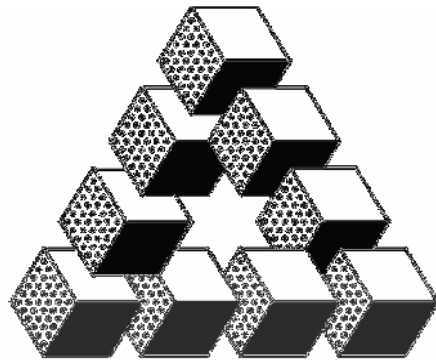


**RANCANG BANGUN APLIKASI SISTEM PAKAR UNTUK PENENTUAN
PRODUK DAN JENIS PERAWATAN TUBUH
PADA KLINIK KECANTIKAN**



STIKOM

JURUVERTAS
Dinamika

Oleh:

Nama : FEMHI NINDYA PUTRI

NIM : 01.41010.0148

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2006



**RANCANG BANGUN APLIKASI SISTEM PAKAR UNTUK PENENTUAN
PRODUK DAN JENIS PERAWATAN TUBUH
PADA KLINIK KECANTIKAN**

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



UNIVERSITAS

Oleh:

Nama : FEMHI NINDYA PUTRI

NIM : 01.41010.0148

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER**

SURABAYA

2006

**RANCANG BANGUN APLIKASI SISTEM PAKAR UNTUK PENENTUAN
PRODUK DAN JENIS PERAWATAN TUBUH
PADA KLINIK KECANTIKAN**

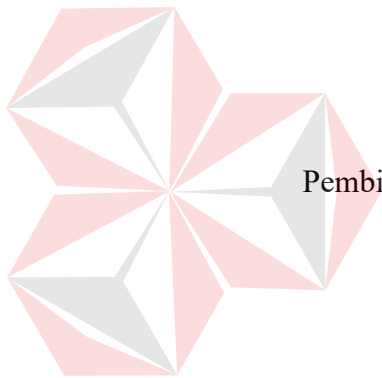
Disusun Oleh :

Nama : Femhi Nindya Putri

NIM : 01.41010.0148

Surabaya, Desember 2006

Telah diperiksa, diuji dan disetujui :



Pembimbing I

Pembimbing II

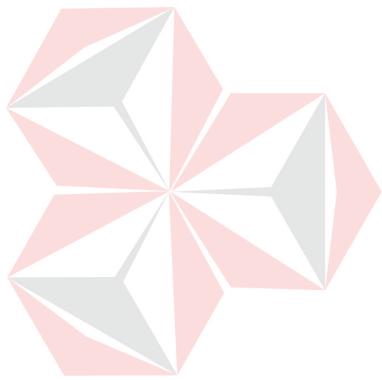
Ahmad Saikhu, S.Si, M.T
NIDN. 0718077101

Asti Dwi Irfianti, M.Kom
NIDN. 0717027301

Mengetahui:

Wakil Ketua Bidang Akademik

Drs. Antok Supriyanto, M. MT
NIDN. 0726106201



UNIVERSITAS
Dinamika

Learn from your failed, be optimistic and never give up.....



Karya ini kupersembahkan kepada :

Tuhan Yesus sebagai bukti nyata karya kasihNya,

kedua orang tuaku tercinta,

saudara-saudara yang mengasihiku,

serta sahabat-sahabatku,

sehingga karya ini dapat terselesaikan dengan baik,

UNIVERSITAS
Dinamika

ABSTRAKSI

Setiap individu terutama wanita tentunya ingin memiliki kulit yang sehat. Adanya jenis kulit, kondisi kulit, dan permasalahan kulit yang bermacam-macam dapat menyebabkan kulit menjadi tidak sehat dan tidak tampak bersinar. Dan ketika kita ingin mengatasi berbagai permasalahan kulit itu, kita sering merasa malas. Salah satu penyebabnya adalah pada saat kita datang pada salah satu klinik kecantikan, kita kurang merasa terpuaskan karena penanganan yang terlalu lama atau antrian yang panjang, dimana hal tersebut dikarenakan dokter yang masih sibuk menangani pasien yang lain atau dokter yang terlambat datang.

Salah satu alternatif pemecahan untuk mengatasi hal tersebut adalah dengan penggunaan sistem pakar yang dapat memberikan solusi berupa produk dan jenis perawatan yang tepat berdasarkan dari karakteristik setiap pasien.

Dengan penggunaan sistem pakar yang menggunakan metode *Forward Chaining* ini diharapkan dapat membantu unit pegawai di klinik kecantikan tersebut dalam mendiagnosa jenis kulit dan permasalahan kulit pasien dengan lebih akurat dan cepat sehingga dapat ditentukan produk-produk yang harus digunakan dan jenis perawatan yang harus dilakukan, tanpa menunggu antrian pemeriksaan dari dokter yang ada.

KATA PENGANTAR

Dengan rasa syukur kepada Tuhan Y.M.E, penulis telah dapat menyelesaikan Tugas Akhir yang merupakan persyaratan dalam menyelesaikan Program Studi Strata Satu di Sekolah Tinggi Manajemen Informatika dan Teknik Komputer Surabaya.

Tugas Akhir ini membahas tentang penggunaan sistem pakar untuk menentukan produk dan jenis perawatan tubuh pada klinik kecantikan “London Beauty Centre” dengan menggunakan metode *Forward Chaining* di dalam aplikasi sistem pakar dan juga menggunakan *Certainty Factor* untuk menghitung tingkat kepastian. Sistem pakar ini mempunyai kemampuan untuk menentukan jenis kulit dari seorang pasien, serta berdasarkan dari kondisi tubuh dan usia pasien tersebut maka sistem mampu untuk menentukan produk dan jenis perawatan tubuh yang tepat bagi pasien.

Selama menyelesaikan Tugas Akhir ini penulis dibantu oleh banyak pihak baik secara moril, materiil, maupun secara rohani berupa doa. Pada kesempatan ini, penulis menyampaikan rasa penghargaan dan terima kasih kepada yang terhormat:

- a. Bapak Ahmad Saikhu, S.Si., M.T sebagai dosen pembimbing I atas segala arahan, bimbingan, dan dorongan yang diberikan.
- b. Ibu Asti Dwi Irfianti, M.Kom sebagai dosen pembimbing II atas segala saran, dukungan, dan bimbingannya.
- c. Seluruh Bapak dan Ibu Dosen yang telah memberikan banyak pengetahuan dan pengalaman.

- d. Kedua orang tuaku untuk segala kasih, pengertian, dan tumpahan perhatian yang selalu dicurahkan kepada penulis setiap saat.
- e. Untuk kekasihku tercinta yang selalu memberikan dukungan dan motivasi baik secara mental maupun spiritual, serta yang telah memberikan waktu dan tenaganya untuk menemaniku.
- f. Untuk temanku Bagus, Dhesy, Meryanti, Genny, Yanti, Dharma, dan semua pihak yang telah membantu dan memberikan dukungan serta saran dalam penyelesaian Tugas Akhir ini.

Penulis berdoa kiranya Tuhan akan memberkati semua pihak yang telah banyak memberikan bantuan, bimbingan, doa, dorongan, dan nasehat.

Penulis menyadari bahwa masih banyak kekurangan pada penulisan Tugas Akhir ini, sebab kesempurnaan hanya milik Allah semata. Namun demikian besar harapan penulis agar Tugas Akhir ini dapat ikut menunjang perkembangan ilmu pengetahuan, khususnya ilmu komputer.

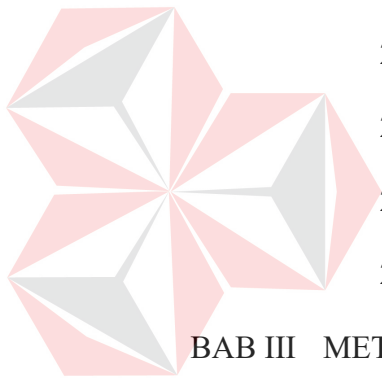
Surabaya, Desember 2006

Penulis

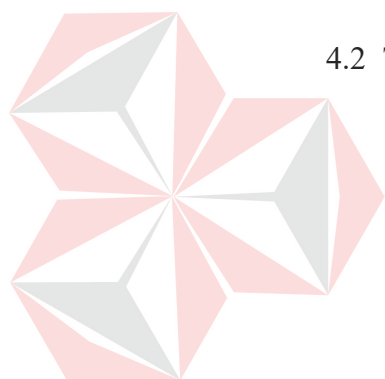
DAFTAR ISI

	Halaman
ABSTRAKSI	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Sistem Pakar	5
2.1.1 Konsep dasar sistem pakar	10
2.1.2 Sistem berbasis aturan	10
2.1.3 Komponen sistem berbasis aturan	11
2.1.4 Knowledge base	12
2.1.5 Working memory	13
2.1.6 Inference engine	14
2.1.7 Metode forward chaining	14
2.1.8 If-Then rules	25

2.1.9 Certainty factor	26
2.1.10 User interface	27
2.1.11 Verifikasi	27
2.2 Jenis dan Kondisi Kulit dari Tinjauan Ilmu Kecantikan	29
2.2.1 Jenis kulit	29
2.2.2 Kondisi kulit	31
2.2.3 Masalah kulit	32
2.2.4 Masalah bibir	33
2.2.5 Jenis kulit tangan	34
2.2.6 Kelainan-kelainan pada tangan dan kuku tangan	34
2.2.7 Perawatan tangan (manicure)	34
2.2.8 Jenis kulit kaki	35
2.2.9 Kelainan-kelainan pada kaki dan kuku kaki	35
2.2.10 Perawatan kaki (pedicure)	35
BAB III METODE PENELITIAN	36
3.1 Permasalahan	36
3.2 Analisa Sistem	36
3.3 Desain Sistem	37
3.3.1 Pengumpulan pengetahuan	38
3.3.2 Block diagram	39
3.3.3 Dependency diagram	48
3.3.4 Graphical representation	53
3.3.5 IF-THEN rules	72
3.3.6 Database	73



3.3.7 Inference engine	77
3.3.8 Faktor kepastian (CF)	79
3.3.9 Fasilitas edit	80
3.3.10 Verifikasi	83
3.3.11 User interface	83
BAB IV IMPLEMENTASI DAN EVALUASI.....	85
4.1 Implementasi	85
4.1.1 Kebutuhan perangkat lunak	85
4.1.2 Kebutuhan perangkat keras	85
4.1.3 Instalasi program dan pengaturan sistem	86
4.2 Tampilan Aplikasi	86
4.2.1 Tampilan form login	86
4.2.2 Tampilan form utama	87
4.2.3 Tampilan form perawatan wajah	89
4.2.4 Tampilan form perawatan mata	90
4.2.5 Tampilan form perawatan bibir	91
4.2.6 Tampilan form perawatan tangan	92
4.2.7 Tampilan form perawatan tubuh	94
4.2.8 Tampilan form perawatan kaki	95
4.2.9 Tampilan form rule	96
4.2.10 Tampilan form premis	98
4.2.11 Tampilan form verifikasi	99
4.2.12 Tampilan form detail penanganan	100
4.2.13 Tampilan form penanganan kondisi	102



UNIVERSITAS
Dinamika

4.2.14 Tampilan form manajemen user	103
4.2.15 Tampilan form ganti password	105
4.2.16 Tampilan form help	105
4.3 Evaluasi Sistem	106
BAB V PENUTUP	110
5.1 Kesimpulan	110
5.2 Saran	110
DAFTAR PUSTAKA	111
BIODATA PENULIS	112
LAMPIRAN	113



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 2.1 Perbandingan antara seorang pakar dan sistem pakar	5
Tabel 2.2 Inisialisasi tabel untuk metode forward chaining	20
Tabel 2.3 Memberikan sebuah value dari sebuah atribut premis	20
Tabel 2.4 Memeriksa kebenaran dari kalusa premis	21
Tabel 2.5 Memberikan tanda M (Marked) pada rule pertama yang statusnya U (Unmarked) dan A (Active)	21
Tabel 2.6 Menghilangkan tanda M (Marked) pada rule dan kembali ke langkah ketiga	22
Tabel 2.7 Mengulangi proses untuk atribut kedua	23
Tabel 2.8 Memulai pemeriksaan untuk atribut ketiga	23
Tabel 2.9 Pada rule 3, statusnya TD (Triggered) bila status semua klausa premisnya adalah benar, TU (True Clause)	24
Tabel 2.10 Bentuk final sebuah tabel yang telah memberikan kesimpulan akhir	24
Tabel 3.1 Tabel jika	74
Tabel 3.2 Tabel maka	74
Tabel 3.3 Tabel detil_penanganan	75
Tabel 3.4 Tabel penanganan	76
Tabel 3.5 Tabel login	76

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Konsep dasar sistem pakar	10
Gambar 2.2 Arsitektur komponen sistem	11
Gambar 3.1 Block diagram untuk perawatan tubuh	40
Gambar 3.2 Block diagram untuk produk perawatan tubuh	41
Gambar 3.3 Block diagram untuk perawatan mata.....	42
Gambar 3.4 Block diagram untuk perawatan tangan.....	43
Gambar 3.5 Block diagram untuk perawatan kaki.....	44
Gambar 3.6 Block diagram untuk produk perawatan bibir.....	45
Gambar 3.7 Block diagram untuk perawatan wajah	46
Gambar 3.8 Block diagram untuk produk perawatan wajah	47
Gambar 3.9 Dependency diagram untuk perawatan tubuh	48
Gambar 3.10 Dependency diagram untuk produk perawatan tubuh	49
Gambar 3.11 Dependency diagram untuk perawatan mata	49
Gambar 3.12 Dependency diagram untuk produk perawatan mata	49
Gambar 3.13 Dependency diagram untuk perawatan tangan	50
Gambar 3.14 Dependency diagram untuk perawatan kaki	50
Gambar 3.15 Dependency diagram untuk produk perawatan bibir	51
Gambar 3.16 Dependency diagram untuk perawatan wajah	51
Gambar 3.17 Dependency diagram untuk produk perawatan wajah	52
Gambar 3.18 Graphical representation untuk perawatan mata	53
Gambar 3.19 Graphical representation untuk produk perawatan mata	53

Gambar 3.20	Graphical representation untuk produk perawatan bibir	53
Gambar 3.21	Graphical representation untuk perawatan tubuh	54
Gambar 3.22	Graphical representation untuk produk perawatan tubuh	55
Gambar 3.23	Graphical representation untuk perawatan tangan	56
Gambar 3.24	Graphical representation untuk perawatan kaki	57
Gambar 3.25	Graphical representation untuk perawatan wajah	59
Gambar 3.26	Graphical representation untuk produk perawatan wajah	67
Gambar 3.27	Diagram alir proses kerja inference engine	78
Gambar 4.1	Tampilan form login	87
Gambar 4.2	Tampilan form utama	88
Gambar 4.3	Tampilan form perawatan wajah	90
Gambar 4.4	Tampilan form perawatan mata	91
Gambar 4.5	Tampilan form perawatan bibir	92
Gambar 4.6	Tampilan form perawatan tangan	93
Gambar 4.7	Tampilan form perawatan tubuh	95
Gambar 4.8	Tampilan form perawatan kaki	96
Gambar 4.9	Tampilan form rule	97
Gambar 4.10	Tampilan form premis	99
Gambar 4.11	Tampilan form verifikasi	100
Gambar 4.12	Tampilan form detail penanganan	102
Gambar 4.13	Tampilan form penanganan kondisi	103
Gambar 4.14	Tampilan form manajemen user	104
Gambar 4.15	Tampilan form ganti password	105

Gambar 4.16	Tampilan form help	106
Gambar 4.17	Pemilihan kriteria	108
Gambar 4.18	Hasil proses inferensi	108



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1	Rules sistem pakar lanjutan dari BAB III 113
Lampiran 2	Nama produk dan jenis perawatan “London Beauty Centre”..... 127



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Setiap individu di dunia ini terutama wanita tentunya ingin memiliki penampilan yang menarik. Penampilan yang menarik tidak hanya dilihat dari pakaian dan pernak-pernik yang dipakai, melainkan ditunjang dengan adanya kulit yang sehat. Oleh karena itu, sangatlah penting merawat kulit sedari dini untuk senantiasa menjaga dan merawat kulit agar tampak bersih, sehat, dan bercahaya.

Selama perjalanan hidup, kulit mengalami berbagai perubahan oleh karena pengaruh buruk lingkungan luar maupun karena perubahan dari dalam tubuh sendiri. Pengaruh buruk lingkungan luar misalnya karena pengaruh sinar matahari, iklim/cuaca, penggunaan AC, polusi, trauma pada kulit, dan penggunaan produk perawatan/tata rias yang tidak sesuai. Sedangkan pengaruh dari dalam antara lain karena pengaruh perubahan hormonal pada saat pubertas, kehamilan, menstruasi, pil KB, pengaruh zat gizi (nutrisi), dan berbagai macam penyakit. Beberapa faktor yang menyebabkan perubahan pada kulit tersebut ada yang dicegah dan ada yang tidak.

Mengingat banyaknya faktor yang dapat mengakibatkan kulit menjadi bermasalah, maka untuk merawat dan mengatasi kulit bermasalah diperlukan suatu produk dan jenis perawatan yang tepat.

Karena adanya jenis kulit, kondisi kulit dan permasalahan kulit yang bermacam-macam tersebut, serta untuk menghindari pasien yang antri terlalu lama dikarenakan dokter terlambat datang atau terlalu sibuk, maka salah satu

alternatif pemecahannya adalah dengan menggunakan suatu sistem pakar berbasis aturan yang dapat membantu unit pegawai di klinik kecantikan tersebut dalam menentukan jenis kulit pasien sehingga sistem pakar berbasis aturan ini pada nantinya mampu memberikan hasil berupa kombinasi produk dan jenis perawatan yang tepat bagi pasien berdasarkan jenis dan kondisi kulitnya.

Klinik kecantikan “London Beauty Centre” ini hadir sebagai salah satu klinik kecantikan dengan menerapkan metode kedokteran modern dari London Research Centre Singapura dan siap untuk membantu setiap individu untuk mendapatkan kulit sehat seperti yang mereka idamkan. Maka untuk meningkatkan pelayanan terhadap pasien, dibuatlah suatu aplikasi sistem pakar yang dapat membantu unit pegawai “London Beauty Centre” dalam menentukan kombinasi produk dan jenis perawatan tubuh pada klinik kecantikan tersebut sehingga pasien semakin terpuaskan.

1.2 Perumusan Masalah

Bagaimana merancang suatu aplikasi sistem pakar yang dapat membantu unit pegawai “London Beauty Centre” dalam menentukan kombinasi produk dan jenis perawatan tubuh pada klinik kecantikan “London Beauty Center”?

1.3 Pembatasan Masalah

Pembatasan masalah untuk Tugas Akhir ini adalah:

- a. Perawatan tubuh yang disediakan oleh klinik kecantikan “London Beauty Centre” meliputi perawatan wajah, mata, bibir, tubuh, kaki, dan tangan.
- b. Masalah-masalah kulit yang sering dijumpai di bidang perawatan kecantikan pada wanita, yaitu jerawat (*acne*), flek, dehidrasi, dan menua (*aging*).

- c. *Software* yang digunakan adalah Microsoft Visual Basic 6.0 untuk membuat *inference engine* dan Microsoft Access untuk menyimpan *database rules* serta *database* produk dan jenis perawatan.
- d. Metode yang digunakan dalam sistem ini adalah *Forward Chaining*.
- e. Menggunakan *Certainty Factor* untuk menghitung tingkat kepastian.
- f. User yang menggunakan sistem ini, yaitu unit pegawai di klinik kecantikan “London Beauty Centre”.

