2.2.



Gambar 2.2. OAV networks

Dari gambar tersebut menunjukkan bahwa objek-nya adalah Pete Jones dengan atribut Pendapatan dan value dari pendapatan tersebut adalah Rp. 5.000.000,-

2.6.1.2. Semantic Networks

Semantik network bisa dianggap tidak jauh berbeda dengan network yang digambarkan dalam multiple AOV triplets dalam bentuk

network yang digambarkan dalam gambar 2.2. Namun tidak hanya menunjuk pada satu atribut untuk sebuah single objek, semantik network dapat dipergunakan untuk merepresentasikan beberapa obyek dan beberapa atribut tiap obyek.

Sebagai catatan bahwa OAV triplets hanyalah sebuah subset dari semantik network yang hanya memiliki sebuah relasi yang mungkin digunakan yaitu "adalah" atau "memiliki". OAV nodes adalah satu diantara tiga tipe sebagai berikut: obyek, atribut atau value.

2.6.1.3. Frames

Semantic network memberikan arti penting dalam representasi pengetahuan, penggunaan frame sebagai pendekatan alternatif representasi yang dipergunakan untuk menangkap sebagian besar feature dari semantik network disamping menyediakan beberapa aspek tambahan. Pada faktanya, kita mungkin akan berfikir sebuah semantik network adalah sebagai subset dari konsep frames.

Penggunaan frame sebagai representasi merupakan cara untuk merepresentasikan pengetahuan. Sebuah frame berisi sebuah obyek dan beberapa slots dan semua informasi yang berkaitan dengan obyek. Isi dari slots itu pada umumnya adalah atribut dan value, dari suatu obyek tertentu. Bagaimanapun dalam menambahkan nilai dari setiap

atribut slot dapat berisi nilai default, pointer pada frame lain, dan set aturan atau prosedur yang mungkin diterapkan.

2.6.1.4. Logic Statement

Bentuk umum dari logika adalah proposional logika. Proporsion adalah statemen yang mungkin true atau false. Proporsion mungkin dihubungkan dengan beberapa operator seperti AND, OR, NOT, dan EQUIVALENT.

Predicate calculus merepresentasikan sebuah proposional logika. Elemen penting dalam predicate kalkulus adalah obyek dan predikat. Sebuah Predikat adalah sebuah statemen tentang sebuah obyek, atau sebuah hubungan yang dimiliki sebuah obyek.

2.6.1.5. Neural Networks

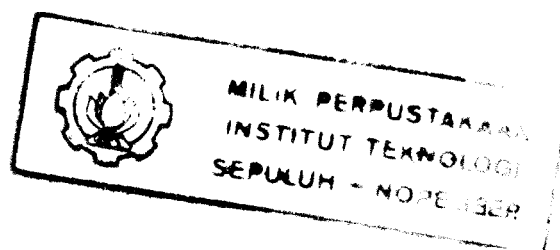
Jelas sekali bahwa, entah bagaimana, dan dengan cara apa, otak manusia menyimpan pengetahuan. Yang tidak jelas adalah tepatnya dalam bentuk apa hal ini dicapai. Neural Network merepresentasikan bagaimana manusia melakukannya, dari sisi hardware, teori yang mengarah pada otak manusia. Secara spesifik pengetahuan manusia disimpan dalam neuron (atau tepatnya dalam hubungan antar neuron).

2.6.1.6. Sistem Berbasis Aturan

Model representasi pengetahuan yang paling populer dalam sistem pakar saat ini, adalah model yang dikembangkan melalui penggunaan aturan, atau sistem berbasis aturan. Rule berarti IF...THEN atau production rule. Kita telah memilih sistem pakar berbasis aturan sebagai pendekatan kita dalam merepresentasikan pengetahuan untuk beberapa alasan, termasuk kepopulerannya dan keluasan penggunaannya. Bagaimanapun, perlu diketahui bahwa keputusan ini tidak menunjukkan bahwa sistem berbasis aturan adalah sebagai pendekatan yang terbaik, atau pendekatan terbaik pada segala situasi.

Pemilihan terhadap model representasi pengetahuan dengan menggunakan rule-base dilakukan karena alasan sebagai berikut:

- Sebagian besar paket pengembangan sistem pakar yang ada (juga termasuk sistem pakar shell) menggunakan basis aturan.
- Paket pengembangan sistem pakar yang menggunakan basis aturan pada umumnya lebih murah jika dibandingkan dengan model alternatif representasi yang lain.
- Keluasan keberadaan rule-base sistem pakar shell memudahkan perekrutan pengetahuan untuk lebih memfokuskan perhatiannya



pada fase yang paling kritis pada pengembangan sistem pakar yaitu pada basis pengetahuan

- Representasi aturan adalah model natural dari representasi pengetahuan. Sebagai konsekuensi, waktu yang dibutuhkan untuk belajar bagaimana mengembangkan rule-base adalah minimum.
- Diperlukan waktu yang lebih sedikit untuk belajar bagaimana menggunakan dan mengimplementasikan sistem pakar berbasis rule-base
- Rule adalah transparant, dan yang pasti lebih transparan dari model representasi yang menggunakan sistem berbasis aturan adalah frame dan neural network. Lebih jauh transparansi sering kali mengarah pada peningkatan keinginan, pada bagian manajemen, untuk menerima solusi yang diinginkan. Dan faktor ini tidak dapat diremehkan.
- Basis aturan dapat secara relatif mudah untuk dimodifikasi.

Model Rule-base yang digunakan dalam representasi pengetahuan yang disebut production rule atau disingkat rule [fikes and Kehler, 1985; Hayes-Roth,1985]. Biasanya aturan yang dimaksud adalah IF..THEN. Bagaimanapun pada beberapa kondisi berkembang termasuk IF..THEN..ELSE aturan.

Untuk memperjelas representasi kita sebaiknya memfokuskan pada IF..THEN aturan. Pada faktanya, secara umum lebih disarankan untuk menghindari penggunaan statemen ELSE pada rule-base sistem pakar. Dengan alasan:

1. Sejumlah sistem pakar komersial tidak mengijinkan penggunaan IF..THEN..ELSE
2. Validasi aturan semacam itu lebih sulit dari pada equivalen-nya dalam IF..THEN
3. Pada saat menjalankan proses inferensi rule semacam itu akan mengarah untuk mencapai kesimpulan (baik pada bagian THEN maupun pada bagian ELSE). Hal ini bisa menimbulkan hasil yang tak terduga.

2.7. AKUISISI PENGETAHUAN

Seperti halnya representasi pengetahuan (knowledge representation), akuisisi pengetahuan (knowledge acquisition) merupakan fase pengembangan sistem yang penting.

Fase akuisisi pengetahuan dalam pengembangan dapat dikategorikan sebagai pekerjaan yang menimbulkan frustrasi dan menghabiskan banyak waktu. Sering kali disebut sebagai *bottle neck* dalam pengembangan sistem pakar. Ketika berurusan dengan orang lain secara umum dapat menyulitkan, berurusan dengan domain kepakaran dapat menyita banyak waktu dan dapat

menimbulkan frustrasi. Untungnya ada beberapa batasan yang dapat mempermudah proses tersebut.

2.7.1. AKUISISI PENGETAHUAN DAN DOMAIN KEPAKARAN

Secara umum dapat dikatakan bahwa, cara yang paling realistis untuk memperoleh basis pengetahuan adalah dengan secara langsung mendapatkannya dari pakar manusia. Namun, terdapat empat alasan mengapa hal tersebut diatas dapat tidak bekerja atau mungkin tidak memberikan hasil yang memuaskan:

- Untuk beberapa persoalan, orang yang kita anggap sebagai pakar ternyata "bukan" pakar yang sebenarnya (dipilih karena atas dasar hanya mampu menyelesaikan suatu persoalan)
- Pakar yang dimaksud diduga keras hanya memiliki kinerja yang cukup bahkan mungkin jelek. Seringkali, seorang pakar hanya dinilai atas keberhasilannya dalam melakukan tugasnya.
- Pakar mungkin tidak bersedia membagi pengetahuan mereka. Dalam beberapa kasus, individu semacam ini menolak untuk bekerja sama sehingga mungkin timbul permasalahan serius - mereka nampak bekerja sama tetapi secara sengaja memberikan informasi yang salah.

- Akhirnya, ada beberapa pakar yang tidak mampu merumuskan pendekatan yang mereka lakukan. Banyak pakar, pada kenyataannya tidak mengerti bagaimana seharusnya ia harus membuat keputusan. Apa yang diuraikannya ternyata hanya merupakan "persepsi" dirinya dan bukan prosedur yang seharusnya dilakukan.

2.7.2. PEMILIHAN DOMAIN

Pemilihan domain adalah hal yang penting dalam pengembangan sistem pakar, kekurangjelasan dalam memilih domain dapat mengakibatkan sistem pakar tidak terfokus yang berakibat melebar, sehingga menyulitkan dalam pengembangan lebih lanjut. Kesalahan dalam memilih domain juga dapat berakibat sistem pakar yang dibangun tidak bermanfaat. Dalam aktifitas pemilihan domain, perencana pengetahuan harus memperhatikan beberapa hal berikut ini:

- Pemilihan domain harus disesuaikan dengan lingkup sistem pakar yang akan dibangun, dan juga memperhatikan kenyataan bahwa sistem pakar yang akan dibangun akan dapat memberikan kelebihan tersendiri dibandingkan dengan alternatif lainnya.

- Keputusan pemilihan domain seharusnya didukung oleh manajemen, bahwa manajemen mempunyai kemauan untuk menyediakan waktu dan sumber daya yang dibutuhkan untuk pengembangan dan implementasi dari sistem pakar.
- Manajemen harus melakukan analisa biaya dan resiko dari pengembangan sistem pakar, Dalam konteks ini, sistem pakar perlu untuk mempekerjakan perekayasa pengetahuan dalam suatu periode waktu tertentu.
- Domain yang dipilih harus relatif stabil dan diharapkan tidak banyak berubah selama periode pengembangan sistem pakar.

2.7.3. PEMILIHAN PEREKAYASA PENGETAHUAN

Perekayasa pengetahuan merupakan bagian penting dalam mengembangkan sistem pakar. Melalui perekayasa pengetahuan-lah seorang pakar berkomunikasi dengan sistem yang akan dibangun. Berikut ini adalah beberapa persyaratan yang perlu dimiliki oleh seorang perekayasa pengetahuan:

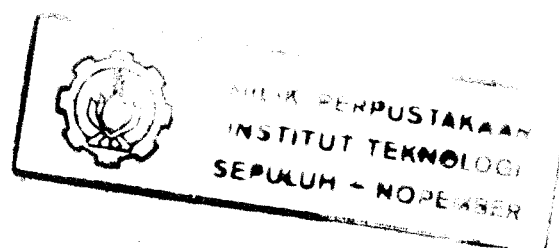
- Idealnya, dibutuhkan 2 orang perekayasa pengetahuan, dimana salah satunya telah berpengalaman dalam membangun dan mengimplementasikan suatu sistem pakar.

- Perekayasa pengetahuan yang dipilih harus mampu untuk memberikan teknik-teknik alternatif dalam melakukan analisa keputusan.
- Skill yang dibutuhkan dari perekayasa pengetahuan utamanya yang berkaitan dengan elisitas pengetahuan dan pembentukan model dari pengetahuan yang diperoleh (yaitu , basis aturan)

2.7.4. PEMILIHAN PAKAR

Sistem pakar yang paling efektif ialah sistem yang digali dari pakar sendiri. Namun kesalahan dalam pemilihan pakar dapat berakibat fatal atau bahkan tidak terbangunnya sistem pakar itu sendiri. Karena itu diperlukan beberapa kriteria dalam memilih pakar, berikut ini adalah kriteria pemilihan pakar:

- Kandidat pakar dari domain dipilih dari mereka yang diyakini mempunyai "kepakaran" yang signifikan dalam domain tersebut.
- Pilih pakar yang secara umum performansinya diakui berada "diatas" rata-rata para pakar lain dalam bidang yang sama.
- Pilih seorang pakar yang mempunyai "track record" keberhasilan yang tinggi.
- Pilih pakar yang sekiranya akan mampu melakukan komunikasi personal yang baik.



- Pilih pakar yang mempunyai kemampuan untuk berkonsentrasi penuh untuk meluangkan waktunya guna mendukung upaya pengembangan.
- Jika tidak ada pakar yang sesuai, gunakan cara alternatif pengembangan sistem berbasis aturan.

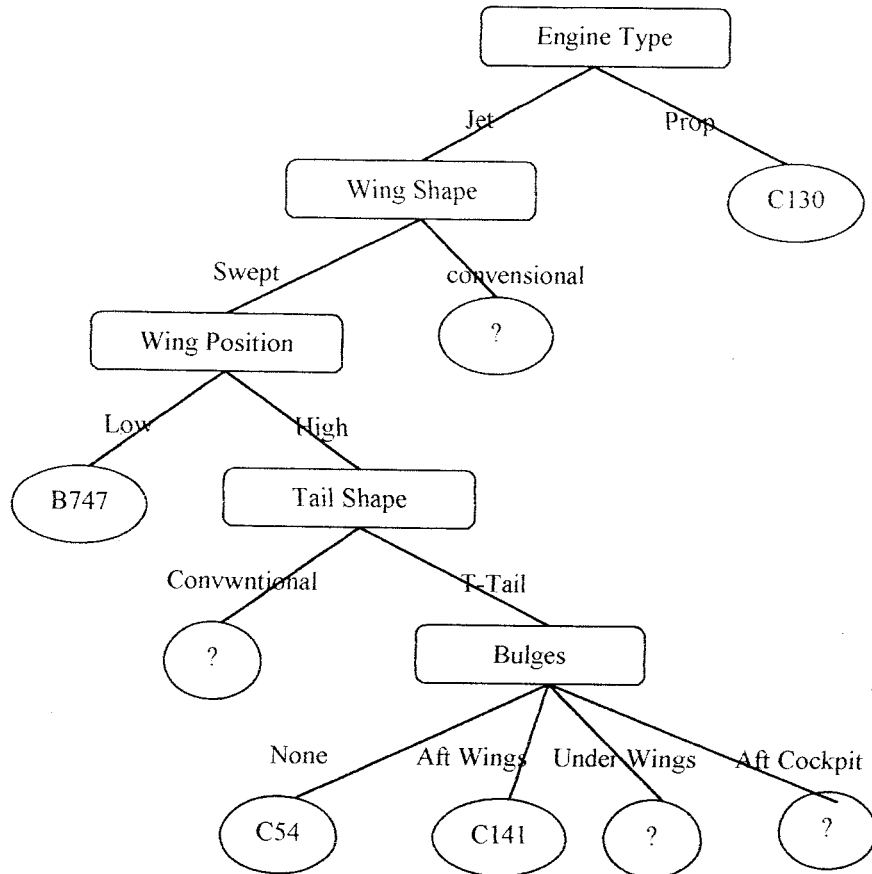
2.7.5. PENGEMBANGAN SEBUAH POHON KEPUTUSAN

Seperti dibahas sebelumnya tentang inference network. Yaitu network yang dipergunakan untuk mengilustrasikan basis aturan dan proses inferensi. Disisi lain network lain yang lebih sederhana yaitu *pohon keputusan*. Yang juga merupakan representasi alternatif dari basis aturan. Network semacam itu kemungkinan merupakan cara terbaik untuk menjelaskan dalam arti sebagai contoh. Contoh pohon keputusan untuk data berikut:

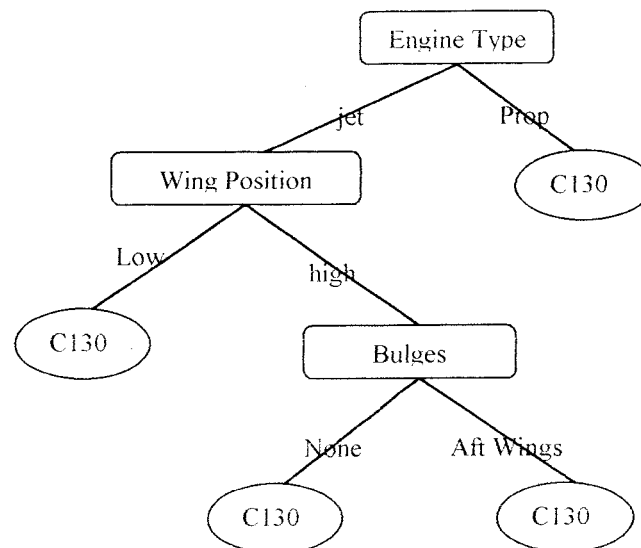
Tabel 2.1
atribut pesawat

Attribut	Tipe Pesawat			
	C130	C141	C5A	B747
Tipe mesin	Prop	Jet	Jet	Jet
Posisi sayap	High	High	High	Low
Whing shape	Conventional	Swept-back	Swept-back	Swept-back
Tail	Conventional	T-tail	T-tail	Conventional
Bulges	Underwings	Aft wings	None	Aft wings

Untuk membangun pohon keputusan yang lebih sederhana dengan menghapus node dan menghilangkan beberapa cabang yang mengarah kepada konklusi yang tidak mungkin.



Gambar 2.3 pohon keputusan yang mungkin dari identifikasi pesawat



Gambar 2.4 tree lebih sederhana

Langkah-langkah yang perlu dilakukan untuk mengenerate pohon keputusan menjadi production rule adalah sebagai berikut:

1. Definisi, sebuah rantai didefinisikan sebagai hubungan dari satu node dalam pohon ke node lain kita sebut sebagai cabang dan hanya memiliki satu arah
2. Langkah satu, mengidentifikasi semua node konklusi yang belum dikaitkan.
3. Langkah dua, melacak rantai dari konklusi node secara backward melalui pohon menuju ke root node.
4. Langkah tiga, rantai seperti yang diidentifikasi pada langkah dua, node yang berbentuk lingkaran adalah sebagai THEN node dan kotak tumpul adalah sebagai IF node.
5. Langkah empat, bangun production rule yang berkaitan dengan pohon keputusan sebagai pertimbangan, ulangi proses ini untuk setiap konklusi node.

2.8. INFERENCE ENGINE

Inference engine bertugas sebagai inference dan mekanisme kontrol bagi sistem pakar dan sebagai semacam bagian yang penting bagi sistem pakar sepenting faktor utama dalam determentasi dari keefektifan dan efisiensi dari sistem semacam ini. Inference adalah proses penggambaran kesimpulan (baik intermediate maupun final) sebagai set dari aturan, sebagai spesifikasi set dari fakta, untuk suatu situasi yang diberikan. Inference adalah sebagai elemen dari pemroses pengetahuan dalam sebuah sistem pakar.

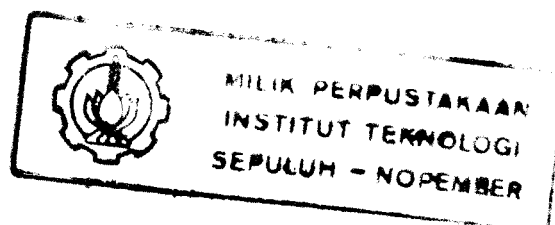
Inference strategi yang paling umum digunakan dalam sistem pakar dikenal sebagai modus ponens. Secara sederhana, modus ponent berarti bahwa jika premis dari sebuah aturan adalah benar maka kesimpulannya juga benar.

Tanggung jawab kontrol dari inference engine adalah segala hal yang berkaitan dengan hal sebagai berikut:

- Bagaimana memulai proses inference
- Aturan yang mana yang ditembakkan jika lebih dari satu yang di trigger.
- Cara yang ditempuh untuk mencari solusi

Seperti basis pengetahuan, inference engine berisi aturan dan fakta. Bagaimanapun aturan dan fakta dari basis pengetahuan berisi domain yang spesifik dari kepakaran ketika aturan dan fakta dari inference engine berkaitan pada kontrol yang lebih umum dan strategi pencarian yang digunakan oleh sistem pakar dalam membangun sebuah solusi. Kedua fakta dan aturan ini secara sengaja dibuat terpisah pada sistem pakar tertentu. Seperti yang dijelaskan sebelumnya, kunci dari sistem pakar adalah penggunaan deferensiasi dari pemrograman heuristic.

Pemisahan ini menghasilkan beberapa manfaat. Pertama, hal itu membolehkan seseorang untuk melakukan perubahan pada basis pengetahuan dengan efek minimum pada inference engine, dan sebaliknya. Kedua, hal itu menyediakan untuk pengembangan dan digunakan untuk shell sistem pakar. Shell sistem pakar, seperti yang didiskusikan, berisi semua komponen yang diperlukan dari sebuah sistem pakar kecuali basis pengetahuan. Seharusnya strategi inferensi dari shell semacam itu sesuai dengan yang dibutuhkan oleh basis pengetahuan, kemudian shell digunakan untuk mengakomodasi basis pengetahuan. Pendekatan ini menghasilkan sebuah tipe plug-in arsitektur dimana kita memasukkan basis pengetahuan tertentu dan menjalankannya melalui host shell sistem pakar. Karenanya, kita bisa menghindari menghabiskan banyak waktu dan usaha untuk membangun sistem pakar kita sendiri dari dasar. Memberikan kapabilitas dari sistem pakar shell yang ada, sehingga pada faktanya, sedikit dibutuhkan bagi perəkayasa



pengetahuan untuk berurusan dengan pembangunan sebuah inference engine - atau bagian manapun dari sistem pakar selain dari basis pengetahuan. Bagaimanapun seorang perancang pengetahuan harus mengerti dasar dari proses inferensi.

BAB III

Perencanaan Produksi Jagung dalam Industri Pertanian

Jagung (*Zea mays L.*) merupakan kebutuhan yang cukup penting bagi kehidupan manusia dan hewan. Sangat cocok sebagai makanan pokok kandungan gizi dan seratnya yang cukup memadai.

Kebutuhan jagung yang akan terus meningkat yang ditunjang dengan semakin luasnya industri yang memanfaatkan jagung sebagai bahan bakunya. Jagung dapat diolah menjadi berbagai bentuk olahan diantaranya adalah minyak goreng, tepung maizena, ethanol, asam organik, gritz, makanan kecil oleh industri makanan dan sebagai bahan dasar industri pakan ternak.

3.1. PERMINTAAN DAN PRODUKSI

Jumlah penduduk Indonesia yang terus meningkat dan kebutuhan bahan baku jagung untuk pakan ternak yang terus meningkat tidak dapat diimbangi oleh meningkatnya produksi jagung dalam negeri. Dalam menutupi kekurangan ini dilakukan impor jagung dari beberapa negara. Sehingga bisa dikatakan bahwa prospek tanaman jagung sangat cerah. Permintaan jagung untuk konsumsi yang terus meningkat dari tahun ketahun dapat dilihat di tabel 3.1. Namun permintaan yang tinggi tersebut tidak diimbangi dengan

produksi yang memadai. Contohnya saja Jawa Timur yang merupakan produsen jagung terbesar se-Indonesia, untuk memenuhi kebutuhan Jawa Timur sendiri saja masih kurang, berdasarkan data BPS Jawa Timur yang dapat dilihat pada tabel 3.2.

Sehingga dari data tersebut dapat dikatakan bahwa permintaan jagung sangat besar sedangkan produksinya rendah.

Tabel 3.1.

Jumlah penduduk, konsumsi per kapita dan tingkat konsumsi jagung di Indonesia tahun 1986 - 1996 [AGR,99]

Tahun	Jumlah penduduk (x1000)	(kg/th)	Konsumsi (ton)
1986	166.489	29,25	4.869.803
1987	169.650	24,71	4.196.994
1988	173.799	30,72	5.339.105
1989	174.730	26,81	4.684.511
1990	178.170	29,68	5.288.086
1991	181.256	28,73	5.207.485
1992	184.491	34,13	6.296.678
1993	187.589	28,86	5.413.819
1994	190.676	32,06	6.113.073
1995	193.750	34,16	6.618.500

Tabel 3.2.

Jumlah produksi dan tingkat konsumsi jagung di Jawa

Tahun Bulan	produksi		Import (ton)	Persediaan (ton)	Export (ton)	Pemakaian (ton)	Pemakaian di propinsi					
	In	Out					Makanan ternak	Untuk bibit	Diolah		tercecer	Tersedia dikonsumsi
									eat	industri		
Jul-Des 92	0	3.023.344	2.241	3.025.585	30.687	2.994.898	268.921	39.121	0	2	149.749	2.537.109
Jan-Jun 93	0	2.363.252	19.242	2.382.494	33.510	2.348.984	199.488	30.362	0	0	117.449	2.001.685
Jul-Des 93	0	2.363.252	128.105	2.491.357	27.565	2.463.792	147.828	25.946	0	0	123.190	2.166.829
Jan-Jun 94	0	2.636.015	46.458	2.682.473	32.024	2.650.449	159.027	34.546	0	0	132.522	2.324.354
Jun-Des 94	0	2.636.015	367.227	3.003.242	18.304	2.984.938	481.611	28.652	0	0	149.247	2.325.428
Jan-Jun 95	0	2.820.868	206.890	3.027.758	18.740	3.009.018	314.720	30.220	0	0	150.451	2.513.827
Jun-Des 95	0	2.820.868	206.116	3.026.984	18.455	3.008.529	592.878	35.991	0	0	150.426	2.229.234
Jan-Jun 96	0	3.417.486	141.381	3.558.870	18.949	3.539.921	679.718	35.568	0	0	176.996	2.647.639
Jun-Des 96	0	3.417.489	141.381	3.558.870	18.949	3.539.921	212.395	35.525	0	619.440	176.996	2.495.565
Jan-Jun 97	0	3.034.805	454.647	3.489.452	13.522	3.475.930	208.556	31.600	0	460.439	173.797	2.601.538

3.2. PERENCANAAN PRODUKSI

Perencanaan produksi adalah menentukan akan dilakukan produksi atau tidak, orientasi apakah yang ingin ditempuh dan menentukan apa yang perlu dilakukan ketika aktifitas produksi dilakukan. Keunikan yang dimiliki industri pertanian untuk komoditi jagung, yaitu sumberdaya yang dipergunakan bukanlah milik industri sendiri, namun sebagian besar merupakan pertanian rakyat yang dikerjasamakan dengan industri pertanian.

Sehingga untuk melakukan aktifitas produksi perlu mempertimbangkan banyak hal yaitu dari sisi ekonomi, dari sisi alam, dari sisi teknologi, tingkat resiko, dan target produksi yang ingin dicapai. Masudnya dalam memproduksi harus dipertimbangkan hal sebagai berikut:

1. **ekonomis**, dipastikan bahwa dari hasil analisa kelayakkan komoditas jagung adalah memuaskan, untuk mengetahui kelayakkan agribisnis jagung, maka perlu dilakukan analisa finansial. Dari hasil analisa tersebut diketahui bahwa agribisnis jagung layak untuk dikembangkan. Hal ini ditunjukkan dengan nilai IRR sebesar 32.32% NPV(DF24%) sebesar 956.538 (rupiah) dan R/C sebesar 1.10. [AGR.99]
2. **alam**, menyesuaikan lokasi dengan kondisi alam yang tepat yaitu curah hujan yang tepat, jenis tanah yang tepat, ketinggian yang tepat.

3. **teknologi**, karena pengusahaan industri pertanian jagung tidak mungkin diusahakan sendiri oleh pengusaha, dan pada umumnya dalam industri pertanian dilakukan kerjasama dengan pertanian rakyat. Sehingga sangat penting untuk mengetahui seberapa jauh masyarakat di daerah pertanian tersebut mengenal pengusahaan pertanian dan prakteknya.

4. **resiko**, Mengembangkan sektor pertanian yang berpotensi dan yang mempunyai keunggulan komparatif adalah tidak mudah karena adanya banyak kenyataan pengembangan sektor pertanian dihadapkan pada masalah resiko (risk) dan ketidakpastian (uncertainty). Masalah iklim seperti musim kemarau panjang; hujan yang tidak menentu; masalah serangan hama dan penyakit tanaman yang sulit diduga sebelumnya; masalah bencana alam banjir; gempa dan gunung berapi; masalah kekurangan air irigasi atau air hujan atau masalah yang lain adalah contoh betapa kehidupan tanaman itu sebenarnya tunduk pada aspek resiko dan ketidakpastian.

Dikatakan resiko (risk) bila kita mengetahui betapa besarnya peluang terjadinya resiko tersebut. Bila tahun depan dikatakan akan ada musim kemarau panjang ; sehingga taksiran produksi menurun 30%; maka secara tidak langsung peluang terhadap besarnya resiko adalah 30 % atau 0.3. jadi peluang terjadinya resiko adalah 0.3 yang dapat diartikan

bahwa bila petani akan menanam padi ; maka mereka mengetahui dan sadar kalau produksi yang akan diperoleh adalah akan berkurang sebesar 30%.

Sebaliknya dikatakan ketidakpastian (uncertainty) bila peluang terjadinya resiko tersebut tidak diketahui; sehingga petani atau produsen bertindak gambling (judi). Bila dilakukan penanaman, mereka juga menyadari adanya ketidakpastian yang tinggi sehingga mereka sadar akan terjadinya kemungkinan yang terjelek misalnya: tidak terjadi hasil panen sekalipun.[SOE-93]

Secara umum pengusaha tidak akan mengambil ketidakpastian, maka disini resiko yang berkaitan dengan topoklimatologi kami atasi terlebih dahulu dengan cara memilih dan menggunakan lokasi yang secara topoklimatologi mendukung sehingga resiko disini dikategorikan dua macam yaitu pasar dan gangguan. Resiko pasar yaitu tingkat kestabilan harga. Resiko gangguan yaitu resiko yang ditimbulkan akibat bencana alam dan hama penyakit.

5. **Target produksi**, yaitu dalam melakukan aktifitas produksi tersebut mana yang hendak diprioritaskan masimum produksi atau maksimum profit. Hal ini terutama berakaitan dengan keterbatasan sumber daya yang dimiliki.

Maksimum profit konsep ini dikembangkan di Barat khususnya setelah munculnya doktrin laba yang diperkenalkan oleh Adam Smith. Saat itu asumsi full-employment memang dapat diberlakukan karena kondisi pertanian dan pasar saat itu memang mendukung. Kemudian setelah revolusi industri di Eropa; dimana peran sektor pertanian sudah semakin berkurang dan digantikan oleh sektor industri maka asumsi full-employment menjadi sulit dipenuhi. Akibat muncul aliran neo-klasik yang ingin memperbaiki asumsi tersebut.

Konsep maksimum profit ini muncul pada usaha tani yang komersial dimana prinsip-prinsip ekonomi sudah diterapkan. Besar kecilnya keuntungan menjadi ukuran dalam mengambil keputusan dan karenanya suatu keputusan diambil atau tidak adalah sangat tergantung pada besar kecilnya keuntungan yang akan diperoleh atau yang dijanjikan oleh komoditas pertanian yang akan diusahakan tersebut. Karena ukuran keputusan ini terletak pada besar kecilnya keuntungan, maka produsen selalu menghitung besar kecilnya cash-flow (angsuran arus uang tunai)

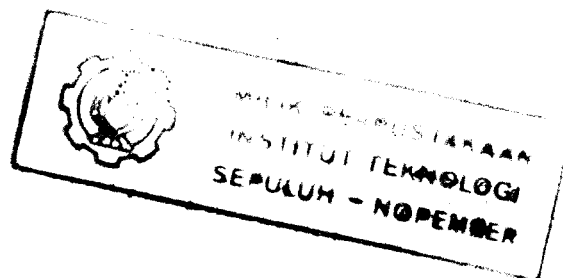
Andaikan:

$$\pi = P_y \cdot Y - \sum_{i=1}^n P_x \cdot X$$

dimana:

π = keuntungan;

P_x = harga output X



P_y = harga output Y

X = macam input;

N = banyaknya input yang dipakai

Maka π akan semakin tinggi bila:

- a. Output tinggi dan P_y tinggi dengan asumsi P_x dan X konstan
atau
- b. P_x dan X rendah dengan asumsi P_y dan Y konstan

Output yang tinggi akan membentuk total penerimaan yang tinggi. Jadi agar keuntungan menjadi tinggi maka perlu diupayakan tindakan yang menyebabkan output menjadi tinggi. Ini berarti perlunya efisiensi teknis karena efisiensi teknis pada dasarnya adalah bagaimana membuat output menjadi setinggi mungkin. Cara lain disamping perlu adanya efisiensi teknis, maka diperlukan efisiensi alokatif atau efisiensi harga; yaitu efisiensi yang dicapai dengan mengkondisikan nilai produk marginal sama dengan harga. Jadi, bila dikehendaki efisiensi alokatif dari penggunaan tenaga kerja

Maksimum produksi, konsep ini didasarkan pada pengertian bahwa dasar pengambilan keputusan adalah ditentukan oleh judgement si pengambil keputusan. Judgement ini tentu saja sudah

mempertimbangkan sampai seberapa besar resiko yang akan diterima jika keputusan tersebut gagal memenuhi kehendak dari si pengambil keputusan. Dengan demikian pendekatan dengan teori pengambilan keputusan (decision theory) dianggap lebih baik dan lebih realistis dibandingkan dengan pendekatan statistik. Konsep ini sering dikembangkan oleh mereka yang tidak banyak pengalaman, sehingga judgement yang dipakai dasar pengambilan keputusan menjadi dasar utama pertimbangan yang dipakai.

Disini dalam menentukan apakah sesuatu produksi itu maksimum produksi atau maksimum profit dengan cara menentukan suatu lokasi apabila lokasi tersebut membutuhkan mengeluarkan modal yang lebih seperti misalnya perlu pengolahan tanah; perlu irigasi; perlu pengelolaan hama; perlu pelatihan; dan lain sebagainya yang mengeluarkan uang. Pakar lebih merekomendasikan untuk berorientasi maksimum profit sedang selain dari diatas sebagai maksimum produksi.

3.3. PROSES YANG DITEMPUH DALAM PENGAMBILAN KEPUTUSAN

Dalam pengambilan keputusan perlu diambil langkah-langkah yang tepat, agar diperoleh keputusan yang mampu menyelesaikan masalah yang

dihadapi. Langkah-langkah yang ditempuh dalam pengambilan keputusan adalah sebagai berikut:

1. mengklasifikasikan persoalan yaitu dengan cara mengklasifikasikan kondisi yang ada termasuk dalam kategori yang mana lalu dibandingkan dengan kondisi yang harus dicapai.
2. Memberikan penyelesaian yang perlu diambil untuk menyesuaikan kondisi dengan standart yang dimiliki
3. Menentukan solusi alternatif
4. Memilih dari beberapa solusi alternatif
5. Menentukan orientasi yang ingin dicapai apakah maksimum profit atau maksimum produksi

BAB IV

DESAIN DAN PEMBUATAN PERANGKAT LUNAK

Dalam bab ini dibahas tentang perancangan perangkat lunak dari Sistem Perencanaan produksi Berbasis Aturan ini, meliputi perancangan komponen sistem, perancangan data, dan perancangan proses. Selain itu juga dibahas tentang implementasi dalam programnya.