1.4 Tujuan

Tujuan dari sistem ini adalah:

- a. Merancang suatu aplikasi sistem pakar yang dapat digunakan untuk membantu unit pegawai di klinik kecantikan “London Beauty Centre” untuk mendiagnosa jenis kulit dan permasalahan kulit yang dihadapi oleh pasien “London Beauty Centre” dengan lebih akurat dan cepat sehingga dapat memberikan produk dan jenis perawatan yang tepat kepada pasien.
- b. Menerapkan metode *Forward Chaining* di dalam aplikasi sistem pakar dan juga menggunakan *Certainty Factor* untuk menghitung tingkat kepastian.

1.5 Sistematika Penulisan

Sistematika penulisan pada laporan ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini membahas tentang latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan dari pembuatan sistem ini, dan sistematika penulisan laporan Tugas Akhir.

BAB II : LANDASAN TEORI

Dalam bab ini, dibahas teori yang mendasari proses analisa, perancangan, dan pembuatan sistem yang mencakup teori tentang sistem pakar, teori tentang jenis, kondisi, dan masalah kulit, teori tentang desain sistem serta metode sistem pakar yang digunakan.

BAB III : METODE PENELITIAN

Pada bab ini dibahas proses perancangan dan pembuatan *software* aplikasi sistem pakar untuk menentukan produk dan jenis perawatan di klinik kecantikan “London Beauty Centre”.

BAB IV : IMPLEMENTASI DAN EVALUASI

Pada bab ini membahas tentang implementasi dari aplikasi yang dibuat secara keseluruhan. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui aplikasi tersebut telah dapat menyelesaikan permasalahan yang dihadapi sesuai dengan yang diharapkan.

BAB V : PENUTUP

Pada bab ini dibahas mengenai kesimpulan Tugas Akhir ini dan saran untuk pengembangan Tugas Akhir ini lebih lanjut.



UNIVERSITAS
Dindamika

BAB II

LANDASAN TEORI

2.1 Sistem Pakar

”Sistem pakar adalah suatu program komputer yang didesain untuk memodelkan/mengemulsi kemampuan seorang pakar dalam memecahkan suatu masalah” (Durkin, 1994:3).

Sistem pakar dimaksudkan untuk menyelesaikan masalah-masalah nyata dimana biasanya akan membutuhkan seorang pakar (seperti dokter, teknisi, dan lainnya). Sistem pakar menghasilkan hasil yang konsisten dan mempunyai kecepatan yang konstan daripada kepakaran seseorang. Ini disebabkan oleh beberapa hal yang bersifat manusiawi dimana dapat mempengaruhi pengambilan keputusan seorang pakar seperti lingkungan kerja, kondisi kesehatan, dan masalah pribadi.

Menurut Durkin (1994:8) ada beberapa keunggulan dan kelemahan sistem pakar dibandingkan dengan seorang pakar, hal ini dapat dilihat pada Tabel 2.1.

Tabel 2.1. Perbandingan antara seorang pakar dan sistem pakar

Faktor	Pakar	Sistem Pakar
Ketersediaan waktu	Hari kerja	Setiap saat
Geografis	Lokal/tertentu	Di mana saja
Keamanan	Tidak tergantikan	Dapat diganti
Dapat habis (<i>Perishable</i>)	Ya	Tidak
Performansi	Variabel	Konsisten
Kecepatan	Variabel	Konsisten
Biaya	Tinggi	Terjangkau

Dari perbandingan di atas, maka dapat diketahui alasan mengapa perlu membangun sebuah sistem pakar, yaitu:

1. Untuk menggantikan seorang pakar
 - a. Memungkinkan untuk mendapatkan kepakaran setelah jam kerja atau di lokasi lain.
 - b. Otomatisasi pekerjaan rutin yang membutuhkan pakar.
 - c. Pakar pensiun atau meninggal.
 - d. Pakar mahal.
 - e. Kepakaran yang diperlukan pada tempat-tempat berbahaya (*hostile environment*).
2. Untuk membantu seorang pakar
 - a. Membantu seorang pakar untuk mengerjakan tugas-tugas rutinnnya, sehingga pakar tersebut dapat meningkatkan produktivitasnya.
 - b. Membantu seorang pakar untuk menyelesaikan tugas-tugas sulit, sehingga pakar tersebut dapat lebih efektif dalam menangani masalah-masalah yang kompleks.
 - c. Menyediakan informasi yang sulit di ingat kembali oleh seorang pakar.

Menurut Ivan Bratko (1990), sistem pakar yang baik harus mampu memenuhi fungsi-fungsi sebagai berikut:

1. Penyelesaian masalah, kemampuan untuk meyelesaikan masalah dengan tepat pada ruang lingkup yang spesifik dengan informasi yang terbatas dan diperlukan (tidak harus lengkap).

2. Berinteraksi dengan *user*, sistem pakar harus dapat berinteraksi dengan *user* dan menjelaskan suatu sistem beserta kesimpulan-kesimpulan selama dan setelah proses penyelesaian masalah.

Hingga sekarang, sistem pakar telah banyak diaplikasikan untuk menyelesaikan masalah dalam bidang yang luas seperti pertanian, bisnis, ilmu kimia, komunikasi, sistem komputer, pendidikan, elektronik, teknik, obat-obatan, dan lain sebagainya.

Berdasarkan dari pemecahan masalah (*problem solving tasks*), sistem pakar dapat dikategorikan menjadi beberapa tipe, yaitu:

- a. Kontrol (*Control*)

Digunakan untuk mengontrol dan mengarahkan kerja sebuah sistem tertentu.

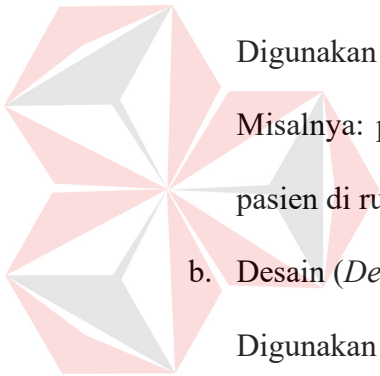
Misalnya: pengontrol mesin-mesin manufaktur atau *treatment* untuk seorang pasien di rumah sakit.

- b. Desain (*Design*)

Digunakan untuk mengkonfirmasi beberapa obyek dengan batasan-batasan masalah tertentu. Misalnya: mendesain sebuah sistem komputer dengan batasan-batasan yang didefinisikan oleh *user*, seperti: batasan harga, kecepatan, kemampuan multimedia, dan lainnya.

- c. Diagnosa (*Diagnosis*)

Digunakan untuk mendeteksi kerusakan di dalam sebuah sistem, dengan cara mengobservasi semua informasi yang didapat dari sistem tersebut. Misalnya: diagnosis penyakit seorang pasien berdasarkan atas gejala yang timbul.



UNIVERSITAS
Dinamika

d. Instruksi (*Instruction*)

Digunakan untuk mendeteksi dan memperbaiki kekurangan perilaku sistem dalam memahami bidang informasi tertentu. Misalnya: petunjuk belajar seorang siswa terhadap topik tertentu yang dipelajari.

e. Interpretasi (*Interpretation*)

Digunakan untuk menghasilkan sebuah pemahaman terhadap suatu situasi tertentu berdasarkan atas informasi yang dihasilkan dari sebuah situasi yang spesifik. Biasanya informasi-informasi tersebut berasal dari sensor, *instrument*, hasil tes, dan lainnya. Misalnya: monitor sensor mesin, sistem pencitraan, atau *speech analysis results*.

f. Pengamatan (*Monitoring*)

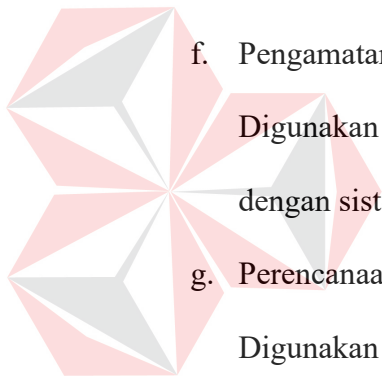
Digunakan untuk memonitor dan membandingkan cara kerja sebuah sistem dengan sistem yang diharapkan. Misalnya: kontrol instalasi nuklir.

g. Perencanaan (*Planning*)

Digunakan untuk melakukan aksi guna mencapai tujuan yang telah ditentukan dengan mengikuti batasan-batasan tertentu. Misalnya: merencanakan beberapa tugas yang berbeda untuk sebuah robot dalam industri manufaktur.

h. Prediksi (*Prediction*)

Digunakan untuk meramalkan aksi yang perlu dilakukan dimasa depan berdasarkan atas informasi yang ada dan model-model yang telah diberikan. Misalnya: prediksi kerusakan pada hasil panen yang diserang hama atau prediksi cuaca.



i. Preskripsi (*Prescription*)

Digunakan untuk menghasilkan rekomendasi solusi akan sebuah malfungsi yang terjadi pada suatu sistem. Misalnya: *Blue Box* yang merekomendasikan terapi untuk pasien yang menderita depresi.

j. Seleksi (*Selection*)

Digunakan untuk dapat mengidentifikasi pilihan terbaik dari suatu daftar kemungkinan atas dasar informasi yang didapat pada suatu kondisi tertentu.

k. Simulasi (*Simulation*)

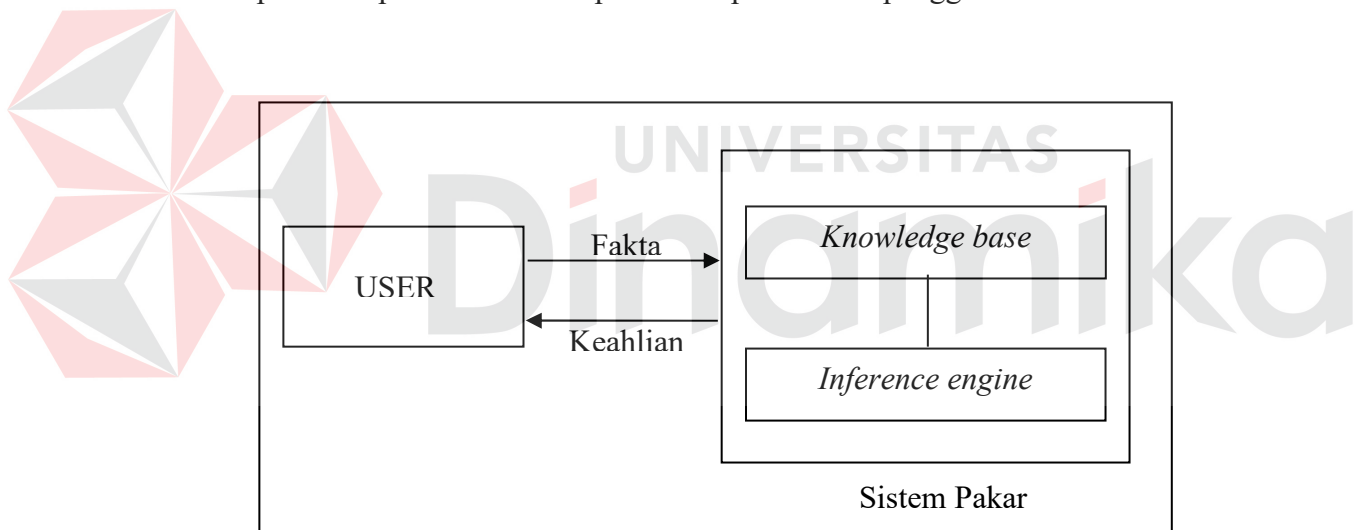
Digunakan untuk memodelkan sebuah proses atau sistem pada bermacam-macam kondisi. Misalnya: pembelajaran bagi para operator sebuah sistem sebelum diterjunkan langsung pada kenyataan.

Adapun langkah-langkah dalam merancang suatu sistem pakar yaitu sebagai berikut:

1. Menentukan batasan-batasan dari suatu sistem pakar yang akan dirancang.
2. Memilih jenis keputusan apa yang akan diambil.
3. Meng-*extract* pengetahuan dari pakar, caranya yaitu dengan *Dependency Diagram* dan *Graphical Representation*.
4. Merepresentasikan pengetahuan dalam sistem pakar (membuat *rules*), salah satu teknik yang dapat digunakan yaitu merepresentasikan ke dalam bentuk *IF-THEN rules*.
5. Membuat *inference engine* dengan menggunakan metode yang sesuai.
6. Merancang *user interface*.

2.1.1 Konsep dasar sistem pakar

Konsep dasar suatu sistem pakar, dapat di lihat pada Gambar 2.1. di bawah ini. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Ada dua kapabilitas penting dari seorang pakar (*expert*), yang dicoba untuk dimodelkan pada sistem pakar yaitu pengetahuan (*knowledge*) dan konsep berpikir (*reasoning*) dari sang pakar. Untuk menghasilkan kedua hal tersebut, sebuah sistem pakar harus terdiri dari dua komponen utama, yaitu *knowledge base* yang berisi *knowledge* dan *inference engine* yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respons dari sistem pakar atas permintaan pengguna.



Gambar 2.1. Konsep dasar sistem pakar

2.1.2 Sistem berbasis aturan

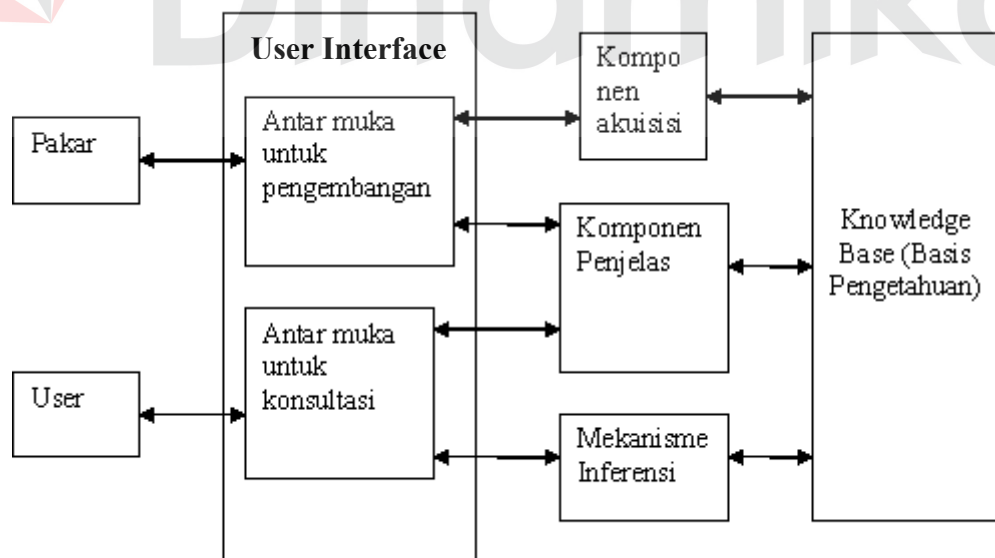
Sistem berbasis aturan merupakan suatu sistem pakar yang menggunakan aturan-aturan untuk menyajikan pengetahuannya, dengan kata lain sistem berbasis aturan ini merupakan suatu perangkat lunak yang menyajikan keahlian pakar dalam bentuk aturan-aturan pada suatu domain untuk menyelesaikan suatu

permasalahan. Menurut Encyclopaedia Britannica (2002), sistem pakar berbasis aturan biasanya dan idealnya, ditujukan untuk menggantikan seorang pakar dengan sebuah mesin.

2.1.3 Komponen sistem berbasis aturan

Sistem berbasis aturan disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar.

Komponen-komponen suatu sistem berbasis aturan dalam kedua bagian tersebut, secara umum dapat dilihat pada Gambar 2.2.



Gambar 2.2. Arsitektur komponen sistem

2.1.4 Knowledge base

“*Knowledge base* adalah bagian dari sistem pakar yang berisi sumber pengetahuan” (Durkin, 1994:28).

Knowledge base berisi pengetahuan spesifik yang disediakan oleh seorang pakar untuk memecahkan suatu masalah tertentu, dimana di dalamnya terdapat fakta, *rules*, konsep, dan hubungan antara *rules* maupun antara fakta dari suatu masalah tersebut.

Untuk mengkodekan pengetahuan dari seorang pakar ke dalam *knowledge base* dapat digunakan beberapa teknik. Salah satu teknik yang paling umum yang digunakan untuk menggambarkan pengetahuan dalam sistem pakar yaitu *rules*.

Pada sistem pakar setiap kasus akan diproses berdasarkan fakta yang ada, baru kemudian dapat diambil suatu kesimpulan. Apabila terjadi suatu kasus yang belum pernah ada, maka sistem pakar tidak mampu mengambil kesimpulan dari kasus tersebut. Untuk itu diperlukan penambahan fakta baru untuk *knowledge base*. Jadi, suatu *knowledge base* dikatakan baik apabila memiliki *rules* yang dapat digunakan dalam setiap kemungkinan kasus dalam ruang lingkup tertentu.

Berdasarkan dari *knowledge base* yang dimilikinya, sistem pakar dapat digolongkan menjadi 5 tingkatan, yaitu:

1. Tingkat 1
 - a. Menggunakan *internal knowledge base*.
 - b. Dapat melakukan penambahan data dan membuat laporan data yang ada pada *knowledge base*.

- c. Menggunakan metode *Backward Chaining* atau metode *Forward Chaining* untuk menyelesaikan suatu kasus.
 - d. Mampu menampilkan hasil kerja dari *inference engine* melalui *user interface*.
2. Tingkat 2
- a. Menggunakan *external knowledge base*.
 - b. Mampu memilih metode *Backward Chaining* atau *Forward Chaining* yang akan digunakan untuk menyelesaikan suatu kasus.
 - c. Mampu melakukan penghitungan matematis.
3. Tingkat 3
- a. mampu melakukan ekspor atau impor suatu *knowledge base*.
 - b. Mampu melakukan penghitungan matematis secara dinamis.
4. Tingkat 4
- a. Dapat dioperasikan pada berbagai *operating system*.
5. Tingkat 5
- a. Mampu merubah atau menambah isi *knowledge base* secara otomatis dan mempelajari suatu pola baru dari beberapa kasus yang pernah dialaminya.

2.1.5 Working memory

“*Working memory* adalah bagian dari sistem pakar yang berisi fakta-fakta dari suatu masalah yang ditemukan selama proses” (Durkin, 1994:29).

Working memory berisi semua informasi mengenai suatu masalah baik yang itu disediakan oleh *user* maupun dari kesimpulan yang diperoleh sistem. Banyak aplikasi sistem pakar dapat menggunakan informasi yang terdapat dalam penyimpanan eksternal seperti *database*, *spreadsheets*, atau *sensors*.

2.1.6 Inference engine

”*Inference engine* adalah *processor* dalam suatu sistem pakar dimana mencocokkan fakta-fakta yang terdapat dalam *working memory* dengan domain pengetahuan yang terdapat dalam *knowledge base*, untuk menggambarkan kesimpulan mengenai suatu masalah” (Durkin, 1994:29).

Inference engine akan mencari *rules* dimana premis dari *rules* tersebut cocok dengan informasi yang terdapat dalam *working memory*. Setelah menemukan kecocokan, *inference engine* akan menambahkan kesimpulan *rules* tersebut ke dalam *working memory* dan melanjutkan mencari *rules* untuk kecocokan yang baru.

2.1.7 Metode forward chaining

”*Forward Chaining* adalah strategi *inference* yang bermula dari sejumlah fakta yang diketahui, dengan menggunakan *rules* yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga *goal* dicapai atau hingga sudah tidak ada *rules* lagi yang premisnya cocok dengan fakta yang diketahui maupun fakta yang diperoleh” (Durkin, 1994).

Pada aplikasi *Forward Chaining* sederhana, *inference engine* menyalakan atau memilih *rule-rule* dimana bagian premisnya cocok dengan informasi yang ada pada *working memory*. Sistem pertama-tama memperoleh informasi masalah dari *user* dan menyimpannya dalam *working memory*. *Inference engine* lalu akan mencari *rules* pada beberapa urutan yang telah ditentukan sebelumnya, dimana premis-premisnya cocok dengan yang terdapat dalam *working memory*. Jika *rule* ditemukan, maka kesimpulan dari *rule* akan diinputkan ke dalam *working memory* (*rule fired*) lalu *cycles* dan cek lagi *rules*

untuk mencari kecocokan baru. Pada *cycle* berikutnya, *rules* yang sebelumnya telah *fired* diabaikan. Proses ini akan terus berlanjut hingga tidak ditemukan lagi adanya kecocokan. Dalam hal ini, *working memory* berisi informasi yang didapat dari *user* dan kesimpulan yang didapat dari sistem.