4.1. PERANCANGAN KOMPONEN-KOMPONEN SISTEM

Sebuah Sistem Berbasis Aturan memiliki komponen-komponen sistem yang harus ada agar memenuhi syarat sebagai sebuah Sistem Berbasis Aturan. Pada Tugas Akhir ini, Sistem Perencanaan Produksi Jagung dibangun dengan komponen-komponen sebagai berikut:

4.1.1 BASIS PENGETAHUAN SISTEM PERENCANAAN PRODUKSI JAGUNG

Basis pengetahuan sistem Perencanaan Produksi Jagung diambil dari pengetahuan seorang pakar yaitu Prof. Dr. Ir. Soemarno, MS. Kepala PUSLIT Pembangunan Wilayah Pedesaan Unibraw Malang.

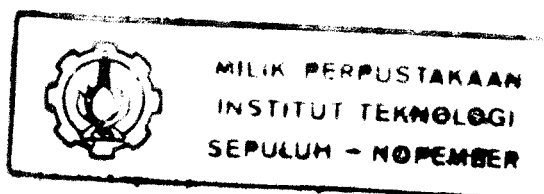
Berdasarkan Proses pengambilan Keputusan yang dilakukan oleh pakar maka dapat dibuat pohon keputusan, yang nantinya digunakan adalah sebagai dasar untuk mengambil keputusan.

Perencanaan produksi adalah menentukan akan dilakukan produksi atau tidak, orientasi apakah yang ingin ditempuh dan penentuan apa yang perlu dilakukan ketika aktifitas produksi dilakukan. Perencanaan dilakukan berdasarkan kondisi yang dihadapi .

4.1.1.1 Asumsi faktor perencanaan

Hal-hal yang dipertimbangkan dalam perencanaan produksi dalam industri pertanian disini dalam basis pengetahuan adalah sebagai berikut:

1. **ekonomis**, dari pembahasan diatas terlihat bahwa jagung berprospek sangat besar. Terlihat dengan semakin banyaknya macam dan jenis industri yang memanfaatkan jagung mulai dari industri makanan sampai pakan ternak. Serta tingginya permintaan dunia terhadap jagung memberikan peluang eskport bagi Indonesia bila terjadi kelebihan produksi dari dalam negeri (sangat kecil kemungkinannya). Jadi dalam basis pengetahuan yang kami bangun diasumsikan permintaan jagung sangat besar.
2. **alam**, kondisi alam yang berkaitan dengan curah hujan, jenis tanah, ketinggian dikelompokkan berdasarkan tingkat kesesuaian untuk mengusahakan jagung biji serta usaha yang dilakukan untuk



memperbaiki kondisi tersebut. Dalam basis pengetahuan ini dikelompokkan menjadi 5 yaitu; sesuai, sesuai dengan pengolahan tanah intensif, untuk produksi batang dan daun (hijauan), diperbaiki dengan irigasi, tidak sesuai. Untuk selanjutnya kondisi tidak sesuai tidak digunakan karena tidak ada pengusaha yang berani mengusahakan suatu usaha pertanian tanaman tertentu padahal sudah jelas kondisi topoklimatologinya tidak sesuai untuk diusahakan tanaman tersebut.

3. **teknologi**, disini diasumsikan pengusaha akan melakukan kerjasama dengan masyarakat dalam mengusahakan jagung yaitu lahan pertanian dan tenaga kerja adalah milik petani. Jika kondisi topoklimatologi telah sesuai maka perlu diketahui tingkat pengetahuan dan penerapan teknologi perusahaan jagung yang dimiliki oleh masyarakat sekitar daerah yang akan dikembangkan. Kondisi teknologi ini dikelompokkan menjadi tiga yaitu teknologi tingkat rendah, teknologi tingkat menengah, teknologi tingkat tinggi.
4. **Resiko**, Secara umum pengusaha tidak akan mengambil ketidakpastian, maka disini resiko yang berkaitan dengan topoklimatologi kami atasi terlebih dahulu dengan cara memilih dan menggunakan lokasi yang secara topoklimatologi mendukung sehingga resiko disini dikategorikan dua macam yaitu pasar dan gangguan. Resiko pasar yaitu tingkat kestabilan harga. Resiko gangguan yaitu resiko yang ditimbulkan akibat bencana alam dan hama penyakit. Resiko disini kami kelompokkan

menjadi 13 yaitu; manual, sedang, intensif, penjadwalan manual, penjadwalan sedang, penjadwalan intensif, scoring (1) manual, scoring (2) sedang, scoring (3) manual, scoring (4) sedang, scoring (5) intensif, scoring (6) intensif, resiko bencana alam tinggi. Untuk selanjutnya resiko bencana alam tinggi tidak digunakan karena tidak ada pengusaha yang mau berusaha di daerah yang sering mengalami bencana alam.

5. **Target produksi**, yaitu dalam melakukan aktifitas produksi tersebut mana yang hendak diprioritaskan maksimum produksi atau maksimum profit. Hal ini terutama berkaitan dengan keterbatasan sumber daya yang dimiliki. Disini dalam menentukan apakah sesuatu produksi itu maksimum produksi atau maksimum profit dengan cara menentukan suatu lokasi apabila lokasi tersebut membutuhkan mengeluarkan modal yang lebih seperti misalnya perlu pengolahan tanah; perlu irigasi; perlu pengelolaan hama; perlu pelatihan; dan lain sebagainya yang mengeluarkan uang. Pakar lebih merekomendasikan untuk berorientasi maksimum profit sedang selain dari diatas sebagai maksimum produksi.

4.1.1.2 Identifikasi Atribut dan value

Berdasarkan Proses pengambilan Keputusan yang dilakukan oleh pakar yang tercantum dalam dokumentasi meeting pada bagian lampiran, maka dapat diketahui macam-macam atribut dan value yang digunakan

dalam sistem berbasis aturan ini yang dapat dilihat pada tabel 4.1, tabel 4.2 dan tabel 4.3. Yang akan digunakan dalam membangun pohon keputusan, yang nantinya digunakan sebagai dasar untuk mengambil keputusan.

Tabel 4.1.
Atribut dan value yang digunakan dalam Tipe Topoklimatologi

Atribut	Value
Curah Hujan	< 1000 mm/th >3500 mm/th antara 1000 mm/th dan 1500 mm/th antara 1500 mm/th dan 2500 mm/th antara 2500 mm/th dan 3500 mm/th
Jenis Tanah	Liat (halus) Pasir (kasar) Lempung (medium) Lempung-berdebu Lempung berpasir
Ketinggian Tempat	Antara 0 m dan 1500 m > 1500 m

Tabel 4.2.
Atribut dan value yang digunakan dalam Tingkat Resiko

Atribut	Value
Bencana Alam	Sering Sedang Jarang
Gangguan Hama Penyakit	Tinggi Sedang Rendah
Tingkat Harga	Stabil Terjadi perubahan kecil Fluktuasi harga yang tinggi

Tabel 4.3
Atribut dan value yang dipergunakan dalam teknologi

Atribut	Value
Pengetahuan Teknologi	Diketahui Tidak diketahui
Penerapan teknologi	Dipraktekkan Tidak dipraktekkan

Tabel. 4.4.
Listing Atribut Tipe Topoklimatologi

	Atribut →	Curah hujan	Jenis tanah	Ketinggian
Tipe Topoklimatologi	Tidak sesuai diusahakan jagung	<1000 mm/th atau >3500 mm/th	Liat(halus) ^a atau pasir (kasar) ^b	>1500
	Tidak sesuai diusahakan jagung	<1000 mm/th atau >3500 mm/th	Liat(halus) atau pasir (kasar)	0 - 1500
	Tidak sesuai diusahakan jagung	<1000 mm/th atau >3500 mm/th	Lempung (medium) ^c atau lempung-berdebu ^d	>1500
	Diperbaiki dengan irigasi	<1000 mm/th atau >3500 mm/th	Lempung (medium) atau lempung-berdebu	0 - 1500
	Tidak sesuai diusahakan jagung ^e	<1000 mm/th atau >3500 mm/th	Lempung berpasir ^e	>1500
	Diperbaiki dengan irigasi ⁴	<1000 mm/th atau >3500 mm/th	Lempung berpasir	0 - 1500
	Untuk produksi batang dan daun (hijauan)	1000-1500 atau 2500-3500	Liat(halus) atau pasir (kasar)	>1500
	Sesuai dengan pengolahan tanah intensif	1000-1500 atau 2500-3500	Liat(halus) atau pasir (kasar)	0 - 1500
	Untuk produksi batang dan daun (hijauan) sesuai	1000-1500 atau 2500-3500	Lempung (medium) atau lempung-berdebu	>1500
	Sesuai	1000-1500 atau 2500-3500	Lempung (medium) atau lempung-berdebu	0 - 1500
	Untuk produksi batang dan daun (hijauan) sesuai	1000-1500 atau 2500-3500	Lempung berpasir	>1500
	Sesuai	1000-1500 atau 2500-3500	Lempung berpasir	0 - 1500
	Untuk produksi batang dan daun (hijauan)	1500 - 2500 mm/th	Liat(halus) atau pasir (kasar)	>1500
	Sesuai dengan pengolahan tanah intensif ²	1500 - 2500 mm/th	Liat(halus) atau pasir (kasar)	0 - 1500
	Untuk produksi batang dan daun (hijauan) sesuai ¹	1500 - 2500 mm/th	Lempung (medium) atau lempung-berdebu	>1500
	Sesuai	1500 - 2500 mm/th	Lempung (medium) atau lempung-berdebu	0 - 1500
	Untuk produksi batang dan daun (hijauan) ³	1500 - 2500 mm/th	Lempung berpasir	>1500
	Sesuai	1500 - 2500 mm/th	Lempung berpasir	0 - 1500

Tabel. 4.5.
Listing Atribut Tingkat Resiko

Tingkat Resiko	Atribut →	Tingkat harga	Gangguan hama penyakit	Bencana alam
	Perlakuan = manual	Stabil	Rendah	jarang
	Perlakuan = sedang	Stabil	Sedang	jarang
	Perlakuan = manual	Stabil	Rendah	sedang
	Perlakuan = sedang	Stabil	Sedang	sedang
	Perlakuan = intensif	Stabil	Tinggi	jarang
	Perlakuan = intensif	Stabil	Tinggi	sedang
	Gangguan bencana= tinggi	Stabil	Sedang	sering
	Gangguan bencana= tinggi	Stabil	rendah	sering
	Gangguan bencana= tinggi	Stabil	tinggi	sering
	Perlakuan = penjadwalan, mahual	Terjadi perubahan kecil	Rendah	jarang
	Perlakuan = penjadwalan, sedang	Terjadi perubahan kecil	Sedang	jarang
	Perlakuan = penjadwalan, manual	Terjadi perubahan kecil	Rendah	sedang
	Perlakuan = penjadwalan, sedang	Terjadi perubahan kecil	Sedang	sedang
	Perlakuan = penjadwalan, intensif	Terjadi perubahan kecil	Tinggi	jarang
	Perlakuan = penjadwalan, intensif	Terjadi perubahan kecil	Tinggi	sedang
	Gangguan bencana= tinggi	Terjadi perubahan kecil	Sedang	sering
	Gangguan bencana= tinggi	Terjadi perubahan kecil	rendah	sering
	Gangguan bencana= tinggi	Terjadi perubahan kecil	tinggi	sering
	Perlakuan = scoring (1), manual	Fluktuasi harga yang tinggi	Rendah	jarang
Perlakuan = scoring (2), sedang	Fluktuasi harga yang tinggi	Sedang	jarang	
Perlakuan = scoring (3), manual	Fluktuasi harga yang tinggi	Rendah	sedang	
Perlakuan = scoring (4), sedang	Fluktuasi harga yang tinggi	Sedang	sedang	
Perlakuan = scoring (5), intensif	Fluktuasi harga yang tinggi	Tinggi	jarang	
Perlakuan = scoring (6), intensif	Fluktuasi harga yang tinggi	Tinggi	sedang	
Gangguan bencana= tinggi	Fluktuasi harga yang tinggi	Sedang	sering	
Gangguan bencana= tinggi	Fluktuasi harga yang tinggi	rendah	sering	
Gangguan bencana= tinggi	Fluktuasi harga yang tinggi	tinggi	sering	

Tabel. 4.6.
Listing Atribut Tipe Teknologi

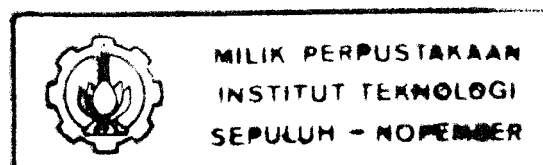
Atribut ↓	Tipe teknologi			
	Tingkat rendah	Tingkat sedang		Tingkat tinggi
Penerapan teknologi	Tidak dipraktekkan	Tidak dipraktekkan	dipraktekkan	dipraktekkan
Pengetahuan teknologi	Tidak diketahui	diketahui	Tidak diketahui	Tidak diketahui

4.1.1.3 Pembentukan Pohon Keputusan

Dengan menggunakan tabel listing atribut diatas akan dibangun pohon keputusan, dan kemudian menyederhanakannya, untuk digunakan membuat production rule. Pohon keputusan untuk menentukan tipe Topoklimatologi yang dihasilkan adalah seperti terlihat pada gambar 4.1 . Pohon Keputusan tersebut dibentuk dengan menggunakan tabel 4.4. Pada pohon keputusan tersebut terlihat bahwa tanda lingkaran menunjukkan konklusi node, sedangkan persegi panjang tumpul menunjukkan atribut dan batangnya menunjukkan value-nya.

Misal pada node paling kiri bernomor 5 jika dilihat pada tabel 4.4 menyatakan topoklimatologi tidak sesuai diusahakan jagung memiliki :

- atribut jenis tanah dengan value a (lihat pada tabel 4.4) menunjukkan value liat (halus)
- atribut curah hujan dengan value < 1000 mm/th
- atribut ketinggian tempat dengan value > 1500 m



Berdasarkan tabel 4.4

- Konklusi node (1) menunjukkan tipe topoklimatologi sesuai untuk ditanami jagung biji
- Konklusi node (2) menunjukkan tipe topoklimatologi sesuai dengan pengolahan tanah intensif.
- Konklusi node (3) menunjukkan tipe topoklimatologi untuk produksi batang dan daun (hijauan)
- Konklusi node (4) menunjukkan tipe topoklimatologi diperbaiki dengan irigasi
- Konklusi node (5) menunjukkan tipe topoklimatologi tidak sesuai diusahakan jagung jenis apapun
- Value atribut jenis tanah (a) menunjukkan liat (halus)
- Value atribut jenis tanah (b) menunjukkan pasir (kasar)
- Value atribut jenis tanah (c) menunjukkan lempung (medium)
- Value atribut jenis tanah (d) menunjukkan lempung-berdebu
- Value atribut jenis tanah (e) menunjukkan lempung berpasir

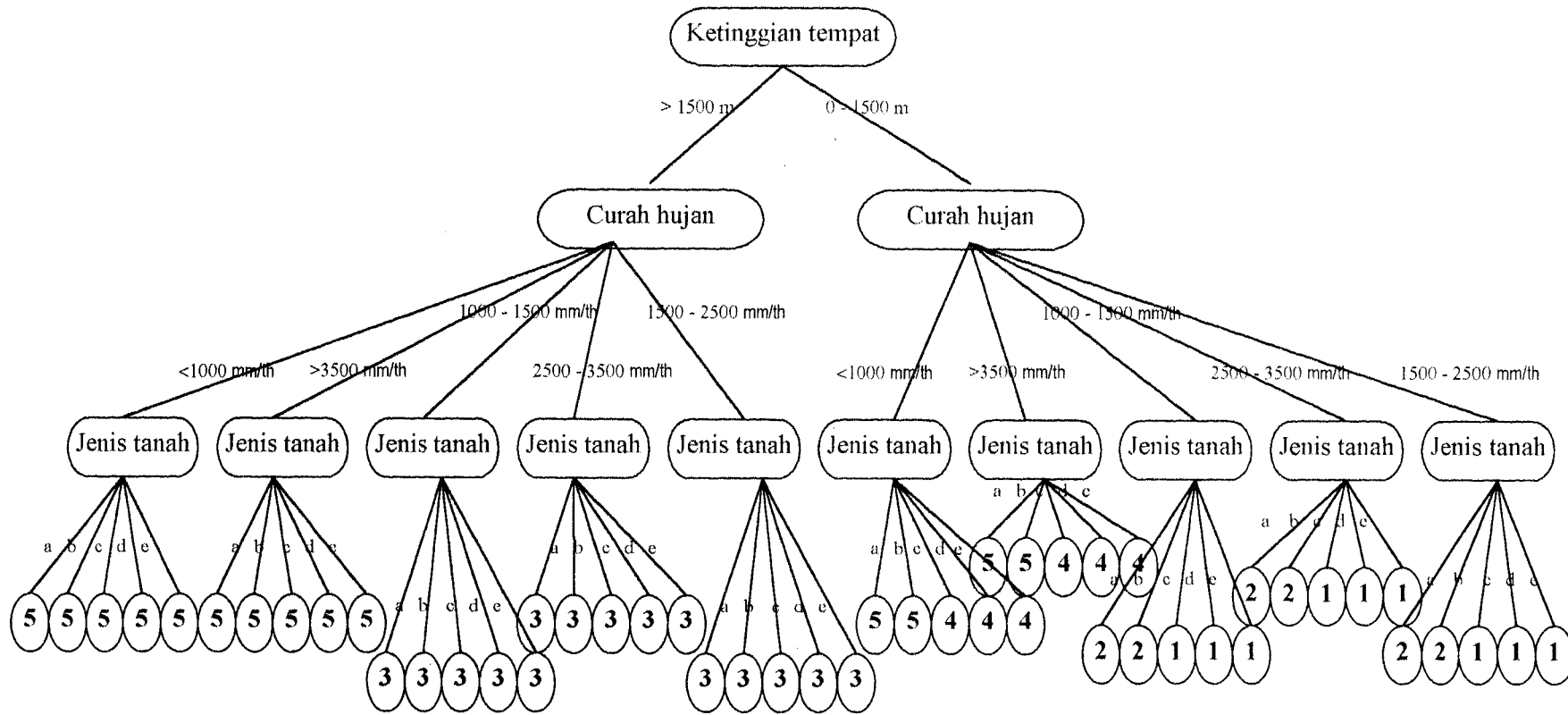
Pohon Keputusan untuk tingkat resiko adalah seperti yang terlihat pada gambar 4.2. Pohon Keputusan tingkat resiko dibentuk berdasarkan tabel listing atribut pada tabel 4.5. Cara membacanya sama dengan pohon

keputusan tipe topoklimatologi, berikut ini adalah penjelasan kode konklusi node:

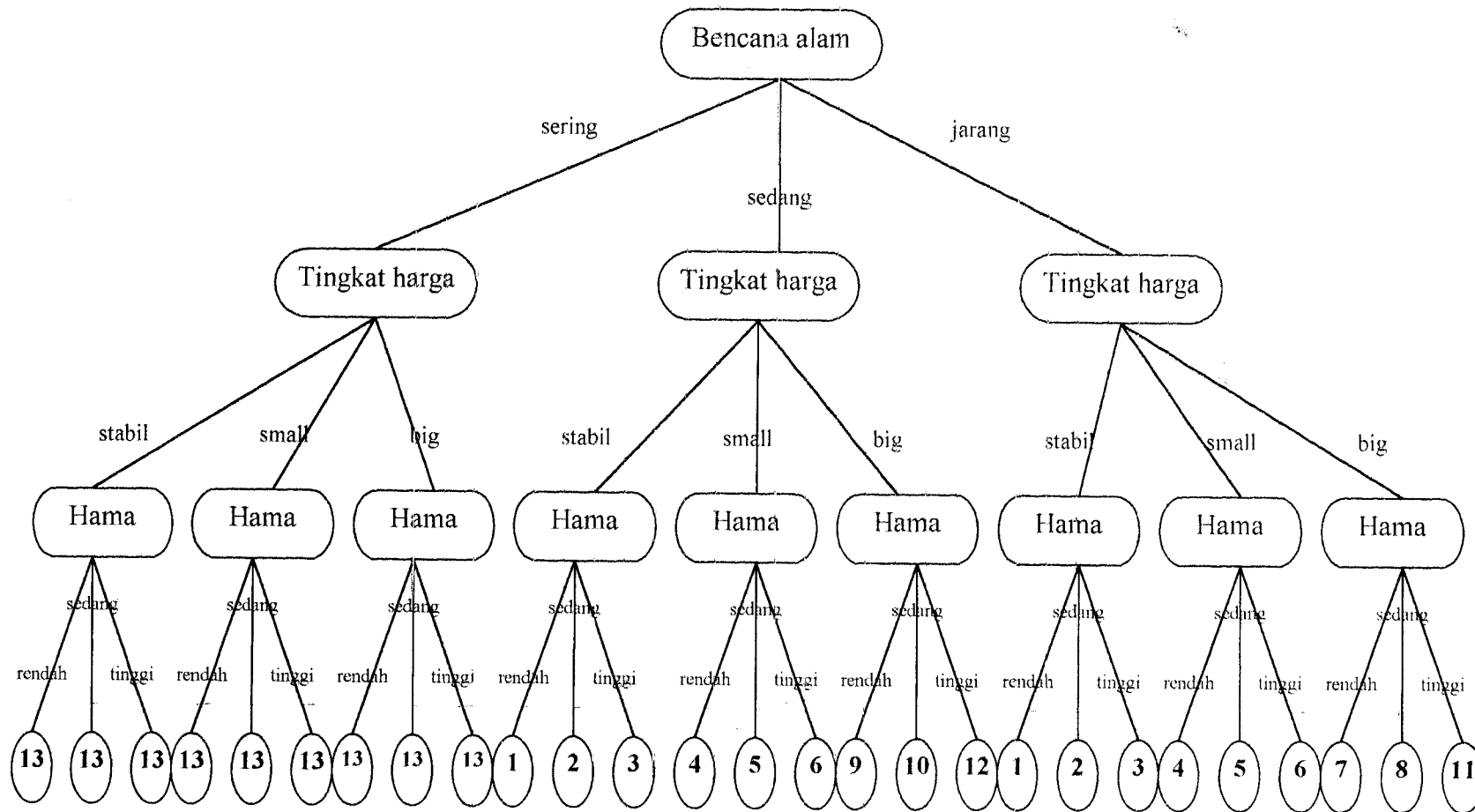
- Konklusi node (1) menunjukkan perlakuan manual
- Konklusi node (2) menunjukkan perlakuan sedang
- Konklusi node (3) menunjukkan perlakuan intensif
- Konklusi node (4) menunjukkan perlakuan penjadwalan produksi dan pengelolaan manual
- Konklusi node (5) menunjukkan perlakuan penjadwalan produksi dan pengelolaan sedang
- Konklusi node (6) menunjukkan perlakuan penjadwalan produksi dan pengelolaan intensif
- Konklusi node (7) menunjukkan perlakuan scoring perusahaan (1) dan pengelolaan manual
- Konklusi node (8) menunjukkan perlakuan scoring perusahaan (2) dan pengelolaan sedang
- Konklusi node (9) menunjukkan perlakuan scoring perusahaan (3) dan pengelolaan manual
- Konklusi node (10) menunjukkan perlakuan scoring perusahaan (4) dan pengelolaan sedang
- Konklusi node (11) menunjukkan perlakuan scoring perusahaan (5) dan pengelolaan intensif

- Konklusi node (12) menunjukkan perlakuan scoring perusahaan (6) dan pengelolaan intensif
- Konklusi node (13) menunjukkan gangguan bencana tinggi

Sedangkan untuk pohon keputusan teknologi cukup sederhana sehingga dengan melihat tabel 4.6 dianggap cukup. Rule yang dihasilkan dapat dilihat pada Lampiran A.



Gambar 4.1. Pohon Keputusan Tipe Topoklimatologi



Gambar 4.2 Pohon Keputusan Tingkat Resiko

4.1.2 MEKANISME INFERENSI

Mekanisme inferensi yang digunakan adalah *forward chaining*, yaitu penelusuran dari data-data yang ada untuk mencapai konklusi. Namun juga memanfaatkan *Backward Chaining* untuk mengecek apakah konklusi yang dihasilkan sesuai dengan fakta yang dihadapi. Pada perangkat lunak ini, mekanisme inferensi dilakukan dengan fasilitas Tlist yang dimiliki bahasa pemrograman Delphi 3.0 dengan cara menelusuri dari pangkal ke node yang belakangnya.

4.1.3 KOMPONEN PENJELAS

Komponen penjelas menyediakan segala penjelasan tambahan yang mungkin dibutuhkan oleh pemakai. Dalam perangkat lunak ini disediakan beberapa komponen penjelas, yaitu:

4.1.3.1 Komponen Susunan Aturan

Komponen ini memperlihatkan bagaimana listing aturan yang telah dibuat sebelumnya. Dengan demikian bisa diketahui aturan-aturan apa saja yang belum ada dan perlu untuk ditambahkan.

4.1.3.2 **Komponen How**

Komponen ini berisi keterangan mengapa suatu kondisi menghasilkan konklusi tertentu. Baik dalam metode inferensi Backward maupun forward chaining. Yang berfungsi menjelaskan pada pemakai alasan keputusan tersebut disarankan. Contoh pada forward chaining, user ingin mengetahui bagaimana bisa ditentukan topoklimatologi sesuai diusahakan batang dan daun (hijauan) sebagai salah satu konklusi dari fakta yang di-inputkannya. Perhatikan gambar 5.6.

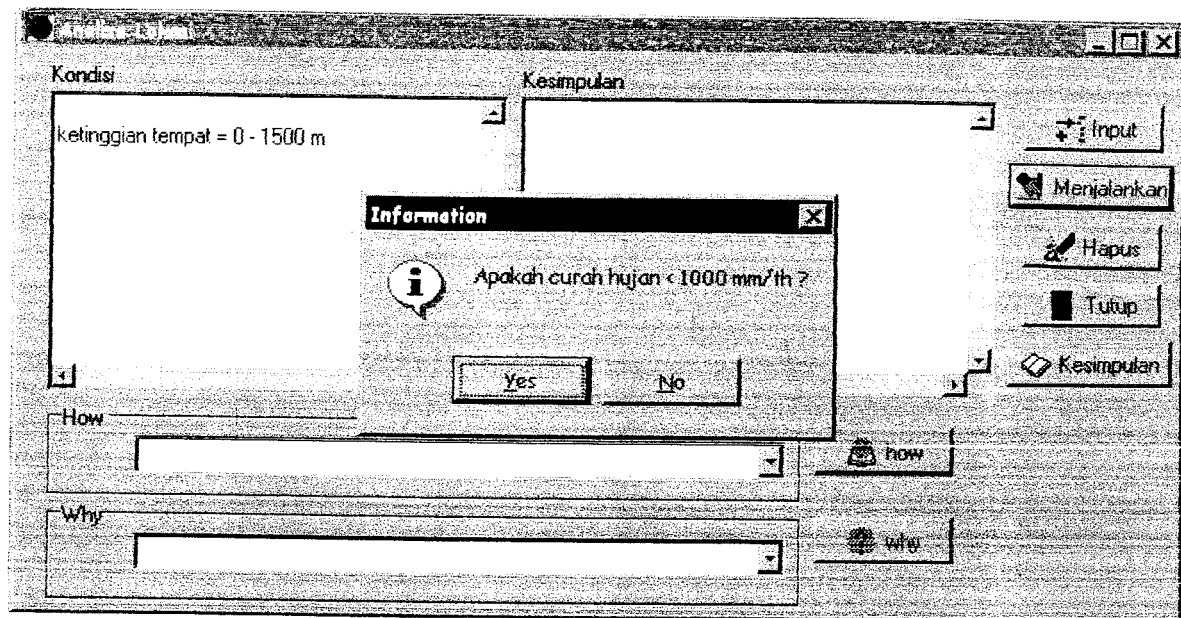
4.1.3.3 **Komponen Why**

Komponen ini memperlihatkan mengapa kondisi tertentu harus terpenuhi untuk suatu konklusi tertentu. Penggunaan komponen *Why* ini lebih jelasnya bisa dilihat pada bagian 5.1. perhatikan gambar 5.5.

4.1.4 **ANTARMUKA PEMAKAI**

Antarmuka pemakai merupakan bagian program yang berhubungan langsung dengan pemakai. Selain yang khusus menangani bagian konsultasi, termasuk antarmuka adalah komponen akuisisi untuk pengembangan sistem/aturan dan komponen penjelas (lihat 4.1.3). Dalam

Sistem Perencanaan Produksi Jagung ini, untuk bagian konsultasi metode tanya jawabnya dilakukan dengan menggunakan *yes-no questions* yaitu pertanyaan yang harus dijawab dengan “ya” atau “tidak”.



Gambar 4.3 Yes-no Question

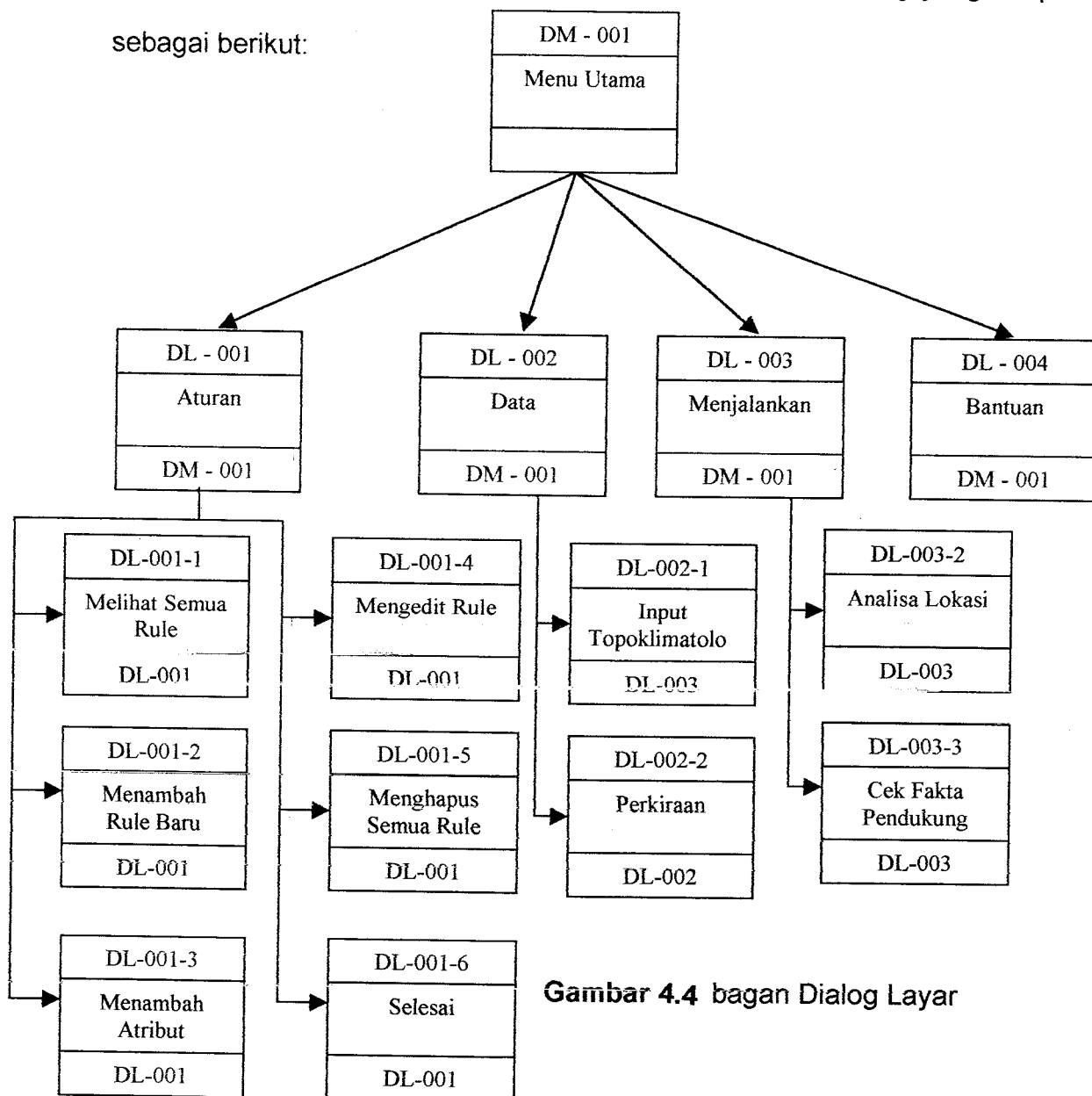
Desain Layar merupakan rancang bangun dari percakapan antara pemakai sistem dengan komputer. Dialog ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan output informasi kepada pemakai sistem atau keduanya melalui layar terminal. Dialog yang telah diidentifikasi didesain sebagai berikut:

Tabel 4.1.
Table Dialog

No/kode dialog	Nama Dialog
DL-001	Aturan
DL-002	Data
DL-003	Menjalankan
DL-004	Bantuan

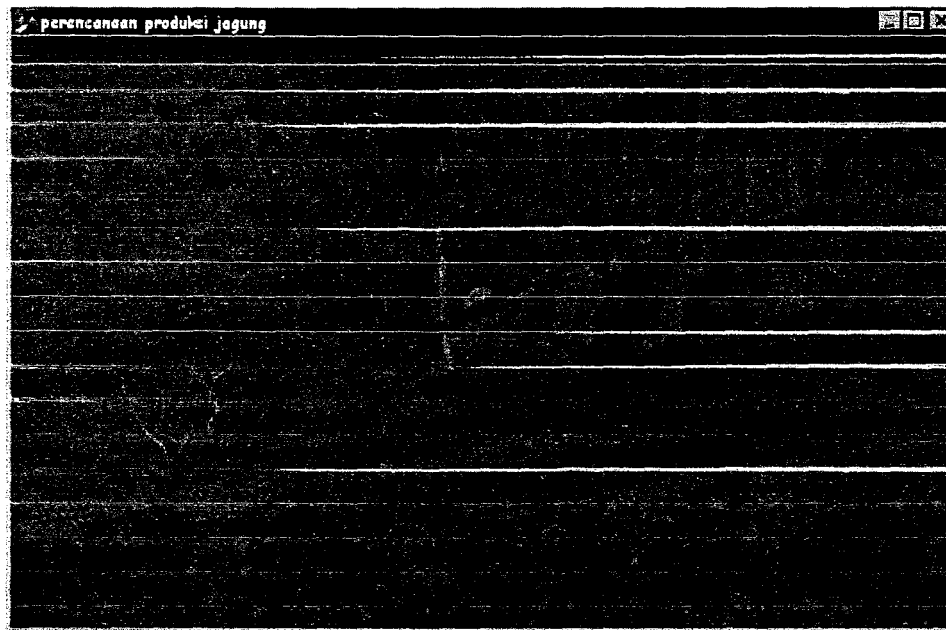
Untuk masing-masing dialog dapat terdiri dari sebuah tampilan saja atau lebih. Tampilan-tampilan layar ini dapat ditampilkan secara bergantian dengan penghapusan tampilan layar sebelumnya atau dapat dilakukan dengan menindih tampilan sebelumnya.

Dialog layar ini dapat digambarkan dengan bagan dialog yang tampak sebagai berikut:

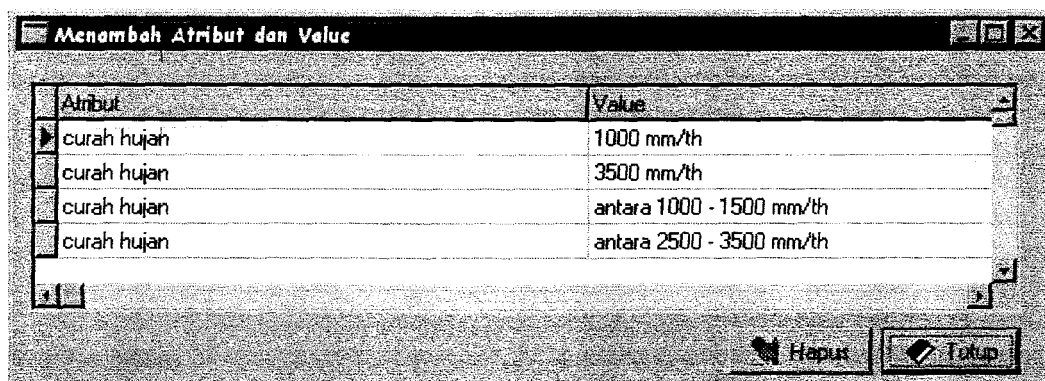


Gambar 4.4 bagan Dialog Layar

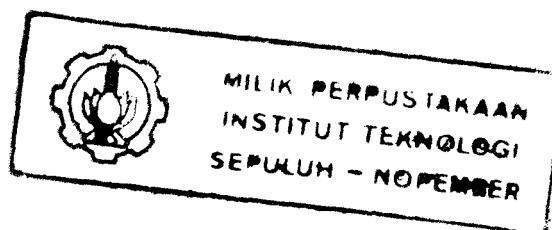
Bentuk dialog layar yang dijelaskan dalam bagan diatas adalah sebagai berikut:



Gambar. 4.5 Menu Utama



Gambar 4.6 Melihat semua Rule



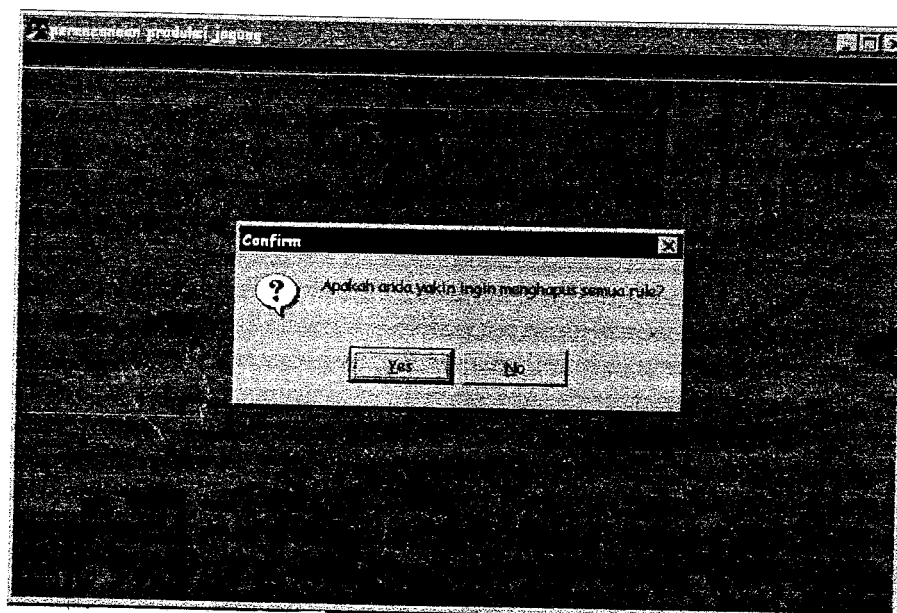
The screenshot shows a window titled "Melihat Semua Aturan" (View All Rules). It contains a table with three columns: "No Rule", "No Claus", and "Klausula IF". The table lists several rules with their respective clause descriptions. On the right side of the window, there are two buttons: "Tutup" (Close) and "Lihat" (View).

No Rule	No Claus	Klausula IF
247	1	perlakuan = penjadwalan, sedang
248	1	perlakuan = penjadwalan, intensif
249	1	perlakuan = scoring (1), manual
250	1	perlakuan = scoring (2), sedang
251	1	perlakuan = scoring (3), manual
252	1	perlakuan = scoring (4), sedang
253	1	perlakuan = scoring (5), intensif
254	1	perlakuan = scoring (6), intensif

Gambar. 4.7 Menambah Atribut dan Value

The screenshot shows a window titled "Mengedit Rule" (Edit Rule). It contains several input fields and buttons for editing a rule. The fields include "Nomor Rule" and "Nomor Klausula IF" for rule identification, and "Cari Nomor Rule" for searching. The main editing area has two rows: "IF" and "THEN". Each row has a "Variabel" (Variable) field, a "simbol" (symbol) dropdown, and a "nilai" (value) dropdown. At the bottom, there are buttons for "Hapus" (Delete), "OK", and "Tutup" (Close).

Gambar. 4.8 Mengedit Rule



Gambar. 4.9 Menghapus Semua Rule

A screenshot of the 'Informasi Harga' window. On the left, there is a table with two columns: 'Harga' and 'Periode'. The table contains 13 rows of data. To the right of the table, there are several interactive elements: a 'Hapus' button, a 'Perkiraan Tahun depan' section with a 'Persamaan Line' option and input fields for 'a=' and 'b=', a 'Perkiraan' button, a 'Periode ke' input field, and a 'Jumlah Perkiraan' input field. Below these is a 'Tingkat Harga' section with an input field and a 'Tingkat Harga' button. On the far right, there is a 'Kesimpulan' section with 'Asumsi nilai rupiah' and 'Point' input fields, an 'Info Harga' button, and 'Harga saat ini' and 'Point' input fields. At the bottom right, there are 'Dk' and 'Tutup' buttons.

Harga	Periode
26999	1
29999	2
28499	3
28449	4
28499	5
28943	6
28943	7
28943	8
28943	9
28493	10
28493	11
28493	12
31422	13

Gambar.4.10 Perkiraan

Kondisi yang diketahui

Fakta Apakah yang anda ketahui

Topoklimatologi

Ketinggian tempat

Curah hujan

Jenis Tanah

Teknologi dalam Masyarakat

Penggunaan Teknologi

Pengetahuan

Topoklimatologi

Ketinggian tempat = m

Curah Hujan = mm/th

Jenis tanah =

Teknologi dalam Masyarakat

Penerapan Teknologi

pengetahuan

Clause	Symbol	Value
▶ ketinggian tempat	=	0 - 1500 m

Gambar. 4.11 Input Topoklimatologi

Analisa Lokasi

Kondisi

ketinggian tempat = 0 - 1500 m
 ketinggian tempat = 0 - 1500 m
 curah hujan < 1000 mm/th
 ketinggian & curah hujan = ketinggian 0 - 1500 d
 jenis tanah = liat (halus)

Kesimpulan

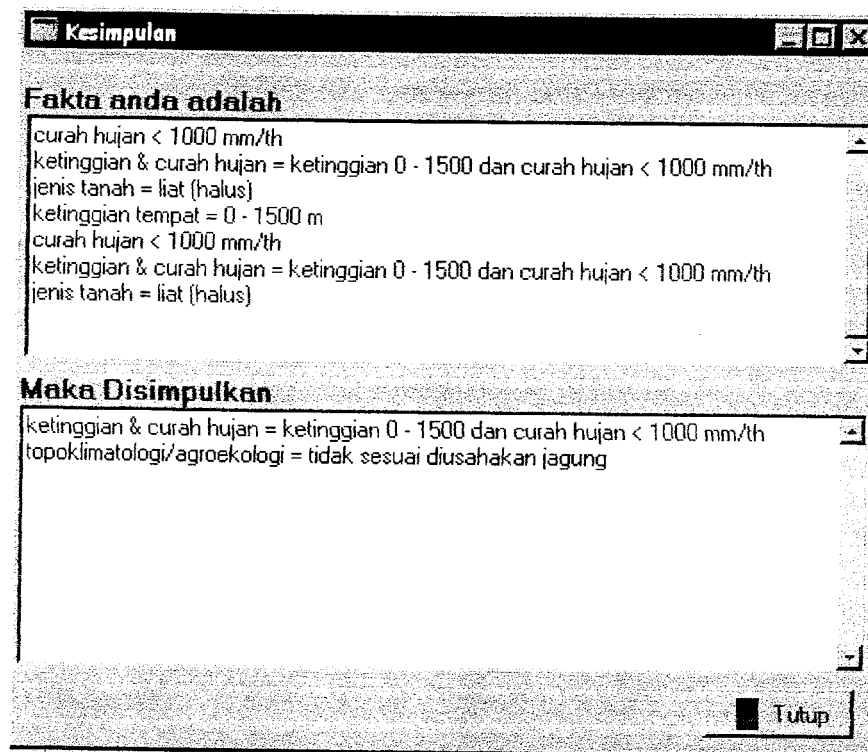
ketinggian & curah hujan = ketinggian 0 - 1500 dar
 topoklimatologi/agroekologi = tidak sesuai diusaha

kesimpulan dari fakta yang anda miliki

How

Why

Gambar 4.12 Analisa Lokasi



Gambar 4.13 Kesimpulan

Antar Muka Pemakai untuk Analisa Lokasi pada gambar 4.11 berfungsi sebagai antar muka dalam memproses aturan yang ada dengan fakta yang dimiliki pemakai, bila pemakai ingin melihat kesimpulan yang dihasilkan maka pemakai dapat mengklik tombol kesimpulan sehingga tampil antar muka seperti pada gambar 4.12, sehingga pemakai memperoleh informasi fakta yang dimilikinya dan kesimpulan yang dihasilkan oleh sistem berkaitan dengan fakta yang dimiliki tersebut. Antar muka pemakai untuk dialog Cek Aturan Terkait pada gambar 5.8 berguna sebagai antarmuka bagi pemakai untuk mengecek aturan yang berkaitan dengan satu atau kesimpulan.

4.1.5. KOMPONEN AKUISISI

Komponen akuisisi dalam Sistem Perencanaan Produksi Jagung Berbasis Aturan ini diimplementasikan dalam bentuk Basis Aturan. Basis Aturan tersebut menggunakan prinsip IF... THEN.

Dengan demikian, penyusunan aturan bisa mengikuti prinsip tersebut sehingga untuk membuat aturan yang berisi banyak premis untuk suatu konklusi bisa diwujudkan dengan menentukan nomor rule dan untuk rule yang bagian if-nya berisi beberapa kondisi yang harus dipenuhi maka memiliki nomor klausa yang berbeda satu sama lain dalam satu rule tersebut. Misal untuk aturan IF A AND B AND C THEN X, direpresentasikan dengan menempatkan B sebagai *If clause dengan nomor clause 2* dan A sebagai *If clause dengan nomor clause 1* dan C sebagai *If clause dengan nomor clause 3* dan seterusnya.

4.2 PIHAK-PIHAK YANG TERLIBAT

Pembangunan dan implementasi Sistem Perencanaan Produksi Jagung ini melibatkan beberapa pihak yang terbagi dalam dua tahapan, yaitu:

4.2.1 TAHAP PERANCANGAN

Tahap Prancangan adalah sewaktu sistem ini pertama kali dibuat. Dalam tahap ini Basis Pengetahuan disediakan sesuai dengan kebutuhan yang dipahami pihak pembangun sistem. Pihak yang terlibat dalam tahapan ini yaitu perekayasa pengetahuan dan seorang pakar yang berperan sebagai pakar. Basis pengetahuan yang digunakan informasi yang diberikan oleh pakar, yaitu prosedur umum yang biasa dilakukan pakar dalam mengambil keputusan untuk menyelesaikan permasalahan Perencanaan Produksi Jagung.

4.2.2 TAHAP IMPLEMENTASI

Tahap implementasi merupakan tahap memanfaatkan Sistem Perencanaan Produksi Jagung untuk menjalankan fungsinya dalam menganalisa fakta yang ada dan menyarankan tindakan yang perlu dilakukan menurut seorang pakar. Pada tahap ini ada dua pihak yang terlibat yaitu:

1. **Perekayasa Pengetahuan**, yang berperan merepresentasikan pengetahuan pakar dan memiliki hak akses terhadap perubahan/penyempurnaan susunan aturan yang ada.
2. **Pemakai biasa**, yaitu orang-orang yang memanfaatkan perangkat lunak ini untuk menyelesaikan masalah perencanaan produksi jagung yang dihadapinya.

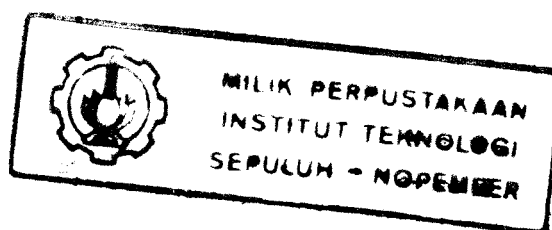
4.3 PERANCANGAN DATA

Pada subbab ini, akan dibahas tentang prosedur dan fungsi yang dipergunakan dalam pembuatan sistem berbasis aturan ini. Serta dibahas tentang tipe yang dimanfaatkan untuk merepresentasikan aturan yang telah dibuat sehingga bisa dimanfaatkan dalam aplikasi yang dibuat.

4.3.1 TLIST

List digunakan untuk mengelola index dari object dari beberapa tipe class. Index itu sendiri menyimpan pointer pada suatu obyek dan secara dynamic dapat di rezised pada saat runtime. Karena kemampuannya untuk tetap memegang segala tipe object, TList dikatakan fleksibel, tetapi membutuhkan lebih banyak pemrograman dari pada VCL List class yang lain untuk didesain mengatasi suatu object tipe tertentu (seperti TString dan TImageList). Tetapi jangan menyalahartikan TList dengan Full-scale collection classes atau Linked-List, karena TList tidak mengatur alokasi atau Dealokasi memory untuk obyek yang ber index tersebut. Adalah tanggung jawab programmer untuk mengelolanya apabila memilih menggunakan TList.

Melalui internal operasi dari TList yang melibatkan manipulasi pointer, Delphi membantu anda pada saat implementasi dengan menyediakan properti yang nyaman dan method untuk mengakses list anda. Beberapa



TList method, bagaimanapun menggunakan generic tipe pointer sebagai parameter. Ingat pada kondisi ini sesuatu yang dinamakan object secara implisit adalah pointer, sehingga tidak diperlukan manipulasi address. Jika anda melihat sebuah parameter bertipe Pointer, anda dapat menggunakannya jika ia bertipe Tobject. Dalam beberapa cara, anda dapat memperlakukan TList sebagai sebuah array pointer, dengan menggunakan Items property untuk mengakses elemen list secara individual.

4.3.2 PROPERTI DAN METHOD YANG DIGUNAKAN DALAM TLIST

TList adalah base class untuk List. Tlist dapat menerapkan banyak Method dan properti yang dapat digunakan untuk memanipulasi list, mengakses dan memodifikasi list items, dan mengumpulkan informasi tentang list dan items-nya.

4.3.2.1. Add

Fungsi : Untuk menambahkan Item baru pada akhir
dari List

Deklarasi :

```
Function Add(item: Pointer): Integer;
```

Contoh Syntax :

```

// prosedur ini membuat sebuah list
// mencetak posisinya pada list (0) sebagai
// caption dari form

procedure TForm1.FormCreate(Sender: TObject);
var
    MyList : TList;
    MyObject : TObject;
    Position : Integer;
begin
    // Membuat List
    MyList := TList.Create;
    // Membuat sebuah object dan menambahkannya
    // pada list
    MyObject := TObject.Create;
    Position := MyList.add(MyObject);
    Form1.Caption := 'The position in the list is '
+IntToStr(Position);
    // jangan lupa me-free object anda
    MyObject.free;
    Mylist.Free;

```

Deskripsi : **add** memperoleh posisi (sebuah nilai integer) pada list ketika sebuah obyek baru dimasukkan. Items pertama pada list selalu berada pada posisi 0, sehingga pada contoh diatas akan tampak hasil "The position in the list is 0" Parameter **item** digunakan untuk memanggil obyek yang diletakkan dalam list.

4.3.2.2. Clear

Fungsi : Clear method berfungsi menghapus semua elemen dalam sebuah list

Deklarasi : Procedure Clear;

Contoh Syntax :

```

var
  Form1 : TForm1;
  MyList : TList;

// Membuat List ketika program dimulai
procedure TForm1.FormCreate(Sender: Object);
begin
  Mylist := Tlist.Create;
end;

// Free List ketika program selesai
procedure TForm1.formClose(Sender: TObject; var
Action TcloseAction);
begin
  MyList.Free;
end;

// Jika Button1 ditekan maka tambahkan sebuah
// elemen dalam list
procedure TForm1.Button1Click(Sender: TObject);
begin
  MyList.Add(MyList);
  Form1.caption := 'Number of items in List = ' +
inttostr(MyList.Count);
end;

// Jika Button2 ditekan clear List
Procedure TForm1.Button2.Click(Sender:TObject):
begin
  MyList.Clear;
  Form1.caption := 'Number of items in List = ' +
inttostr(MyList.Count);
end;

```

Deskripsi : Gunakan **Clear** untuk menghapus semua elemen dalam list. Program ini juga men set **count** dan **capacity** properti menjadi 0. Membebaskan memori yang dipergunakan oleh elemen dari list.

Perhatian : **Clear** tidak menghapus object asli yang ditunjuk oleh elemen dari list index, clear hanya menghapus pointer antar index. Jika anda ingin menghapus object tersebut, maka anda perlu melakukannya sendiri.

4.3.2.3. Count

Fungsi : Count Properti berisi jumlah elemen dalam sebuah list

Deklarasi :

```
property count: Integer read FCount write SetCount
```

Contoh Syntax :

```
var
    Form1 : TForm1;
    MyList : TList;

// Membuat List ketika program dimulai
procedure TForm1.FormCreate(Sender: Object);
begin
    Mylist := Tlist.Create;
end;

// Free List ketika program selesai
procedure TForm1.formClose(Sender: TObject; var
Action TcloseAction);
begin
    MyList.Free;
end;

// Jika Button1 ditekan maka tambahkan sebuah
// elemen dalam list
procedure TForm1.Button1Click(Sender: TObject);
begin
    MyList.Add(MyList);
    Form1.caption := 'Number of items in List = ' +
inttostr(MyList.Count);
end;
```

```
// Jika Button2 ditekan clear List
Procedure TForm.Button2.Click(Sender:Tobject):
Begin
    MyList.Clear;
    Form1.caption := 'Number of items in List = ' +
    inttostr(MyList.Count);
End;
```

Deskripsi : **Count** adalah sebuah run-time dan read-only property yang berisi jumlah elemen dalam sebuah list. Sebuah List dengan **count** value 3 akan memiliki elemen dengan posisi 0, 1 dan 2 dalam list. Yaitu nomor aktual yang dipergunakan oleh elemen dari list, bukan nomor alokasi dari elemen. **Capacity** properti berisi nomor dari alokasi elemen.

4.3.2.4. Delete

Fungsi : Delete Properti berfungsi menghapus sebuah elemen dari list

Deklarasi : Procedure Delete(Index: Integer);

Contoh Syntax :

```
Procedure TForm1.FormCreate(Sender: Tobject);
Var
    X: integer;
    MyObject: Tobject;
Begin
    // membuat List dan Object
    MyList := Tlist.Create;
    MyObject := Tobject.Create;
    // tambahkan 10 items dalam List
    For x:= 1 to 10 do MyList.Add(MyObject);
```

```
// Hapus items ke-dua dari list  
MyList.Delete(1);  
End;
```

Deskripsi : **Delete** memindahkan elemen dari list yang ditunjuk oleh **index** parameter. Sebagai catatan list index dimulai dengan 0. Sehingga dalam contoh diatas, 1 yang digunakan untuk mengisi parameter index berfungsi untuk menghapus item ke-2 dalam list. Ketika sebuah item dihapus dari sebuah list nilai dari items ke-2 menjadi nil. Untuk menghapus nil dari list gunakan **Pack**.

Perhatian : **Delete** tidak menghapus secara phisical object asli yang ditunjuk oleh elemen dari list index, delete hanya menghapus pointer antar index. Jika anda ingin menghapus object tersebut, maka anda perlu melakukannya sendiri.

4.3.2.5. Items

Fungsi : **Items** property memperbolehkan anda untuk mengakses secara individual items dalam sebuah list.

Deklarasi :

```
property items[index: integer]:pointer;
```

Coptoh Syntax :

```
// prosedur ini membuat sebuah object baru,  
// TnewObject, dan meng-assign-nya pada elemen  
// pertama dari list MyList  
  
procedure TForm1.Button1.Click(Sender:Tobject);  
var  
    MyNewObject: Tobject;  
begin  
    MyNewObject := MyList.Items[0];  
    //...  
end;
```

Deskripsi : **Items** digunakan untuk mengubah item tertentu dalam list, dengan menggunakan list index untuk menentukan object yang mana yang akan dirubah.**Items** mengembalikan pointer pada list item. Elemen pertama dari list adalah **items[0]**

4.3.3. WORKING MEMORI

Agar tiap-tiap aturan yang telah dikonfirmasi oleh user tersimpan, diperlukan satu tempat untuk menyimpannya. Untuk itu, diperlukan suatu *working memory*, yang menyimpan setiap data yang diperoleh selama sesi konsultasi.

Untuk Menyimpan seluruh Rule yang ada, yang akan dibandingkan dengan fakta yang dimasukkan oleh User. VariableList mencatat hal-hal yang ada dibagian IF dan hal-hal dibagian THEN.

```
PVariableList = ^VariableList;
VariableList = record
    RuleNum      : integer;
    clauseNum    : integer;
    IFVar        : string;
    IFSymbol     : ShortStr;
    IFValue      : string;
    THENVar      : string;
    THENSymbol   : ShortStr;
    THENValue    : string;
end;
```

Untuk menyimpan Nomor Rule dari Konklusi yang dihasilkan dari proses inferensi.

```
PConclution = ^Conclution;
Conclution = record
    result       : string;
    ruleNumber   : integer;
end;
```

Untuk menyimpan inputan kondisi dari user. Pada saat Inferensi Forward Chaining.

```
PCondition = ^Condition;
Condition = record
    VarCond      : string;
    VarSym       : ShortStr;
```

```

    VarVal : string;
    sign : boolean;
end;

```

Untuk menyimpan inputan konklusi dari user. Pada saat Inferensi Backward Chaining.

```

PTempTHEN = ^TempTHEN;
TempTHEN = record
    posRule : integer;
    variable : string;
    symbol : ShortStr;
    value : string;
    sign : boolean;
end;

```

Untuk menyimpan seluruh Rule yang dilewati pada saat proses inferensi.

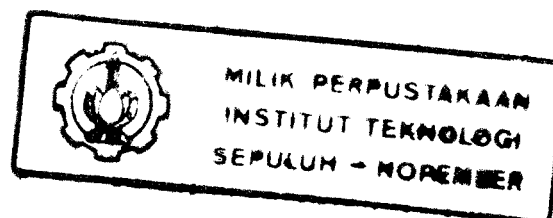
```

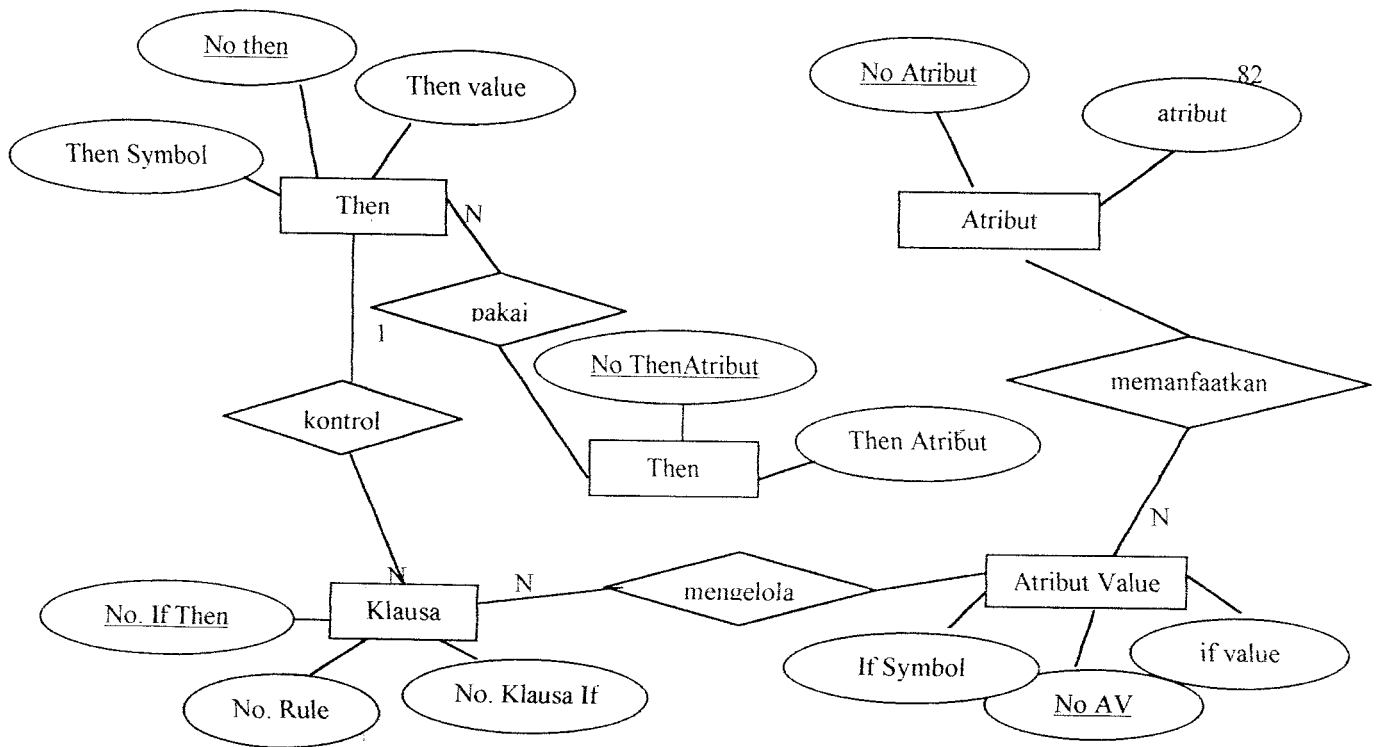
PHowWhy = ^HowWhy;
HowWhy = record
    NumRule : integer;
    IFVar : string;
    IFSymbol : ShortStr;
    IFValue : string;
    THENVar : string;
    THENSymbol : ShortStr;
    THENValue : string;
end;

```

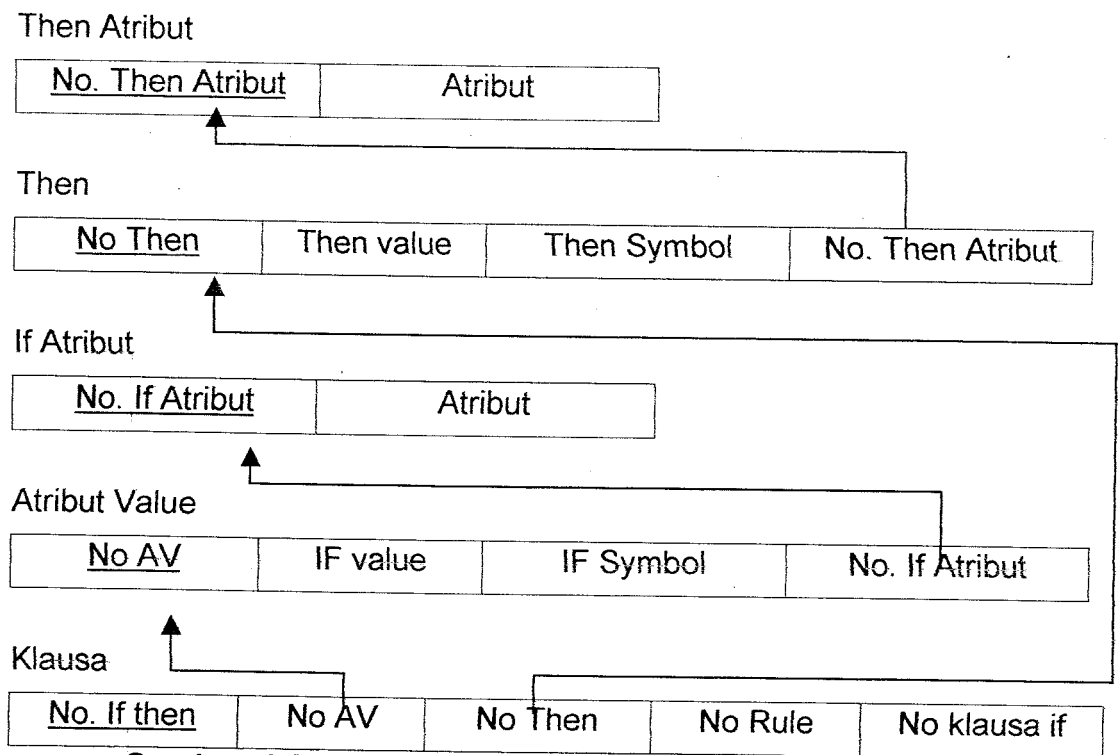
4.3.4. BASIS PENGETAHUAN

Salah satu cara untuk menyimpan rule secara permanen adalah dengan menggunakan basis data. Berikut ini adalah rancangan Basis data yang mungkin digunakan dalam menyimpan aturan yang dipergunakan secara permanen. ER-Diagram untuk basis pengetahuan tersebut adalah seperti yang terlihat pada gambar 4.13. Pada gambar 4.14 anda dapat melihat Skema Relasi Basis Pengetahuan dari rancangan basis data tersebut





Gambar 4.14 ER-Diagram Basis Pengetahuan



Gambar 4.15 Skema Relasi Basis Pengetahuan

Deskripsi dari masing-masing tabel tersebut adalah sebagai berikut:

Then Atribut

No	Nama Data	Tipe	Lebar	Keterangan
1	No. Then Atribut	Integer		Nomor atribut untuk bagian then
2	Atribut	String	50	Nama dari atribut yang dipergunakan pada bagian then

Then

No	Nama Data	Tipe	Lebar	Keterangan
1	No. Then	Integer		Nomor identifikasi dari gabungan atribut dan value bagian then
2	Then Value	String	50	Nilai dari atribut pada bagian then tersebut
3	Then Symbol	String	1	Sym bol dari hubungan atribut dan value
4	No. Then Atribut	Integer		Nomor daro atribut yang digunakan dalam bagian then ini

If Atribut

No	Nama Data	Tipe	Lebar	Keterangan
1	No. If Atribut	Integer		Nomor atribut yang digunakan pada bagian if
2	Atribut	String	50	Nama dari atribut yang dipergunakan pada bagian if

Atribut dan Value

No	Nama Data	Tipe	Lebar	Keterangan
1	No. AV	Integer		Nomor identifikasi dari gabungan atribut dan value bagian if
2	If Value	String	50	Nilai dari atribut pada bagian if tersebut
3	If Symbol	String	1	Sym bol dari hubungan atribut dan value
4	No. If Atribut	Integer		Nomor dari atribut yang digunakan dalam bagian if ini

Klausa

No	Nama Data	Tipe	Lebar	Keterangan
1	No. If then	Integer		Nomor dari klausa
2	No. Rule	Integer		Nomor Rule dari suatu klausa
3	No. Klausa If	Integer		Nomor klausa if dari suatu rule tertentu
4	No. AV	Integer		Nomor dari atribut dan value pada bagian if
5	No. Then	Integer		Nomor dari atribut dan value pada bagian then

Dalam sistem ini basis pengetahuan secara permanen disimpan dalam sebuah tabel dengan nama 'Jagung-Biji' . Tabel tersebut terdiri dari 8 field. Deskripsi dari field-field yang dipergunakan dalam tabel tersebut adalah sebagai berikut:

Number : berisi field nomor dari rule dengan tipe integer.

- If Clause** : berisi field klausa bagian if (atribut bagian if) bertipe string .
- If Symbol** : berisi field simbol dari klausa if , simbol disini hanya ada tiga pilihan yaitu (<, >, =) pemakai dalam menambahkannya hanya dengan memilih dari commbobox yang ada.
- If Value** : berisi field nilai dari klausa if
- Then Clause** : berisi field dari klausa then (yaitu atribut dari bagian then)
- Then Symbol** : berisi field simbol dari klausa then, seperti halnya if Symbol pada Then Symbol memiliki tiga pilihan simbol.
- Then Value** : berisi field nilai dari klausa then
- NumClause** : berisi field nomor klausa dari suatu rule tertentu yang berada dalam hubungan and, untuk menentukan NumClause sistem melakukannya secara otomatis ketika pemakai menekan tombol And. misalnya rule 1 adalah if A=a and B=a and C=a then D=a, maka rule 1 akan direpresentasikan sebagai berikut:

No Rule	If Clause	If Symbol	If Value	Then Clause	Then Symbol	Then Value	Num Clause
1	A	=	a	D	=	a	1
1	B	=	a	D	=	a	2
1	C	=	a	D	=	a	3

4.4 PERANCANGAN PROSES

Adapun proses yang terjadi selama penggunaan sistem ini, mulai dari awal hingga mendapatkan hasil, adalah sebagai berikut:

1. Inisialisasi meliputi mengambil seluruh rule yang ada dibasis pengetahuan kedalam working memori.
2. Proses konsultasi.
 - a. Menganalisa data permintaan dan harga yang dimiliki oleh User (bila ada)
 - b. Konsultasi tentang kondisi yang dimiliki.
3. Proses penyimpanan fakta dalam *working memory*.
4. Proses Penjelasan, mengapa suatu saran atau kesimpulan diajukan.
5. Selesai → mengembalikan setiap nilai dan data ke dalam bentuk awal agar siap untuk digunakan dalam proses berikutnya.