Karena metode *Forward Chaining* berangkat dari kiri ke kanan, yaitu dari premis menuju kepada kesimpulan akhir, maka seringkali pula disebut *data driven* (yaitu, pencarian dikendalikan oleh data yang diberikan). Metode ini lebih baik digunakan apabila memiliki sedikit premis dan banyak kesimpulan.

Menurut Durkin (1994) adapun kelebihan dan kelemahan metode *Forward Chaining* yaitu:

a. Kelebihan:

1. Kelebihan utama dari *Forward Chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan/menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.
2. Metode ini mampu menyediakan banyak sekali informasi dari hanya sejumlah kecil data.
3. Merupakan pendekatan paling sempurna untuk beberapa tipe dari *problem solving tasks*, yaitu *planning*, *monitoring*, *control*, dan *interpretation*.

b. Kelemahan:

1. Kelemahan utama dari metode ini yaitu kemungkinan tidak adanya cara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
2. Sistem bisa saja menanyakan pertanyaan yang tidak berhubungan. Walaupun jawaban dari pertanyaan tersebut penting, namun hal ini akan

membingungkan *user* untuk menjawab pertanyaan pada subjek yang tidak berhubungan.

Menurut Ignizio (1991) pada metode *Forward Chaining*, ada 2 cara yang dapat dilakukan untuk melakukan pencarian, yaitu:

- a. Pertama, dengan memasukkan semua data yang tersedia ke dalam sistem pakar pada satu kesempatan dalam sesi konsultasi. Cara ini banyak berguna pada sistem pakar yang termasuk dalam proses terautomatisasi dan menerima data langsung dari komputer yang menyimpan *database*, atau dari satu set sensor.
- b. Kedua, dengan hanya memberikan elemen spesifik dari data yang diperoleh selama sesi konsultasi kepada sistem pakar. Cara ini mengurangi jumlah data yang diminta, sehingga data yang diminta hanyalah data-data yang benar-benar dibutuhkan oleh sistem pakar dalam mengambil kesimpulan.

Menurut Ignizio (1991) algoritma untuk metode *Forward Chaining* adalah sebagai berikut:

1. Inisialisasi. Dapat dibuat tiga tabel kosong, yaitu tabel *Working Memory*, tabel *Attribute Queue*, dan tabel *Rule/Premis Status*. Tabel *Working Memory* berguna untuk menyimpan setiap *input*, yaitu semua fakta yang disimpulkan selama proses konsultasi. Tabel *Attribute Queue* berguna untuk menyimpan semua atribut dari *value* yang sedang diperiksa. Atribut pada awal tabel adalah atribut yang sedang menjalani proses pemeriksaan. Tabel *Rule/Premis Status* menyimpan status dari *rule-rule* yang ada yaitu *Active*, *Marked*, *Unmarked*, *Discarded*, *Triggered*, *Fired*. Status setiap premis pada saat inisialisasi adalah bebas (*Free*) dan status setiap *rule* pada saat inisialisasi adalah tidak bertanda

(*Unmarked*) dan aktif (*Active*). Notasi yang digunakan untuk merepresentasikan keadaan suatu *rule* dan premis adalah sebagai berikut:

A = *Active Rule*

D = *Discarded Rule*

U = *Unmarked Rule*

M = *Marked Rule*

TD = *Triggered Rule*

FD = *Fired Rule*

FR = *Free Clause*

FA = *False Clause*

TU = *True Clause*

2. Memulai proses pengambilan keputusan. Sebuah *value* dari sebuah atribut premis diambil. Di mana atribut tersebut tidak boleh ada pada klausa kesimpulan. Atribut ini disimpan pada bagian teratas tabel *Attribute Queue*.

Juga simpan atribut ini beserta *value*-nya pada bagian terbawah tabel *Working Memory*.

3. Penelitian satu per satu *rule* yang ada untuk memeriksa ada tidaknya kesamaan. Periksa tabel *Rule/Premis Status*, jika tidak ada *rule* yang statusnya '*Active*', pencarian dihentikan. Bila ada, dilakukan penelitian bagian klausa premis *rule* yang statusnya '*Active*' untuk mencocokkan klausa premis yang sesuai dengan *value* dari atribut pada bagian teratas tabel *Attribute Queue*. Simpan perubahan status klausa premis dari sekumpulan *rule* yang statusnya *Active*. Pada tabel *Rule/Premis Status* diberikan tanda FA (*False Clause*) pada status klausa premis yang bernilai salah dan tanda TU (*True Clause*) pada

status klausa premis yang bernilai benar. Periksa status *Rule* pada tabel *Rule/Premis Status*:

- a. Bila ada premis dari sebuah *rule* yang bernilai salah, maka diberi tanda D (*Discarded*) pada *rule* tersebut untuk menunjukkan bahwa *rule* tersebut bernilai salah dan tidak dipakai lagi. Langkah tersebut dilakukan pada setiap *rule* yang memiliki premis yang bernilai salah.
 - b. Bila ada sebuah *rule* yang semua premisnya dari bernilai benar, diberi tanda TD (*Triggered*) pada *rule status*. Kemudian dilanjutkan ke langkah 3c.
 - c. Bila tidak ada *rule* yang statusnya TD (*Triggered*), dilanjutkan ke langkah ke-5, bila ada satu atau lebih *rule* yang statusnya TD (*Triggered*), dilanjutkan ke langkah ke-4.
4. *Rule firing* atau menyatakan *rule* tersebut benar dan mengambil klausa kesimpulan *rule* tersebut sebagai kesimpulan akhir. Coretlah atribut pada bagian teratas tabel *Attribute Queue*, kemudian status *rule* tersebut diganti dan ditempatkan atribut kesimpulan pada bagian terbawah tabel *Attribute Queue* dan *value*-nya pada tabel *Working Memory*.
 5. Status antrian. Bagian teratas tabel *Attribute Queue* dicoret.
 6. Menandai *rule*. Telitilah kumpulan *rule active* untuk mencari *rule* yang statusnya U (*Unmarked*) dan A (*Active*). Bila tidak ditemukan, pencarian dihentikan. Bila ada, *rule* pertama yang ditemui ditandai dengan M (*Marked*).
 7. *Query*. Pada *rule* yang baru saja diberi tanda M (*Marked*), ditanyakan pada *user* untuk memperoleh *input*. Apabila *user* memberikan jawaban, dilanjutkan ke langkah 8. Sedangkan bila *user* tidak memberikan jawaban atau bila atribut

rule yang ditanyakan tersebut tidak memerlukan jawaban dari *user*, dilanjutkan langkah ini pada setiap klausa premis pada *rule* yang diberi tanda M (*Marked*) tersebut. Apabila setiap klausa premis pada *rule* yang diberi tanda M (*Marked*) tersebut telah diperiksa, kembali ke langkah 6.

8. Menghilangkan tanda M (*Marked*) pada *rule*. Atribut dan nomor *rule* diletakkan pada bagian teratas tabel *Attribute Queue*. Atribut ini diletakkan juga beserta *value*-nya pada bagian terbawah tabel *Working Memory*. Pada *rule* yang baru saja diberi tanda M (*Marked*) diberikan tanda U (*Unmarked*), dan kembali ke langkah 3.

Untuk memperjelas algoritma metode *Forward Chaining*, dapat diambil contoh sebagai berikut:

Dalam menentukan jenis sebuah pesawat terbang yang disimpan pada suatu hangar, diperlukan *rule-rule* sebagai berikut:

Rule 1: If engine type is propeller

Then plane is C130

Rule 2: If engine type is jet

and wing position is low

Then plane is B747p

Rule 3: If engine type is jet

and wing position is high

and bulges are none

Then plane is C5A

Rule 4: If engine type is jet

and wing position is high

and bulges are aft wing

Then plane is C141

Berikut adalah contoh tabel yang disusun untuk menentukan jenis pesawat dari jenis mesin, posisi mesin dan tonjolan yang dimiliki berdasarkan contoh soal diatas.

Tabel 2.2. Inisialisasi tabel untuk metode forward chaining

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U	(1)-1	FR
(2)	A, U	(2)-1	FR
		(2)-2	FR
(3)	A, U	(3)-1	FR
		(3)-2	FR
		(3)-3	FR
(4)	A,U	(4)-1	FR
		(4)-2	FR
		(4)-3	FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.3. Memberikan sebuah value dari sebuah atribut premis

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U	(1)-1	FR
(2)	A, U	(2)-1	FR
		(2)-2	FR
(3)	A, U	(3)-1	FR
		(3)-2	FR
		(3)-3	FR

(4)	A,U	(4)-1 (4)-2 (4)-3	FR FR FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type		Engine Type = jet	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.4. Memeriksa kebenaran dari klausa premis

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR , FA
(2)	A, U	(2)-1 (2)-2	FR , TU FR
(3)	A, U	(3)-1 (3)-2 (3)-3	FR , TU FR FR
(4)	A,U	(4)-1 (4)-2 (4)-3	FR , TU FR FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type		Engine Type = jet	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.5. Memberikan tanda M (Marked) pada rule pertama yang statusnya U (Unmarked) dan A (Active)

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR , FA
(2)	A, U , M	(2)-1 (2)-2	FR , TU FR

(3)	A, U	(3)-1 (3)-2 (3)-3	FR , TU FR FR
(4)	A,U	(4)-1 (4)-2 (4)-3	FR , TU FR FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type		Engine Type = jet	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.6. Menghilangkan tanda M (Marked) pada rule dan kembali ke langkah ketiga

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR , FA
(2)	A, U , M , U, D	(2)-1 (2)-2	FR , TU FR , FA
(3)	A, U	(3)-1 (3)-2 (3)-3	FR , TU FR , TU FR
(4)	A,U	(4)-1 (4)-2 (4)-3	FR , TU FR , TU FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type Wing Position (rule 2)		Engine Type = jet Wing Position = high	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.7. Mengulangi proses untuk atribut kedua

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR, FA
(2)	A, U, M, U, D	(2)-1	FR, TU
		(2)-2	FR, FA
(3)	A, U, M	(3)-1	FR, TU
		(3)-2	FR, TU
		(3)-3	FR
(4)	A,U	(4)-1	FR, TU
		(4)-2	FR, TU
		(4)-3	FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type Wing Position (rule 2)		Engine Type = jet Wing Position = high	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.8. Memulai pemeriksaan untuk atribut ketiga

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR, FA
(2)	A, U, M, U, D	(2)-1	FR, TU
		(2)-2	FR, FA
(3)	A, U, M, U	(3)-1	FR, TU
		(3)-2	FR, TU
		(3)-3	FR
(4)	A,U	(4)-1	FR, TU
		(4)-2	FR, TU
		(4)-3	FR
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type Wing Position (rule 2) Bulges (rule 3)		Engine Type = jet Wing Position = high Bulges = none	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.9. Pada rule 3, statusnya TD (Triggered) bila status semua klausa premisnya adalah benar, TU (True Clause)

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR, FA
(2)	A, U, M, U, D	(2)-1	FR, TU
		(2)-2	FR, FA
(3)	A, U, M, U, TD	(3)-1	FR, TU
		(3)-2	FR, TU
		(3)-3	FR, TU
(4)	A,U, D	(4)-1	FR, TU
		(4)-2	FR, TU
		(4)-3	FR, FA
Attribute – Queue Table (attribute listing)		Working Memory Table (attribute = value)	
Engine Type Wing Position (rule 2) Bulges (rule 3) Plane (rule 3)		Engine Type = jet Wing Position = high Bulges = none	

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

Tabel 2.10. Bentuk final sebuah tabel yang telah memberikan kesimpulan akhir

Rule / Premise Table			
Rule Number	Rule Status	Premise Clause Number	Premise Clause status
(1)	A, U, D	(1)-1	FR, FA
(2)	A, U, M, U, D	(2)-1	FR, TU
		(2)-2	FR, FA
(3)	A, U, M, U, TD, FD	(3)-1	FR, TU
		(3)-2	FR, TU
		(3)-3	FR, TU
(4)	A,U, D	(4)-1	FR, TU
		(4)-2	FR, TU
		(4)-3	FR, FA

Attribute – Queue Table (attribute listing)	Working Memory Table (attribute = value)
Engine Type Wing Position (rule 2) Bulges (rule 3) Plane (rule 3)	Engine Type = jet Wing Position = high Bulges = none Planes = C5A

Sumber: Ignizio, James P. Introduction to Expert System: The Development and implementation of Rule-Based Expert System. Singapore: McGrawHill, 1991.

2.1.8 IF-Then rules

Rule dimana secara logika menghubungkan informasi di bagian *IF* dengan informasi di bagian *THEN*.

IF : Sebagai penanda awal kondisi pada sebuah *rule*.

THEN : Sebagai penanda awal kesimpulan pada sebuah *rule*.

ELSE : Sebagai penanda awal alternatif kesimpulan pada sebuah *rule*, bersifat *optional*.

Operator yang dapat digunakan pada *IF-THEN rule* adalah:

AND : Semua kondisi yang dihubungkan oleh operator *AND* harus bernilai benar, agar kondisi keseluruhan *rule* tersebut bernilai benar. Bila ada satu kondisi yang bernilai salah, keseluruhan *rule* tersebut bernilai salah.

OR : Semua kondisi yang dihubungkan oleh operator *OR* harus bernilai salah, agar kondisi keseluruhan *rule* tersebut bernilai salah. Bila ada satu kondisi atau lebih yang bernilai benar, keseluruhan *rule* tersebut bernilai benar.

Contoh *IF-THEN rules*:

Diagnosa masalah pada mobil

RULE 1

IF The car will not start

THEN The problem may be in the electrical system

RULE 2

IF The problem may be in the electrical system

AND The battery voltage is below 10 volts

THEN The fault is a bad battery

2.1.9 Certainty factor

Dalam kenyataan sehari-hari, pakar seringkali menjumpai fakta-fakta yang tidak menentu dan tidak pasti. Karena itu, sistem pakar yang dibuat harus dapat menangani masalah ini. Salah satu alternatif untuk menangani masalah ini adalah dengan penggunaan faktor kepastian. Faktor kepastian tersebut menggambarkan beragam tingkat dari kepastian pakar yang mempertimbangkan konklusi tersebut.

Menurut Gunawan (2000) ada 3 macam tingkat kepastian, yaitu:

a. Faktor kepastian *rule*

Faktor kepastian *rule* merupakan kepastian dari *knowledge engineer* akan kebenaran dari *rule* tersebut.

b. Faktor kepastian *premise*

Faktor kepastian *premise* didapat dari kepastian *user* akan kebenaran dari setiap jawaban yang diberikannya.

c. Faktor kepastian konklusi

Faktor kepastian konklusi merupakan kepastian akan konklusi yang dihasilkan, yang diperoleh dari hasil perhitungan faktor kepastian *premise* dan faktor kepastian *rule*.

2.1.10 User interface

User interface merupakan interaksi antara seorang pakar dengan *user* yang disajikan dalam *natural language style*, dimana *user* memasukkan *input* dan akan menerima *output*. *User interface* memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi. Pada umumnya, *user interface* juga berfungsi untuk menginputkan pengetahuan baru ke dalam basis pengetahuan sistem pakar, menampilkan fasilitas penjelasan sistem dan memberikan tuntunan penggunaan sistem secara menyeluruh langkah demi langkah sehingga *user* mengerti apa yang harus dilakukan terhadap sistem. Karena dimaksudkan untuk *user*, maka *interface* harus dirancang sebisa mungkin memberi kemudahan kepada *user* dalam mengoperasikan sistem dan sejelas mungkin sehingga tidak terjadi salah interpretasi. Jadi, *interface* haruslah *user friendly*.

2.1.11 Verifikasi

Suatu kualitas dari basis pengetahuan dapat dilihat dari ukuran, kompleksitas dan sifat kritikal dari aplikasi-aplikasi yang ada. Semuanya itu dapat diwujudkan dari proses-proses verifikasi. Verifikasi adalah membangun sistem yang benar yang terdiri dari 2 proses yaitu:

- a. Memeriksa pelaksanaan suatu sistem secara spesifik.
- b. Memeriksa konsistensi dan kelengkapan dari basis pengetahuan.

Verifikasi dijalankan ketika ada penambahan atau perubahan pada *rule*, karena *rule* tersebut sudah ada pada sistem. Berikut ini adalah yang harus dicek dalam suatu basis pengetahuan :

1. *Redundant Rules*

Dikatakan *redundant rule* jika dua *rule* atau lebih mempunyai *premise* dan *conclusion* yang sama.

2. *Conflicting Rules*

Terjadi ketika dua buah *rule* atau lebih mempunyai *premise* yang sama, tetapi mempunyai *conclusion* yang berlawanan.

3. *Subsumed Rules*

Terjadi jika *rule* tersebut mempunyai *constraints* yang lebih atau kurang tetapi mempunyai *conclusion* yang sama.

4. *Circular Rules*

Circular rules adalah suatu keadaan dimana terjadinya proses perulangan dari suatu *rule*. Ini dikarenakan suatu *premise* dari salah satu *rule* merupakan *conclusion* dari *rule* yang lain atau sebaliknya.

5. *Unnecessary IF Conditions*

Terjadi ketika dua *rule* atau lebih mempunyai *conclusion* yang sama, tetapi salah satu dari *rule* tersebut mempunyai *premise* yang tidak perlu dikondisikan dalam *rule* karena tidak mempunyai pengaruh apapun.

6. *Dead-End Rules*

Dead-end rules adalah suatu *rule* yang *conclusionnya* tidak diperlukan oleh *rule-rule* lainnya.

2.2 Jenis dan Kondisi Kulit dari Tinjauan Ilmu Kecantikan

Kulit adalah organ tubuh yang terletak paling luar. Kulit merupakan organ yang esensial dan vital serta merupakan cermin kesehatan dan kehidupan.

Kulit terbagi menjadi 3 lapisan utama, yaitu epidermis, dermis dan subcutis, serta memiliki berbagai aksesori pendukung seperti kuku, folikel rambut, kelenjar keringat, kelenjar minyak, pembuluh darah, pembuluh limfatik dan saraf.

Kulit sehat adalah kulit yang kenyal dan kencang, lembab, tampilan warna kulit sempurna, bersih dari noda, sedikit kerutan halus dan segar bercahaya.

2.2.1 Jenis kulit

Jenis kulit lebih bersifat menetap dan tidak mudah dipengaruhi oleh perubahan lingkungan luar. Ada 4 jenis kulit, antara lain:

1. Kulit normal

Kulit normal memiliki ciri-ciri yaitu:

- a. Tidak terlalu berminyak dan tidak terlalu kering.
- b. Tampak sehat dan segar, sedikit bersinar atau tampak transparan.
- c. Pori-pori hampir tidak kelihatan, kecuali pada area sentral yang tampak sedikit besar.
- d. Sedikit kerutan-kerutan pada mata, mulut dan leher.
- e. Lembut dan kenyal pada perabaan.
- f. Memberikan respon yang baik terhadap perawatan yang minimal.

2. Kulit berminyak

Kulit berminyak memiliki ciri-ciri yaitu:

- a. Tampak berminyak/berkilau segera setelah dibersihkan.
- b. Pori-pori tampak jelas/lebar dan biasanya tersumbat terutama di hidung, dagu dan dahi (*T zone*).
- c. Mempunyai kecenderungan berjerawat.
- d. Mempunyai sedikit kerutan kecuali garis-garis ekspresi.
- e. Sering terdapat sisik-sisik halus pada hidung (akibat pemakaian produk yang mengandung alkohol atau sabun yang berlebihan).
- f. Mempunyai tekstur kulit yang kasar (kulit jeruk) tetapi teraba lembut dan lentur.