Proses secara detail yang terjadi dalam sistem adalah sebagai berikut:

4.4.1. FORWARD CHAINING

Secara mendasar , terdapat dua filosofi yang berbeda tentang forward chaining pada satu sisi , semua data yang diketahui diberikan pada sisitem pakar pada saat yang sama ketika sesi konsultasi. Sistem pakar hanya

meminta elemen tertentu dari data yang diminta pada saat sesi konsultasi.

Kedua pendekatan tersebut mempunyai keuntungan dan kerugiannya masing-masing. Pada pendekatan yang pertama sistem akan berjalan dengan baik apabila sistem memperoleh data secara otomatis dari basisdata fakta yang dimiliki. Sedangkan pada sistem yang kedua sistem hanya menanyakan hal-hal yang diperlukan saja untuk mencapai suatu kesimpulan yang tentu saja berdasarkan data aturan yang dimiliki. Dalam Sistem berbasis aturan yang dibangun ini menggunakan pendekatan yang kedua (perhatikan gambar 4.12). Algoritma untuk inferensi forward chaining adalah sebagai berikut:

1. Identifikasi Kondisi
2. Variabel kondisi ditempatkan pada Concluci Var. Queue dan nilainya dicatat pada variabel list.
3. Diadakan pencarian pada clause varlist untuk variabel yang namanya sama dengan nama pada awal queue. Jika ketemu, nomor rule dan nomor clause diisikan pada clause var .pointer. Jika tidak ketemu, ke step 6.
4. Setiap variabel dalam if clause dari rule yang belum diisi, selanjutnya diisi. Variabel-variabel ditempatkan dalam clause var.list. Jika semua clause benar kondisinya bagian then dijalankan.

5. Pengisian variabel then pada variabelnya ditempatkan pada bagian belakang di conclusion var. queue.
6. Jika tidak ada lagi statemen if yang mengandung variabel yang berada di awal conclusion var.queue, maka variabel tersebut dihapus.
7. Jika tak ada lagi variabel pada conclusion var queue, pencarian berakhir; jika masih ada variabel yang lain, kembali ke step 3.

Tipe sistem yang dapat dicari dengan forward Chaining:

1. Sistem yang dipresentasikan dengan satu atau beberapa kondisi
2. Untuk setiap kondisi, sistem mencari rule-rule pada basis pengetahuan untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian if.
3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian THEN. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi, sistem akan kembali ke step 2 dan mencari rule-rule dalam basis pengetahuan kembali. Jika tidak ada kondisi baru sesi ini berhenti.

4.4.2. BACKWARD CHAINNING

Backward chainning merupakan pelacakan aturan dengan dimulai dari konklusi kemudian mencek clause-clause pendukung konklusi tersebut

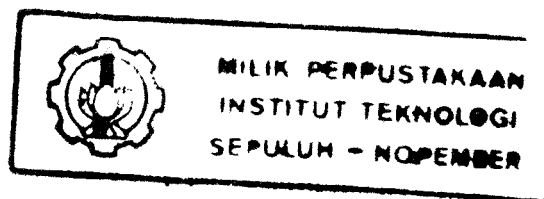
(perhatikan gambar 4.13). Berikut ini adalah algoritma inferensi backward chaining:

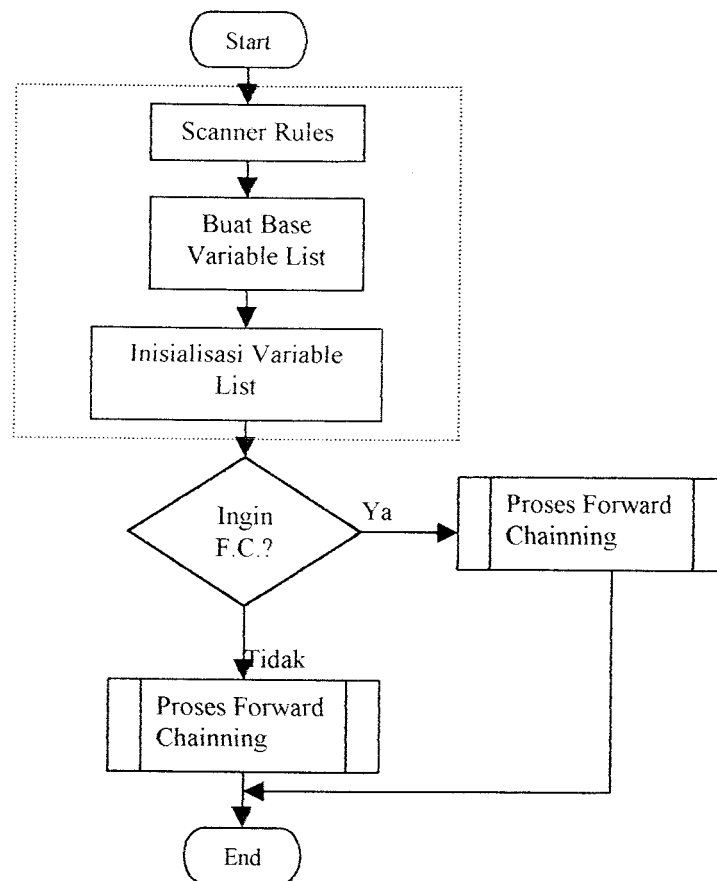
1. Identifikasi konklusi
2. Cari pada conclusion list untuk pengisian pertama dari nama konklusi. Jika ketemu, tempatkan rule pada conclusion stack berdasarkan nomor rule dan 1 yang merepresentasikan nomor clause. Jika tak ketemu, beritahu user bahwa jawaban tersebut tidak ada.
3. Isi if clause (yaitu, setiap variabel kondisi) dari statemen.
4. Jika satu dari variabel pada IF clause belum diisi, yang diindikasikan oleh varlist, dan bukan merupakan variabel konklusi, yaitu tak ada dalam conclusion list, tanyakan user untuk memasukkan satu nilai.
5. Jika satu dari clause adalah variabel konklusi, tempatkan nomor rule dari variabel konklusi di top of stack dan kembali ke step 3.
6. Jika statemen pada top of stack tak dapat di instantiate menggunakan statemen IF-THEN yang ada, hapus dari top of stack dan cari pada conclusion List untuk pengisian lain dari nama variabel konklusi
7. Jika Suatu statemen ditemukan, kembali ke step 3
8. Jika tak ada konklusi yang tersisa pada conclusion stack rule untuk konklusi sebelumnya adalah salah jika tak ada konklusi sebelumnya , maka beritahu user jawaban tak ditemukan, jika ada konklusi sebelumnya kembali ke step 6

9. Jika rule pada top of stack dapat di-instantiate, hapus rule tersebut dari stack. Jika ada variabel konklusi lain dibawahnya (pada stack), increment nomor clause, dan untuk clause yang tersisa kembali ke step 3. Jika tak ada variabel konklusi lain di bawahnya, didapatkan jawabannya

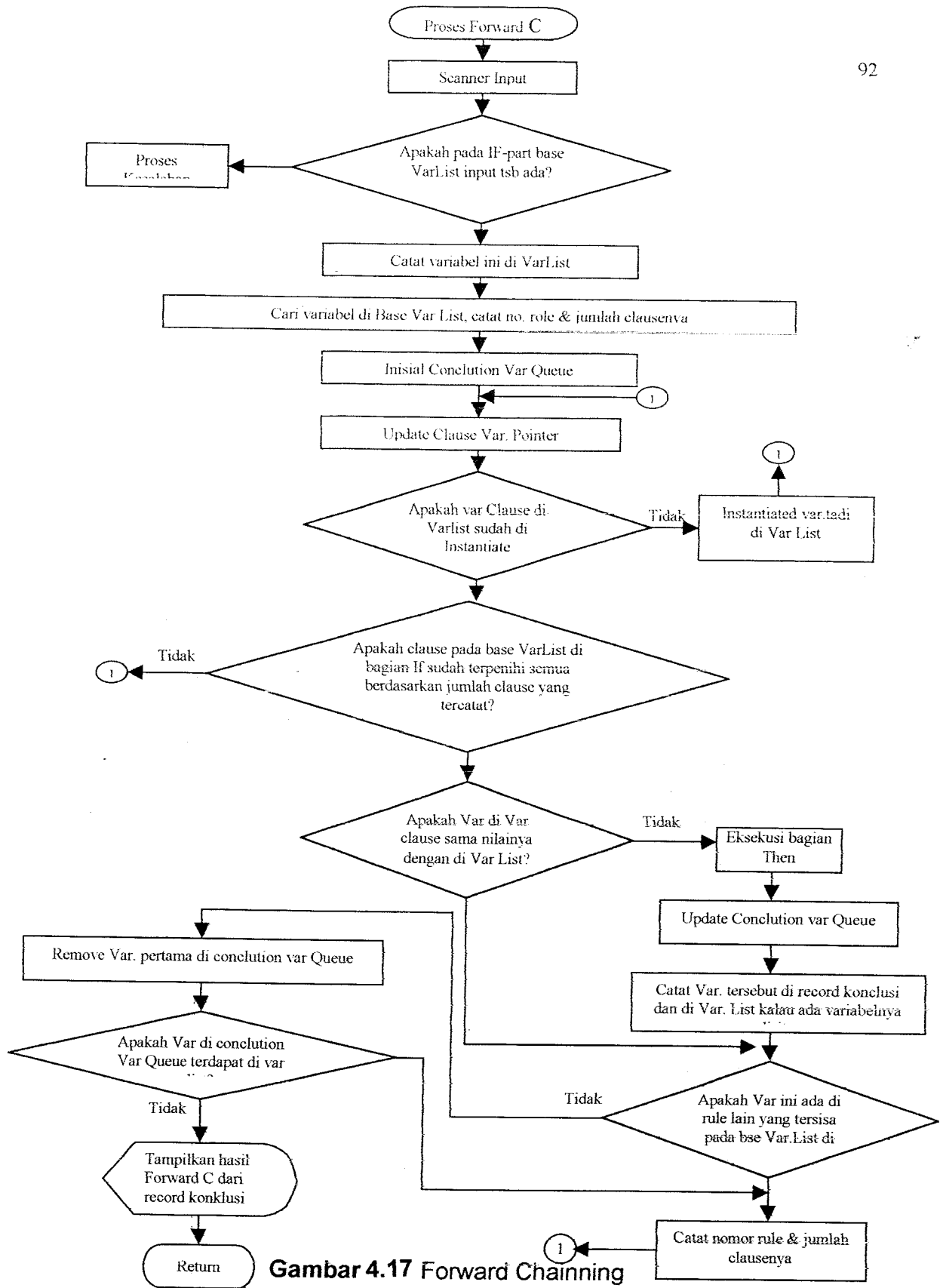
Tipe sistem yang dapat dicari dengan Backward Chaining:

1. Sistem yang di presentasikan dengan satu atau beberapa kondisi
2. Untuk setiap konklusi, sistem mencari rule-rule dalam basis pengetahuan untuk rule-rule yang berkorespondensi dengan konklusi pada bagian then
3. Setiap konklusi dihasilkan dari kondisi-kondisi yang terdapat pada bagian IF. Selanjutnya kondisi-kondisi tersebut menjadi konklusi baru yang dimasukkan kedalam stack diatas konklusi yang sudah ada
4. Setiap konklusi yang ditambahkan kesistem akan di proses. Jika ditemui suatu konklusi, sistem akan kembali ke step 2 dan mencari rule-rule dalam basis pengetahuan kembali. Jika tidak ada konklusi baru, sesi ini berhenti.

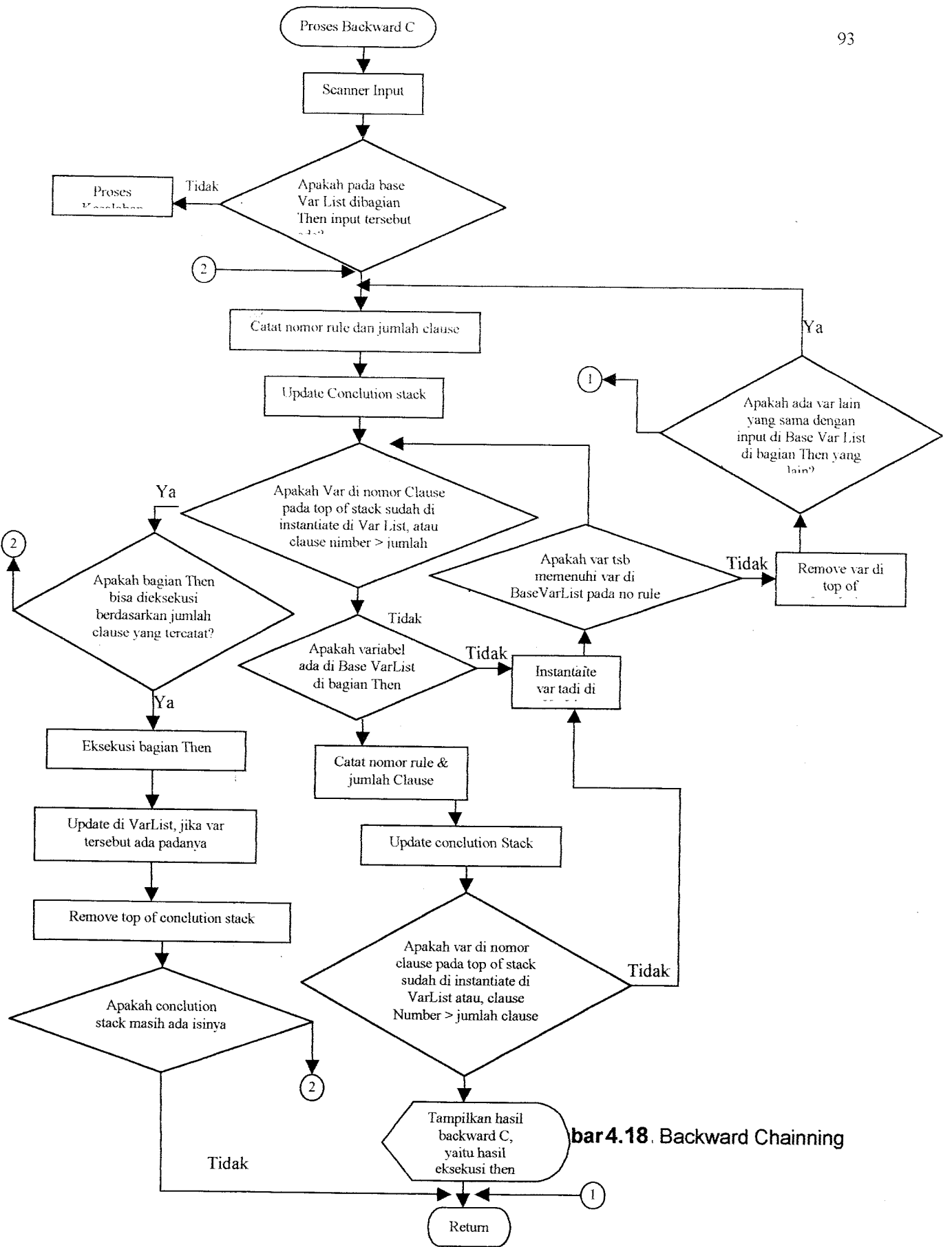




Gambar 4.16 Proses pengambilan Rule ke working memory



Gambar 4.17 Forward Chaining

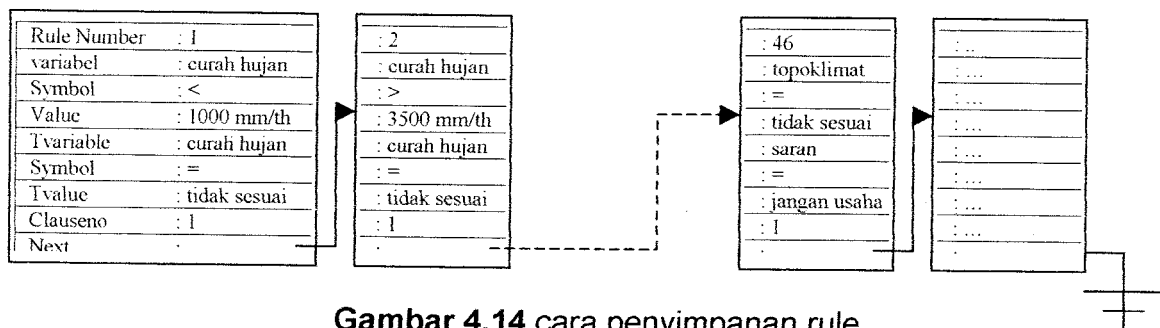


bar 4.18. Backward Chaining

4.5 IMPLEMENTASI PENYAJIAN ATURAN

Cara yang dipakai untuk menyajikan aturan dalam Sistem Perencanaan Produksi Jagung ini adalah dengan menggunakan fasilitas TList yang ada dalam bahasa pemrograman Delphi 3.0. lihat bagian 4.3.

Rule yang dibuat secara permanen akan disimpan dalam tabel, pada saat dibutuhkan untuk melakukan proses inferensi. Rule tersebut diambil dan disimpan dalam working memori dengan memanfaatkan fasilitas TList. Lihat gambar 4.14.



Gambar 4.14 cara penyimpanan rule

Proses penyimpanan setelah dilakukan perubahan item-item dalam TList bisa dilihat pada prosedur-prosedur berikut ini:

*) Untuk mengambil file berisi Rule dan disimpan dalam working memori :

```
.....
while not TableData.eof do

begin
  New(Rule);
  Rule^.RuleNum := TableDataNUMBER.value;
  Rule^.clauseNum := TableDataNUMCLAUSE.value;
```

```
Rule^.IFVar := TableDataIFCLAUSE.value;  
Rule^.IFSymbol := TableDataIFSMBOL.value;  
Rule^.IFValue := TableDataIFVALUE.value;  
Rule^.THENVar := TableDataTHENCLAUSE.value;  
Rule^.THENSymbol := TableDataTHENSYMBOL.value;  
Rule^.THENValue := TableDataTHENVALUE.value;  
VarList.Add(Rule);  
TableData.next;  
end;
```

....

4.6 IMPLEMENTASI PENCARIAN DAN PENYIMPANAN DATA

Data rule yang ada secara permanen disimpan didalam sebuah tabel. Sedangkan untuk proses inferensi data yang berada dalam tabel tersebut diambil dan disimpan dalam list dengan menggunakan fasilitas Tlist. Penyimpanan data secara permanen pada tabel dilakukan untuk menjaga agar Aturan yang telah dimasukkan dan diedit oleh pemakai, ketika sistem dimatikan tidak hilang bersamaan dengan penghapusan alokasi memori yang diberikan saat penggunaan Tlist. Pada saat proses inferensi digunakan Tlist agar memori yang dipergunakan saat proses inferensi seminim mungkin yaitu disesuaikan dengan kebutuhan inferensi. Berikut ini adalah beberapa prosedur dalam sistem berbasis aturan yang dibangun untuk perencanaan produksi jagung dalam industri pertanian.

1. Proses Pencarian dan penyimpanan data dalam forward chaining

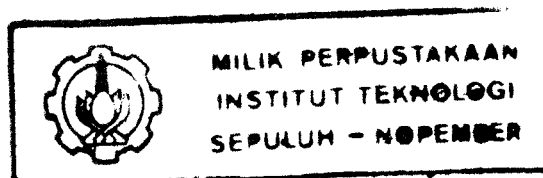
- *) Fungsi untuk mengecek apakah input yang diberikan oleh user yang disimpan dalam inputlist sama dengan data yang dimiliki. Memberikan nilai true apabila sama, dan false bila tidak

```
FunctionCheckInputSign(dataVar:string;dataSym:shortStr;dataVal:string):boolean;

var i:byte;
begin
  CheckInputSign := false;
  for i:=0 to InputList.count-1 do
  begin
    input1 := InputList.items[i];
    if(dataVar=input1^.VarCond) and (dataSym=input1^.varSym) and (dataVal=
input1^.varVal) then CheckInputSign := true;
  end;
end;
```

- *) Prosedur untuk mencari rule dalam VarList dengan menggunakan rule number sebagai kunci pencarian, bila ada yang sama maka rule tersebut disimpan dalam TempList.

```
Procedure CheckAllIF(num : integer);
var i:integer;
begin
  for i:=0 to VarList.count-1 do
  begin
    Rule := Varlist.items[i];
    if(Rule^.Rulenum=num) then
    begin
      new(tempTH);
      TempTH^.posRule := i;
      TempTH^.variable := Rule^.IFVar;
      TempTH^.symbol := Rule^.IFSymbol;
      TempTH^.value := Rule^.IFValue;
      TempList.Add(TempTH);
    end;
  end;
end;
```



```

    end;
  end;
end;

```

*) Fungsi untuk mengecek semua isi dari InputList dengan yang terdapat dalam TempList bila sama berikan Sign True. Bila tidak sama. Cek apakah benar kondisi yang ada dalam TempList sesuai dengan kenyataan? Jika tidak maka hapus isi TempList. Jika ya, maka masukkan items dari TempList tersebut kedalam InputList.

```

function CheckInput:integer;
var i,j: integer;
    cek ,cektotal: boolean;
    k,jml : byte;
begin
    jml := 0;
    cektotal := true;
    for i:=0 to TempList.count-1 do
    begin
        cek := false;
        TempTH:= TempList.items[i];
        for j:=0 to InputList.count-1 do
        begin
            input1 := InputList.items[j];

            if(CheckInputSign(TempTH^.variable,TempTH^.symbol,TempTH^.value)=true)
            then begin
                NumIF := j;
                input1^.sign:= true;
                cek := true;
            end;
            if(cek=false)then
            begin
                if MessageDlg('Apakah '+ (TempTH^.variable)+'
                '+ (TempTH^.symbol)+' '+'+(TempTH^.value)+' ?',
                mtInformation, [mbYes, mbNo], 0) = mrNo
                then begin
                    if(jml>0)then
                    begin
                        for k:=0 to jml-1 do
                        begin

```

```

FowForm.ListBoxCondition.items.delete(inputList.count-1);
    InputList.delete(InputList.count-1);
    jml := jml-1;
    end;
    cehtotal := false ;
    end;
    for k:=Templist.count-1 downto 0 do
        Templist.delete(0);
        CheckInput := -1;
        exit;
    end
    else begin
        new(input1);
        input1^.varcond:=TempTH^.variable;
        input1^.varsym:=TempTH^.symbol;
        input1^.varval:=TempTH^.value;
        input1^.sign := true;
        inputList.add(input1);
        FowForm.ListBoxCondition.items.add((TempTH^.variable)+'
'+(TempTH^.symbol)+' '(TempTH^.value));
        jml := jml+1;
    end;
    end;
    end;

    end;
    if cehtotal=true then CheckInput := TempTH^.posRule;
    if cehtotal=false then CheckInput := -1;
    for i:=Templist.count-1 downto 0 do
        Templist.delete(0);
    end;
end;

```

*) Prosedur untuk mencari if dan meletakkan bagian then dari if yang ketemu dalam conclusion combo box.

```

Procedure TFowForm.searchIF;
var i,j:byte;
    ok : boolean;
    number : integer;
begin
    for i:=0 to VarList.count-1 do
        begin
            rule := VarList.items[i];

```



```

if
(rule^.IFVar=CompVar) and (rule^.IFSymbol=CompSym) and (rule^.IFValue=
CompVal)
then begin
number := Rule^.RuleNum;
new(HW);
HW^.NumRule := Rule^.RuleNum;
HWList.add(HW);
CheckAllIF(number);
if (TempList.count=1) then
begin
input1 := InputList.items[NumIF];
input1^.sign := true;
TempList.delete(0);
Rule := VarList.items[checkinput];
New(conclu);
Conclu^.result := rule^.THENVar+' '+rule^.THENSymbol+'
'+rule^.THENValue;
ConcluList.add(conclu);
SearchIF(rule^.THENVar, rule^.THENSymbol, rule^.THENValue);
end
else if (TempList.count>1) then
begin
if (CheckInput>=0) then
begin
Rule := Varlist.items[checkinput];
New(conclu);
Conclu^.result := rule^.THENVar+' '+rule^.THENSymbol+'
'+rule^.THENValue;
ConcluList.add(conclu);
SearchIF(rule^.THENVar, rule^.THENSymbol, rule^.THENValue);
end
end;
end;
end;
end;

```

*) Prosedur untuk menjalankan kondisi yang telah diinputkan oleh pemakai dengan menggunakan forward chaining. Sebelumnya diawali dengan mengecek apakah ada rule dalam sistem. Kemudian mengecek apakah ada inputan kondisi dari user.

```

procedure TFowForm.RunBitBtn1Click(Sender: TObject);
var i :byte;
    valid : boolean;

```

```

begin
  If (Rule=nil)then
  begin
    MessageDlg('There is no Rule in the system, please cek in View
All Rule in main menu, thank you!', mtInformation,
    [mbOk], 0);
    Close;
  end

  else if (input1=nil)then
  begin
    MessageDlg('Enter Input Condition!', mtInformation,
    [mbOk], 0);
    Close;
  end
  else begin
    NumIF := 0;
    valid :=true;

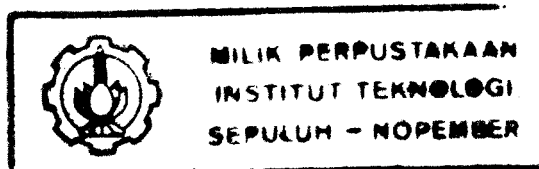
    for i:= 0 to InputList.count-1 do
    begin
      input1 := InputList.items[i];
      ListBoxCondition.items.add(input1^.varcond+' '+input1^.varsym+'
'+input1^.varval);
    end;

    input1 := InputList.items[NumIF];
    SearchIF(input1^.varcond,input1^.varsym,input1^.varval);

    for i:=0 to InputList.count-1 do..
    begin
      input1 := InputList.items[i];
      if(input1^.sign = false)then valid := false;
    end;

    if (valid=false)then
    begin
      MessageDlg('Tidak ada kesimpulan!', mtInformation,
      [mbOk], 0);
      while(InputList.count>0)do
      begin
        input1 := InputList.items[0];
        ListBoxCondition.items.delete(0);
        InputList.delete(0);
      end;
      exit;
    end
    else begin
      for i:=0 to ConcluList.count-1 do
      begin
        conclu := ConcluList.items[i];

```



```

        ListBoxConclution.items.add(conclu^.result);
end;
    end;
end;
clearbitbtn1.enabled := true;

for i:=0 to ConcluList.count-1 do
begin
    conclu := ConcluList.items[i];
    HowComboBox.items.add(conclu^.result)
end;
for i:=0 to ConcluList.count-2 do
begin
    conclu := ConcluList.items[i];
    WhyComboBox.items.add(conclu^.result)
end;
end;
end;

```

*) Prosedur untuk menginputkan kondisi untuk diproses secara forward chaining.

```

....
New(input1);
input1^.varcond := Variable.Text;
input1^.varsym  := Symbol.Text;
input1^.varval  := Value.text;
input1^.sign   := false;
InputList.add(input1);
.....

```

*) Prosedur untuk menyelesaikan permasalahan "How"

```

procedure TFowForm.HowBitbtn1Click(Sender: TObject);
var i: integer;
temp : string;
numTemp : integer;
begin
    temp := '';
    for i:=0 to VarList.count-1 do
    begin
        Rule := VarList.items[i];
        if (HowComboBox.text=Rule^.THENVar+' '+Rule^.THENSymbol+'
'+Rule^.THENValue)
        then begin
            numtemp := Rule^.RuleNum;

```

```

    temp := temp + ' '(Rule^.IFVar)+' '(Rule^.IFSymbol)+'
'+(Rule^.IFValue)+',';
    end;
end;
    MessageDlg(dengan mengetahui bahwa '+temp+' We conclude that
'+HowComboBox.text+' (Rule '+ (IntToStr(Numtemp))+')', mtInformation,
    [mbOk], 0);
    exit;
end;

```

***) Prosedur untuk menyelesaikan permasalahan "Why"**

```

procedure TFowForm.WhyBitbtn1Click(Sender: TObject);
var i: integer;
begin
    for i:=0 to VarList.count-1 do
    begin
        Rule := VarList.items[i];
        if(WhyComboBox.text=Rule^.IFVar+' '(Rule^.IFSymbol)+'
'+Rule^.IFValue)
        then begin
            MessageDlg('Kami menyimpulkan bahwa'+WhyComboBox.text+' karena
'+(Rule^.THENVar)+' '(Rule^.THENSymbol)+' '(Rule^.THENValue)+'
(Rule '+ (IntToStr(Rule^.RuleNum))+')', mtInformation,
                [mbOk], 0);
            exit;
        end;
    end;
end;
end;

```

2. Proses Pencarian dan penyimpanan data dalam backward chaining

***) Fungsi untuk mengecek keberadaan beberapa if dari suatu then.**

```

function
CheckMultiIF(compvar:string;compsym:ShortStr;compVal:string):boolean;
var i:integer;
    cek : boolean;
begin
    cek := true;
    for i:=0 to VarList.count-1 do
    begin
        Rule := Varlist.items[i];

        if(Rule^.THENVar=compvar) and (Rule^.THENSymbol=compsym) and (Rule^.THENValue=co
mpval)
        then begin
            New(tempTH);
            tempTH^.variable := Rule^.IFVar;

```

```

tempTH^.symbol := Rule^.IFSymbol;
tempTH^.value := Rule^.IFValue;
TempList.add(tempTH);
if MessageDlg('Is '+ (Rule^.IFVar) + ' '+ (Rule^.IFSymbol) +
'+(Rule^.IFValue)+' ?',
mtInformation, [mbYes, mbNo], 0) = mrNo
then cek := false;
end;
end;
if (cek=false)then CheckMultiIF := false
else CheckMultiIF := true;
end;

```

*) Prosedur untuk mencari bagian THEN dari suatu IF yang berkaitan dengan THEN yang dimasukkan oleh user.

```

procedure TBackwardForm.SearchTHEN;
var i,j,t : integer;
begin
for i:=0 to VarList.count-1 do
begin
Rule := VarList.items[i];
if (Rule^.THENVar=CompVar) and (Rule^.THENSymbol=CompSym) and (Rule^.THENValue=CompVal)
then begin
input1 := InputList.items[NumIF];

new(HW);
HW^.NumRule := Rule^.RuleNum;
HW^.IFVar := Rule^.IFVar;
HW^.IFSymbol := Rule^.IFSymbol;
HW^.IFValue := Rule^.IFValue;
HW^.THENVar := Rule^.THENVar;
HW^.THENSymbol := Rule^.THENSymbol;
HW^.THENValue := Rule^.THENValue;
HWList.add(HW);

if (TempList.count>0) then
begin
for t:=TempList.count-1 downto 0 do
begin
TempTH := TempList.items[t];

if (TempTH^.variable=CompVar) and (TempTH^.Symbol=CompSym) and (TempTH^.Value=CompVal)
then TempList.delete(t);
end;
end;
if (i=0) then j:=0
else j:= i-1;
Rule := VarList.items[j];
LastRuleNum := Rule^.RuleNum;
Rule := VarList.items[i];
if (Rule^.RuleNum<>LastRuleNum) then
begin
ok := CheckMultiIF(Rule^.THENVar, Rule^.THENSymbol, Rule^.THENValue);
if (ok=false) then
begin

```

```

        New(conclu);
        Conclu^.result := 'You are wrong,
'+(input1^.VarCond)+'<>'+(input1^.VarVal);
        ConcluList.add(conclu);
        exit;
    end;
    for j:= 0 to TempList.count-1 do
    begin
        TempTH := TempList.items[j];
        SearchTHEN(TempTH^.variable, tempTH^.symbol, tempTH^.value);
    end;
    end;
end;
end;
end;

```

*) Prosedur untuk menjalankan backward chaining berdasarkan inputan bagian THEN yang diberikan oleh user.

```

procedure TBackwardForm.RunBitbtn1Click(Sender: TObject);
var i : byte;
begin
    If (Rule=nil)then
    begin
        MessageDlg('There is no Rule in the system, please cek in View All Rule
in main menu, thank you!', mtInformation,
        [mbOk], 0);
        Close;
    end
    else
    begin
        New(input1);
        input1^.Varcond := Variable.text;
        input1^.VarSym := Symbol.text;
        input1^.VarVal := Value.text;
        input1^.sign := false;
        InputList.add(input1);

        ok := true;
        SearchTHEN(input1^.Varcond, input1^.VarSym, input1^.VarVal);

        if(ConcluList.count=0)then
        begin
            if(TempList.count=0)then
            begin
                MessageDlg('Not in Rules. No Conclusion!', mtInformation,
                [mbOk], 0);
                while(InputList.count>0)do
                begin
                    input1 := InputList.items[0];
                    InputList.delete(0);
                end;
                variable.text := '';
                symbol.text := '';
            end;
        end;
    end;
end;

```

```

    value.text := '';
    exit;
end
else begin
    New(conclu);
    Conclu^.result := 'You are right,
'+(input1^.VarCond)+(input1^.VarSym)+(input1^.VarVal);
    ConcluList.add(conclu);
    for i:=0 to ConcluList.count-1 do
    begin
        conclu := ConcluList.items[i];
        Conclution.text := (conclu^.result);
    end;
end;
end
else begin
    for i:=0 to ConcluList.count-1 do
    begin
        conclu := ConcluList.items[i];
        Conclution.text := (conclu^.result);
    end;
end;
clearbitbtn2.enabled := true;
for i:=1 to HWList.count-1 do
begin
    HW:= HWList.items[i];
    HowComboBox.items.add(HW^.THENVar+' '+HW^.THENSymbol+' '+HW^.THENValue);
end;
for i:=0 to HWList.count-1 do
begin
    HW:= HWList.items[i];
    WhyComboBox.items.add(HW^.IFVar+' '+HW^.IFSymbol+' '+HW^.IFValue);
end;
end;
end;

```

*) Prosedur untuk menyelesaikan permasalahan "How" dalam Backward

Chaining.

```

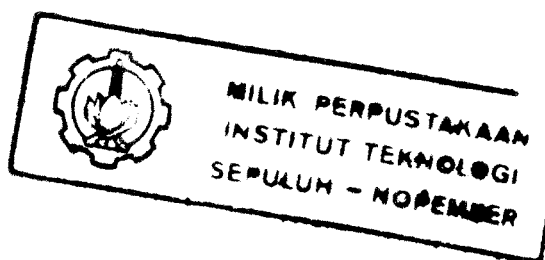
procedure TBackwardForm.HowBitbtn4Click(Sender: TObject);
var i:byte;
    temp : string;
begin
    temp := '';
    for i:=0 to HWList.count-1 do
    begin
        HW := HWList.items[i];
        if(HowComboBox.text= HW^.THENVar+' '+HW^.THENSymbol+'
'+HW^.THENValue)
        then
            temp := temp +HW^.IFVar+' '+HW^.IFSymbol+' '+HW^.IFValue+', ';
    end;
end;

```

```
MessageDlg('By knowing that '+temp+' We conclude that  
'+HowComboBox.text , mtInformation,  
  [mbOk], 0);  
  exit;  
end;
```

*) Prosedur untuk menyelesaikan permasalahan "Why" dalam Backward Chaining.

```
procedure TBackwardForm.WhyBitbtn5Click(Sender: TObject);  
var i:byte;  
begin  
  for i:=0 to HWList.count-1 do  
  begin  
    HW := HWList.items[i];  
    if(WhyComboBox.text= HW^.IFVar+' '+HW^.IFSymbol+' '+HW^.IFValue)  
    then begin  
      MessageDlg('We conclude that '+WhyComboBox.text+' because  
'+HW^.THENVar+' '+HW^.THENSymbol+' '+HW^.THENValue , mtInformation,  
        [mbOk], 0);  
      exit;  
    end;  
  end;  
end;  
end;
```



3. Proses pengolahan data harga

- *) Prosedur untuk menghasilkan nilai a dan b dalam persamaan linear untuk data harga yang diinputkan oleh pemakai.

```

procedure TForm2.BtnLinearClick(Sender: TObject);
var
  a,b,xy : integer;
  JumlahData,xtotal,ytotal,totalxy,xkuadrat,totalxkuadrat : integer;
begin
  xtotal := 0;
  ytotal := 0;
  totalxy := 0;
  totalxkuadrat := 0;
  JumlahData := 0;
  Table1.first;
  while not Table1.eof do
    Begin
      JumlahData := JumlahData+1;
      xtotal := xtotal+Table1tahun.value;
      ytotal := ytotal+Table1harga.value;
      xy := Table1Tahun.value*Table1harga.value;
      totalxy := totalxy + xy;
      xkuadrat := table1Tahun.value*table1Tahun.value;
      totalxkuadrat := totalxkuadrat + xkuadrat;
      Table1.next;
    end;
  b:= ((JumlahData*totalxy)-(ytotal*xtotal))div((JumlahData*totalxkuadrat)-(
  xtotal*xtotal));
  a:= ((ytotal div JumlahData)-b*(xtotal div JumlahData));

  edita.Text := inttostr(a);
  editb.text := inttostr(b);

end;

```

- *) Prosedur untuk menentukan kondisi fluktuasi harga untuk data harga yang diinputkan oleh pemakai.

```

procedure TForm2.BitBtnTkHarggaClick(Sender: TObject);
var
  y,a,b : integer;
  stabil, low, high : integer;

begin
  stabil := 0;
  low := 0;
  high := 0;
  a := strtoint(edita.text);

```

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini akan diuraikan tentang hasil uji coba perangkat lunak dengan berbagai kasus perencanaan produksi jagung yang ada serta evaluasi dan pembahasannya. Pengambilan contoh tersebut diambil dari informasi pakar.

5.1 KASUS PERTAMA

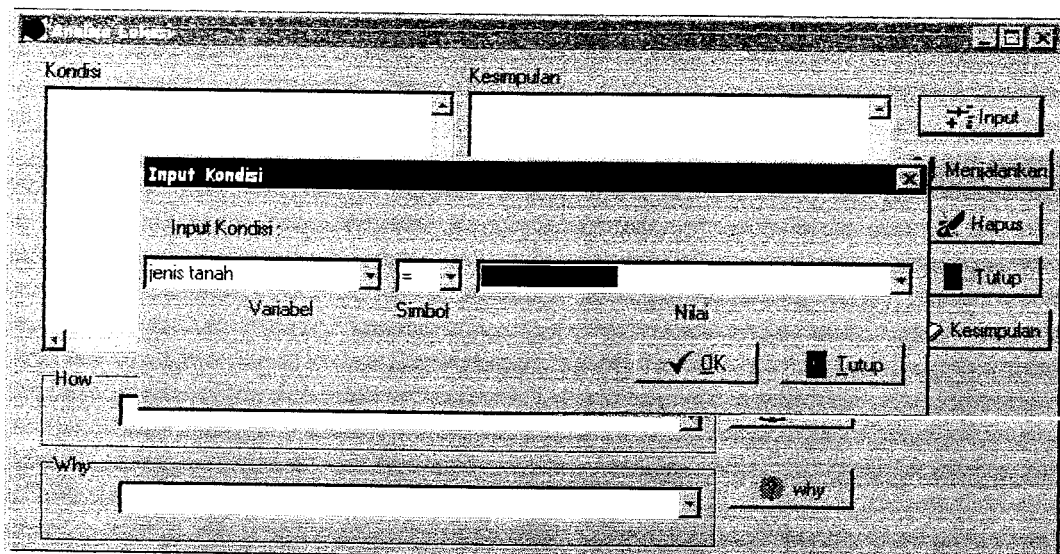
Untuk kasus pertama akan diambil kasus perencanaan produksi jagung dengan diketahui salah satu faktor topoklimatologi misalnya "jenis tanah = lempung berdebu", sedangkan faktor topoklimatologi yang lain , yaitu jika diketahui Ketinggian tempat < 1500 m, dan curah hujan antara 1500 - 2500 mm/th akan dijawab pemakai dalam proses konsultasi. Maka proses-proses yang terjadi di dalam sistem adalah sebagai berikut:

1. Proses Pengisian Data
2. Proses Konsultasi
3. Pemberian informasi mengapa saran tersebut diberikan, dan bagaimana bisa disarankan.

Masing-masing proses bisa dilihat dari hasil *running* program sebagai berikut:

1. *Proses Pengisian Data*

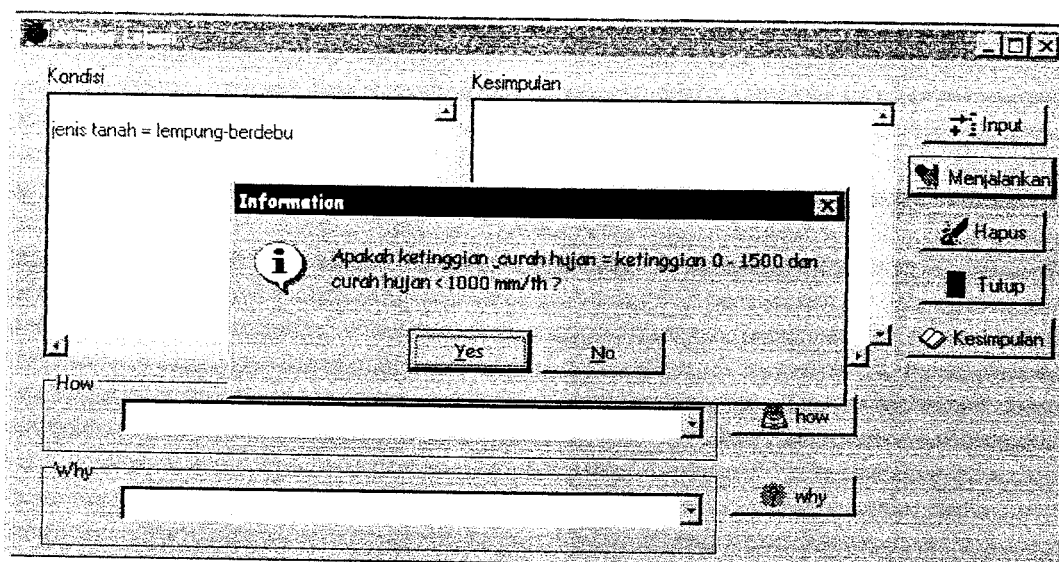
Dalam lembar data ini, pemakai dapat memasukkan beberapa data yang diketahuinya, dengan memilih pada combobox atribut dan value dari fakta yang diketahui, kemudian secara berkala sistem menanyakan beberapa hal yang perlu diketahui oleh sistem. Dalam menginputkan kondisi disini pemakai hanya menginputkan satu buah kondisi yaitu "jenis tanah = lempung berdebu" untuk kondisi lain dimasukkan pada sesi konsultasi. Lembar tersebut disajikan dalam gambar 5.1 berikut ini:



Gambar 5.1 Lembar Data

2. Proses Konsultasi

Setelah lembar data diisi dengan benar, maka sistem akan mengisi listbox 'condition' dengan fakta yang telah dimasukkan oleh pemakai dan membawa pemakai pada lembar konsultasi, sebagai berikut ini (gambar 5.2):

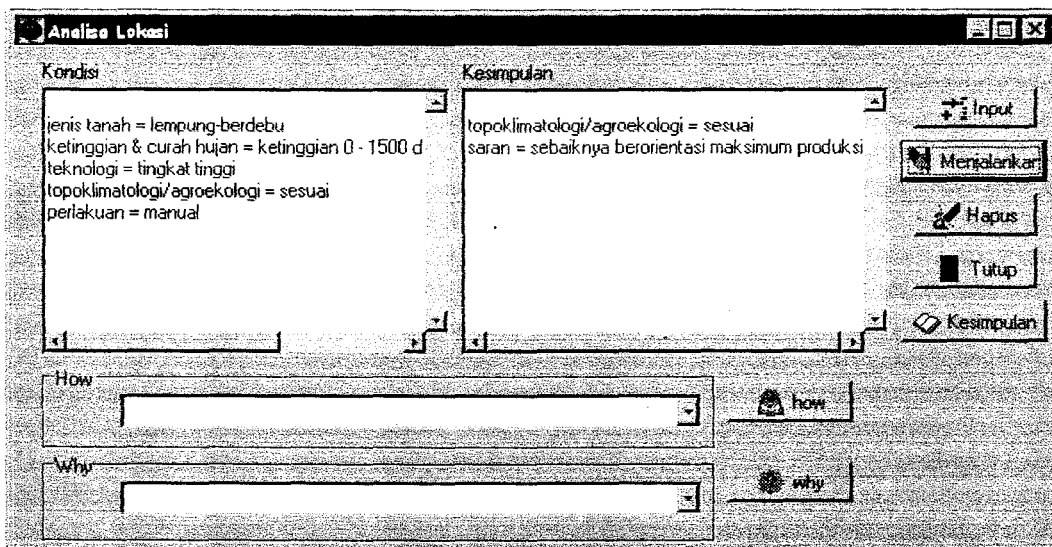


Gambar 5.2 Lembar Pertanyaan

Proses ini akan menunggu respon dari pemakai. Untuk kasus ini maka pemakai akan menjawab "tidak" untuk pertanyaan "Apakah ketinggian curah hujan = ketinggian 0 - 1500 m dan curah hujan < 1000 mm/th?". Setelah pemakai memberikan respon, maka akan muncul pertanyaan berikutnya. "Apakah ketinggian curah hujan = ketinggian 0 - 1500 m dan curah hujan > 3500 mm/th?". Kemudian sistem akan melanjutkan dengan pertanyaan berikutnya "Apakah ketinggian curah hujan = ketinggian 0 - 1500 m dan curah hujan 1000 - 1500?" dan seterusnya. Jawab yes atau

no untuk pertanyaan yang muncul sesuai dengan fakta yang diketahui seperti diatas.

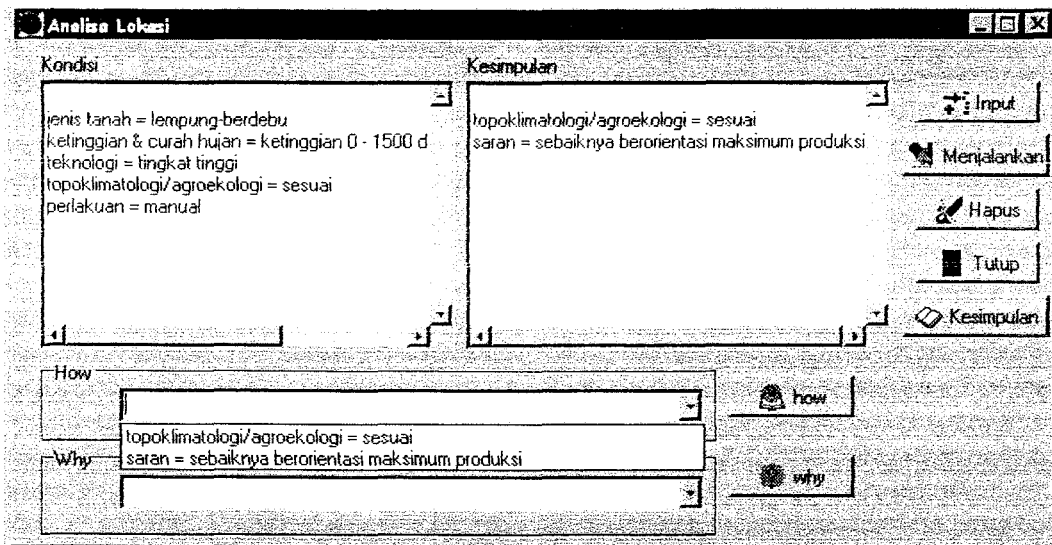
Kemudian setelah anda memberikan konfirmasi terhadap semua pertanyaan yang muncul sesuai dengan fakta yang dimiliki, maka akan dihasilkan kesimpulan seperti yang terlihat pada listbox 'conclution', dan seluruh input yang telah dikonfirmasi dengan pemakai ditampilkan pada listbox 'condition'. Sebagaimana dalam gambar 5.3 berikut ini:



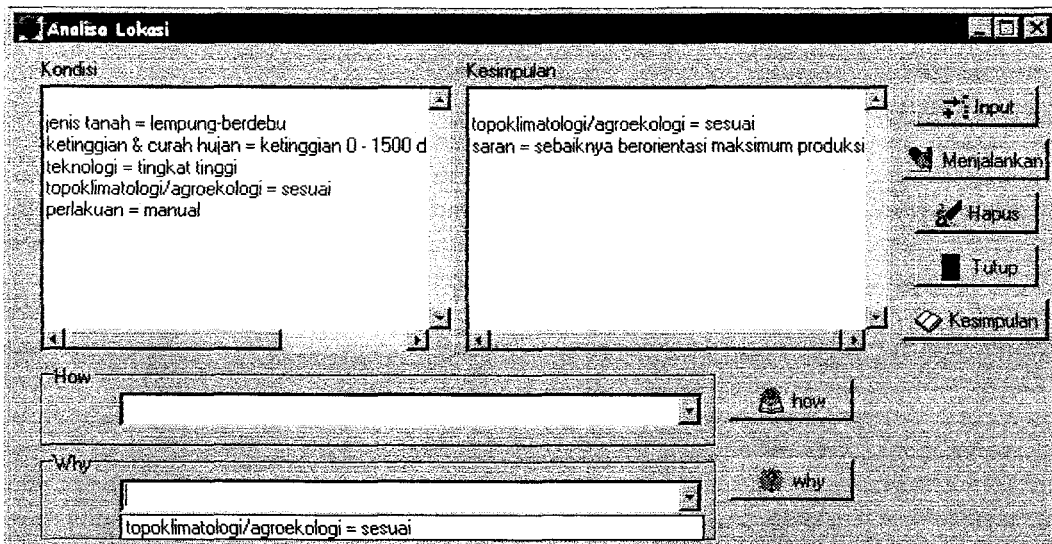
Gambar 5.3 Lembar kesimpulan

Dalam membantu menjelaskan kepada pemakai mengapa dan bagaimana fakta yang dimasukkan bisa dikategorikan seperti kesimpulan yang diberikan oleh sistem, pemakai dapat menggunakan bagian 'how' dan 'why' untuk mendapatkan penjelasan. Bersamaan dengan dihasilkannya kesimpulan, maka combobox 'how' dan combobox 'why' diisi sesuai

dengan kesimpulan yang dihasilkan, berikut ini adalah combobox 'how' dan combobox 'why' (lihat gambar 5.4 dan gambar 5.5).



Gambar 5.4 combobox 'How'

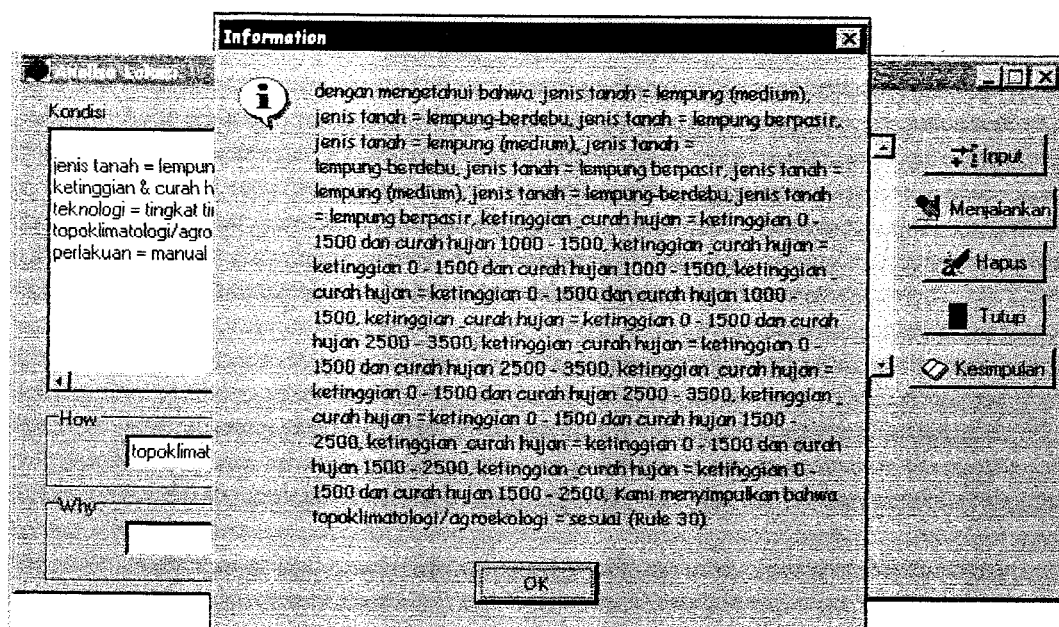


Gambar 5.5 combobox 'Why'

3. Pemberian informasi mengapa saran tersebut diberikan, dan bagaimana bisa disarankan

Dari gambar 5.3 terlihat bahwa fakta yang dimiliki adalah seperti yang terlihat dengan kesimpulan yang ada pada bagian

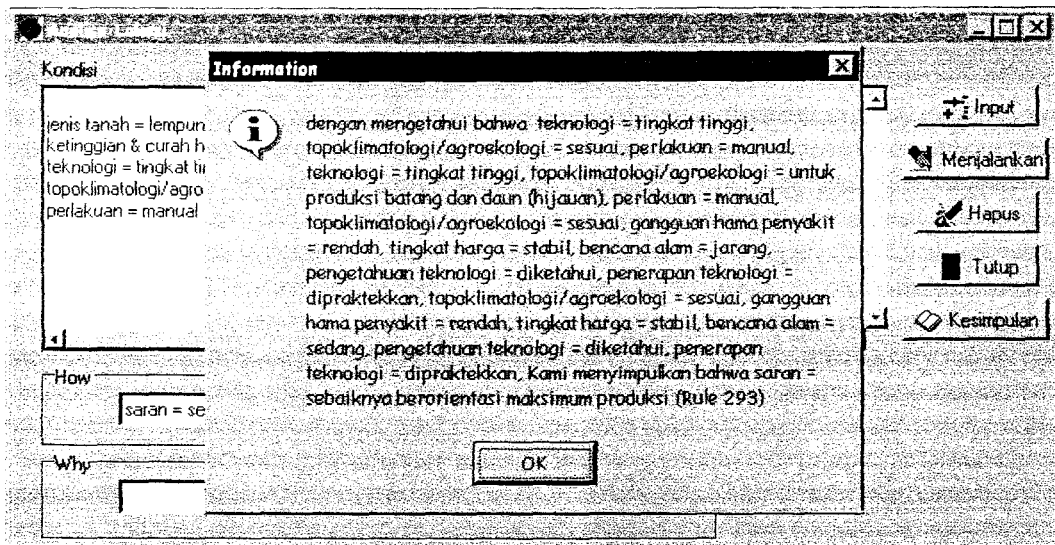
konklusi. Pemakai perlu mengetahui mengapa kondisi yang dimilikinya menghasilkan kesimpulan yang ada maka pemakai dapat menggunakan tombol "HOW" dan "WHY" dengan sebelumnya memilih kesimpulan yang ingin mendapat penjelasan untuk mengetahui alasan pengambilan kesimpulan tersebut. Pada combobox 'How' terdapat dua pilihan. Untuk memperoleh penjelasan bagaimana kondisi yang ada bisa disimpulkan sebagai tipoklimatologi/agroekologi untuk produksi batang dan daun (hijauan) pemakai dapat memilih pilihan pertama, sehingga sistem menampilkan gambar 5.6 yang menginformasikan kepada pemakai bagaimana kesimpulan tersebut diambil.



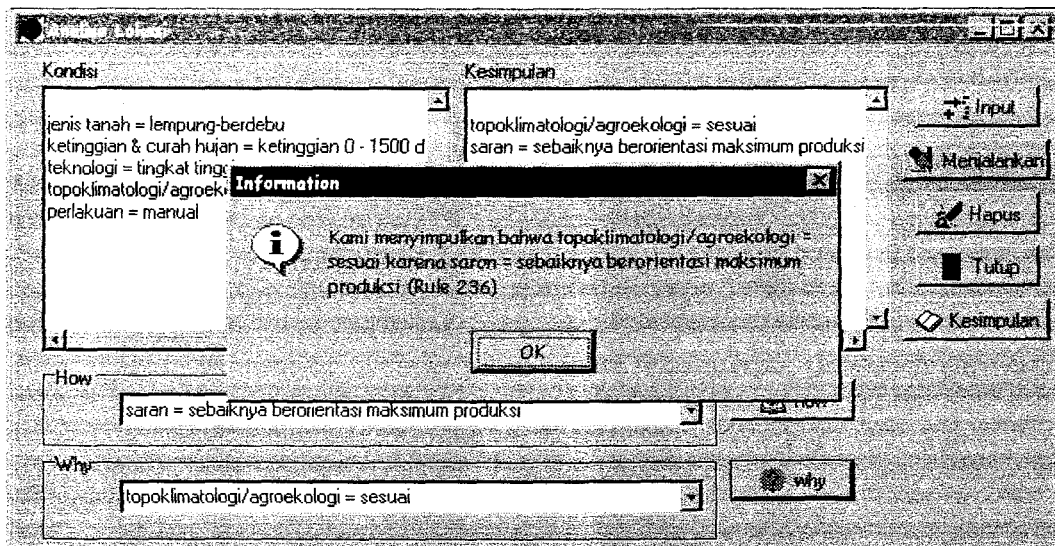
Gambar 5.6 Hasil How bagian pertama Kasus Pertama

Kemudian apabila ingin mengetahui bagaimana bisa disimpulkan saran = sebaiknya berorientasi maksimum produksi, maka pemakai bisa memilih

value dari combobox yang kedua sehingga ditampilkan seperti gambar 5.7.



Gambar 5.7 Hasil How bagian kedua Kasus Pertama



Gambar 5.8 Hasil Why Kasus Pertama

Hasil tersebut telah sama dengan hasil analisa pakar.

5.2 KASUS KEDUA (menguji kesimpulan yang diambil)

Kasus kedua merupakan pengujian kesimpulan yang dibuat terhadap kondisi yang ada. Dengan cara, user memasukkan kesimpulan yang dihasilkan, kemudian sistem mengujinya dengan mengajukan beberapa pertanyaan untuk pengecekan kesimpulan. Untuk menginputkan kesimpulan user dari program bisa dilihat pada gambar 5.8 berikut ini:

Gambar 5.8 Input Kesimpulan Kasus Kedua

Bila pemakai memberikan informasi fakta yang sesuai dengan kesimpulan tersebut, maka sistem akan membenarkan dan pemakai bisa mencari tahu mengapa kesimpulan dengan fakta yang dimasukkan oleh pemakai bisa dibenarkan oleh sistem, melalui tombol "How" dan "Why". Seperti yang terlihat dalam gambar 5.9:

Cek Aturan Terkait

Input

teknologi = tingkat tinggi

Variabel Simbol Nilai

Kesimpulan

Anda Benar, teknologi=tingkat tinggi

Jalankan

Kosongkan

Tutup

How

how

Why

why

Gambar 5.9 Kesimpulan Benar

Information

Anda: Dengan mengetahui bahwa pengetahuan teknologi = diketahui, penerapan teknologi = dipraktikkan, kami menyimpulkan bahwa teknologi = tingkat tinggi

OK

How

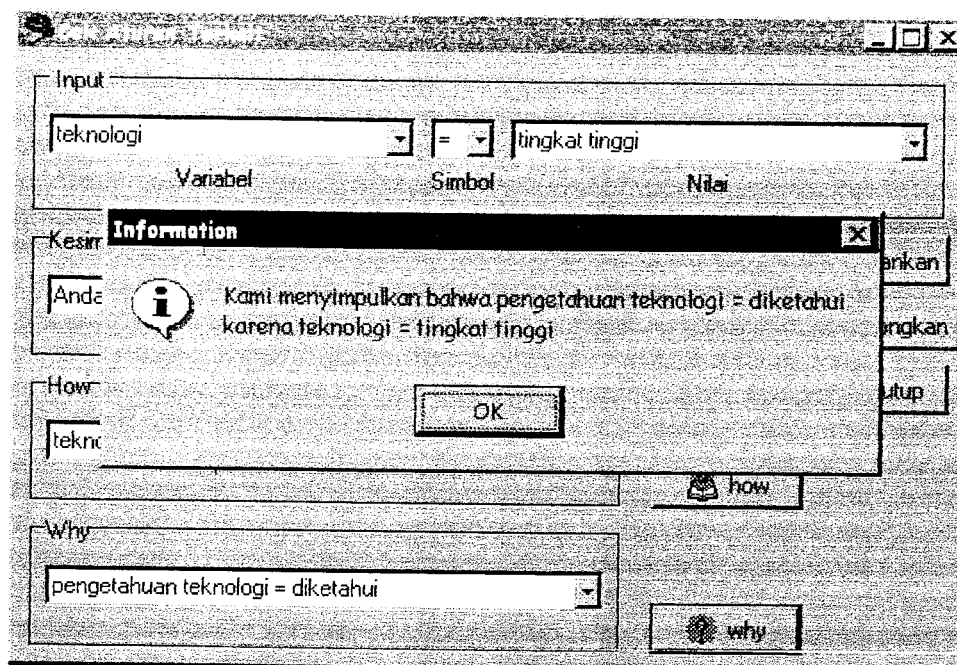
tekn

Why

why

Gambar 5.10 'How' pada bagian backward

Tetapi bila kesimpulan tidak sesuai dengan fakta yang dimasukkan maka sistem akan menyatakan bahwa kesimpulan yang diambil oleh pemakai adalah salah. Seperti halnya pada forward disini juga terdapat fungsi 'how' dan 'why', karena fungsinya untuk mengecek kebenaran rule yang dimasukkan. Hasilnya dapat dilihat pada gambar 5.10 dan gambar 5.11 berikut ini.



Gambar 5.11 'Why' pada bagian backward

Pada kasus pengecekan kesimpulan semacam kasus dua ini, seluruh rule yang berkaitan dengan rule yang akan dicek kebenarannya ditanyakan, namun apabila pemakai kurang teliti dan memberikan jawaban yang bukan seharusnya, meskipun hanya satu pertanyaan saja yang salah dijawab maka sistem akan memberikan kesimpulan salah untuk kesimpulan yang diambil oleh pemakai.

5.3. KASUS KETIGA (analisa harga)

Kasus ketiga merupakan kasus bila yang diketahui adalah data tentang harga produk jagung dengan selang bulan satuan harga per 100 kg jagung pipilan. Dengan menggunakan data yang bersumber dari BPS (Badan Pusat Statistik) Sidoarjo, sistem Perencanaan produksi Pertanian ini mencoba menganalisisnya dan memberikan saran sebaiknya apa yang perlu dilakukan bila dihadapkan pada kondisi tersebut.

Tahun	Jan	Peb	Mar	Apr	Mai	Jun	Jul	Ags	Sep	Okt	Nop	Des
1994	26999	29999	28499	28449	28499	28943	28943	28943	28943	28943	28943	28943
1995	31422	31837	33078	35146	35927	29770	41348	41348	41348	45483	45483	45483
1996	51108	49618	45483	45483	49618	49618	49618	49618	49618	41348	49618	49618
1997	49618	53753	53753	53753	46618	49618	51277	51686	53753	56233	56647	57888
1998	62022	62022	59641	66157	70291	82698	99237	124047	115773	107507	107507	107507

Tabel 5.1 Tingkat harga jagung pipilan di Sidoarjo 1994 - 1998¹

Dengan menggunakan data yang ada pemakai memasukkan data tersebut kedalam sistem melalui menu utama dengan mengklik 'data' kemudian pilih 'perkiraan'. Kemudian Pemakai memasukkan data tersebut dengan menyebutkan urutan data. Setelah semua data dimasukkan akan tampak seperti gambar 5.12.

¹ Sumber BPS (Badan Pusat Statistik) Sidoarjo

Informasi Harga

Harga	Periode
26999	1
29999	2
28499	3
28449	4
28499	5
28943	6
28943	7
28943	8
28943	9
28943	10
28943	11
28943	12
31422	13

Hapus

Perkiraan Tahun depan

Persamaan Line

a= b=

Perkiraan

Periode ke

Jumlah Perkiraan

Tingkat Harga

Tingkat Harga

Kesimpulan

Asumsi nilai rupiah

Point

Info Harga

Harga saat ini

Point

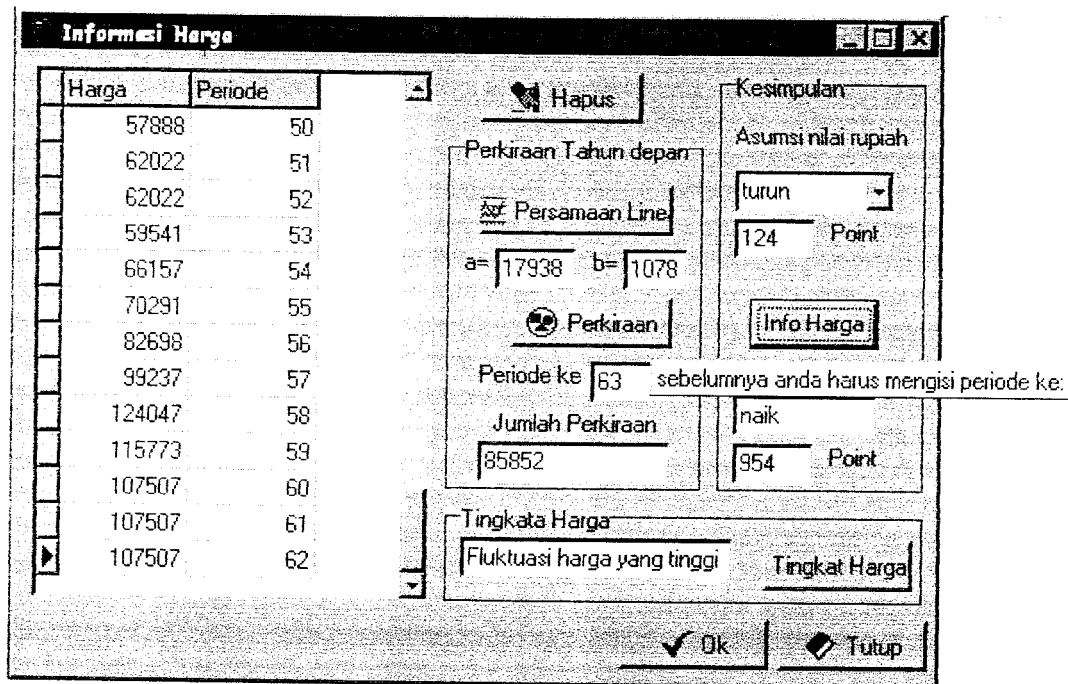
Ok

Tutup

Gambar 5.12 Input data harga

Kemudian pemakai harus mengklik tombol 'persamaan linear' untuk memperoleh persamaan umum dari data tersebut.

Apabila pemakai ingin mengetahui harga pada tahun dan bulan tertentu maka pemakai harap memasukkan urutan ke- dari harga yang ingin diketahui. Misalnya harga pada tahun 1999 bulan maret yaitu urutan ke 63. Kemudian pemakai harap menekan tombol 'perkiraan' maka hasilnya akan tampak pada jumlah perkiraan.

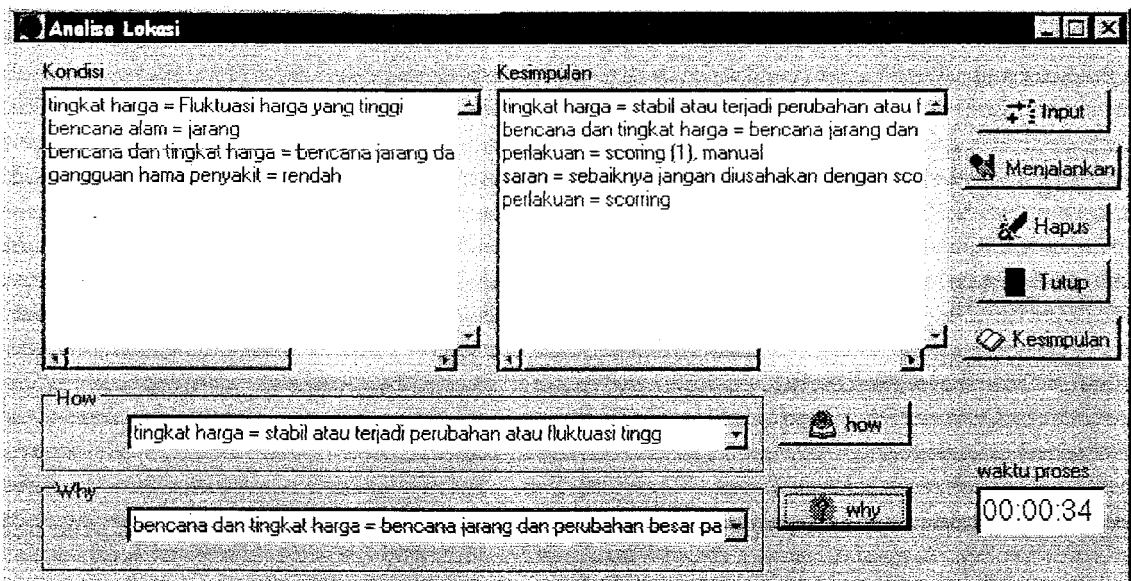


Gambar 5.13 Hasil dari pengolahan data harga

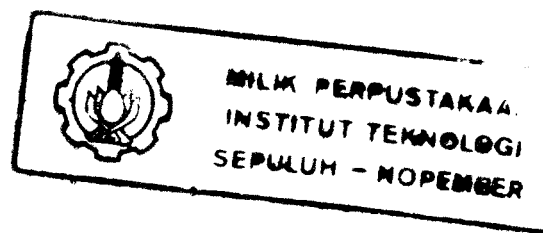
Apabila pemakai ingin mengetahui point (dalam Rp.) kenaikan dan penurunan harga pada tahun tertentu, dengan pemakai mengetahui kenaikan atau penurunan nilai rupiah pada saat yang ingin diketahui. Maka pemakai perlu memasukkan asumsi nilai rupiah (naik, stabil atau turun) dan besarnya point perubahan tersebut. Misalnya Nilai Rupiah 'Turun' dengan point = '124' point. Untuk mengetahui hasilnya pemakai perlu mengklik tombol info harga. Hasilnya akan tampak pada Gambar 5.13.