3. Kulit kering

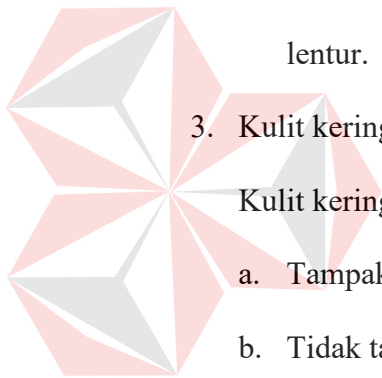
Kulit kering memiliki ciri-ciri yaitu:

- a. Tampak kusam, bersisik dan tipis.
- b. Tidak tampak berkilauan kecuali daerah hidung.
- c. Kerutan-kerutan halus sekitar mata, mulut dan leher tampak jelas.
- d. Tekstur teraba halus.
- e. Kulit terasa kencang/tertarik terutama setelah dibersihkan.
- f. Memberikan respons yang baik dengan krim perawatan yang mengandung minyak.

4. Kulit kombinasi

Kulit kombinasi memiliki ciri-ciri yaitu:

- a. Kombinasi normal cenderung kering.
- b. Kombinasi normal cenderung berminyak.



2.2.2 Kondisi kulit

Berbeda halnya dengan jenis kulit, kondisi kulit mudah berubah-ubah, karena dapat dipengaruhi oleh lingkungan luar maupun dari dalam tubuh sendiri.

Kondisi kulit ada 3 macam, yaitu:

1. Sensitif

Kondisi sensitif mempunyai ciri-ciri antara lain:

- a. Kondisi kulit seringkali dimiliki oleh orang yang mudah cemas.
- b. Biasanya kering tetapi juga dapat berminyak.
- c. Tampak kemerahan atau mudah menjadi merah karena sentuhan.
- d. Umumnya memiliki tekstur halus.
- e. Tidak tahan produk perawatan yang mengandung *astringent* ataupun krim yang terlalu berminyak.
- f. Umumnya berwarna putih/pucat.
- g. Sensitif terhadap dingin, panas atau angin.
- h. Seringkali mengeluh gatal.

2. Dehidrasi

Kondisi dehidrasi mempunyai ciri-ciri antara lain:

- a. Tampak garis-garis halus ada hampir seluruh wajah, dapat dijumpai pada jenis kulit berminyak sekalipun.
- b. Kulit wajah tampak kusam.
- c. Wajah terasa kaku/tertarik/kencang.
- d. Keadaan ini disebabkan karena kadar air yang rendah di dalam kulit.

3. Kulit menua (*aging skin*)

Kondisi kulit menua mempunyai ciri-ciri antara lain:

- a. Kulit tampak menipis.
- b. Disertai kekendoran pada kulit.
- c. Tampak adanya kerutan-kerutan halus maupun tajam.
- d. Timbul beberapa problem yang menyertai proses penuaan seperti adanya *hiperpigmentasi*, beberapa tumor jinak kulit seperti *seboroik keratosis* atau *skin tag*.
- e. Seringkali disertai dehidrasi.

2.2.3 Masalah kulit

Adapun macam-macam permasalahan kulit yang sering dijumpai di bidang perawatan kecantikan, yaitu:

- a. Jerawat (*acne vulgaris*) : sumbatan pada muara kelenjar minyak kulit yang dapat disertai peradangan.
- b. *Black head* : komedo terbuka, sering dijumpai di hidung.
- c. *White head* atau *milia* : komedo tertutup.
- d. *Flek/hiperpigmentasi* : noda atau bercak berwarna coklat atau kehitaman biasanya simetris, terdapat pada wajah akibat pigmentasi kulit yang berlebihan.
- e. *Vitiligo* : bercak (*makula*) berwarna putih.
- f. *Dermatitis* atau *eksema* : reaksi peradangan kulit terhadap adanya rangsangan dari dalam maupun luar tubuh.

- g. *Keloid* : pertumbuhan berlebihan dari jaringan ikat kulit, padat dan biasanya terbentuk setelah penyembuhan luka pada kulit. Pertumbuhannya meluas melebihi batas-batas luka asli.
- h. *Urtikaria (bidur)* : salah satu bentuk reaksi alergi dengan gambaran pembengkakan pada kulit dengan warna kemerahan dan disertai rasa gatal.
- i. *Xerosis* : keadaan kulit yang sangat kering, dapat disebabkan karena kekurangan minyak dan air.
- j. *Seboroik keratosis* atau *kutil* : tumor jinak kulit, berwarna coklat, datar, berbatas tegas dengan permukaan yang kasar.
- k. *Melasma (khloasma)* : bercak *hiperpigmentasi* yang sering didapatkan pada wajah, yaitu pada pipi, dahi, dagu, di atas bibir dan dapat meluas sampai ke leher.
- l. *Efelis* atau *frekles* : bercak *hiperpigmentasi* berukuran kecil (3-5 mm), berwarna coklat muda hingga coklat tua dan seringkali terdapat pada bagian yang terkena sinar matahari seperti wajah, lengan dan leher.
- m. *Pytiriasis alba* : bercak putih dengan batas tegas, yang lebih banyak dijumpai pada kulit dengan warna gelap, sampai sekarang tidak diketahui penyebabnya. Biasanya dijumpai pada umur 3-16 tahun.
- n. *Skin tag* : tumor jinak pada kulit yang berwarna seperti kulit serta bertangkai. *Skin tag* ini seringkali dijumpai di leher.
- o. *Xanthelasma* : timbunan lemak.

2.2.4 Masalah bibir

Adapun macam-macam permasalahan bibir yang sering dijumpai di bidang perawatan kecantikan, yaitu:

- a. Bibir kering.
- b. Bibir pecah-pecah.
- c. Bibir hitam, hal ini bisa disebabkan karena lipstik maupun bekas rokok.

2.2.5 Jenis kulit tangan

Kulit tangan dibedakan atas 3 jenis. Berikut jenis-jenis kulit tangan beserta cirinya masing-masing:

- a. Berminyak : mengkilat, tebal, kasar dan lengket.
- b. Normal : segar, bagus, kenyal, tidak bercacat.
- c. Kering : kusam, keriput, bersisik halus.

2.2.6 Kelainan-kelainan pada tangan dan kuku tangan

Beberapa kelainan-kelainan pada tangan dan kuku yang sering dijumpai adalah sebagai berikut:

- a. Telapak tangan kasar.
- b. Kapalan.
- c. Kuku tangan rapuh (mudah patah dan robek).

2.2.7 Perawatan tangan (manicure)

”Perawatan tangan atau *manicure* diambil dari bahasa Latin “*menusi*” yang artinya tangan dan “*cura*” yang artinya adalah perawatan. Sehingga *manicure* berarti perawatan tangan dimana termasuk kuku. Maksud *manicure* adalah untuk meningkatkan penampilan tangan dan kuku. Wajah yang cantik dan terpelihara akan kurang nilainya apabila tangan dan kukunya tidak terawat” (Kel, 2001).

2.2.8 Jenis kulit kaki

Kulit kaki dapat dibedakan menjadi 3 jenis dengan cirinya masing-masing, yaitu sebagai berikut:

- a. Berminyak : mengkilat, tebal, kasar dan lengket.
- b. Normal : segar, bagus, kenyal, tidak bercacat.
- c. Kering : kusam, keriput, bersisik halus.

2.2.9 Kelainan-kelainan pada kaki dan kuku kaki

Beberapa kelainan-kelainan yang sering dijumpai pada kaki dan kuku, yaitu sebagai berikut:

- a. Pecah-pecah pada tumit kaki : karena tebalnya dan keringnya lapisan tanduk telapak kaki, kulit tumit menjadi retak (*Regaden*).
- b. Kapalan.
- c. Mata ikan : pertandukan seperti duri ke dalam kulit karena tekanan menahun.
- d. Lepuh ringan : berupa gelembung yang berisi cairan.
- e. Kuku kaki rapuh.
- f. Jamuran (*Ring worm/Athlete's foot*).

2.2.10 Perawatan kaki (pedicure)

“Istilah *pedicure* diambil dari bahasa Latin “*pes*” yang artinya kaki, dan “*cura*” yang artinya adalah perawatan. Sehingga *pedicure* berarti perawatan kaki dan kuku dimana merupakan bagian dari peningkatan penampilan seseorang” (Kel, 2001).

BAB III

METODE PENELITIAN

3.1 Permasalahan

Permasalahan yang dihadapi dalam membangun sistem pakar yang dapat menentukan produk dan jenis perawatan tubuh pada klinik kecantikan “London Beauty Centre” dengan menggunakan *Forward Chaining* adalah bagaimana cara merancang dan membangun aplikasi sistem pakar tersebut sehingga dapat memberikan data mengenai jenis dan kondisi kulit pasien di klinik kecantikan “London Beauty Centre” sehingga dapat ditentukan produk dan jenis perawatan tubuh, yang dapat meliputi perawatan wajah, mata, bibir, badan, kaki, dan tangan, yang harus dilakukan oleh pasien di klinik kecantikan “London Beauty Centre” tersebut sesuai dengan kebutuhannya.

3.2 Analisa Sistem

Untuk membangun sistem pakar yang mampu menentukan produk dan jenis perawatan yang tepat terhadap pasien, maka dilakukan analisa terhadap permasalahan yang ada.

Analisa dilakukan dengan menggunakan data-data yang didapat dari hasil wawancara dengan dokter, terutama mengenai bagaimana menentukan jenis kulit, kondisi kulit pasien, dan parameter yang dibutuhkan untuk menentukan produk dan perawatan yang tepat, yang dapat digunakan untuk mendesain struktur sistem pakar sesuai dengan cara penentuan produk dan perawatan. Untuk memperkuat analisa, dilakukan studi literatur dengan menggunakan buku-buku yang terkait

dengan jenis kulit dan kondisi atau masalah kulit. Literatur sebagai pendukung wawancara yang telah dilakukan yaitu mengenai data produk dan perawatan.

Berdasarkan dari *knowledge base* yang dimilikinya, aplikasi sistem pakar ini dapat digolongkan sebagai aplikasi sistem tingkat 1, yaitu dengan kriteria:

- a. Menggunakan *internal knowledge base*. *Knowledge base* aplikasi sistem pakar untuk menentukan kombinasi produk dan perawatan yang sesuai ini terhubung langsung dengan *database* dan telah diprogramkan sebelumnya dan bersifat dinamis.
- b. Dapat melakukan penambahan, pengurangan maupun mengubah data yang ada pada *knowledge base*.
- c. *Inference engine* aplikasi sistem pakar untuk membantu menentukan produk dan jenis perawatan ini menggunakan metode *Forward Chaining*.
- d. Melakukan perhitungan *Certainty Factor* untuk tiap kesimpulan yang dihasilkan sistem.
- e. Mampu menampilkan hasil kerja dari *inference engine* melalui *user interface*.

Hasil atau kesimpulan nantinya akan disajikan dalam bentuk *form* berupa rekomendasi mengenai kombinasi produk dan jenis perawatan yang disediakan di klinik kecantikan “London Beauty Centre”, dimana yang sesuai untuk setiap orang.

3.3 Desain Sistem

Perancangan dan pembuatan aplikasi sistem pakar untuk menentukan produk dan jenis perawatan di klinik kecantikan “London Beauty Centre” ini dibagi menjadi 9 bagian utama, yaitu:

- a. Pengumpulan pengetahuan.
- b. Pembuatan *Block Diagram*.
- c. Pembuatan *Dependency Diagram* dan *Graphical Representation* sebagai landasan dalam membuat *knowledge based system*.
- d. Melakukan konversi dari *Dependency Diagram* dan *Graphical Representation* menjadi *IF-THEN Rules*, yang berperan penting dalam proses pengambilan keputusan.
- e. Perancangan *database* sebagai salah satu komponen penunjang *knowledge based system*.
- f. Perancangan *inference engine* yang menggunakan metode *Forward Chaining*, sebagai pusat pemrosesan aplikasi sistem pakar.
- g. Perhitungan *Certainty Factor* untuk tiap kesimpulan yang dihasilkan sistem.
- h. Perancangan fasilitas *edit* dan verifikasi
- i. Perancangan *user interface*.

Adapun penjelasan lebih lanjut dari perencanaan aplikasi sistem pakar untuk menentukan produk dan jenis perawatan di klinik kecantikan "London Beauty Centre" ini sebagai berikut:

3.3.1 Pengumpulan pengetahuan

Suatu sistem pakar memerlukan basis pengetahuan yang baik. Pengetahuan harus diperoleh, disusun dan direpresentasikan. Dalam sistem pakar, pengumpulan pengetahuan adalah hal yang terpenting dalam menentukan *knowledge based* aplikasi sistem pakar yang akan dibuat. Juga untuk menentukan batasan-batasan sistem pakar ini. Untuk itu diperlukan sumber bagi pembuatnya.

Dari sumber yang ada barulah data-data dapat diperoleh dengan lebih jelas dan lengkap, yaitu dengan cara:

- a. Mencari seorang pakar (*human expert*) atau orang yang ahli di bidangnya.

Yaitu dengan melakukan wawancara dengan dokter di klinik kecantikan “London Beauty Centre” mengenai masalah yang ada, kemudian mendaftar jawabannya.

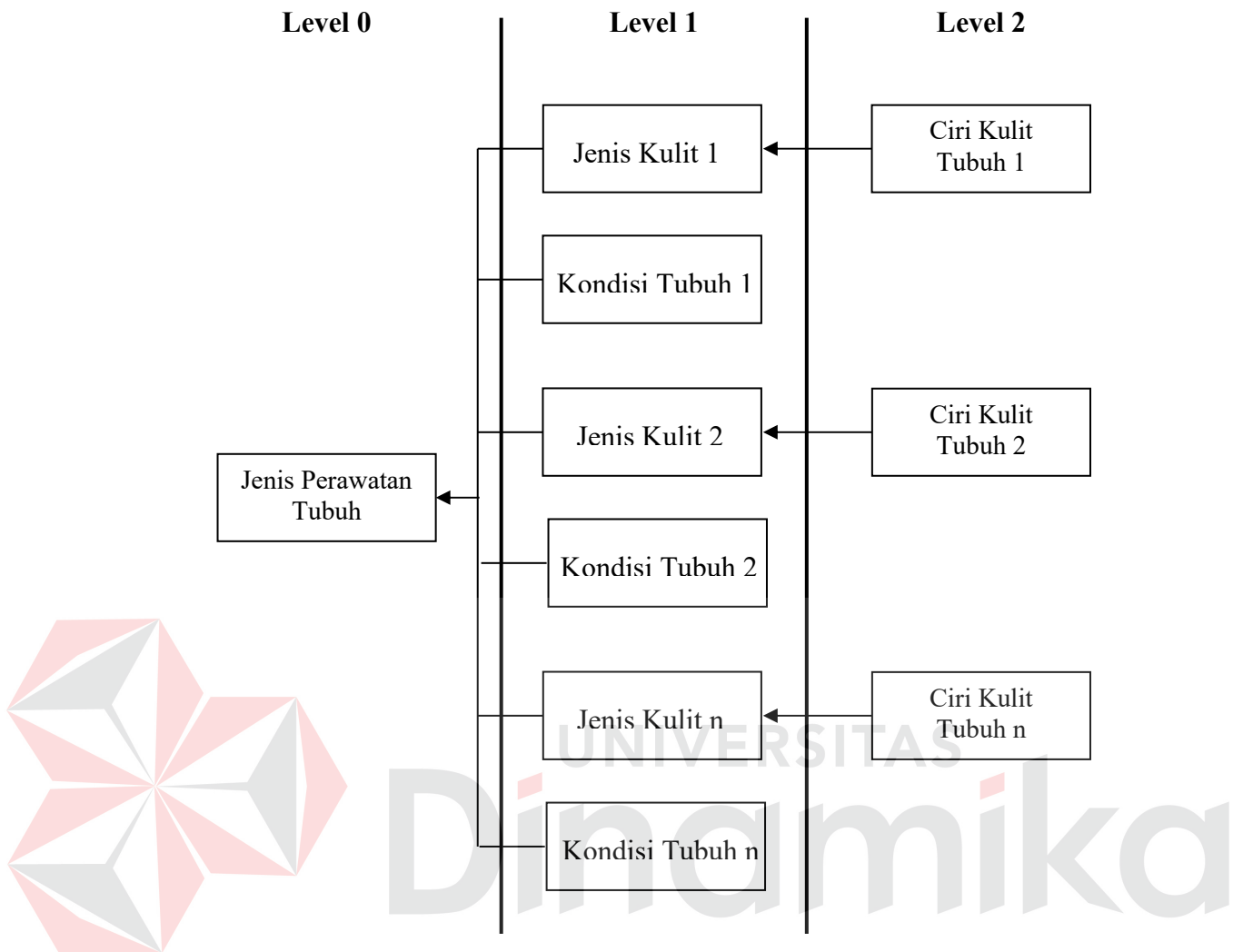
- b. Mencari buku-buku yang tepat.

Yaitu dengan mencari buku-buku yang sesuai, kemudian mempelajari teori-teori maupun masalah-masalah yang terdapat dalam buku tersebut.

Pengumpulan pengetahuan ini digunakan untuk menentukan ruang lingkup atau batasan-batasan yang diperlukan. Setelah mengumpulkan pengetahuan maka dapat dibuat *knowledge base* serta *rules* dari aplikasi sistem pakar ini.

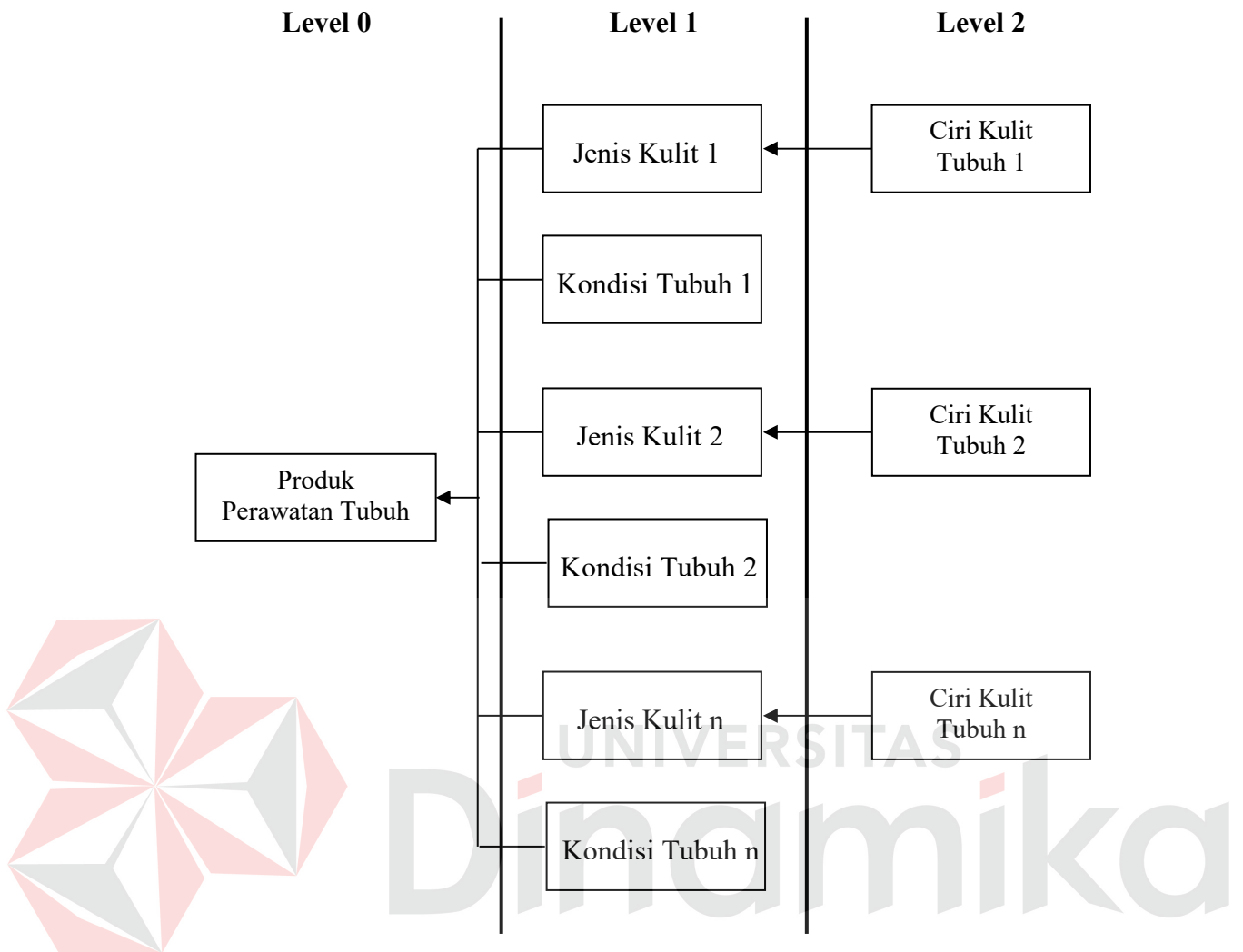
3.3.2 Block diagram

Block Diagram diperlukan untuk mengetahui urutan kerja sistem untuk mencapai keputusan. Perancangan aturan untuk menentukan jenis perawatan dan produk diambil dari parameter-parameter opsional yang ada, seperti pada gambar di bawah ini.



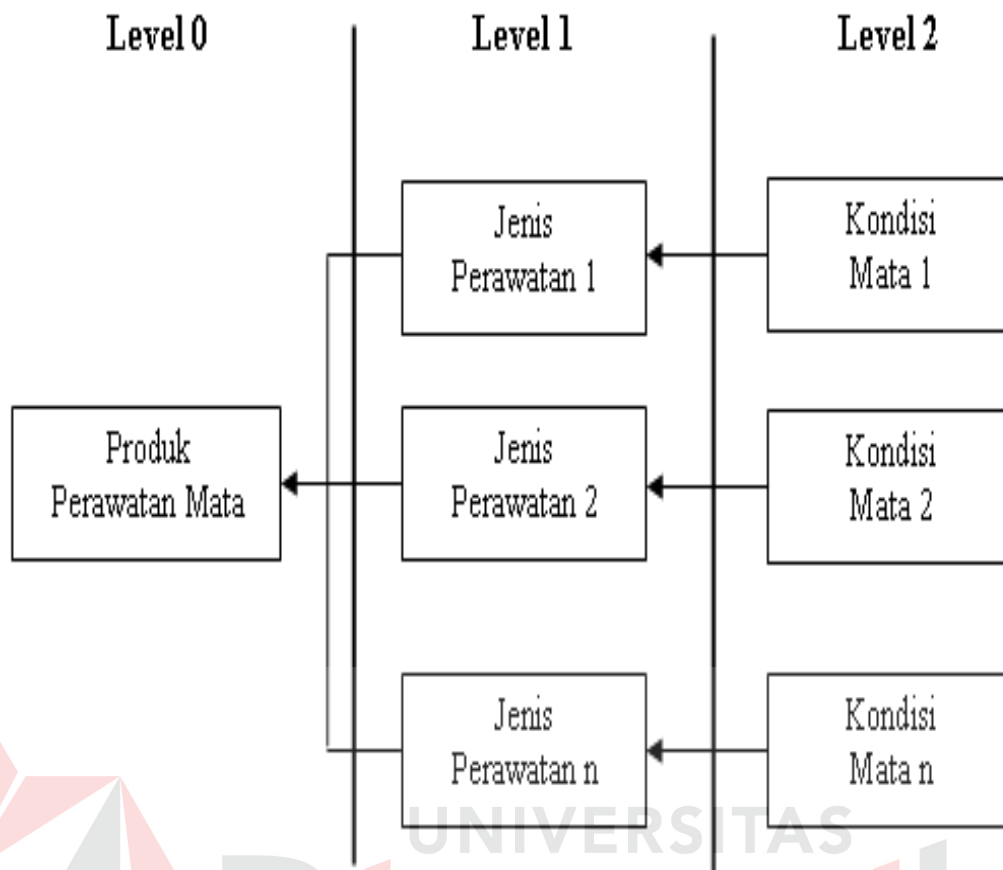
Gambar 3.1 Block diagram untuk perawatan tubuh

Pada Gambar 3.1 dapat dijelaskan bahwa untuk menentukan jenis perawatan tubuh terdiri dari dua tingkatan. Pada level 1, terdiri dari parameter jenis kulit dan parameter kondisi tubuh. Pada level 2 parameter jenis kulit terdiri dari sub parameter ciri kulit tubuh. Parameter jenis kulit, parameter kondisi tubuh, dan sub parameter ciri kulit tubuh merupakan karakteristik dari pasien, yang akan menentukan jenis perawatan apa yang tepat bagi pasien.



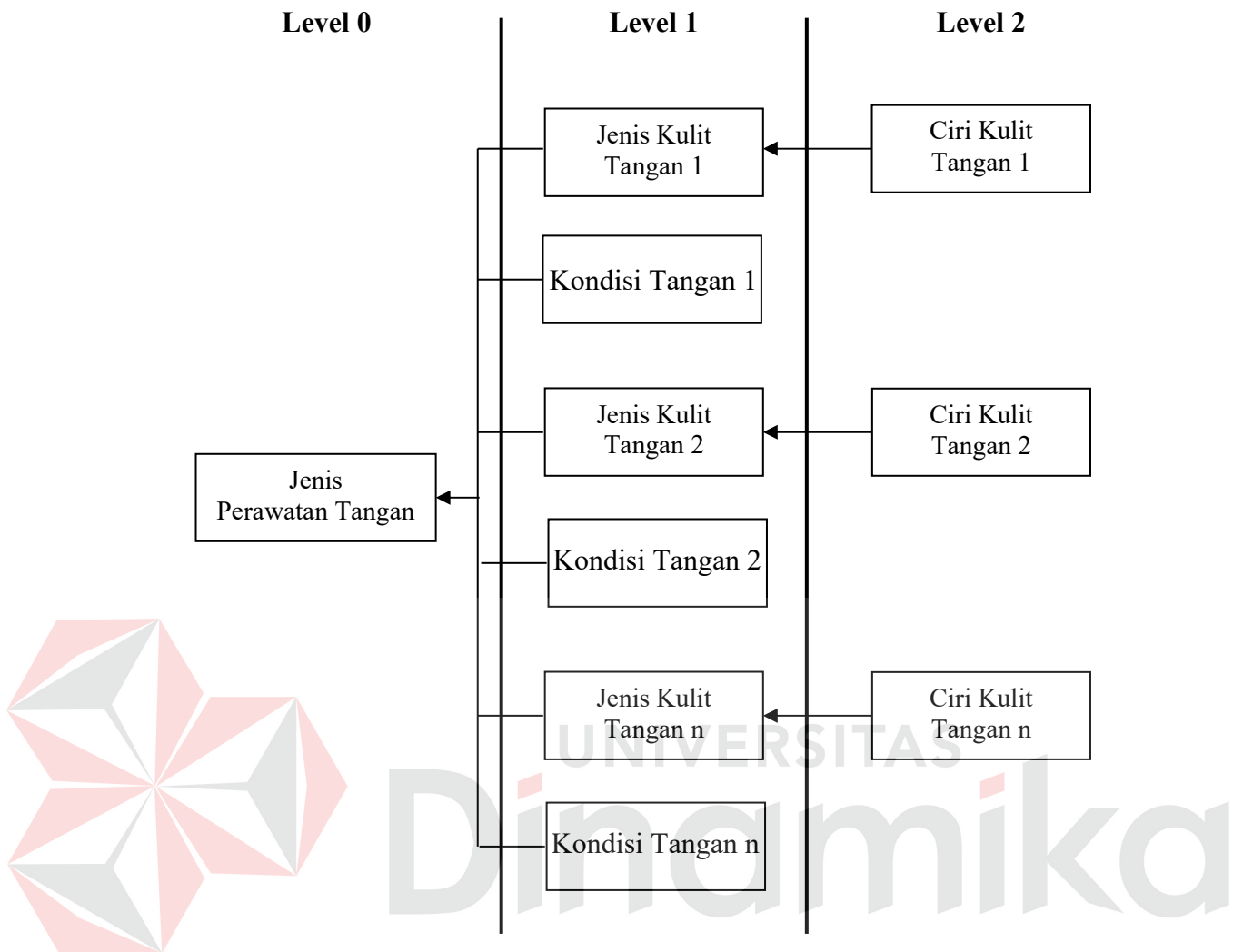
Gambar 3.2 Block diagram untuk produk perawatan tubuh

Pada Gambar 3.2 dapat dijelaskan bahwa untuk menentukan produk perawatan tubuh terdiri dari dua tingkatan. Pada level 1, terdiri dari parameter jenis kulit dan parameter kondisi tubuh. Pada level 2 parameter jenis kulit terdiri dari sub parameter ciri kulit tubuh. Parameter jenis kulit, parameter kondisi tubuh, dan sub parameter ciri kulit tubuh merupakan karakteristik dari pasien, yang akan menentukan produk perawatan apa yang tepat bagi pasien.



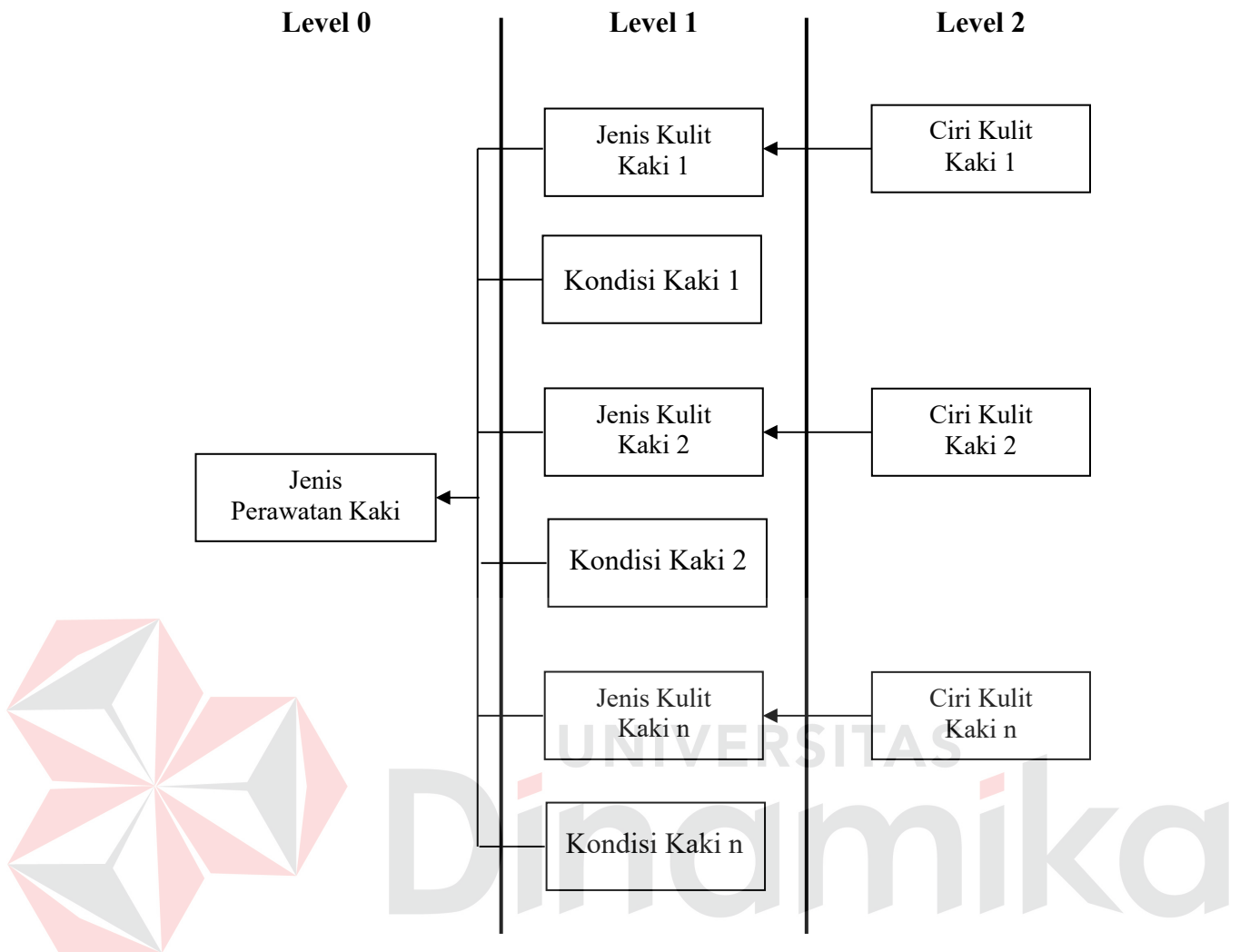
Gambar 3.3 Block diagram untuk perawatan mata

Pada Gambar 3.3 dapat dijelaskan bahwa dalam perawatan mata terdapat 2 level. Level 1 terdiri dari parameter jenis perawatan mata, dimana parameter ini digunakan untuk menentukan produk perawatan mata yang tepat bagi pasien. Sedangkan pada level 2, jenis perawatan mata terdiri dari sub parameter kondisi mata.



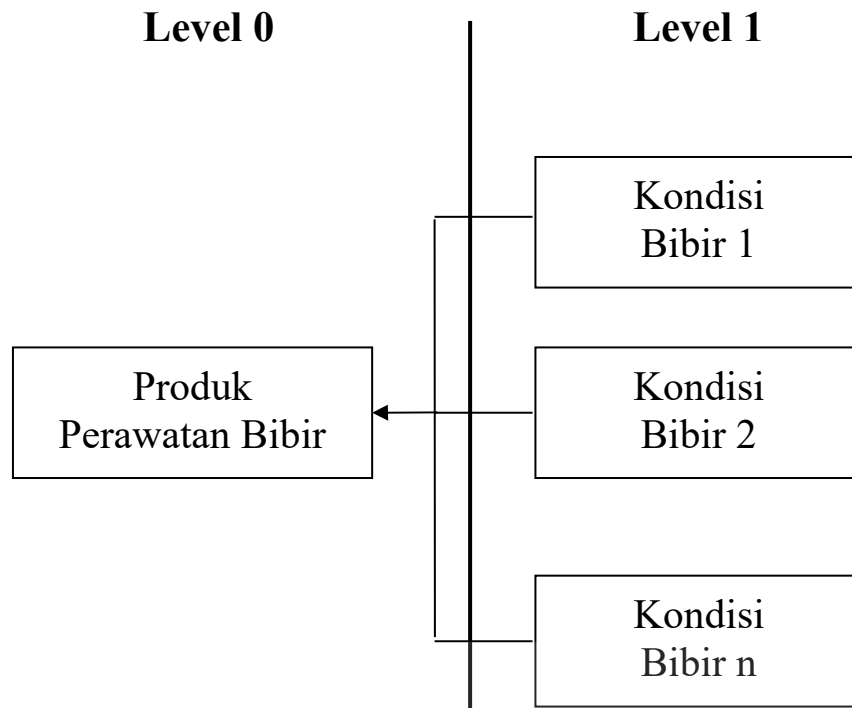
Gambar 3.4 Block diagram untuk perawatan tangan

Pada Gambar 3.4 dapat dijelaskan bahwa untuk menentukan perawatan tangan terdiri dari dua tingkatan. Pada level 1, terdiri dari parameter jenis kulit tangan dan parameter kondisi tangan. Pada level 2 parameter jenis kulit tangan terdiri dari sub parameter ciri kulit tangan. Parameter jenis kulit tangan, parameter kondisi tangan, dan sub parameter ciri kulit tangan merupakan karakteristik dari pasien, yang akan menentukan jenis perawatan tangan yang tepat bagi pasien.



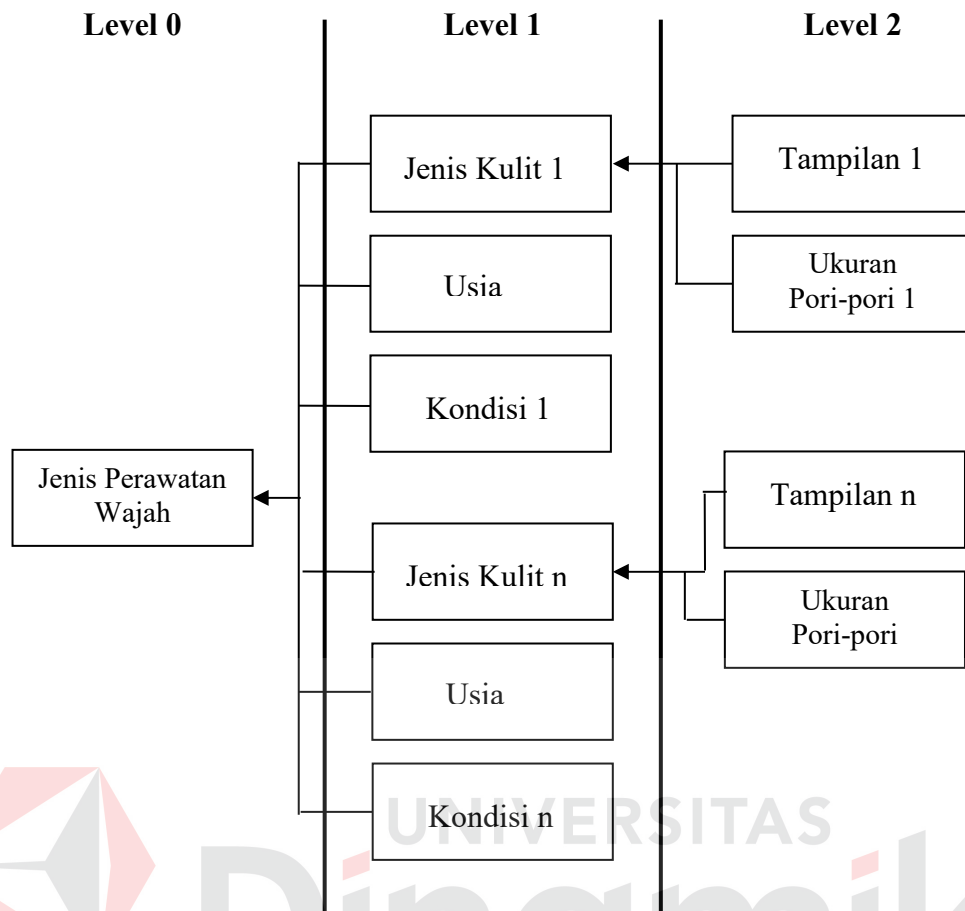
Gambar 3.5 Block diagram untuk perawatan kaki

Pada Gambar 3.5 dapat dijelaskan bahwa untuk menentukan perawatan kaki terdiri dari dua tingkatan. Pada level 1, terdiri dari parameter jenis kulit kaki dan parameter kondisi kaki. Pada level 2 parameter jenis kulit kaki terdiri dari sub parameter ciri kulit kaki. Parameter jenis kulit kaki, parameter kondisi kaki, dan sub parameter ciri kulit kaki merupakan karakteristik dari pasien, yang akan menentukan jenis perawatan kaki yang tepat bagi pasien.