Apabila pemakai ingin mengetahui apa yang akan disarankan oleh sistem Perencanaan Produksi Pertanian apabila kondisi harga seperti yang dimasukkan oleh pemakai. Maka pemakai perlu menekan tombol 'Tingkat Harga'. Kemudian Tekan 'OK' untuk menyimpan informasi

tersebut, untuk dipergunakan dalam proses selanjutnya. Lalu buka forward Chaining. Tekan Tombol 'Input', saat muncul informasi 'Apakah anda ingin memasukkan fakta yang anda masukkan sebelumnya?' jawab 'Yes'. Kemudian tekan tombol 'Menjalankan'. Lanjutkan proses sesuai dengan yang anda ketahui sesuai dengan kondisi Resiko. Misalnya Bencana alam = jarang, gangguan hama = rendah. Maka akan diperoleh hasil seperti terlihat pada gambar 5.14.



Gambar 5.14 Saran dan Kesimpulan dari data harga dan berbagi faktor yang mempengaruhi saran tersebut yang dimasukkan

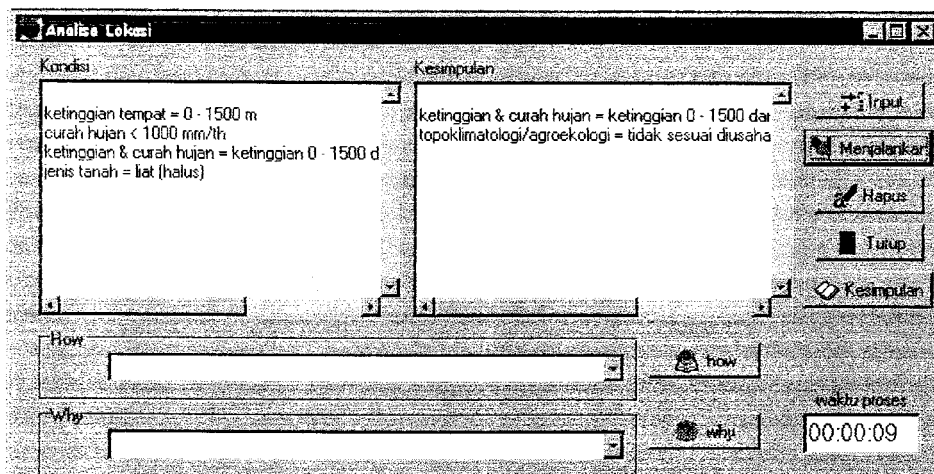


5.4. EVALUASI SISTEM

Untuk beberapa kasus yang telah ada dalam basis pengetahuan sistem ini dapat menyelesaikannya dengan baik. Sistem ini menyimpan rule dengan cara menyimpan per klausa. Sehingga jumlah yang mampu ditampung dalam satuan klausa bukan satuan rule. Sistem ini mampu menampung sebanyak 2.147.483.647 klausa. Satu rule dapat berisi satu atau lebih klausa.

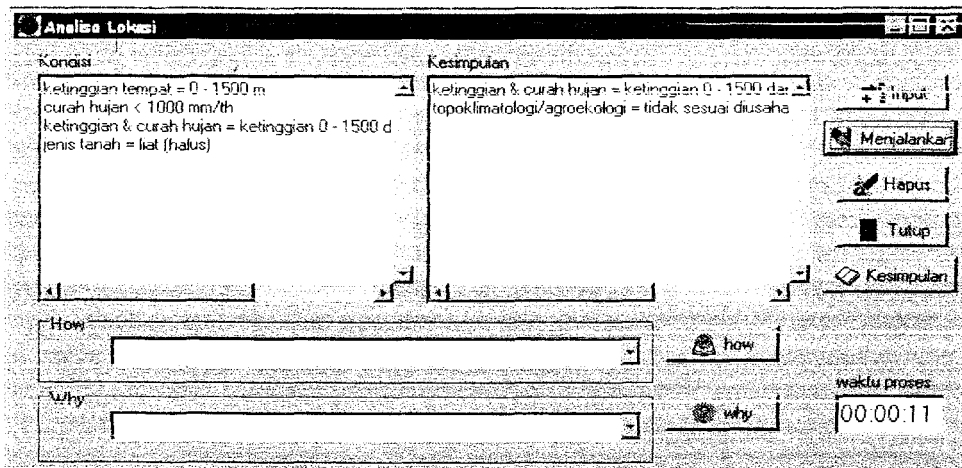
Waktu yang diperlukan untuk menyelesaikan suatu kasus pada sistem ini sangat tergantung pada:

- a. Jumlah rule yang terdapat dalam basis pengetahuan. Semakin kecil jumlah rule yang terdapat pada basis pengetahuan maka semakin kecil waktu yang diperlukan untuk proses. Contoh pada kasus A klausa yang digunakan sebanyak 664 klausa lihat gambar 5.15. dengan kasus yang sama tetapi jumlah klausa sebanyak 953 lihat pada gambar 5.16. Perhatikan waktu proses.

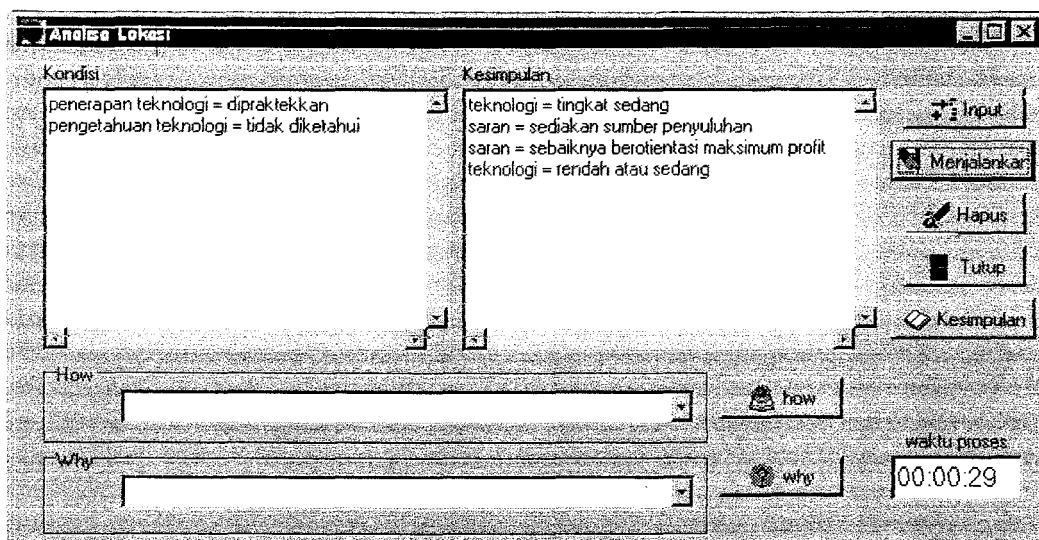


Gambar 5.15 hasil proses inferensi kasus A dengan 664 klausa

- b. Fakta yang dimasukkan oleh pemakai, semakin banyak rule yang memiliki klausa yang sama dengan fakta yang dimasukkan oleh pemakai maka semakin banyak waktu yang dibutuhkan untuk melakukan proses inferensi.



Gambar 5.16 hasil proses inferensi kasus A dengan 953 klausa



Gambar 5.17 hasil proses inferensi kasus B

Contoh: Dengan menggunakan 664 klausa dengan pemakai yang sama, sistem berusaha menyelesaikan 2 buah kasus. Kasus Kasus B diketahui "penerapan teknologi = tidak dipraktekkan" dalam basis

aturan fakta ini terkait dengan 25 Rule yang memiliki 145 klausa. Kasus A diketahui "ketinggian tempat = antara 0 - 1500 m" dalam basis aturan fakta ini terkait dengan 10 Rule yang memiliki 20 klausa. Hasil dari kedua proses inferensi untuk kedua kasus tersebut adalah seperti yang terlihat pada gambar 5.15 dan 5.17. Yang menunjukkan bahwa waktu yang diperlukan untuk kasus pertama lebih banyak karena rule yang terkait dengan fakta tersebut lebih banyak.

- c. Penguasaan pemakai terhadap fakta yang dimiliki sangat mempengaruhi waktu yang dibutuhkan dalam proses konsultasi. Padahal jawaban pada proses konsultasi yang sangat berperan terhadap proses inferensi. Sehingga semakin pemakai menguasai fakta yang dimilikinya maka semakin kecil waktu prosesnya.

Jadi bisa disimpulkan bahwa kondisi terburuk dari sistem berkaitan dengan waktu yang dibutuhkan untuk mengambil kesimpulan adalah ketika:

- a. Rule dalam basis aturan ada dalam jumlah besar
- b. Fakta yang dimasukkan dalam sistem berkaitan dengan banyak rule
- c. Pengguna yang tidak menguasai fakta yang dimiliki

Sedangkan kondisi terbaik dari sistem berkaitan dengan waktu yang dibutuhkan untuk mengambil kesimpulan adalah ketika:

- a. Rule dalam basis aturan ada dalam jumlah kecil, namun sebagai sistem berbasis aturan merupakan suatu hal yang normal memiliki banyak rule.
- b. Fakta yang dimasukkan dalam sistem hanya berkaitan dengan sedikit rule
- c. Pengguna menguasai fakta yang dimiliki

BAB VI

PENUTUP

Secara umum, Sistem Perencanaan Produksi Pertanian Berbasis Aturan ini telah berfungsi sebagaimana diharapkan. Namun tentu saja, masih perlu dilakukan beberapa perbaikan yang akan dipaparkan dalam pembahasan bab ini.

6.1. KESIMPULAN

Berdasarkan ujicoba yang dilakukan didapatkan beberapa kesimpulan dari Tugas Akhir ini adalah sebagai berikut:

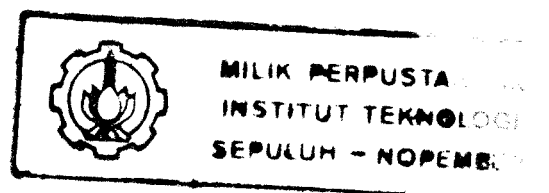
- a. Dalam Sistem Pakar ini apabila ada kasus-kasus yang belum ada di dalam sistem sebelumnya, maka seorang pakar perencanaan produksi pertanian jagung bisa menyempurnakannya. Sedangkan untuk kasus-kasus yang aturannya sudah ada sebelumnya berdasarkan hasil percobaan, sistem dapat menyelesaikannya sesuai dengan analisa pakar. Selain analisa terhadap kondisi, sistem pakar juga melakukan analisa terhadap harga yang bermanfaat sebagai bahan pertimbangan dalam merencanakan aktifitas produksi yaitu:
 - informasi harga pada periode tertentu berdasarkan data pemakai yang diolah dengan persamaan linear
 - informasi poin fluktuasi harga terhadap fluktuasi nilai rupiah
 - saran bila kondisi harga seperti yang di masukkan oleh pemakai.

- b. Dalam sistem ini pemakai dapat menambah dan mengedit aturan yang ada sehingga memungkinkan untuk memperbanyak kasus yang mampu diselesaikan oleh sistem. Selain fasilitas pengeditan kasus sistem pakar ini memberikan fasilitas informasi harga yang memungkinkan pemakai untuk memperoleh informasi berdasarkan data yang dimiliki tentang harga.
- c. Dengan mekanisme *reasoning* yaitu *forward chaining* dan *backward chaining* yang ditawarkan oleh sistem pemakai dapat memilihnya berdasarkan kebutuhan pemakai, yaitu:
- *Forward chaining* membantu pemakai untuk menganalisa fakta yang dimiliki dan mengategorikannya sesuai dengan aturan yang ada, serta memberikan saran kepada pemakai tentang apa yang sebaiknya dilakukan.
 - *Backward chaining* membantu pemakai untuk mengecek apakah kesimpulan yang diambil oleh pemakai sesuai dengan kesimpulan yang diambil sistem pada kondisi yang sama.
 - komponen penjelas 'How' dan 'Why' yang diharapkan mampu memberikan penjelasan kepada pemakai mengapa kondisi yang dimilikinya dikategorikan dalam kesimpulan tertentu.

6.2. KEMUNGKINAN PENGEMBANGAN LEBIH LANJUT

Adapun kemungkinan pengembangan terhadap Sistem Perencanaan Produksi Jagung dalam Industri Pertanian Berbasis Aturan ini adalah sebagai berikut:

- a. Sebaiknya dilakukan penambahan fasilitas yaitu:
 - disediakan sarana bagi pemakai untuk memberikan pertanyaan pada setiap rule, sehingga dengan fasilitas ini pemakai sistem pakar dapat lebih mudah dalam mengkonfirmasi fakta yang dimiliki pemakai.
 - Hasil akhir saran beserta kondisi yang dimiliki seharusnya bisa dilaporkan sebagai sebuah *report* untuk kemudian bisa diproses selanjutnya (misal untuk dicetak).
- b. Sistem Perencanaan Produksi ini dalam memperkirakan harga pada tahun tahun tertentu menggunakan regresi linear sederhana, sehingga tingkat validitasnya rendah. Apabila diketahui harga dari tahun ketahun pemakai dapat menggunakan perangkat lunak ini sebagai pertimbangan tingkat fluktuasi yang mungkin timbul. Untuk pengembangannya, masalah ini akan lebih baik jika menggunakan analisa harga dengan menggunakan metode analisa yang hasil analisisnya lebih mendekati kebenaran misalnya dengan menggunakan time series analisis. Yaitu sesuai dengan kondisi harga dengan kecenderungan fluktuasinya dalam skala waktu, karena metode ini adalah yang paling sesuai untuk menganalisa data kondisi harga dengan kecenderungan fluktuasinya dalam sekala waktu.



DAFTAR PUSTAKA

- [ADW-74] Prof. Ir. Anwas Adiwilaga; *Ilmu Usaha Tani*, Penerbit Alumni 1974.
- [AGR-99] Badan Agribisnis Departemen Pertanian, *Investasi Agribisnis Komoditas Unggulan Tanaman Pangan dan Hortikultura*, Kanisius 1999
- [CHR-90] Dimitris N. Chorafas, *Knowledge Engineering: Knowledge Acquisition, Knowledge Representation, the Role of the Knowledge Engineer, and Domains Fertile for Ai Implementation*, Van Nostrand Reinhold, 1990
- [GGC-94] Guida, Giovanni dan Carlo Tasso; *Design And Development Of Knowledge Based Systems*, John Wiley & Sons Ltd, 1994
- [HAR-89] Anna Hart; *Knowledge Acquisition for Expert Systems*, Kogan Page Ltd, 1989
- [IGN-91] Ignizio, James P, *Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems*, McGraw-Hill, Inc, 1991
- [LEG&DOH-89] William E. Leigh, Michael E. Doherty; *Decision Support and Expert Systems*, South-Western Publishing Co. 1986

- [PIF&BAR-85] D.V. Pigford, Greg Baur; *Expert System for Business Concepts and Applications*, boyd & fraser publishing company, 1985
- [RIC-91] Elaine Rich, Kevin Knight, *Artificial Inteligence, Second Edition*, Mc Graw-Hill, Inc., 1991
- [SOE-93] Soekartawi, Rusmadi, Effi Damijati. *Resiko dan ketidakpastian dalam Agribisnis, manajemen* PT. Rajawali Grafindo Persada Jakarta 1993
- [SPY&MGE-98] Spyros Makridakis, Steven C. Wheelwright dan Victor E. McGee; *Forecasting: Methods and Application, Secind Edition*, John Wiley & Sons, Inc., 1998
- [THO-83] Ir. Kaslan A. Thohir; *Bagian 2, edisi Pertama Bina Aksara*, Jakarta 1983.

LAMPIRAN A:

ATURAN

Pada lampiran ini dijelaskan tentang beberapa rule yang dimanfaatkan oleh sistem berbasis aturan yang dibangun. Satu rule dapat berisi sebuah *if clause* atau lebih, minimal sebuah *if clause*.

Berdasarkan pohon keputusan yang dibangun maka dapat dibangun rule, berikut ini adalah set rule yang telah dibangun. Rule 8 sampai dengan rule 30 merupakan Set Rule untuk Tipe Topoklimatologi, set rule tersebut diperoleh dari pohon keputusan seperti yang terlihat pada gambar 4.1 pada bagian 4.1.1.3. Untuk Rule 70 sampai dengan rule 73 merupakan set rule untuk Tipe Teknologi yang diperoleh dari indentifikasi atribut dan value pada bagian 4.1.1.2.

Untuk Rule saran pada bagian 4.1.1 tidak dibahas sama sekali maka dilampiran ini dapat dilihat bagaimana suatu saran dihasilkan berdasarkan dari rule yang ada.

Rule yang dipergunakan

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
8	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
8	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
9	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
9	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
10	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
10	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
13	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
13	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
14	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
14	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
15	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
15	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	diperbaiki dengan irigasi
16	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
16	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
17	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
17	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
18	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	sesuai

rule num	clau se num	if clause	if sy	if value	then clause	th cy	then value
18	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	sesuai
19	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	sesuai
19	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	sesuai
20	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	sesuai
20	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	sesuai
21	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
21	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
22	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
22	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
23	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	sesuai
23	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	sesuai
24	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	sesuai
24	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	sesuai
25	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	sesuai
25	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	sesuai
26	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
26	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
27	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif
27	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	sesuai dengan pengolahan tanah intensif

rule num	clau se num	if clause	if sy	if value	then clause	tr: sy	then value
28	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500	topoklimatologi/agro ekologi	=	sesuai
28	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	sesuai
29	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500	topoklimatologi/agro ekologi	=	sesuai
29	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	sesuai
30	1	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500	topoklimatologi/agro ekologi	=	sesuai
30	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	sesuai
41	1	bencana alam	=	sering	gangguan bencana alam	=	tinggi
41	2	tingkat harga	=	stabil atau terjadi perubahan atau fluktuasi tingg	gangguan bencana alam	=	tinggi
41	3	gangguan hama penyakit	=	rendah atau sedang atau tinggi	gangguan bencana alam	=	tinggi
70	1	penerapan teknologi	=	tidak dipraktekkan	teknologi	=	tingkat rendah
70	2	pengetahuan teknologi	=	tidak diketahui	teknologi	=	tingkat rendah
71	1	penerapan teknologi	=	dipraktekkan	teknologi	=	tingkat sedang
71	2	pengetahuan teknologi	=	tidak diketahui	teknologi	=	tingkat sedang
72	1	penerapan teknologi	=	tidak dipraktekkan	teknologi	=	tingkat sedang
72	2	pengetahuan teknologi	=	diketahui	teknologi	=	tingkat sedang
73	1	pengetahuan teknologi	=	diketahui	teknologi	=	tingkat tinggi
73	2	penerapan teknologi	=	dipraktekkan	teknologi	=	tingkat tinggi
83	1	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	lakukan perbaikan sarana irigasi
84	1	teknologi	=	tingkat rendah	saran	=	Sediakan sumber penyuluhan secara intensif
85	1	teknologi	=	tingkat sedang	saran	=	sediakan sumber penyuluhan
86	1	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	lakukan pengolahan tanah secara intensif
90	1	intensitas serangan hama	=	tinggi	saran	=	lakukan aplikasi pestisida tingkat intensif

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
91	1	intensitas serangan hama	=	sedang	saran	=	lakukan aplikasi pestisida tingkat menengah
92	1	tingkat harga	=	terjadi perubahan kecil	saran	=	lakukan penjadwalan produksi
100	1	teknologi	=	tingkat rendah	saran	=	sebaiknya berorientasi maksimum profit
101	1	teknologi	=	tingkat sedang	saran	=	sebaiknya berorientasi maksimum profit
102	1	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berorientasi maksimum profit
103	1	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berorientasi maksimum profit
104	1	intensitas serangan hama	=	tinggi	saran	=	sebaiknya berorientasi maksimum profit
105	1	intensitas serangan hama	=	sedang	saran	=	sebaiknya berorientasi maksimum profit
106	1	tingkat harga	=	terjadi perubahan kecil	saran	=	sebaiknya berorientasi maksimum profit
110	1	perlakuan	=	scoring (1), manual	saran	=	sebaiknya jangan diusahakan dengan score (1)
111	1	perlakuan	=	scoring (2), sedang	saran	=	sebaiknya jangan diusahakan dengan score (2)
112	1	perlakuan	=	scoring (3), manual	saran	=	sebaiknya jangan diusahakan dengan score (3)
113	1	perlakuan	=	scoring (4), sedang	saran	=	sebaiknya jangan diusahakan dengan score (4)
114	1	perlakuan	=	scoring (5), intensif	saran	=	sebaiknya jangan diusahakan dengan score (5)
115	1	perlakuan	=	scoring (6), intensif	saran	=	sebaiknya jangan diusahakan dengan score (6)
125	1	curah hujan	<	1000 mm/th	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th
125	2	ketinggian tempat	>	1500 m	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th
126	1	ketinggian tempat	>	1500 m	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th
126	2	curah hujan	>	3500 mm/th	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
127	1	ketinggian tempat	>	1500 m	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500
127	2	curah hujan	=	antara 1000 - 1500 mm/th	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500
128	1	ketinggian tempat	>	1500 m	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 2500 - 3500
128	2	curah hujan	=	antara 2500 - 3500 mm/th	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 2500 - 3500
129	1	ketinggian tempat	>	1500 m	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1500 - 2500
129	2	curah hujan	=	1500 - 2500 mm/th	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1500 - 2500
130	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
130	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
131	1	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
131	2	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
132	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
132	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
133	1	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
133	2	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
134	1	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
134	2	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan < 1000 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
135	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
135	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
136	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
136	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
137	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
137	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
138	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
138	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
139	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
139	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
141	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
141	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
142	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
142	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
143	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
143	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
144	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
144	2	jenis tanah	=	lempung-berdebu	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
145	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 1000 - 1500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
145	2	jenis tanah	=	lempung berpasir	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
146	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
146	2	jenis tanah	=	liat (halus)	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
147	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
147	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
148	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan 2500 - 3500	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)
148	2	jenis tanah	=	lempung (medium)	topoklimatologi/agro ekologi	=	untuk produksi batang dan daun (hijauan)

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
167	1	ketinggian & curah hujan	=	ketinggian > 1500 m dan curah hujan > 3500 mm/th	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
167	2	jenis tanah	=	pasir (kasar)	topoklimatologi/agro ekologi	=	tidak sesuai diusahakan jagung
175	1	bencana alam	=	sedang	bencana alam	=	jarang atau sedang
176	1	gangguan hama penyakit	=	tinggi	intensitas serangan hama	=	tinggi
177	1	gangguan hama penyakit	=	sedang	intensitas serangan hama	=	sedang
180	1	gangguan hama penyakit	=	tinggi	gangguan hama penyakit	=	rendah atau sedang atau tinggi
181	1	bencana alam	=	jarang	bencana alam	=	jarang atau sedang
191	1	ketinggian tempat	=	0 - 1500 m	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan < 1000 mm/th
191	2	curah hujan	<	1000 mm/th	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan < 1000 mm/th
192	1	ketinggian tempat	=	0 - 1500 m	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan > 3500 mm/th
192	2	curah hujan	>	3500 mm/th	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan > 3500 mm/th
193	1	ketinggian tempat	=	0 - 1500 m	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500
193	2	curah hujan	=	antara 1000 - 1500 mm/th	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1000 - 1500
194	1	ketinggian tempat	=	0 - 1500 m	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500
194	2	curah hujan	=	antara 2500 - 3500 mm/th	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 2500 - 3500
195	1	ketinggian tempat	=	0 - 1500 m	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500
195	2	curah hujan	=	antara 1500 - 2500 mm/th	ketinggian & curah hujan	=	ketinggian 0 - 1500 dan curah hujan 1500 - 2500
196	1	tingkat harga	=	stabil	tingkat harga	=	stabil atau terjadi perubahan atau fluktuasi tinggi
197	1	tingkat harga	=	terjadi perubahan kecil	tingkat harga	=	stabil atau terjadi perubahan atau fluktuasi tinggi
198	1	tingkat harga	=	Fluktuasi harga yang tinggi	tingkat harga	=	stabil atau terjadi perubahan atau fluktuasi tinggi
199	1	gangguan hama penyakit	=	rendah	gangguan hama penyakit	=	rendah atau sedang atau tinggi

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
200	1	gangguan hama penyakit	=	sedang	gangguan hama penyakit	=	rendah atau sedang atau tinggi
201	1	bencana alam	=	sedang	bencana dan tingkat harga	=	bencana sedang dan tingkat harga stabil
201	2	tingkat harga	=	stabil	bencana dan tingkat harga	=	bencana sedang dan tingkat harga stabil
202	1	bencana alam	=	sedang	bencana dan tingkat harga	=	bencana sedang dan perubahan kecil pada harga
202	2	tingkat harga	=	terjadi perubahan kecil	bencana dan tingkat harga	=	bencana sedang dan perubahan kecil pada harga
203	1	gangguan hama penyakit	=	sedang	bencana dan tingkat harga	=	bencana sedang dan perubahan besar pada harga
203	2	tingkat harga	=	Fluktuasi harga yang tinggi	bencana dan tingkat harga	=	bencana sedang dan perubahan besar pada harga
204	1	bencana alam	=	jarang	bencana dan tingkat harga	=	bencana jarang dan tingkat harga stabil
204	2	tingkat harga	=	stabil	bencana dan tingkat harga	=	bencana jarang dan tingkat harga stabil
205	1	bencana alam	=	jarang	bencana dan tingkat harga	=	bencana jarang dan perubahan kecil pada harga
205	2	tingkat harga	=	terjadi perubahan kecil	bencana dan tingkat harga	=	bencana jarang dan perubahan kecil pada harga
206	1	bencana alam	=	jarang	bencana dan tingkat harga	=	bencana jarang dan perubahan besar pada harga
206	2	tingkat harga	=	Fluktuasi harga yang tinggi	bencana dan tingkat harga	=	bencana jarang dan perubahan besar pada harga
210	1	bencana dan tingkat harga	=	bencana sedang dan tingkat harga stabil	perlakuan	=	manual
210	2	gangguan hama penyakit	=	rendah	perlakuan	=	manual
211	1	bencana dan tingkat harga	=	bencana sedang dan tingkat harga stabil	perlakuan	=	sedang
211	2	gangguan hama penyakit	=	sedang	perlakuan	=	sedang
212	1	bencana dan tingkat harga	=	bencana sedang dan tingkat harga stabil	perlakuan	=	intensif
212	2	gangguan hama penyakit	=	tinggi	perlakuan	=	intensif

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
213	1	bencana dan tingkat harga	=	bencana sedang dan perubahan kecil pada harga	perlakuan	=	penjadwalan, manual
213	2	gangguan hama penyakit	=	rendah	perlakuan	=	penjadwalan, manual
214	1	bencana dan tingkat harga	=	bencana sedang dan perubahan kecil pada harga	perlakuan	=	penjadwalan, sedang
214	2	gangguan hama penyakit	=	sedang	perlakuan	=	penjadwalan, sedang
215	1	bencana dan tingkat harga	=	bencana sedang dan perubahan kecil pada harga	perlakuan	=	penjadwalan, intensif
215	2	gangguan hama penyakit	=	tinggi	perlakuan	=	penjadwalan, intensif
221	1	bencana dan tingkat harga	=	bencana jarang dan tingkat harga stabil	perlakuan	=	manual
221	2	gangguan hama penyakit	=	rendah	perlakuan	=	manual
222	1	bencana dan tingkat harga	=	bencana jarang dan tingkat harga stabil	perlakuan	=	sedang
222	2	gangguan hama penyakit	=	sedang	perlakuan	=	sedang
223	1	bencana dan tingkat harga	=	bencana jarang dan tingkat harga stabil	perlakuan	=	intensif
223	2	gangguan hama penyakit	=	tinggi	perlakuan	=	intensif
224	1	bencana dan tingkat harga	=	bencana jarang dan perubahan kecil pada harga	perlakuan	=	penjadwalan, manual
224	2	gangguan hama penyakit	=	rendah	perlakuan	=	penjadwalan, manual
225	1	bencana dan tingkat harga	=	bencana jarang dan perubahan kecil pada harga	perlakuan	=	penjadwalan, sedang
225	2	gangguan hama penyakit	=	sedang	perlakuan	=	penjadwalan, sedang
226	1	bencana dan tingkat harga	=	bencana jarang dan perubahan kecil pada harga	perlakuan	=	penjadwalan, intensif
226	2	gangguan hama penyakit	=	tinggi	perlakuan	=	penjadwalan, intensif



rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
230	1	bencana dan tingkat harga	=	bencana sedang dan perubahan besar pada harga	perlakuan	=	scoring (3), manual
230	2	gangguan hama penyakit	=	rendah	perlakuan	=	scoring (3), manual
231	1	bencana dan tingkat harga	=	bencana sedang dan perubahan besar pada harga	perlakuan	=	scoring (4), sedang
231	2	gangguan hama penyakit	=	sedang	perlakuan	=	scoring (4), sedang
232	1	bencana dan tingkat harga	=	bencana sedang dan perubahan besar pada harga	perlakuan	=	scoring (6), intensif
232	2	gangguan hama penyakit	=	tinggi	perlakuan	=	scoring (6), intensif
233	1	bencana dan tingkat harga	=	bencana jarang dan perubahan besar pada harga	perlakuan	=	scoring (1), manual
233	2	gangguan hama penyakit	=	rendah	perlakuan	=	scoring (1), manual
234	1	bencana dan tingkat harga	=	bencana jarang dan perubahan besar pada harga	perlakuan	=	scoring (2), sedang
234	2	gangguan hama penyakit	=	sedang	perlakuan	=	scoring (2), sedang
235	1	bencana dan tingkat harga	=	bencana jarang dan perubahan besar pada harga	perlakuan	=	scoring (5), intensif
235	2	gangguan hama penyakit	=	tinggi	perlakuan	=	scoring (5), intensif
236	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum produksi
236	2	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berorientasi maksimum produksi
236	3	perlakuan	=	manual	saran	=	sebaiknya berorientasi maksimum produksi
237	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum produksi
237	2	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berorientasi maksimum produksi
237	3	perlakuan	=	manual	saran	=	sebaiknya berorientasi maksimum produksi
238	1	ketinggian tempat	=	lebih besar 1500 m	ketinggian tempat	>	1500 m
239	1	ketinggian tempat	=	lebih kecil 1500 m	ketinggian tempat	=	0 - 1500 m

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
240	1	curah hujan	=	antara 1000 dan 1500 mm/th	curah hujan	=	antara 1000 - 1500 mm/th
241	1	curah hujan	=	antara 2500 dan 3500 mm/th	curah hujan	=	antara 2500 - 3500 mm/th
242	1	curah hujan	=	antara 1500 dan 2500 mm/th	curah hujan		antara 1500 - 2500 mm/th
244	1	teknologi	=	tingkat rendah	teknologi	=	rendah atau sedang
245	1	teknologi	=	tingkat sedang	teknologi	=	rendah atau sedang
246	1	perlakuan	=	penjadwalan, manual	perlakuan	=	penjadwalan
247	1	perlakuan	=	penjadwalan, sedang	perlakuan	=	penjadwalan
248	1	perlakuan	=	penjadwalan, intensif	perlakuan	=	penjadwalan
249	1	perlakuan	=	scoring (1), manual	perlakuan	=	scoring
250	1	perlakuan	=	scoring (2), sedang	perlakuan	=	scoring
251	1	perlakuan	=	scoring (3), manual	perlakuan	=	scoring
252	1	perlakuan	=	scoring (4), sedang	perlakuan	=	scoring
253	1	perlakuan	=	scoring (5), intensif	perlakuan	=	scoring
254	1	perlakuan	=	scoring (6), intensif	perlakuan	=	scoring
256	1	perlakuan	=	intensif	perlakuan	=	sedang atau intensif
257	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
257	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berotientasi maksimum profit
257	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
258	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
258	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berotientasi maksimum profit
258	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berotientasi maksimum profit
259	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
259	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berotientasi maksimum profit
259	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berotientasi maksimum profit
260	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
260	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berotientasi maksimum profit
260	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berotientasi maksimum profit
261	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
261	2	perlakuan	=	scoring	saran	=	sebaiknya berotientasi maksimum profit
261	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
262	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
262	2	perlakuan	=	scoring	saran	=	sebaiknya berotientasi maksimum profit
262	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berotientasi maksimum profit
263	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
263	2	perlakuan	=	scoring	saran	=	sebaiknya berotientasi maksimum profit
263	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berotientasi maksimum profit
264	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
264	2	perlakuan	=	scoring	saran	=	sebaiknya berotientasi maksimum profit
264	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berotientasi maksimum profit
265	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
265	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
265	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
266	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
266	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
266	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
267	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
267	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
267	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
268	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
268	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
268	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berotientasi maksimum profit
269	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
269	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
269	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berotientasi maksimum profit
270	1	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
270	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
270	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berorientasi maksimum profit
271	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
271	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berorientasi maksimum profit
271	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berorientasi maksimum profit
272	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
272	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berorientasi maksimum profit
272	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berorientasi maksimum profit
273	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
273	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berorientasi maksimum profit
273	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berorientasi maksimum profit
274	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
274	2	perlakuan	=	penjadwalan	saran	=	sebaiknya berorientasi maksimum profit
274	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berorientasi maksimum profit
275	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
275	2	perlakuan	=	scoring	saran	=	sebaiknya berorientasi maksimum profit
275	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berorientasi maksimum profit
276	1	teknologi	=	tingkat tinggi	curah hujan	=	sebaiknya berorientasi maksimum profit
276	2	perlakuan	=	scoring	curah hujan	=	sebaiknya berorientasi maksimum profit
276	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	curah hujan	=	sebaiknya berorientasi maksimum profit
277	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
277	2	perlakuan	=	scoring	saran	=	sebaiknya berorientasi maksimum profit
277	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berorientasi maksimum profit
278	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berorientasi maksimum profit
278	2	perlakuan	=	scoring	saran	=	sebaiknya berorientasi maksimum profit
278	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berorientasi maksimum profit

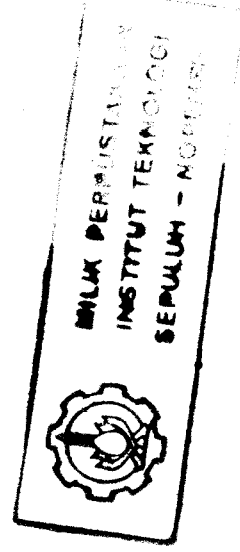
rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
279	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berotientasi maksimum profit
279	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
279	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
280	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berotientasi maksimum profit
280	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
280	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berotientasi maksimum profit
281	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berotientasi maksimum profit
281	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
281	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berotientasi maksimum profit
282	1	teknologi	=	tingkat tinggi	saran	=	sebaiknya berotientasi maksimum profit
282	2	perlakuan	=	sedang atau intensif	saran	=	sebaiknya berotientasi maksimum profit
282	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berotientasi maksimum profit
283	1	perlakuan	=	manual	saran	=	sebaiknya berotientasi maksimum profit
283	2	teknologi	=	tingkat tinggi	saran	=	sebaiknya berotientasi maksimum profit
283	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berotientasi maksimum profit
284	1	perlakuan	=	manual	saran	=	sebaiknya berotientasi maksimum profit
284	2	teknologi	=	tingkat tinggi	saran	=	sebaiknya berotientasi maksimum profit
284	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berotientasi maksimum profit
285	1	perlakuan	=	manual	saran	=	sebaiknya berotientasi maksimum profit
285	2	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
285	3	topoklimatologi/agroekologi	=	sesuai	saran	=	sebaiknya berotientasi maksimum profit
286	1	perlakuan	=	manual	saran	=	sebaiknya berotientasi maksimum profit
286	2	teknologi	=	rendah atau sedang	saran	=	sebaiknya berotientasi maksimum profit
286	3	topoklimatologi/agroekologi	=	diperbaiki dengan irigasi	saran	=	sebaiknya berotientasi maksimum profit
287	1	perlakuan	=	manual	saran	=	sebaiknya berotientasi maksimum profit

rule num	clau se num	if clause	if sy	if value	then clause	th sy	then value
287	2	teknologi	=	rendah atau sedang	saran	=	sebaiknya berorientasi maksimum profit
287	3	topoklimatologi/agroekologi	=	sesuai dengan pengolahan tanah intensif	saran	=	sebaiknya berorientasi maksimum profit
288	1	perlakuan	=	manual	saran	=	sebaiknya berorientasi maksimum profit
288	2	teknologi	=	rendah atau sedang	saran	=	sebaiknya berorientasi maksimum profit
288	3	topoklimatologi/agroekologi	=	untuk produksi batang dan daun (hijauan)	saran	=	sebaiknya berorientasi maksimum profit

LAMPIRAN B:

DOKUMENTASI MEETING

Pada lampiran ini berisi tentang dokumentasi meeting yaitu dokumentasi yang diperoleh dari konsultasi dengan pakar. Seperti dijelaskan pada bagian 6.2.1 pada tahap perancangan melibatkan seorang pakar dalam bidang pertanian. Pada tahap perancangan ketika merancangan basis aturan diperlukan keterlibatan seorang pakar, untuk menghindari distorsi pada tahap ini, maka dilakukan aktifitas dokumentasi dari setiap pertemuan yang dilakukan bersama dengan pakar, baik secara tertulis maupun berupa tape. Namun dalam lampiran ini hanya dilampirkan duplikat dari dokumentasi yang secara tertulis. Pada saat pertemuan dengan pakar memungkinkan terjadinya koreksi terhadap hasil dari meeting yang terjadi sebelumnya, sehingga terdapat koreksi dari pakar terhadap hasil dari dokumentasi meeting sebelumnya. Berikut ini adalah kopi dari beberapa dokumentasi meeting.