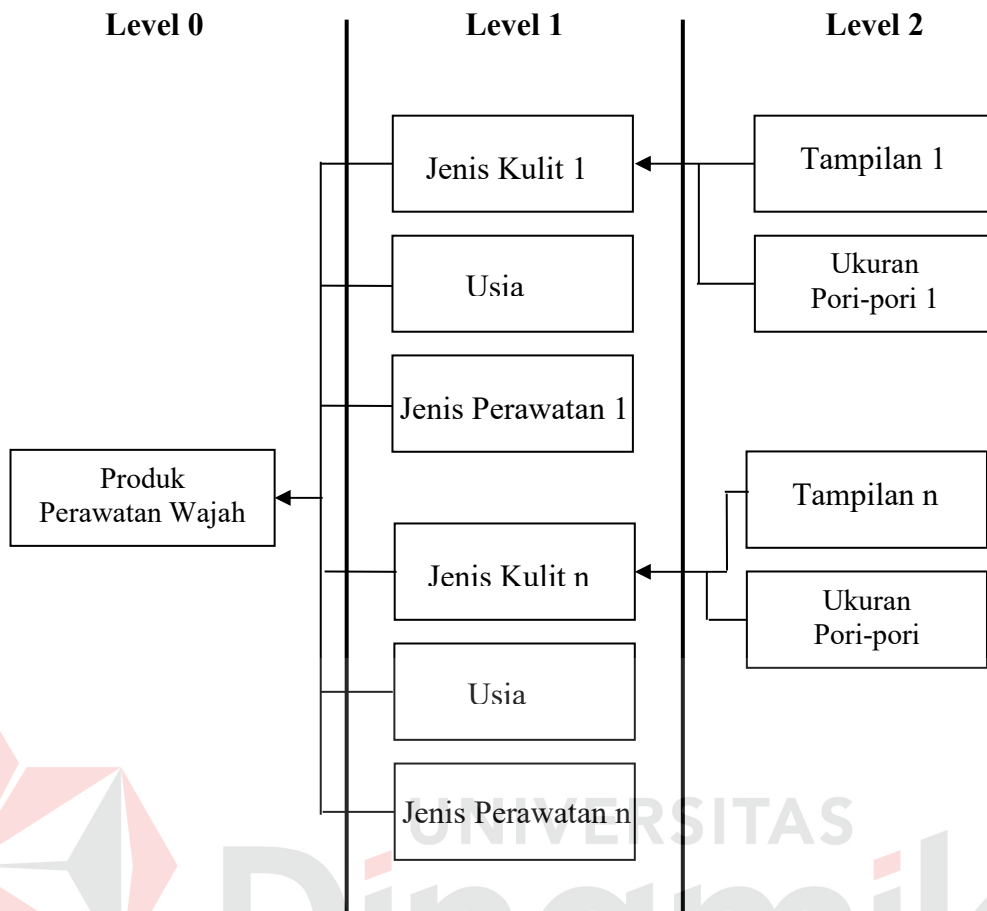
Gambar 3.6 Block diagram untuk produk perawatan bibir

Pada Gambar 3.6 dapat dijelaskan bahwa untuk menentukan produk perawatan bibir terdiri dari satu tingkatan. Dimana pada level 1 terdiri dari parameter kondisi bibir, yang nantinya akan digunakan untuk menentukan produk perawatan bibir yang tepat bagi pasien.



Gambar 3.7 Block diagram untuk perawatan wajah

Pada Gambar 3.7 dapat dijelaskan bahwa untuk menentukan jenis perawatan wajah terdiri dari dua tingkatan. Pada level 1, terdiri dari parameter jenis kulit, usia, dan parameter kondisi wajah. Pada level 2 parameter jenis kulit wajah terdiri dari sub parameter tampilan dan ukuran pori-pori. Parameter jenis kulit, parameter usia, parameter kondisi wajah, dan sub parameter tampilan serta ukuran pori-pori merupakan karakteristik dari pasien, yang akan menentukan jenis perawatan wajah yang tepat bagi pasien.



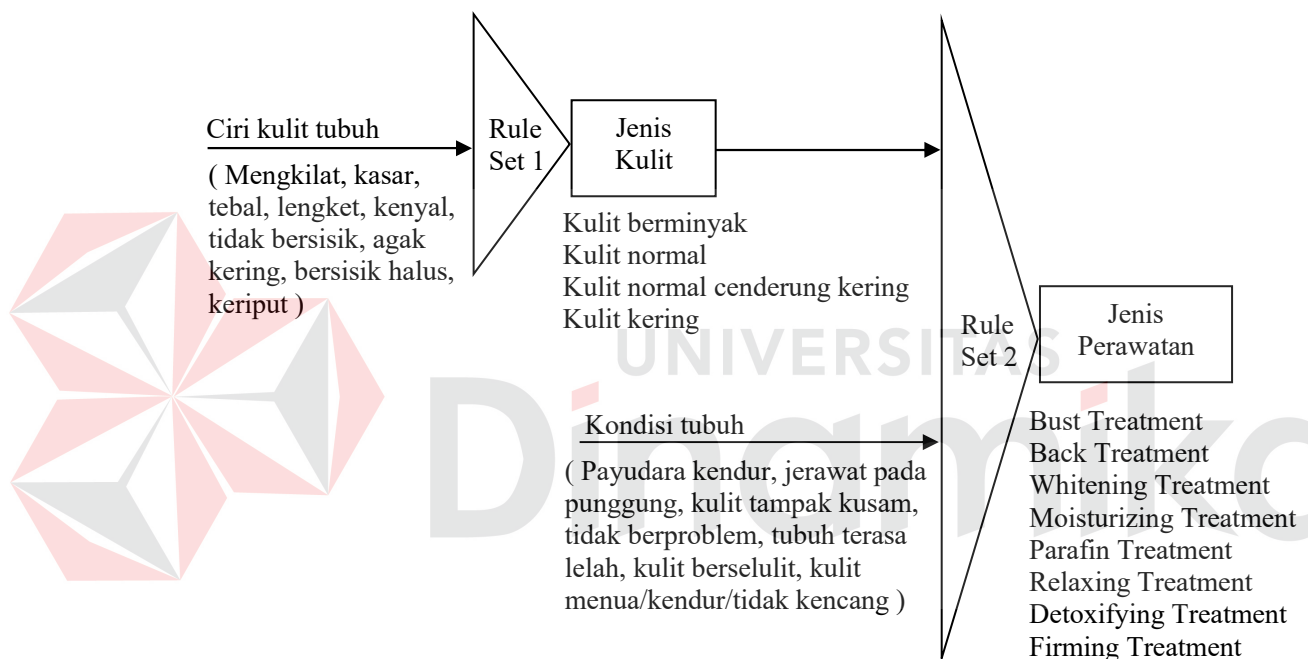
Gambar 3.8 Block diagram untuk produk perawatan wajah

Pada Gambar 3.8 dapat dijelaskan bahwa untuk menentukan produk perawatan wajah terdiri dari dua tingkatan. Pada level 1, terdiri dari parameter jenis kulit, usia, dan jenis perawatan wajah. Pada level 2 parameter jenis kulit wajah terdiri dari sub parameter tampilan dan ukuran pori-pori. Parameter jenis kulit, parameter usia, parameter jenis perawatan wajah, dan sub parameter tampilan serta ukuran pori-pori merupakan karakteristik dari pasien, yang akan menentukan produk perawatan wajah yang tepat bagi pasien.

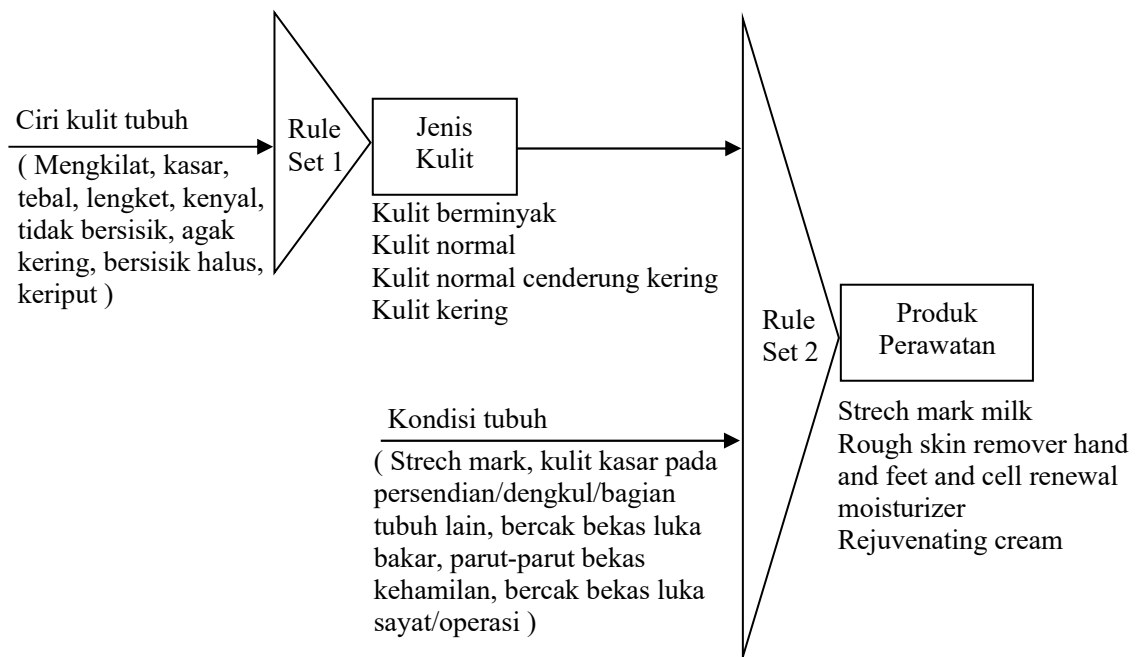
3.3.3 Dependency diagram

Pengumpulan pengetahuan yang telah dilakukan akan digambarkan dalam *Dependency Diagram*. Dengan *Dependency Diagram*, dapat dilihat hubungan antara *rules* dalam menyusun *rules final*.

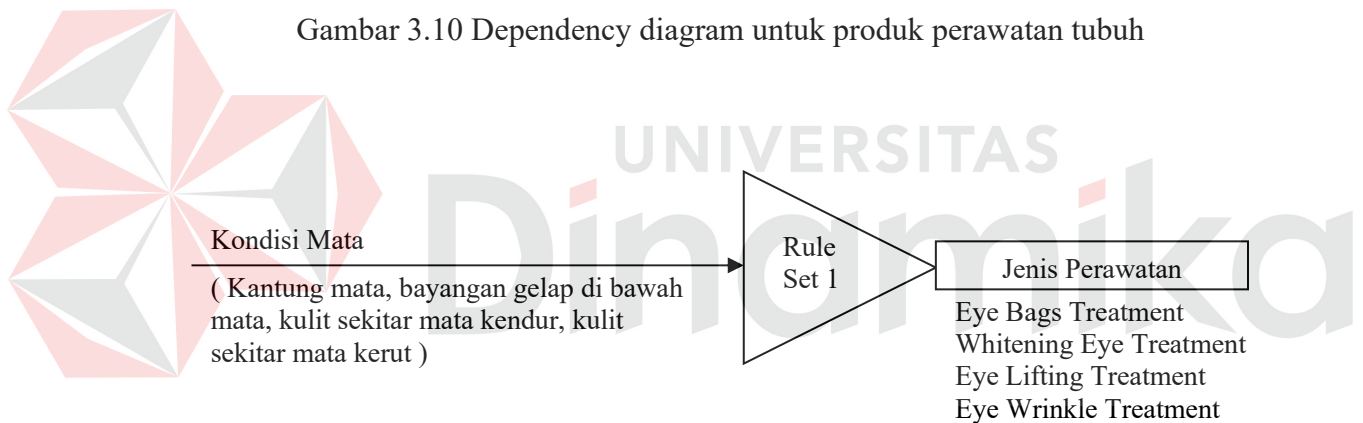
Dependency Diagram untuk aplikasi sistem pakar ini dapat dilihat pada gambar berikut ini:



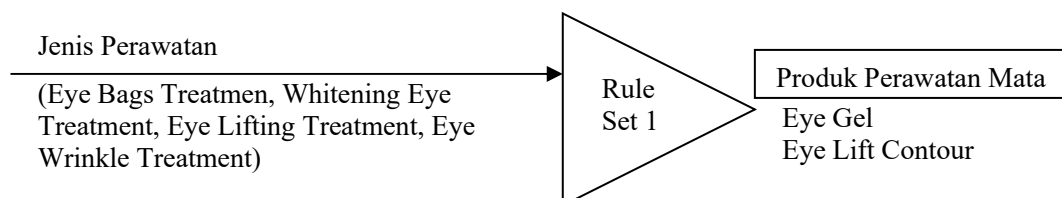
Gambar 3.9 Dependency diagram untuk perawatan tubuh



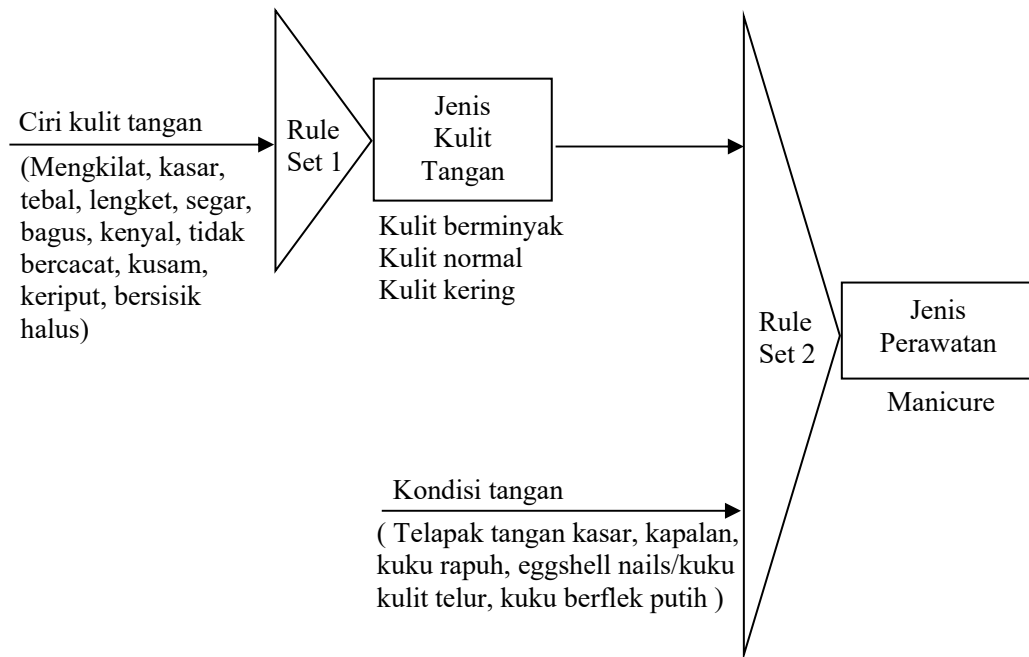
Gambar 3.10 Dependency diagram untuk produk perawatan tubuh



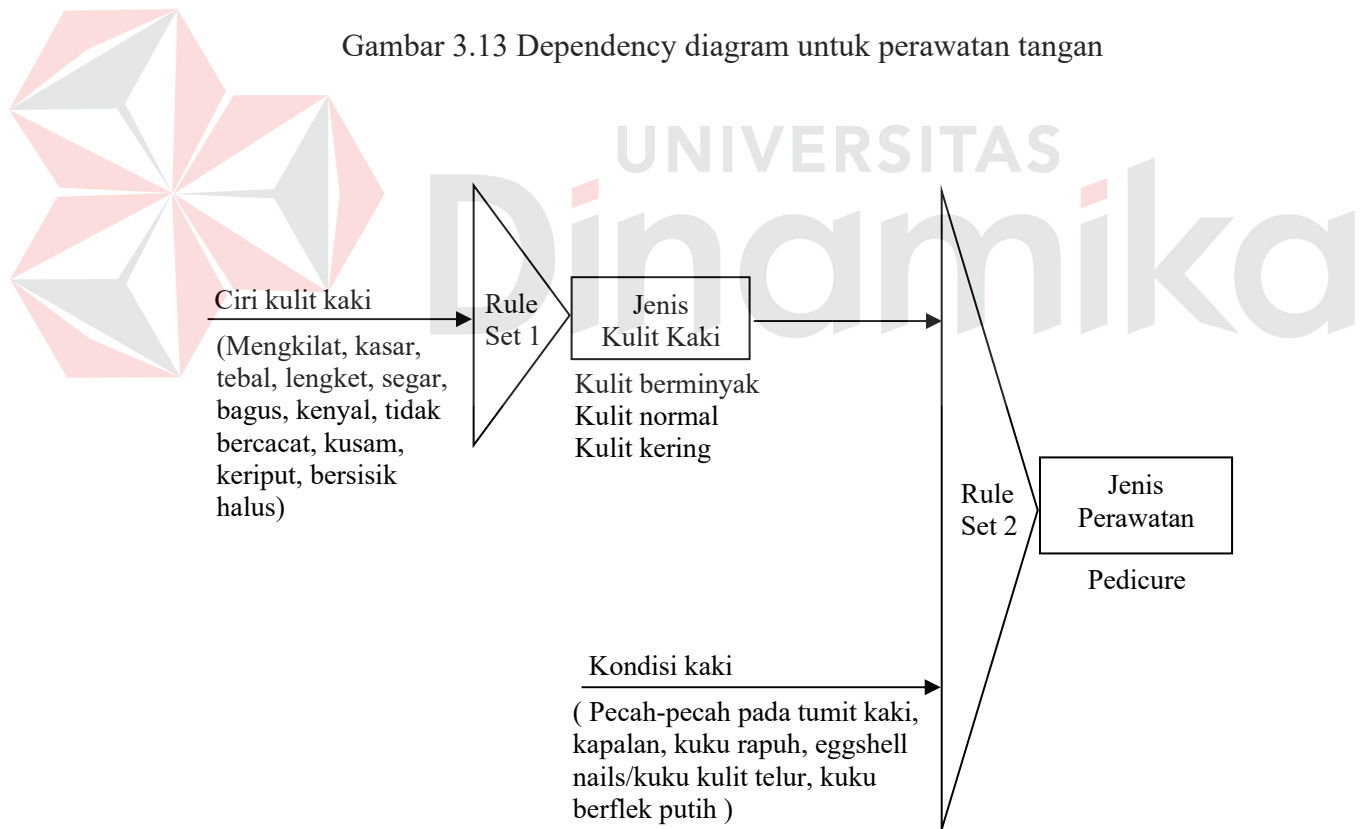
Gambar 3.11 Dependency diagram untuk perawatan mata



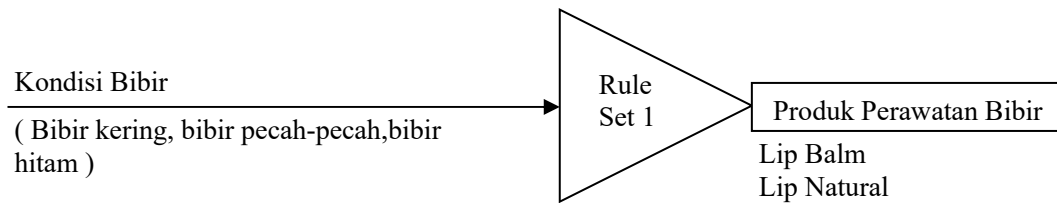
Gambar 3.12 Dependency diagram untuk produk perawatan mata



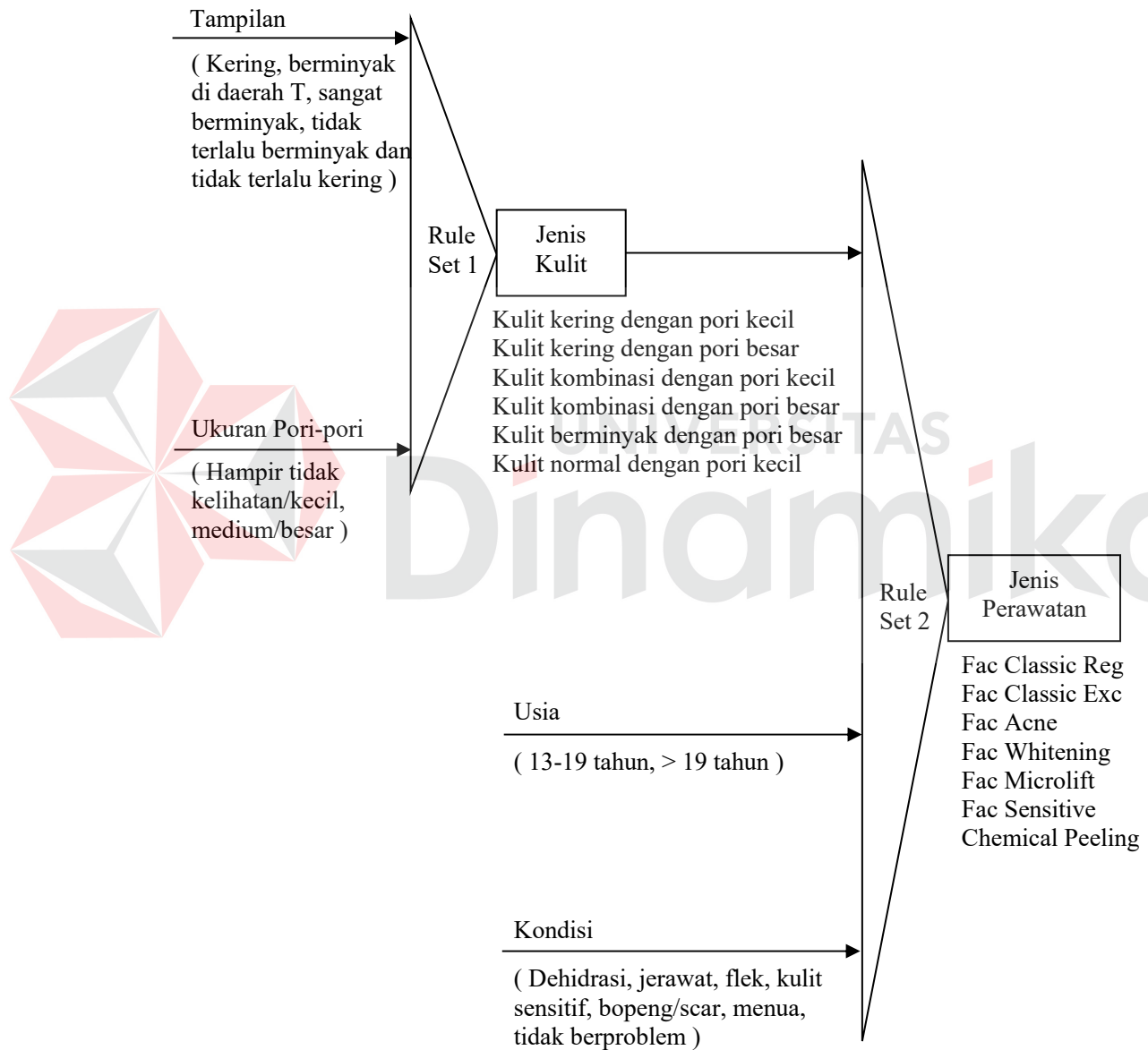
Gambar 3.13 Dependency diagram untuk perawatan tangan



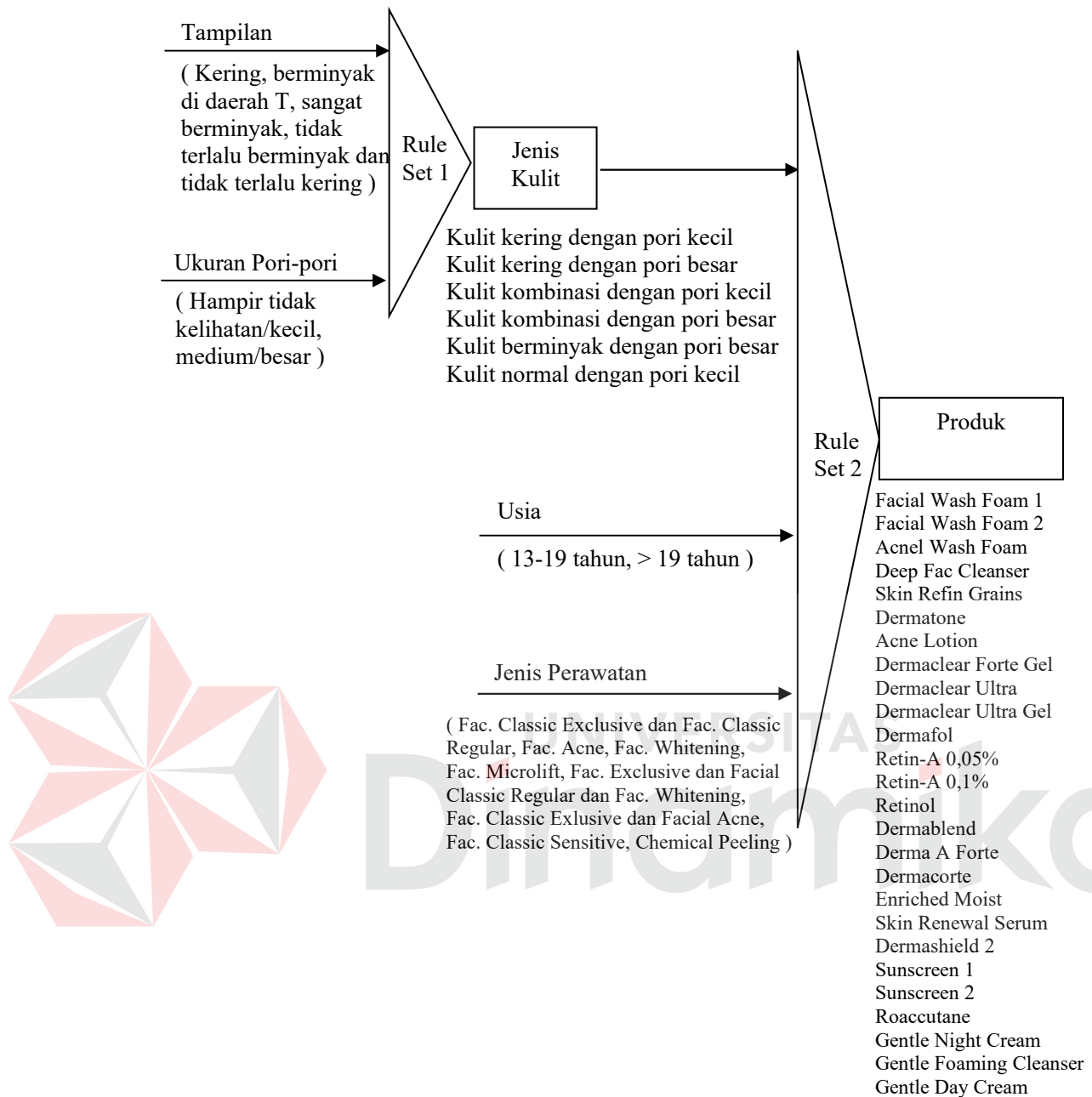
Gambar 3.14 Dependency diagram untuk perawatan kaki



Gambar 3.15 Dependency diagram untuk produk perawatan bibir



Gambar 3.16 Dependency diagram untuk perawatan wajah



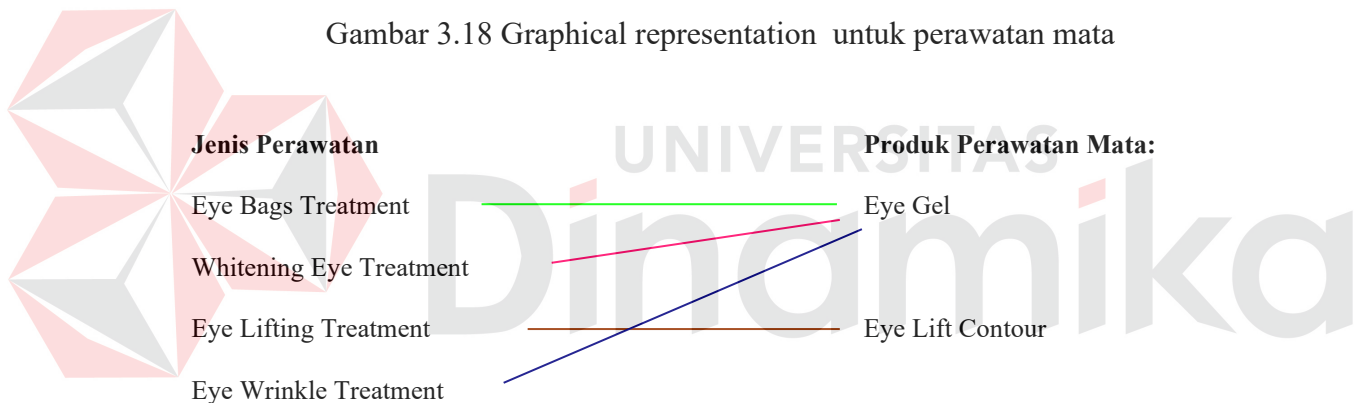
Gambar 3.17 Dependency diagram untuk produk perawatan wajah

3.3.4 Graphical representation

Dari *Dependency Diagram*, digambarkan *Graphical Representation* untuk memperlihatkan hubungan yang lebih jelas antara *rules* dalam menyusun *rules final*. *Graphical Representation* yang telah dibuat untuk aplikasi sistem pakar ini dapat dilihat pada gambar berikut ini:

Kondisi Mata:		Jenis Perawatan:
Kantung mata	_____	Eye Bags Treatment
Bayangan gelap di bawah mata	_____	Whitening Eye Treatment
Kulit sekitar mata kendur	_____	Eye Lifting Treatment
Kulit sekitar mata kerut	_____	Eye Wrinkle Treatment

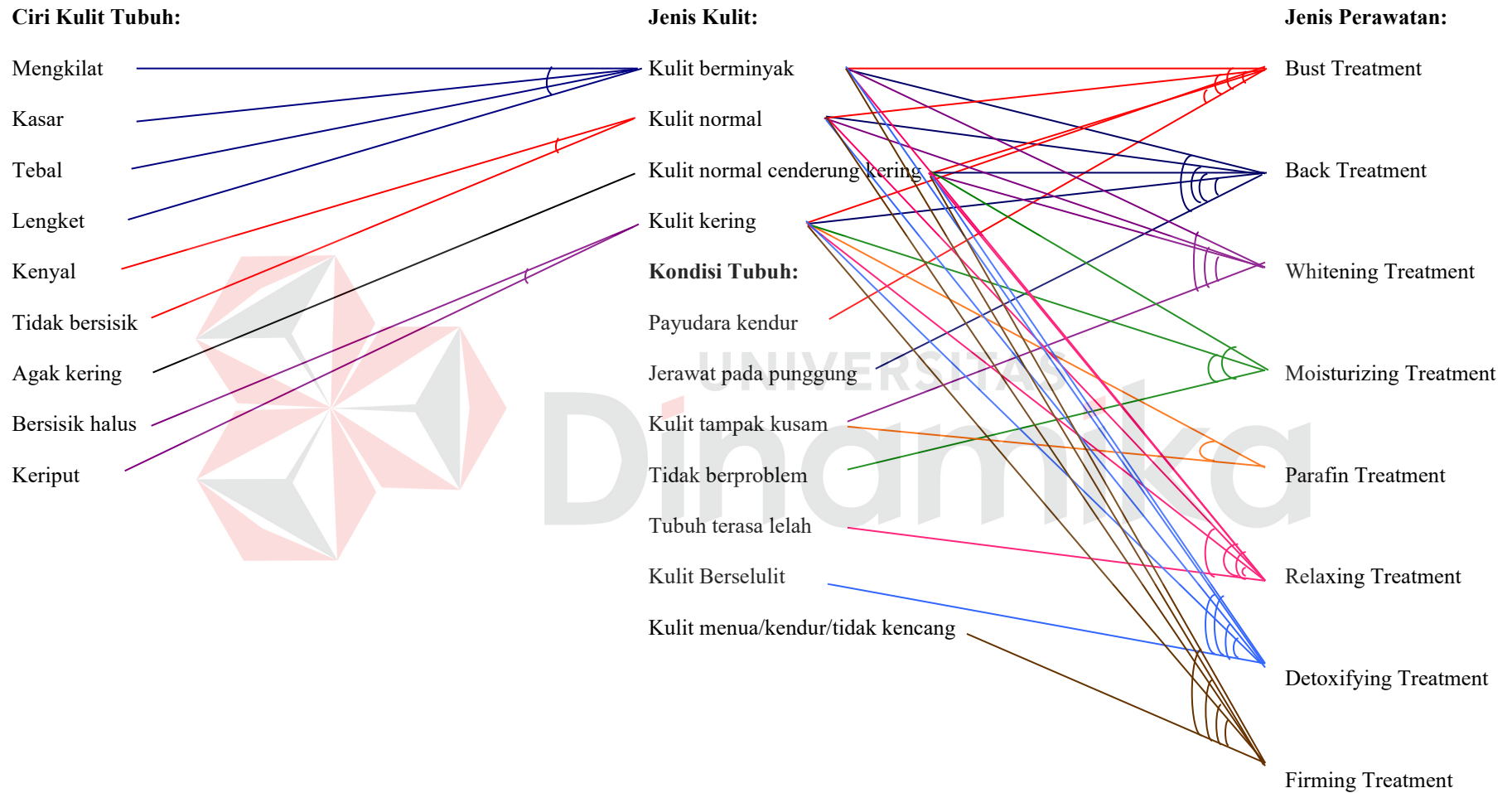
Gambar 3.18 Graphical representation untuk perawatan mata



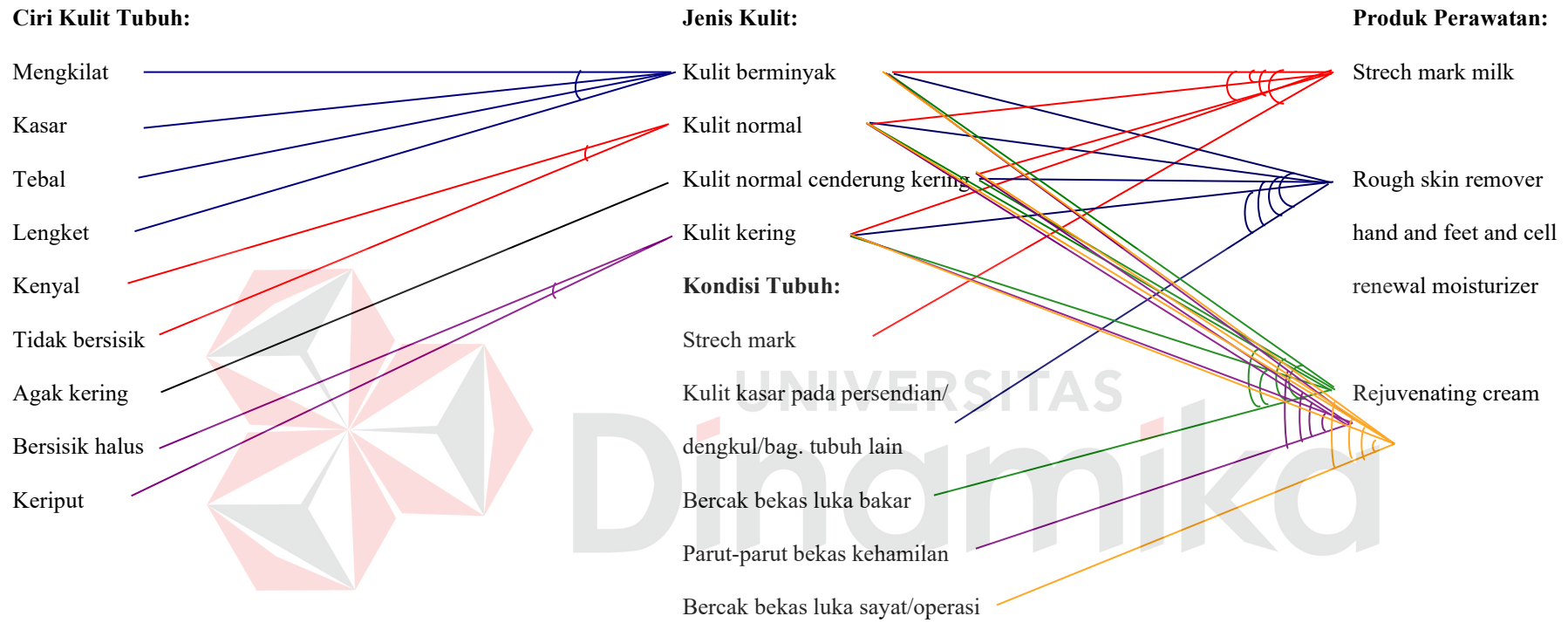
Gambar 3.19 Graphical representation untuk produk perawatan mata

Kondisi Bibir		Produk Perawatan Bibir:
Bibir kering	_____	Lip Balm
Bibir pecah-pecah	_____	
Bibir hitam	_____	Lip Natural

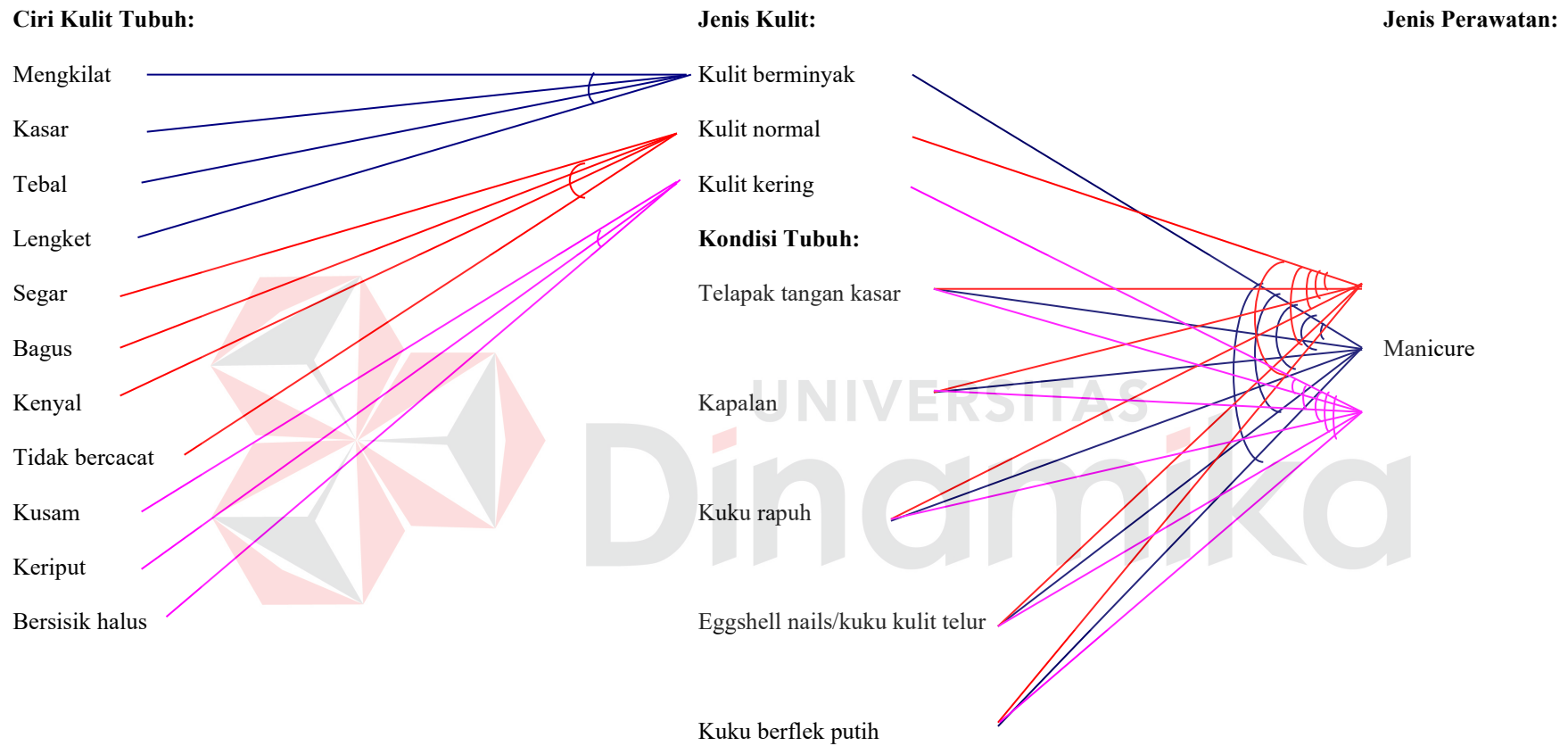
Gambar 3.20 Graphical representation untuk produk perawatan bibir