Dokumentasi Meeting

1. Waktu
 - a. Tanggal : Selasa, 10 Nopember 1999
 - b. Jam : 11.00 wib
2. Lokasi : Lembaga Penelitian Unibraw Malang
3. Nama Pakar : Prof. DR. Ir. Soemarno, MS
4. Pengambilan keputusan dalam perencanaan industri pertanian dapat dikategorikan sebagai pengambilan keputusan yang kompleks. *kelemahannya fungsional.*
5. Permasalahan yang dihadapi dalam pengambilan keputusan perencanaan pertanian adalah:
 - a. Secara umum dalam industri pertanian banyak dilakukan kerjasama dengan pertanian rakyat hal ini menyulitkan dalam mengkoordinir
 - b. Apabila industri juga melakukan suply terhadap pupuk dan bibit tingkat resiko yang diperhitungkan juga meningkat

Sehingga dalam pengambilan keputusan untuk merencanakan pertanian kita perlu memperhitingkan kondisi yang ada kemudian kita tentukan sejauh mana tingkat keterlibatan kita terhadap aktifitas produksi.
6. Tingkat kebutuhan jagung dalam pasar diasumsikan tinggi sekali sehingga berapapun besarnya hasil produksi dapat diasumsikan mampu diserap oleh pasar. *jenis → big scale → job part sup*
7. Faktor-faktor yang berpengaruh dalam pengambilan keputusan ini adalah: *Rakyat → small → tidak ada keahlian*
 - a. Musim¹ *alam*
 - iklim (kelembaban, curah hujan, ketinggian dari permukaan laut, posisi geografis) *Resiko rendah*
 - kondisi tanah
 - waktu tanam dan waktu panen
 - b. Teknologi² *prosedur*
 - c. Resiko³ *Resiko rendah*

Proses yg diukur → bibit, pupuk, tanah } ilustrasi 2 memperluas sayap
8. Jenis keputusan yang diambil dalam perencanaan industri pertanian khususnya jagung adalah melingkupi segala lingkup metode pengambilan keputusan mulai dari terstruktur, semi terstruktur dan unstruktur. Sehingga tiap pengambilan keputusan sangat tergantung dengan permasalahan yang dihadapi bisa jadi dalam suatu kondisi tertentu bisa menggunakan yang terstruktur bisa jadi tidak.
9. Dalam pengambilan keputusan dilakukan klasifikasi persoalan yaitu dengan cara mengklasifikasikan kondisi yang ada termasuk dalam kategori yang mana (ditentukan kemudian) lalu membandingkan kondisi tersebut dengan kondisi yang harus dicapainya. *(yang perlu diperbandingkan?)⁴*
10. Ciri-ciri dan batasan klasifikasi menyusul.*
11. Syarat kondisi yang harus dipenuhi untuk mencapai
12. kondisi tertentu menyusul.*
13. identifikasi objek dan atribut semetara

Atribut		Tipe kondisi						
		A	B	C	D	E	F	G
a)	Musim	Value?						
	• iklim							
	➢ kelembaban							
	➢ curah hujan							
	➢ ketinggian dari permukaan laut							
	➢ posisi geografis							
	➢ kondisi tanah							
	• waktu tanam dan waktu panen							
	b. Teknologi							
	c. Resiko							

¹ Apa saja yang dapat dikategorikan sebagai musim
² apa yang dimaksud dengan teknologi disini
³ Gambaran resiko apakah juga diklasifikasikan?
⁴ Gambaran umum rule

a) Agroklimatologi

1. Curah hujan
2. Jenis tanah B.2
3. Ketinggian tempat

1) Tdk sama
 2) Cc...
 3) Jml...
 Cc...
 Signa

Dokumentasi Meeting

1. Waktu
 - a. Tanggal : Selasa, 24 Nopember 1999
 - b. Jam : 10.30 wib
2. Lokasi : Lembaga Penelitian Unibraw Malang
3. Nama Pakar : Prof. DR. Ir. Soemarno, MS
4. Preview pertemuan sebelumnya (beberapa pertanyaan belum terjawab)
Perubahan terhadap obyek dan atribut sementara

Topoklimatologi	Tidak sesuai	Sesuai	Sangat sesuai
1. Curah Hujan (mm/th) (sumber data dari kecamatan/dinas pengairan)	< 1000 >3500	1000 - 1500 2500 - 3500	1500 - 2500
2. Jenis tanah (sumber data dinas pertanian)	Liat (halus) Pasir (kasar)	Lempung (medium) Lempung-berdebu	Lempung berpasir
3. Ketinggian tempat	> 1500 m	0 - 1500	0 - 1500

teknologi	Tidak sesuai	Sesuai	Sangat sesuai
1. Sosio-teknologi (teknologi yang sudah diketahui dan dipraktekkan dalam masyarakat)	Tidak diketahui dan tidak dipraktekkan	Sudah diketahui atau dipraktekkan	Diketahui dan dipraktekkan dalam masyarakat
2. Sumber inovasi teknologi (penyuluhan pertanian.) • PPL / pemerintah • Swasta (industri pertanian yang berkepentingan) • Pedagang (aktifitas promosi)	Tidak ada	Ada satu atau dua diantara tiga	Keseluruhan komponen tersebut ada

Resiko	rendah < 5%	Sedang 5% - 25%	Tinggi >25%
1. Pasar: ➢ Finansial ➢ harga	Tingkat harga stabil	Terjadi perubahan - perubahan tingkat harga	Fluktuasi harga yang tinggi
2. Agroklimatologi	Kategori topoklimatologi sangat sesuai	Kategori sesuai	Kategori tidak sesuai
3. Gangguan ➢ Hama penyakit ➢ Bencana alam (sumber data dari penduduk setempat)	Kategori sangat sesuai Data historis bencana alam jarang	Kategori sesuai Data historis bencana alam sedang	Kategori tidak sesuai Data historis bencana alam sering

5. Metode pengambilan keputusan (umum)?
 - a. menentukan klasifikasi kondisi
 - b. memberikan penyelesaian (keputusan) yang perlu diambil untuk menyesuaikan kondisi dengan standar yang dimiliki.
 - c. Menentukan solusi alternatif
 - d. Memilih dari beberapa solusi alternatif yang muncul.
6. Adakah cara tersendiri untuk menyelesaikan permasalahan pada pertemuan lalu no. 5
Ada, setiap kondisi memiliki cara yang spesifik dalam setiap solusi yang dimiliki.

7. Solusi-solusi alternatif semacam shortcut (selain metode yang dipakai sebelumnya) dalam pengambilan keputusan? ⁵
8. Type permasalahan dan karakteristik?
- a. How much to produce?
Produksi akan dilakukan dengan tiga kategori (tingkat kesuksesannya disesuaikan dengan topoklimatologi)⁶:

	Tidak sesuai	Sesuai	Sangat sesuai
Produksi (data tersebut diperoleh dari dinas pertanian)	<60%	60% - 80%	80% - 100%

- b. How to produce?
Dari klasifikasi dihasilkan perlakuan dalam produksi (dijelaskan berikut)*
- c. What to Produce?
Yang mampu dihasilkan dalam industri pertanian jagung adalah
1. Biji jagung kering
 2. Produksi jagung muda
 3. Batang dan daun jagung
9. Economics problem?
- a. goals or objectives to be obtained
- Max produksi
 - Max profit
- Merupakan pertimbangan terakhir dari proses pengambilan keputusan setelah alternatif-alternatif dalam aktifitas produksi tergambar jelas baru diperhitungkan mana yang lebih diprioritaskan max profit atau max produksi.⁷
- b. a limited amount of resource to use in reaching these goal and objectives
dalam produksi jagung sumber daya yang terbatas pada umumnya adalah pupuk, bibit dan obat-obatan permasalahan ini dipertimbangkan sebelum menentukan goal yang ingin dicapai.⁸
- c. A number of alternative ways to use the limited resources in attempting to obtain the goal and objectives
Alternatif solusi untuk penyediaan sumberdaya setelah identifikasi permasalahan (pengklasifikasian kondisi) telah selesai.
10. Decision-making process (konfirmasi)
- a. identify and define the problem
 - b. collect relevant data, fact, and information
 - c. identify and analyze alternative solutions
 - d. make the decision ----select the best alternative
 - e. implement the decision
 - f. observe the result and bear responsibility for the outcome
- a dan b bukan merupakan langkah sekuensial bisa dibolak-balik
11. Rule sementara yang dihasilkan:

⁵ berikut setelah pengklasifikasian selesai

⁶ dari tiap bibit mempunyai deskripsi produksi perhektarnya dengan nilai prosentasi diatas maksudnya a% dari produksi perhektarnya tiap bibit.

⁷ Goal yang ingin dicapai sangat tergantung dengan kondisi yang ada

⁸ Diperhitungkan sebelum penentuan goal yang ingin dicapai

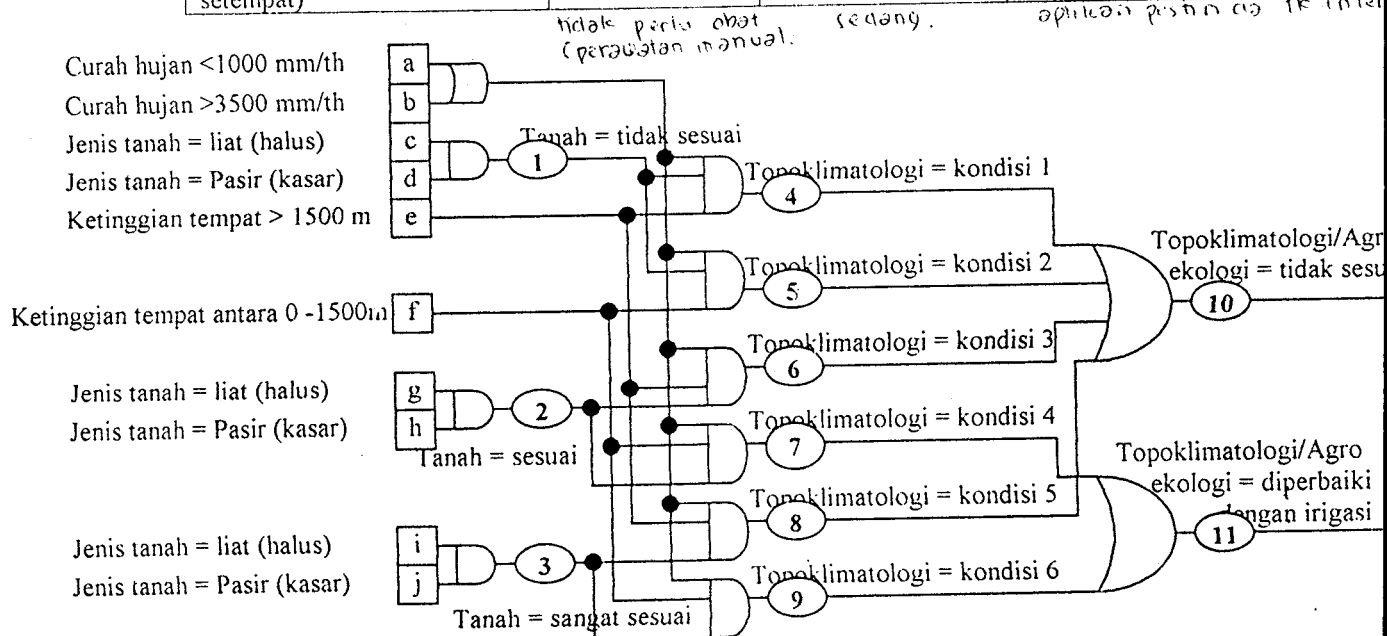
Dokumentasi Meeting

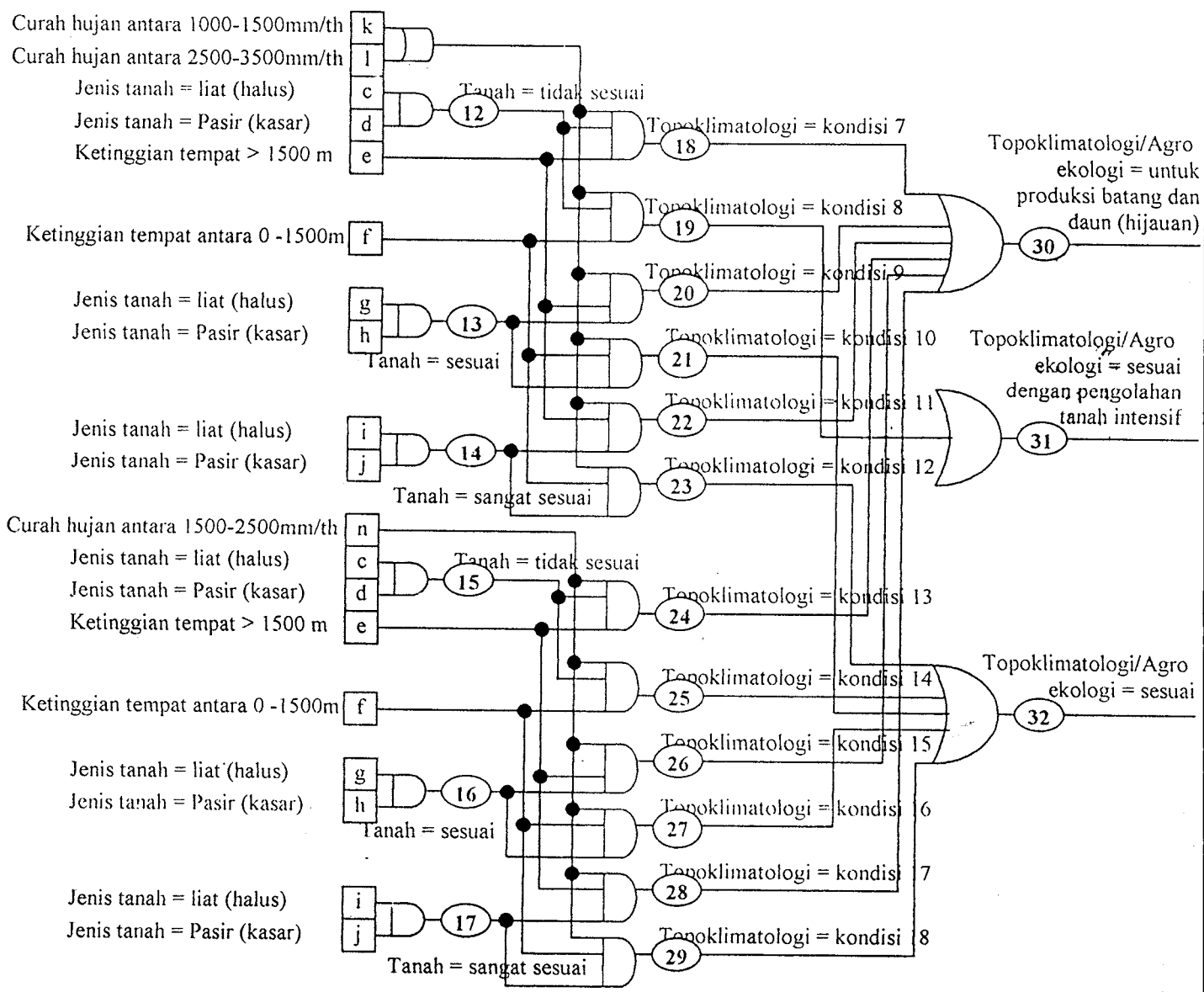
1. Waktu
 - c. Tanggal : Selasa, 15 Desember 1999
 - d. Jam : 08.00 wib
2. Lokasi : Lembaga Penelitian Unibraw Malang
3. Nama Pakar : Prof. DR. Ir. Soemarno, MS
- 4.

Topoklimatologi	Tidak sesuai	Sesuai	Sangat sesuai
1. Curah Hujan (mm/th) (sumber data dari kecamatan/dinas pengairan)	< 1000 > 3500	1000 - 1500 2500 - 3500	1500 - 2500
2. Jenis tanah (sumber data dinas pertanian)	Liat (halus) Pasir (kasar)	Lempung (medium) Lempung-berdebu	Lempung berpasir
3. Ketinggian tempat	> 1500 m	0 - 1500	0 - 1500

teknologi	Tidak sesuai	Sesuai	Sangat sesuai
1. Sosio-teknologi (teknologi yang sudah diketahui dan dipraktekkan dalam masyarakat.	Tidak diketahui dan tidak dipraktekkan	Sudah diketahui atau dipraktekkan	Diketahui dan dipraktekkan dalam masyarakat

Resiko	rendah < 5%	Sedang 5% - 25%	Tinggi > 25%
1. Pasar: ➢ Finansial ➢ harga	Tingkat harga stabil	Terjadi perubahan - perubahan tingkat harga (p.0) stabil	Fluktuasi harga yang tinggi (scoring) (-)
2. Gangguan ➢ Hama penyakit ➢ Bencana alam (sumber data dari penduduk setempat)	Kategori sangat sesuai Data historis bencana alam jarang	Kategori sesuai Data historis bencana alam sedang	Kategori tidak sesuai Data historis bencana alam sering





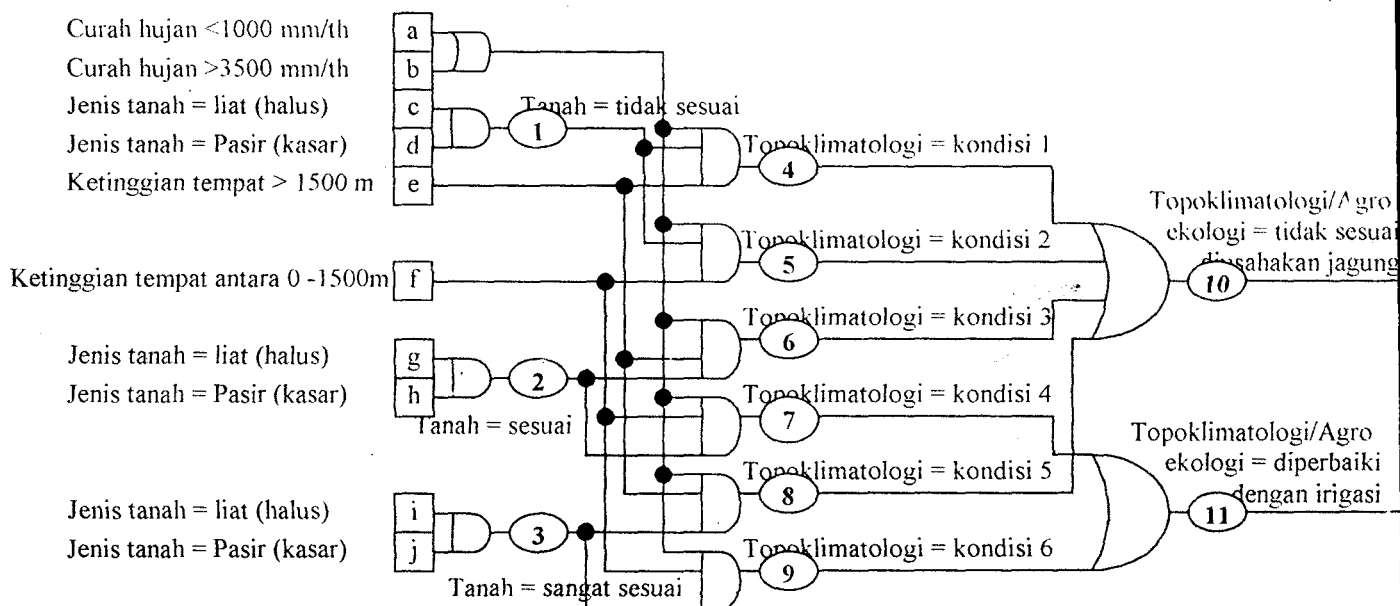
Dokumentasi Meeting

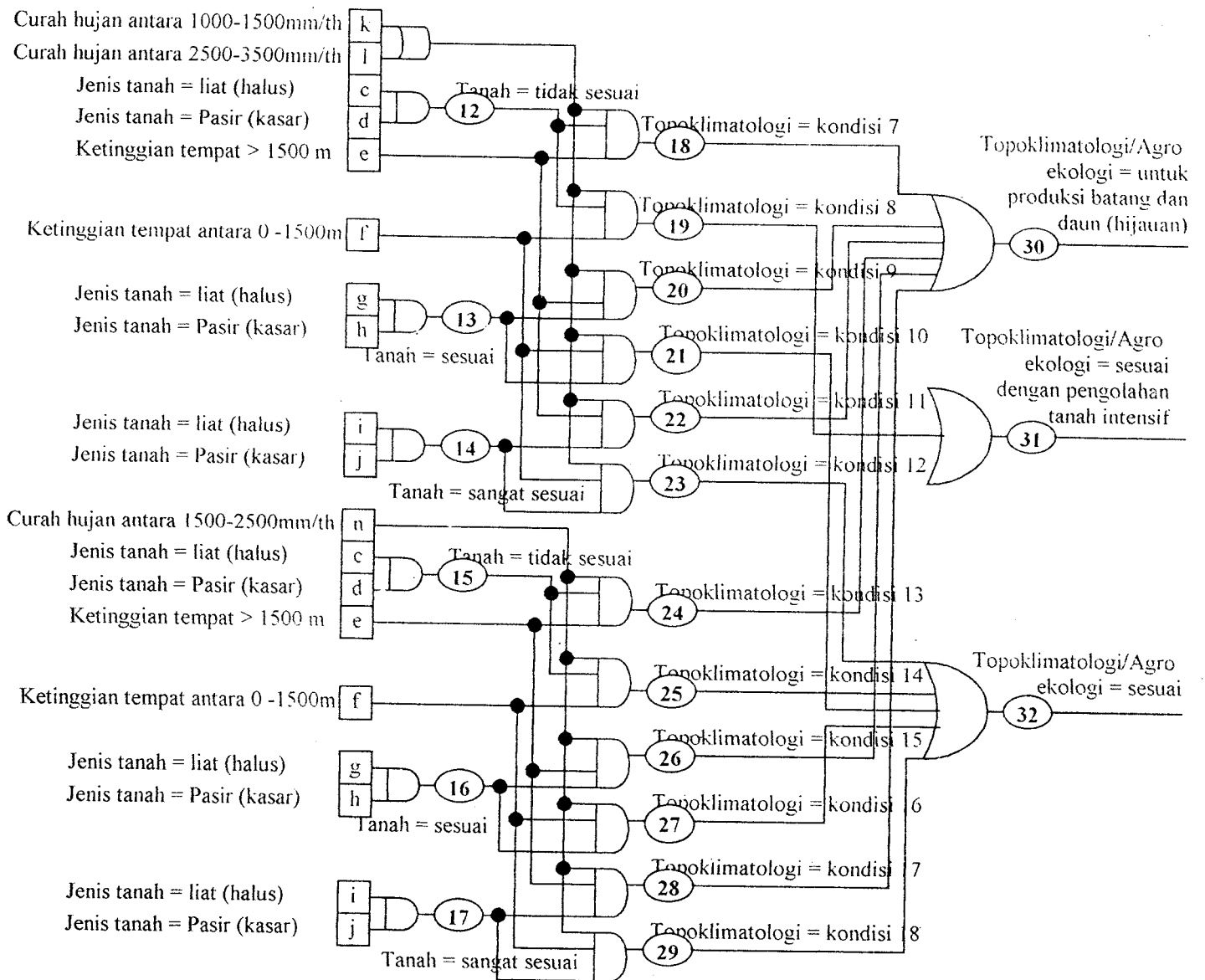
1. Waktu
 - c. Tanggal : Selasa, 22 Desember 1999
 - d. Jam : 08.00 wib
2. Lokasi : Lembaga Penelitian Unibraw Malang
3. Nama Pakar : Prof. DR. Ir. Soemarno, MS
- 4.

Topoklimatologi	Tidak sesuai	Sesuai	Sangat sesuai
1. Curah Hujan (mm/th) (sumber data dari kecamatan/dinas pengairan)	< 1000 >3500	1000 - 1500 2500 - 3500	1500 - 2500
2. Jenis tanah (sumber data dinas pertanian)	Liat (halus) Pasir (kasar)	Lempung (medium) Lempung-berdebu	Lempung berpasir
3. Ketinggian tempat	> 1500 m	0 - 1500	0 - 1500

teknologi	Tidak sesuai	Sesuai	Sangat sesuai
1. Sosio-teknologi (teknologi yang sudah diketahui dan dipraktekkan dalam masyarakat.	Tidak diketahui dan tidak dipraktekkan	Sudah diketahui atau dipraktekkan	Diketahui dan dipraktekkan dalam masyarakat

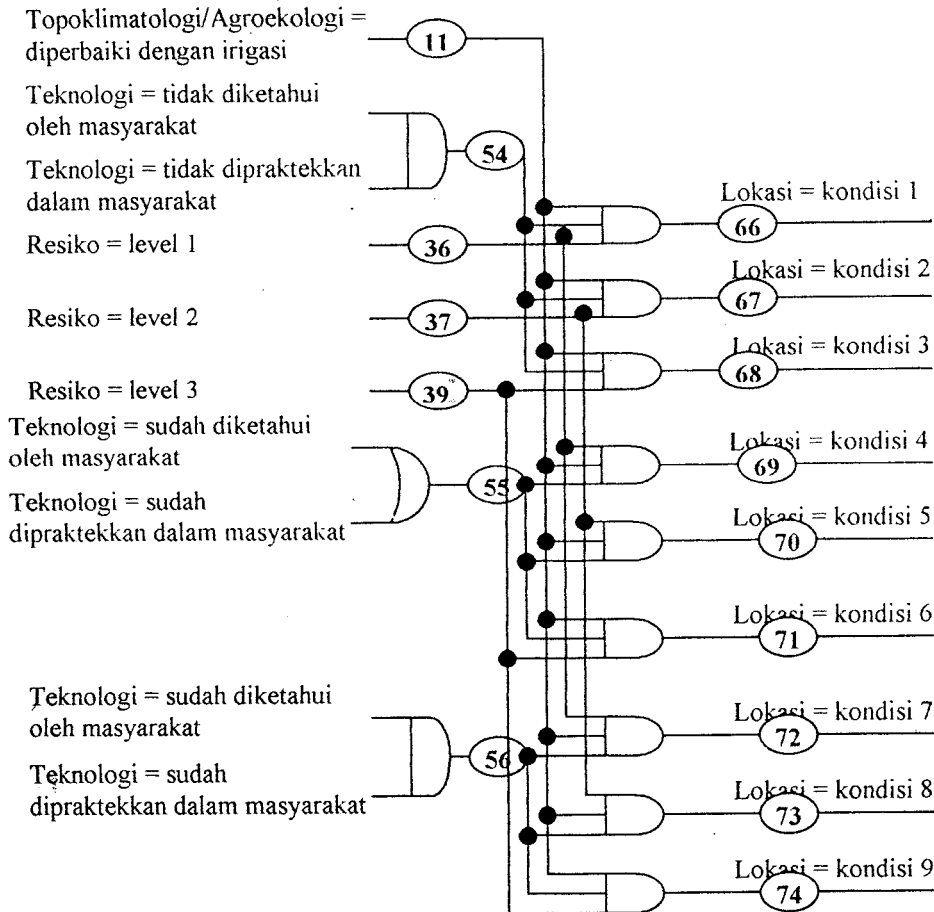
Resiko	rendah < 5%	Sedang 5% - 25%	Tinggi >25%
1. Pasar: ➢ Finansial ➢ harga	Tingkat harga stabil	Terjadi perubahan - perubahan tingkat harga	Fluktuasi harga yang tinggi
2. Gangguan ➢ Hama penyakit ➢ Bencana alam (sumber data dari penduduk setempat)	Kategori sangat sesuai Data historis bencana alam jarang	Kategori sesuai Data historis bencana alam sedang	Kategori tidak sesuai Data historis bencana alam sering





Dokumentasi Meeting

1. Waktu
 - a. Tanggal : Selasa, ^{22 Nov}~~10~~ Nopember 1999
 - b. Jam : 11.00 wib
2. Lokasi : Lembaga Penelitian Unibraw Malang
3. Nama Pakar : Prof. DR. Ir. Soemarno, MS

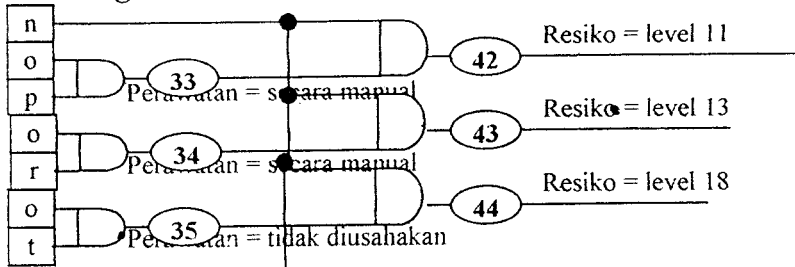


Saran = memperbaiki kondisi tanah dengan irigasi -

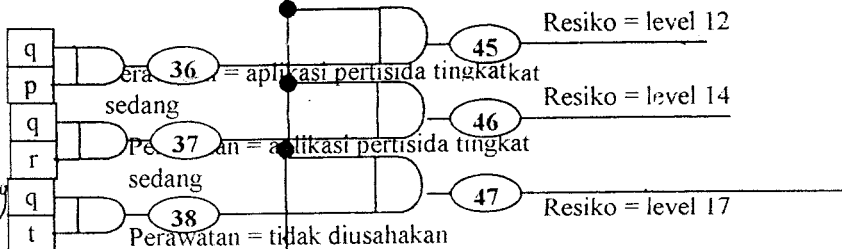
- * Berikan sarana
 - PPL / Pemerintah
 - Swasta (industri pertanian yang berkepentingan)
 - pedagang (aktivitas promosi)
- * tingkat harga stabil
- * lakukan perawatan secara manual.

Tingkat Resiko

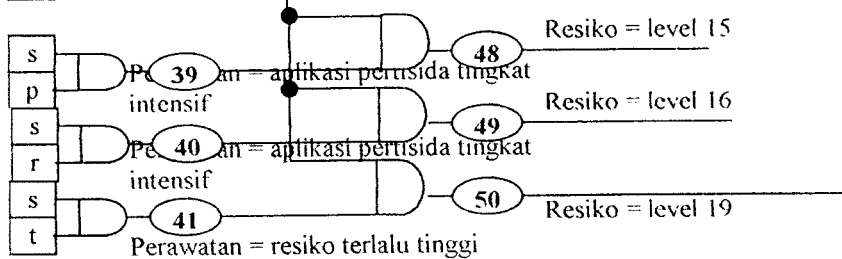
Tingkat harga = stabil
 Gangguan hama penyakit = rendah
 Data historis bencana alam = jarang
 Gangguan hama penyakit = rendah
 Data historis bencana alam = sedang
 Gangguan hama penyakit = rendah
 Data historis bencana alam = sering



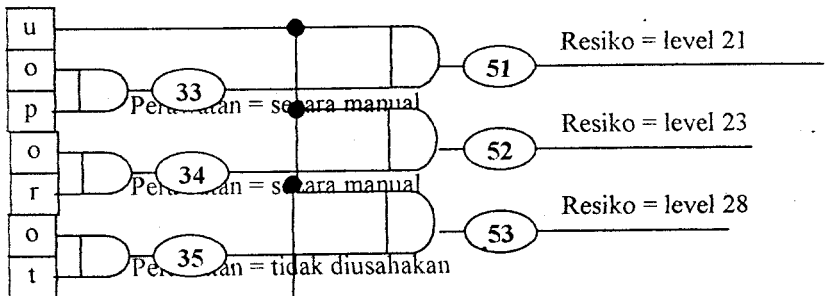
Gangguan hama penyakit = sedang
 Data historis bencana alam = jarang
 Gangguan hama penyakit = sedang
 Data historis bencana alam = sedang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sering



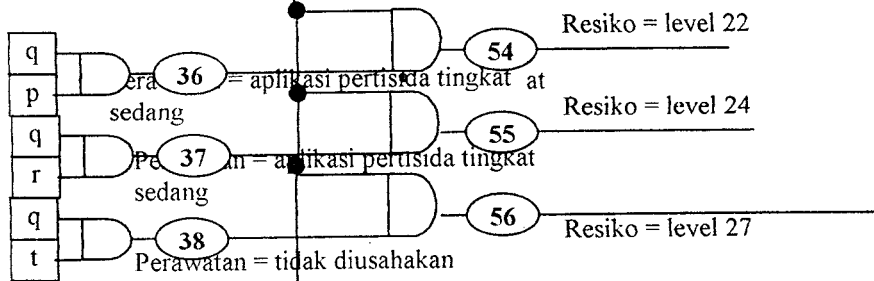
Gangguan hama penyakit = tinggi
 Data historis bencana alam = jarang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sedang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sering



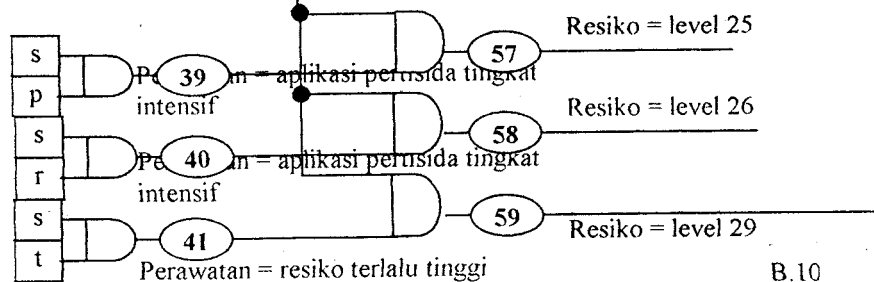
Tingkat harga = terjadi perubahan kecil
 Gangguan hama penyakit = rendah
 Data historis bencana alam = jarang
 Gangguan hama penyakit = rendah
 Data historis bencana alam = sedang
 Gangguan hama penyakit = rendah
 Data historis bencana alam = sering



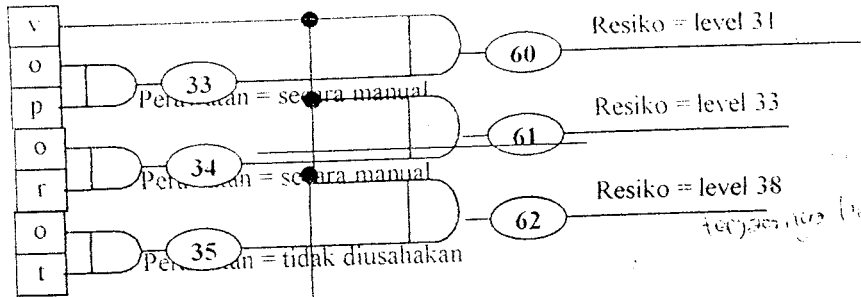
Gangguan hama penyakit = sedang
 Data historis bencana alam = jarang
 Gangguan hama penyakit = sedang
 Data historis bencana alam = sedang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sering



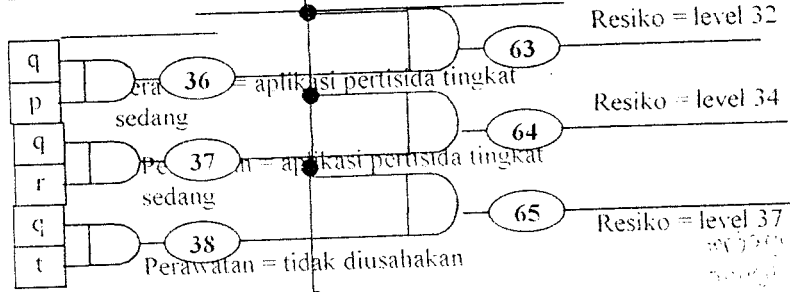
Gangguan hama penyakit = tinggi
 Data historis bencana alam = jarang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sedang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sering



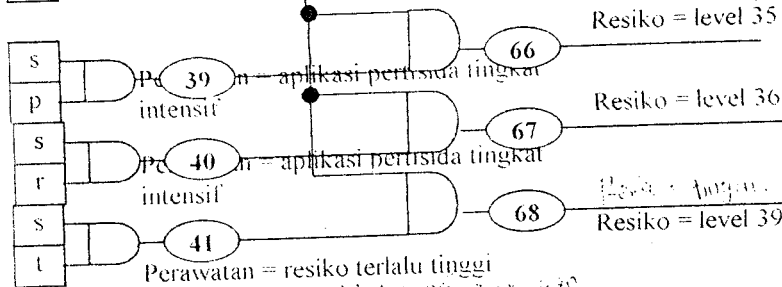
Tingkat harga = terjadi perubahan besar
 Gangguan hama penyakit = rendah
 Data historis bencana alam = jarang
 Gangguan hama penyakit = rendah
 Data historis bencana alam = sedang
 Gangguan hama penyakit = rendah
 Data historis bencana alam = sering



Gangguan hama penyakit = sedang
 Data historis bencana alam = jarang
 Gangguan hama penyakit = sedang
 Data historis bencana alam = sedang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sering

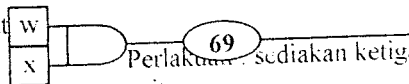


Gangguan hama penyakit = tinggi
 Data historis bencana alam = jarang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sedang
 Gangguan hama penyakit = tinggi
 Data historis bencana alam = sering



Teknologi pengusahaan jagung

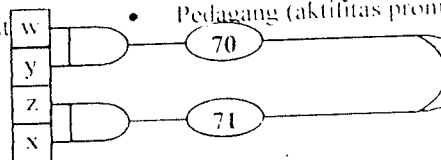
Teknologi = tidak dipraktekkan dalam masyarakat
 Teknologi = tidak diketahui oleh masyarakat



Perlakuan : sediakan ketiga sumber penyuluh lapangan yaitu:

- PPL (pemerintah)
- Swasta (industri pertanian yang berkepentingan)
- Pedagang (aktifitas promosi)

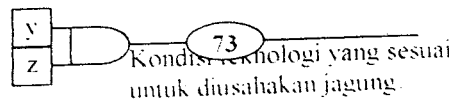
Teknologi = tidak dipraktekkan dalam masyarakat
 Teknologi = diketahui oleh masyarakat
 Teknologi = dipraktekkan dalam masyarakat
 Teknologi = tidak diketahui oleh masyarakat



Perlakuan : sediakan dua diantara ketiga sumber penyuluh lapangan yaitu

- PPL (pemerintah)
- Swasta (industri pertanian yang berkepentingan)
- Pedagang (aktifitas promosi)

Teknologi = dipraktekkan dalam masyarakat
 Teknologi = diketahui oleh masyarakat



Kondisi teknologi yang sesuai untuk diusahakan jagung.

LAMPIRAN C:

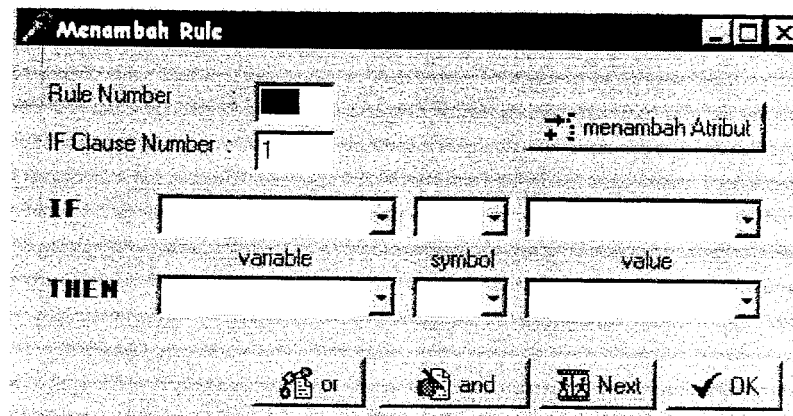
PETUNJUK PENGGUNAAN

Sistem pakar untuk perencanaan produksi pertanian ini dibagi dalam dua bagian utama yaitu untuk pakar dan atau untuk perekayasa pengetahuan dan bagian yang lain untuk pemakai yang ingin memanfaatkan sistem pakar ini untuk membantu menyelesaikan permasalahannya. Bagian pertama yaitu untuk pakar dan atau untuk perekayasa pengetahuan berfungsi untuk mengubah aturan yang ada baik untuk menambah, mengedit dan menghapus aturan yang ada. Yang selanjutnya disebut sebagai *bagian pengembangan*. Sedangkan bagian yang kedua disebut sebagai *bagian pemanfaatan*.

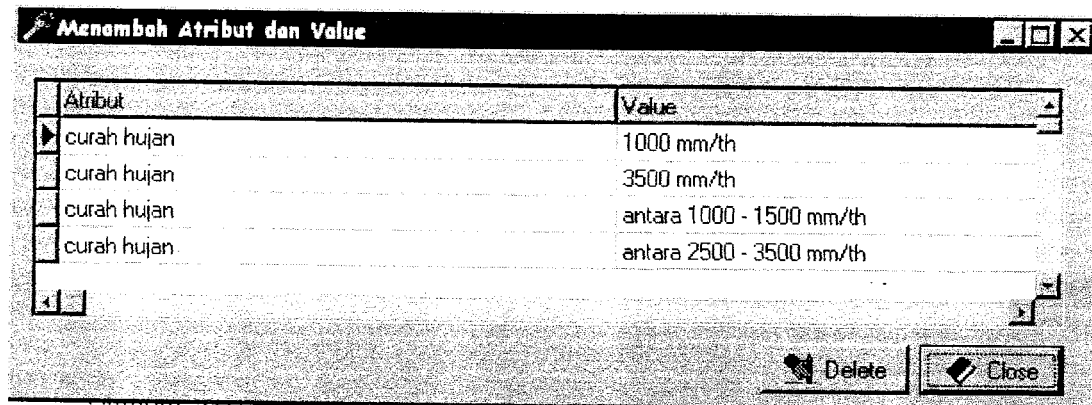
A.1. Bagian Pengembangan

Pakar atau perekayasa pengetahuan dapat menambahkan aturan pada sistem ini dengan cara pada menu utama memilih '*Rule*' kemudian pilih '*menambah rule baru*' maka akan terlihat antar muka seperti tampak pada gambar C.1. Apabila '*Rule*' yang ingin ditambahkan tersebut atribut dan valuenya sudah ada maka pemakai bisa langsung menambahkan '*Rule*' dengan menggunakan combobox di masing-masing bagian yang harus diisi. Tetapi apabila belum ada atribut dan valuenya pemakai tidak dapat menambahkan rule yang diinginkan, maka pemakai harus menambahkan atribut dan value dengan cara mengklik tombol '*menambah atribut*' kemudian akan muncul antar muka seperti pada gambar C.2. Untuk

mengisikannya pemakai langsung mengetikkan atribut dan valuenya yang ingin ditambahkan.



Gambar C.1. Antar muka menambah rule baru



Gambar C.2. Antar muka menambah atribut

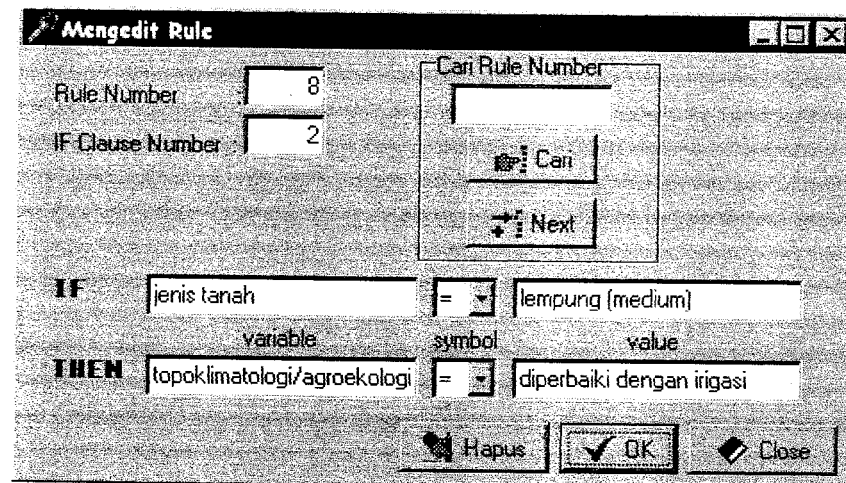
Dalam pengembangan aturan memungkinkan pemakai melakukan kesalahan dalam menambahkan aturan maka dalam sistem ini disediakan fasilitas mengedit aturan, untuk melakukannya pemakai perlu memilih 'mengedit rule'. Langkah yang perlu dilakukan dalam mengedit rule adalah sebagai berikut.

1. Tentukan terlebih dahulu nomor rule yang ingin diedit.
2. Ketikkan nomor rule pada textbox cari rule number

3. Klik cari
4. Setelah rule tersebut ditemukan pemakai dapat mengedit rule yang ingin diedit tersebut.

Antar muka untuk mengedit rule tersebut dapat dilihat pada gambar c.3. Pada antar muka '*mengedit rule*' pemakai hanya bisa mengedit dan menghapus rule sedangkan untuk menambah rule dilakukan pada antar muka '*manambah rule baru*'.

Dalam sistem ini juga disediakan fasilitas untuk menghapus semua rule yang ada dalam 'working memory' apabila pemakai menginginkan untuk menghapus seluruh rule yang ada dalam 'working memory' . antar muka fasilitas ini dapat dilihat pada gambar C.4. Pada fasilitas ini terdapat konfirmasi dari sistem apakah pemakai benar-benar ingin menghapus aturan yang ada dalam 'working memori'.



Gambar C.3. antar muka untuk mengedit rule



Gambar C.4. antar muka konfirmasi untuk menghapus semua rule

Selain mempunyai hak pada bagian pengembangan pemakai pakar dan perekayasa pengetahuan berhak untuk memanfaatkan bagian pemanfaatan untuk menguji apakah aturan yang dimasukkan dapat bekerja dengan baik sesuai dengan yang diharapkan.

A.2. Bagian Pemanfaatan

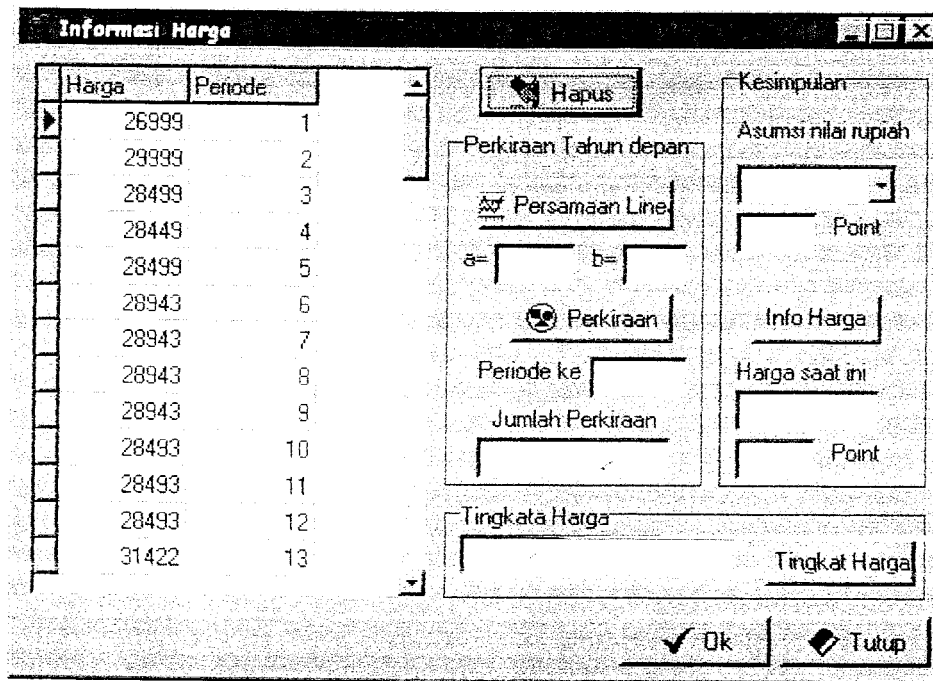
Pada bagian ini pemakai berhak untuk menggunakan aturan yang telah dibuat oleh pakar dan atau perekayasa pengetahuan untuk berkonsultasi. Perlu diingat oleh pemakai sebelum pemakai memanfaatkan sistem berbasis aturan ini harus melihat dulu rule yang ada dengan mengklik bagian '*melihat semua rule*' maka akan muncul antar muka seperti

pada gambar C.5 dan kemudian mengklik tombol 'view' untuk memastikan apakah ada rule dalam 'working memory'. Apabila dalam 'working memori' tidak terdapat rule sedangkan pada basis aturan dalam sistem terdapat rule maka ketika pemakai mengklik tombol 'view', sistem langsung mengambil rule yang ada dalam basis aturan dan meletakkannya pada 'working memori'. Bila dalam basis aturan tidak terdapat aturan sama sekali maka pemakai pakar dan atau perekayasa pengetahuan harus mengisi basis aturan dengan rule.

Setelah pemakai memastikan memang ada rule dalam 'working memori' pemakai dapat mulai menggunakan sistem berbasis aturan ini.

No. Rule	No. Clause	IF Clause
261	1	perlakuan = scoring
261	2	teknologi = rendah atau sedang
262	1	perlakuan = scoring
262	2	teknologi = rendah atau sedang
263	1	perlakuan = scoring
263	2	teknologi = rendah atau sedang
263	3	topoklimatologi/agroekologi = sesuai dengan penc
264	1	...

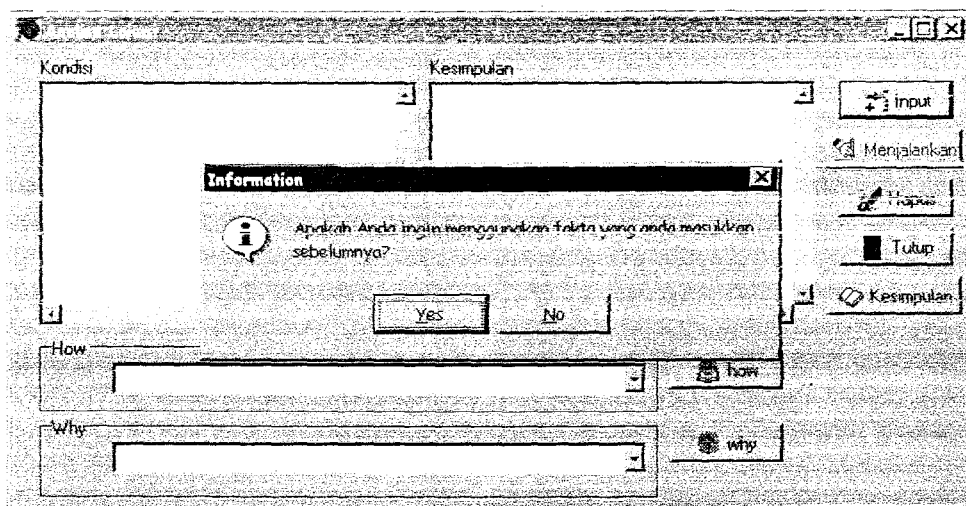
Gambar C.5. Antar muka melihat semua rule



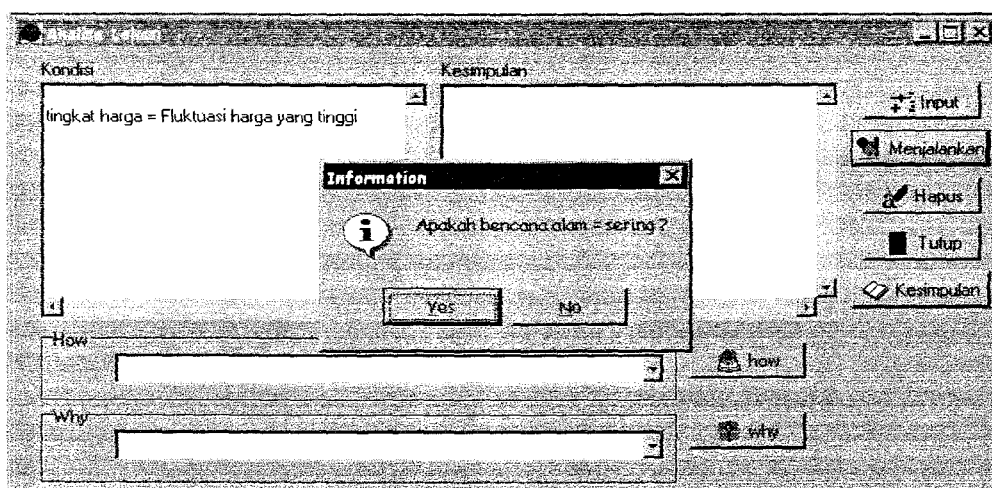
Gambar C.6. Antar muka untuk mengolah data harga

Untuk menganalisa data harga yang dimiliki oleh pemakai pemakai dapat memilih bagian 'data' pada menu utama dan memilih 'perkiraan' sehingga muncul antar muka pemakai seperti pada gambar C.6. Pada antar muka tersebut pemakai dapat mengisi tabel yang ada dengan data harga yang dimiliki, dengan range yang bebas baik tahun, atau bulan tentu saja dengan interval yang sama antara satu data dengan data yang lainnya. Data yang dimasukkan harus berupa data harga jagung per 100 kg. Setelah pemakai selesai memasukkan data harga pemakai dapat mengetahui harga a (nilai awal) dan b (nilai beda) dari data yang dimasukkan dengan menggunakan fungsi linear, dengan cara mengklik tombol 'persamaan linear'.

Bila pemakai ingin memanfaatkan data yang dimasukkannya maka pemakai dapat mengklik tombol 'tingkat harga' sistem akan membawa pemakai memasuki antramuka 'forward'. Namun sebelumnya pemakai harus memastikan bahwa bagian 'input topoklimatologi' pemakai telah mengklar semua data yang ada dalam tabel input sebelum pemakai mengklik tombol 'tingkat harga'.



Gambar C.7. Antar muka konfirmasi penggunaan data yang ada



Gambar C.8. Antar muka konfirmasi terhadap informasi pendukung

Saat pemakai memasuki antar muka 'forward' , pemakai harus mengawali dengan mengklik tombol 'input' yang kemudian akan muncul antar muka konfirmasi seperti yang terlihat pada gambar C.7 untuk menanyakan apakah pemakai akan menggunakan data input yang telah ada? Konfirmasi tersebut muncul karena sebelumnya pemakai telah memasukkan informasi analisa data kedalam sistem. Karena pemakai ingin memperoleh saran terhadap kondisi harga yang dimiliki maka harus menjawab 'yes'. Kemudian klik tombol 'run' maka akan muncul beberapa pertanyaan seperti yang terlihat pada gambar C.8 yang perlu dijawab pemakai untuk membantu sistem ini memperoleh kesimpulan.

Kondisi yang diketahui

Fakta Apakah yang anda ketahui

Topoklimatologi

Ketinggian tempat

Curah hujan

Jenis Tanah

Teknologi dalam Masyarakat

Penggunaan Teknologi

Pengetahuan

Topoklimatologi

Ketinggian tempat = [] m OK

Curah Hujan = [] mm/th

Jenis tanah = []

Teknologi dalam Masyarakat

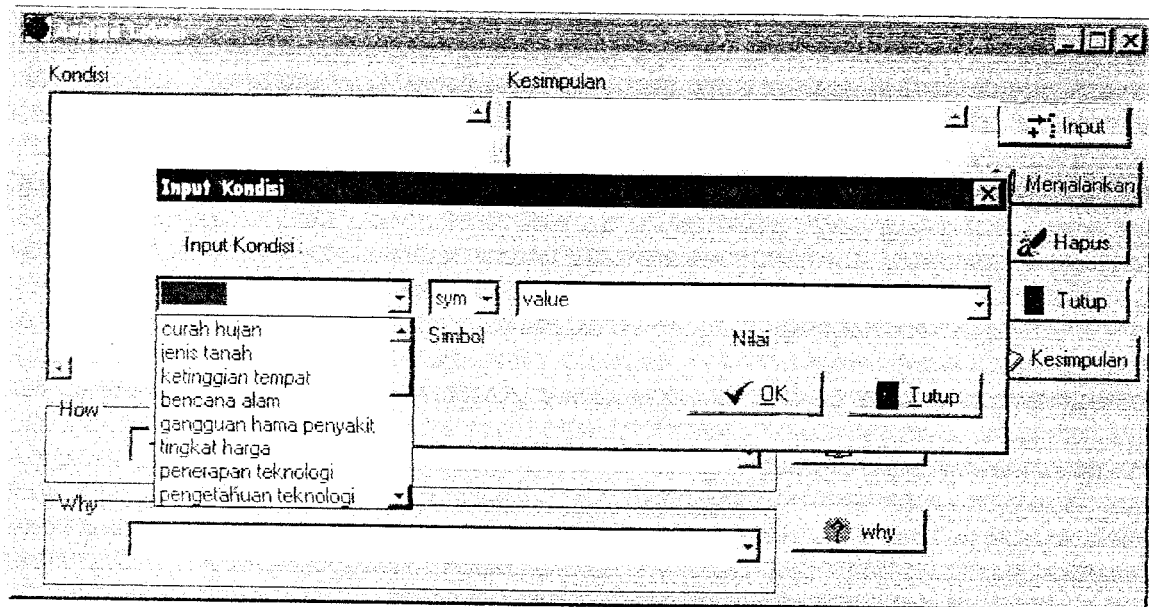
Penerapan Teknologi []

pengetahuan []

OK Clear Close

Clause	Symbol	Value

Gambar C.9. Antar muka untuk Input topoklimatologi

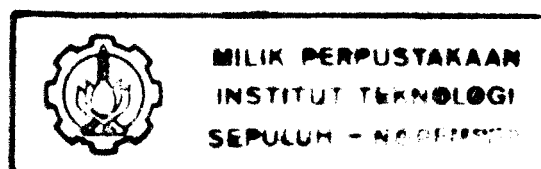


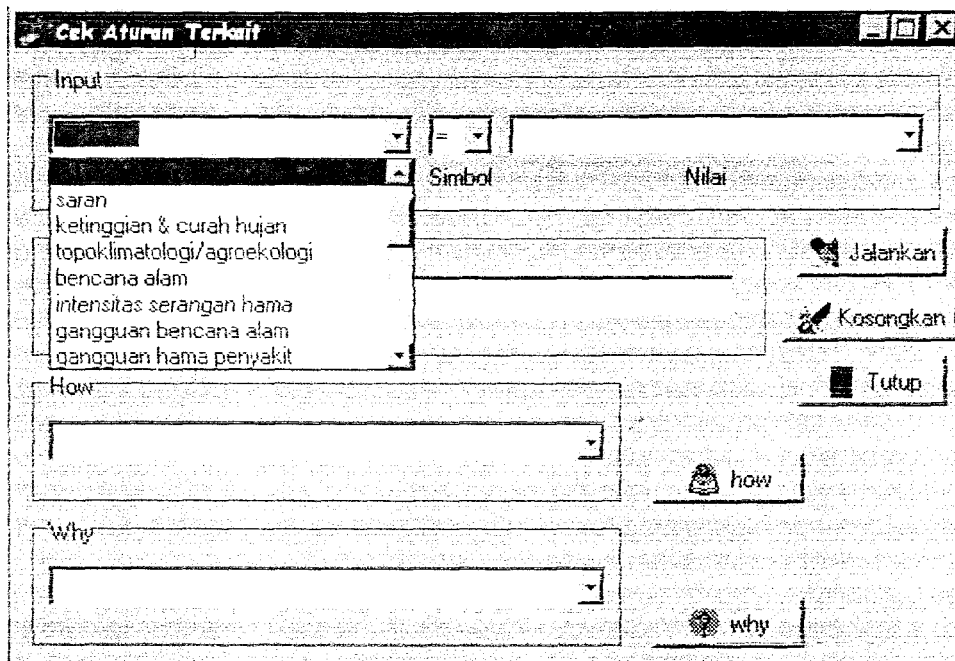
Gambar C.10. Antar muka untuk menginputkan kondisi

Untuk menganalisa informasi topoklimatologi pengguna dapat melakukannya dengan dua cara pertama dengan melalui 'data' dan kemudian memilih 'input topoklimatologi' maka akan tampak antar muka seperti pada gambar C.9, sebelum semua input yang lalu dibersihkan dari basis input dengan menggunakan tombol 'clear' pengguna tidak dapat menggunakan memasukkan inputannya. Setelah semua input yang lalu dihilangkan pengguna dapat memasukkan salah satu data topoklimatologi yang dimiliki. Ketika pemakai mengklik 'OK' maka sistem akan membawa kelembar 'forward' pemakai harus mengawali dengan mengklik tombol 'input' yang kemudian akan muncul antar muka konfirmasi seperti yang terlihat pada gambar C.7 untuk menanyakan apakah pemakai akan menggunakan data input yang telah ada? Konfirmasi tersebut muncul karena sebelumnya pemakai telah memasukkan informasi analisa data

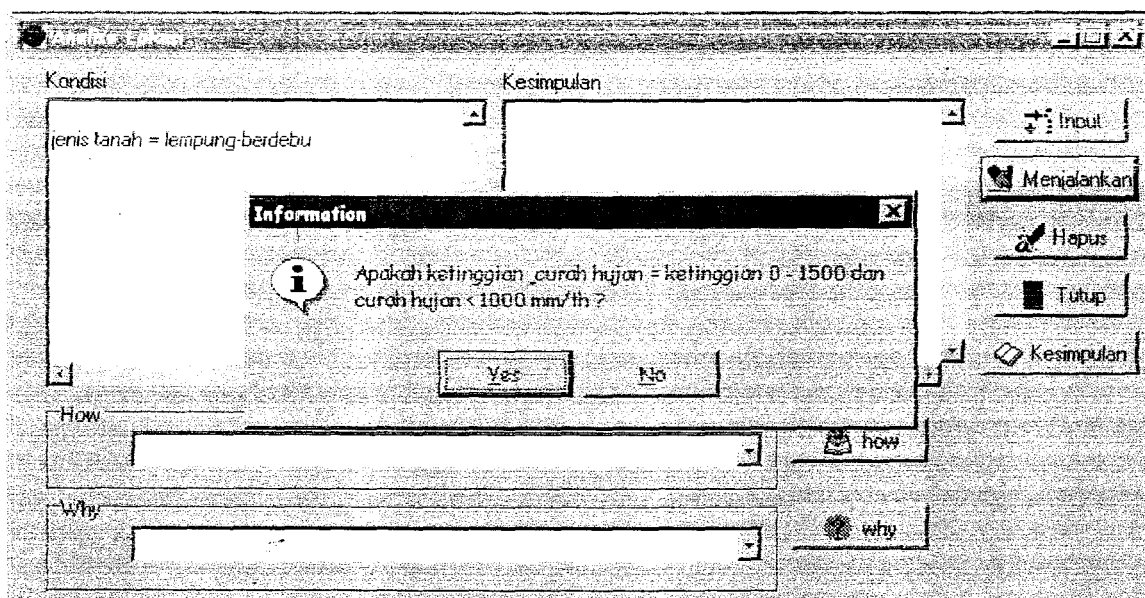
kedalam sistem. Karena pemakai ingin memperoleh saran terhadap kondisi harga yang dimiliki maka harus menjawab 'yes'. Kemudian klik tombol 'run' maka akan muncul beberapa pertanyaan seperti yang terlihat pada gambar C.8 yang perlu dijawab pemakai untuk membantu sistem ini memperoleh kesimpulan.

Cara yang kedua dengan langsung melalui bagian 'menjalankan' pada menu utama dan pilih 'forward' sehingga muncul antar muka seperti pada gambar C.6. Pada cara yang kedua inipun dapat dipergunakan untuk inputan selain topoklimatologi selama fakta yang diinputkan tersebut ada dalam basis atribut dan value yang dimiliki sistem ini. Pemakai harus mengawali dengan mengklik tombol 'input' yang kemudian akan muncul antar muka konfirmasi seperti yang terlihat pada gambar C.7 untuk menanyakan apakah pemakai akan menggunakan data input yang telah ada? Konfirmasi tersebut muncul karena sebelumnya pemakai telah memasukkan informasi analisa data kedalam sistem. Karena pemakai tidak memasukkan kondisi apapun sebelumnya, berarti data yang berada di basis input bukan data yang dimaksud pemakai, maka pemakai harus menjawab 'no'. Sehingga akan muncul antar muka seperti pada gambar C.10. Pada antar muka ini pemakai harap mengisikan salah satu kondisi yang diketahuinya, kemudian klik 'OK'. Setelah menginputkan fakta yang diketahui sudah dilakukan kemudian klik tombol 'run' maka akan muncul beberapa pertanyaan seperti yang terlihat pada gambar C.8 yang perlu dijawab pemakai untuk membantu sistem ini memperoleh kesimpulan.





Gambar C.11. Antar muka untuk proses backward



Gambar C.12. Antar muka untuk konfirmasi kondisi

Apabila pengguna ingin mengetahui fakta-fakta yang mendukung suatu kesimpulan dan memcocokkannya dengan yang dimilikinya pemakai dapat menggunakan 'backward' yang antarmukanya tampak seperti

gambar C.11. Pemakai harus memasukkan kesimpulan yang ingin dicek faktor-faktor yang mendukung kesimpulan itu diambil pada group input. Kemudian muncul antar muka untuk konfirmasi tentang kondisi yang dimiliki oleh pemakai yang tampak seperti gambar C.12 pemakai tinggal menjawab 'yes' dan 'no' saja sesuai dengan kondisi yang dimilikinya tidak ada jawaban tidak tahu jadi pemakai harus benar-benar mengetahui faktanya. Kesimpulan akhirnya adalah informasi apakah kesimpulan yang diambil itu benar atau salah.

Jika pemakai ingin mengetahui mengapa ditarik suatu kesimpulan tertentu dari fakta yang dimiliki, pemakai dapat mengklik tombol 'how' atau 'why' pada bagian 'forward' atau 'backward', pada bagian 'backward' pemakai dapat menggunakan penjelasan 'how' dan 'why' apabila kesimpulan yang diambil benar. Antar muka bagian 'how' dan 'why' berada pada antarmuka 'forward' dan 'backward' yang tampak pada gambar C11 dan C.6.