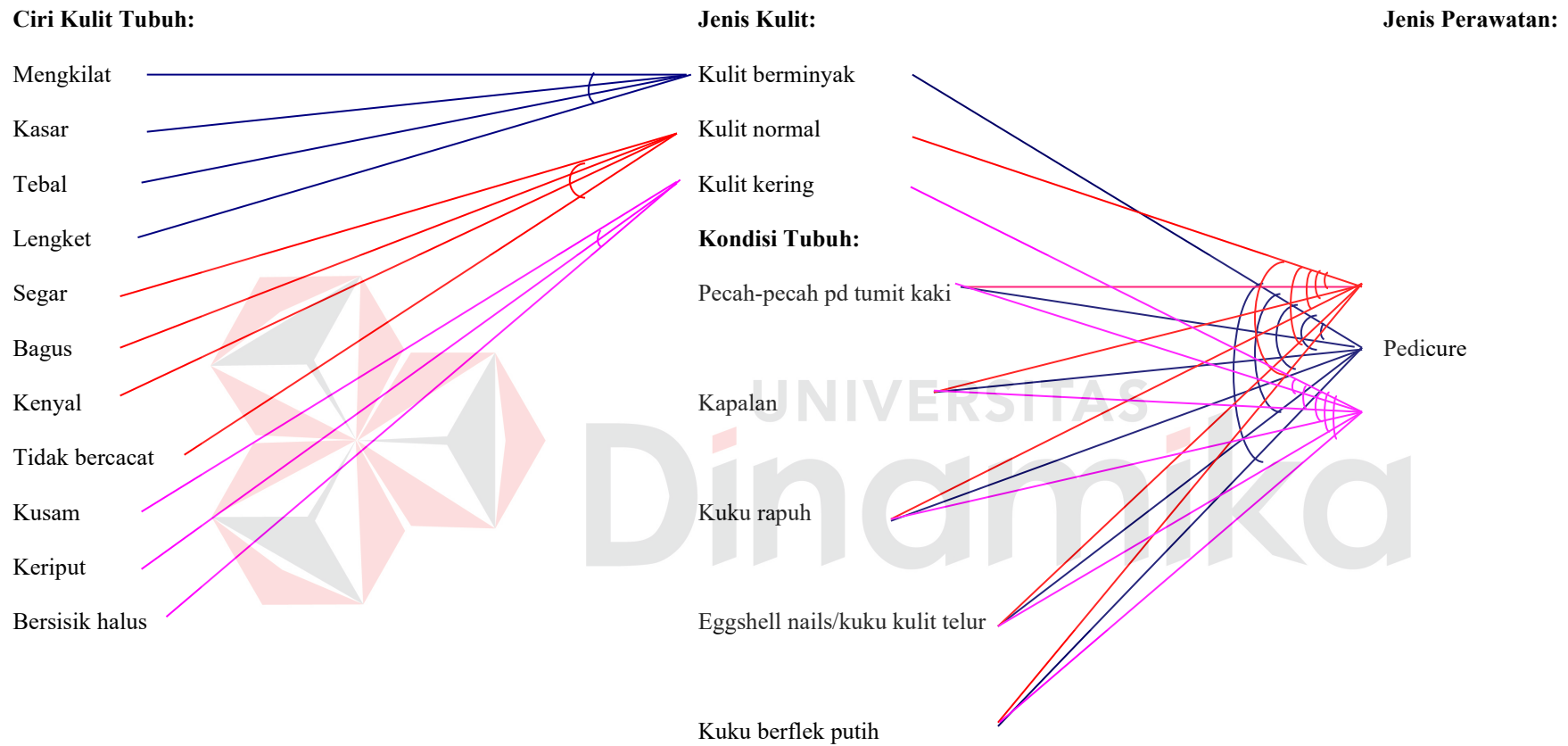
Gambar 3.21 Graphical representation untuk perawatan tubuh



Gambar 3.22 Graphical representation untuk produk perawatan tubuh



Gambar 3.23 Graphical representation untuk perawatan tangan



Gambar 3.24 Graphical representation untuk perawatan kaki

Keterangan Kode Perawatan Wajah:

A : Facial Classic Exclusive dan Facial Classic Regular

B : Facial Acne

C : Facial Whitening

D : Facial Microlift

E : Facial Classic Exclusive dan Facial Classic Regular dan Facial Whitening

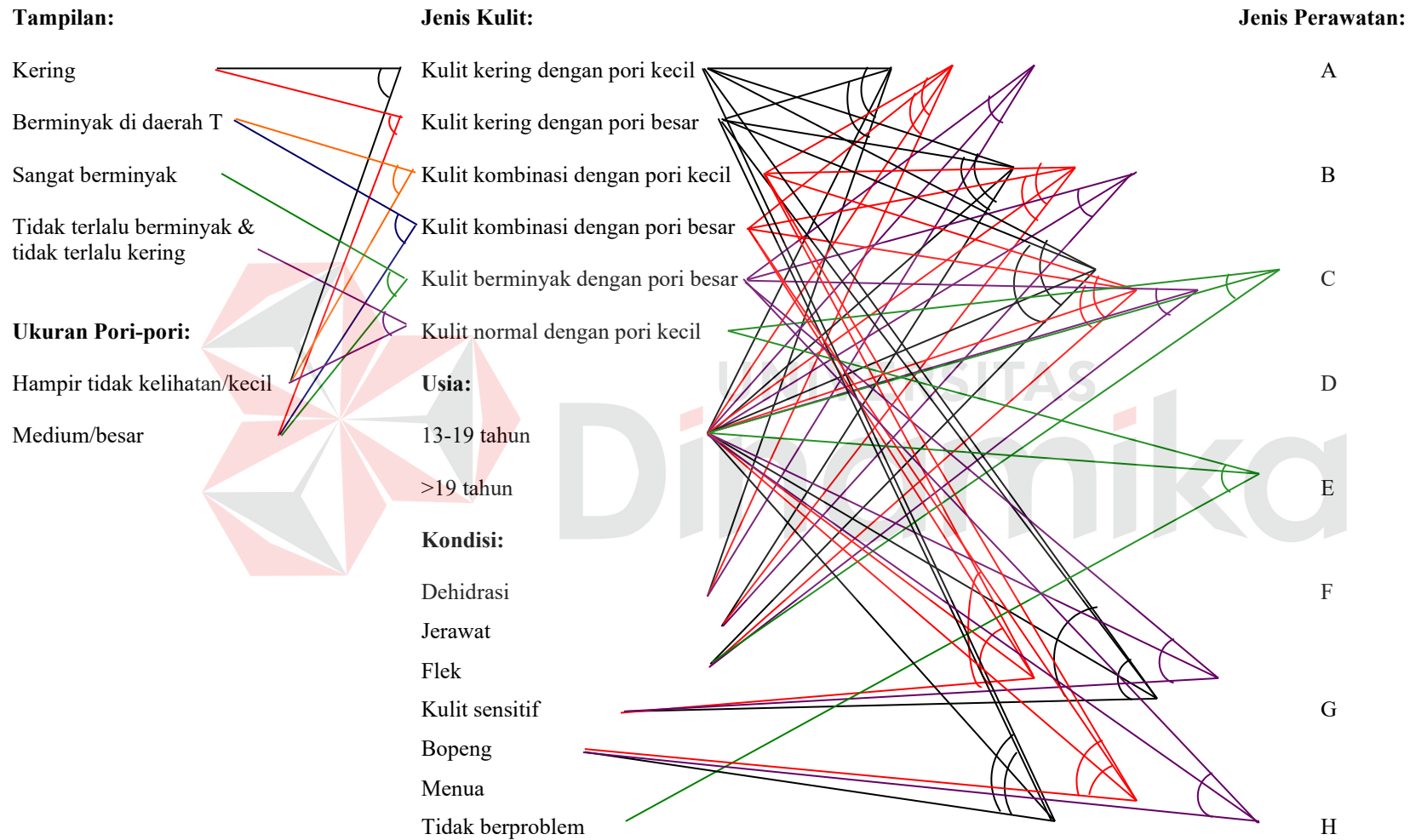
F : Facial Classic Exclusive dan Facial Acne

G : Facial Classic Sensitive

H : Chemical Peeling



UNIVERSITAS
Dinamika



Gambar 3.25 Graphical representation untuk perawatan wajah

Tampilan:

- Kering
- Berminyak di daerah T
- Sangat berminyak
- Tidak terlalu berminyak & tidak terlalu kering

Ukuran Pori-pori:

- Hampir tidak kelihatan/kecil
- Medium/besar

Jenis Kulit:

- Kulit kering dengan pori kecil
- Kulit kering dengan pori besar
- Kulit kombinasi dengan pori kecil
- Kulit kombinasi dengan pori besar
- Kulit berminyak dengan pori besar
- Kulit normal dengan pori kecil

Usia:

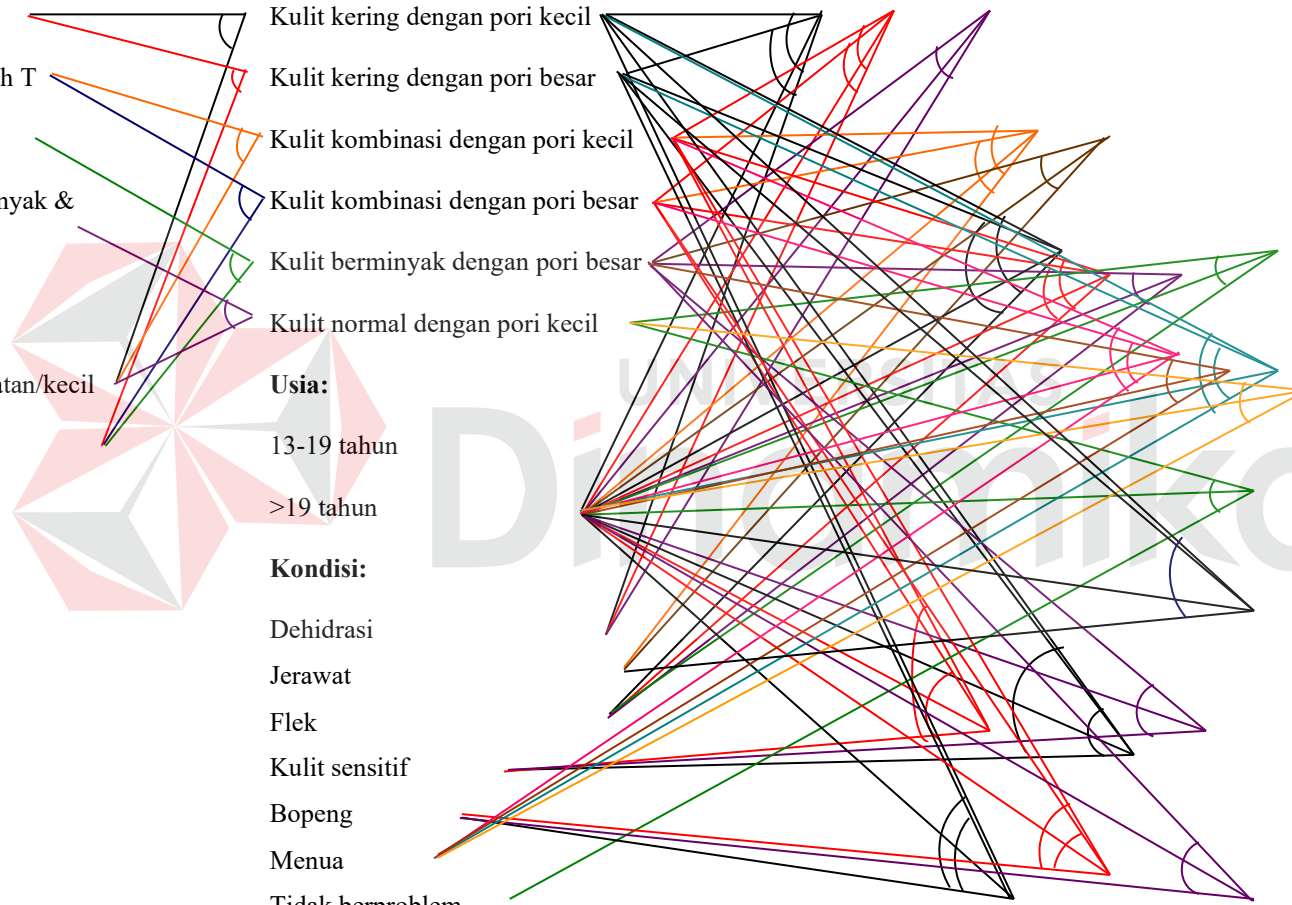
- 13-19 tahun
- >19 tahun

Kondisi:

- Dehidrasi
- Jerawat
- Flek
- Kulit sensitif
- Bopeng
- Menua
- Tidak berproblem

Jenis Perawatan:

- A
- B
- C
- D
- E
- F
- G
- H



Keterangan Kode Produk Wajah :

a : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermablend

Retin-A 0,05 %

Dermacorte

Sunscreen I / II

b : Acne Wash Foam

Dermatone

Acne lotion

Dermablend

Retin-A 0,05 %

Dermacorte

Sunscreen I / II

c : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra

Dermablend

Retin-A 0,05 %

Dermacorte

Sunscreen I / II

d : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermafol

Dermablend

Retin-A 0,05 %

Dermacorte

Sunscreen I / II

e : Acne Wash Foam

Dermatone

Acne lotion

Dermafol

Dermablend

Retin-A 0,05 %

Dermacorte

Sunscreen I / II

f : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra

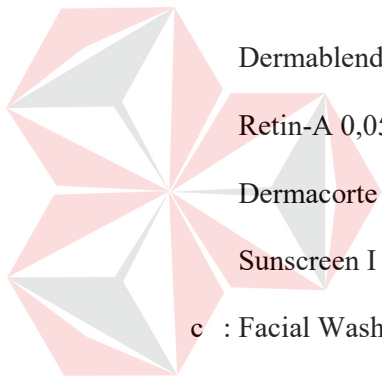
Dermafol

Dermablend

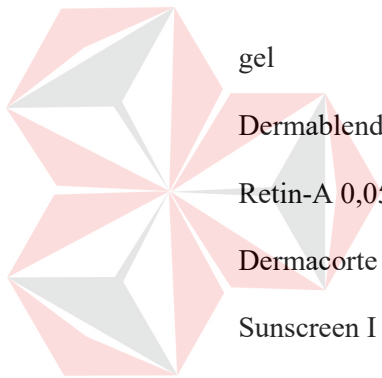
Retin-A 0,05 %

Dermacorte

Sunscreen I / II



g : Facial Wash Foam II	Dermaclear ultra / dermaclear ultra
Refining grains (seminggu sekali)	gel
Dermatone	Dermafol
Dermablend	Dermablend
Retin-A 0,05 %	Retin-A 0,05 %
Dermacorte	Dermacorte
Sunscreen I / II	Sunscreen I / II
h : Facial Wash Foam II	k : Acne Wash Foam
Refining grains (seminggu sekali)	Dermatone
Dermatone	Acne lotion
Dermaclear ultra / dermaclear ultra	Dermafol
gel	Dermablend
Dermablend	Retin-A 0,05 %
Retin-A 0,05 %	Dermacorte
Dermacorte	Sunscreen I / II
Sunscreen I / II	Roaccutane
i : Facial Wash Foam II	l : Facial Wash Foam II
Refining grains (seminggu sekali)	Refining grains (seminggu sekali)
Dermatone	Dermatone
Dermafol	Dermaclear ultra gel
Dermablend	Dermafol
Retin-A 0,05 %	Dermablend
Dermacorte	Retin-A 0,05 %
Sunscreen I / II	Dermacorte
j : Facial Wash Foam II	Sunscreen I / II
Refining grains (seminggu sekali)	
Dermatone	



m : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermaclear ultra

Dermablend

Retin-A 0,05 %

Dermacorte

Sunscreen I / II

n : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermacorte

Sunscreen I / II

o : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Enriched moisturizer

Dermablend

Retin-A 0,1 %

Dermacorte

Dermashield II

p : Acne Wash Foam

Dermatone

Acne lotion

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

q : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra / dermaclear forte

gel

Dermablend

Retin-A 0,1 %

Dermacorte

Dermashield II

r : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Skin renewal serum

Dermablend

Retin-A 0,1 %

Dermacorte

Dermashield II

s : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Enriched moisturizer

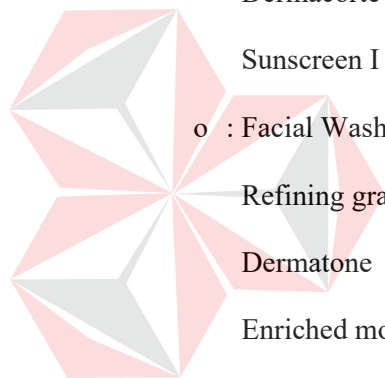
Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Dermashield II



t : Acne Wash Foam

Dermatone

Acne lotion

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

u : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra / dermaclear forte

gel

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Dermashield II

v : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Skin renewal serum

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Dermashield II

w : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

x : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra / dermaclear ultra

gel / dermaclear forte gel

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

y : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone

Skin renewal serum

Dermablend

Retin-A 0,1 %

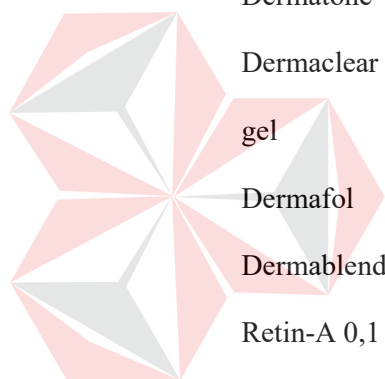
Dermacorte

Sunscreen I / II

z : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone



Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

1 : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra / dermaclear ultra

gel / dermaclear forte gel

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

2 : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone

Skin renewal serum

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

3 : Acne Wash Foam

Dermatone

Acne lotion

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

Roaccutane

4 : Facial Wash Foam II

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra gel / dermaclear

forte gel

Dermafol

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

5 : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Dermaclear ultra gel / dermaclear

ultra

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II



6 : Facial Wash Foam I

Refining grains (seminggu sekali)

Dermatone

Skin renewal serum

Dermablend

Retin-A 0,1 %

Dermacorte

Sunscreen I / II

7 : Deep facial cleansing

Refining grains (seminggu sekali)

Dermatone

Enriched moisturizer

Dermacorte

Sunscreen I / II

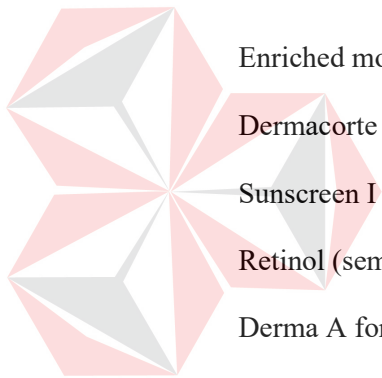
Retinol (seminggu sekali)

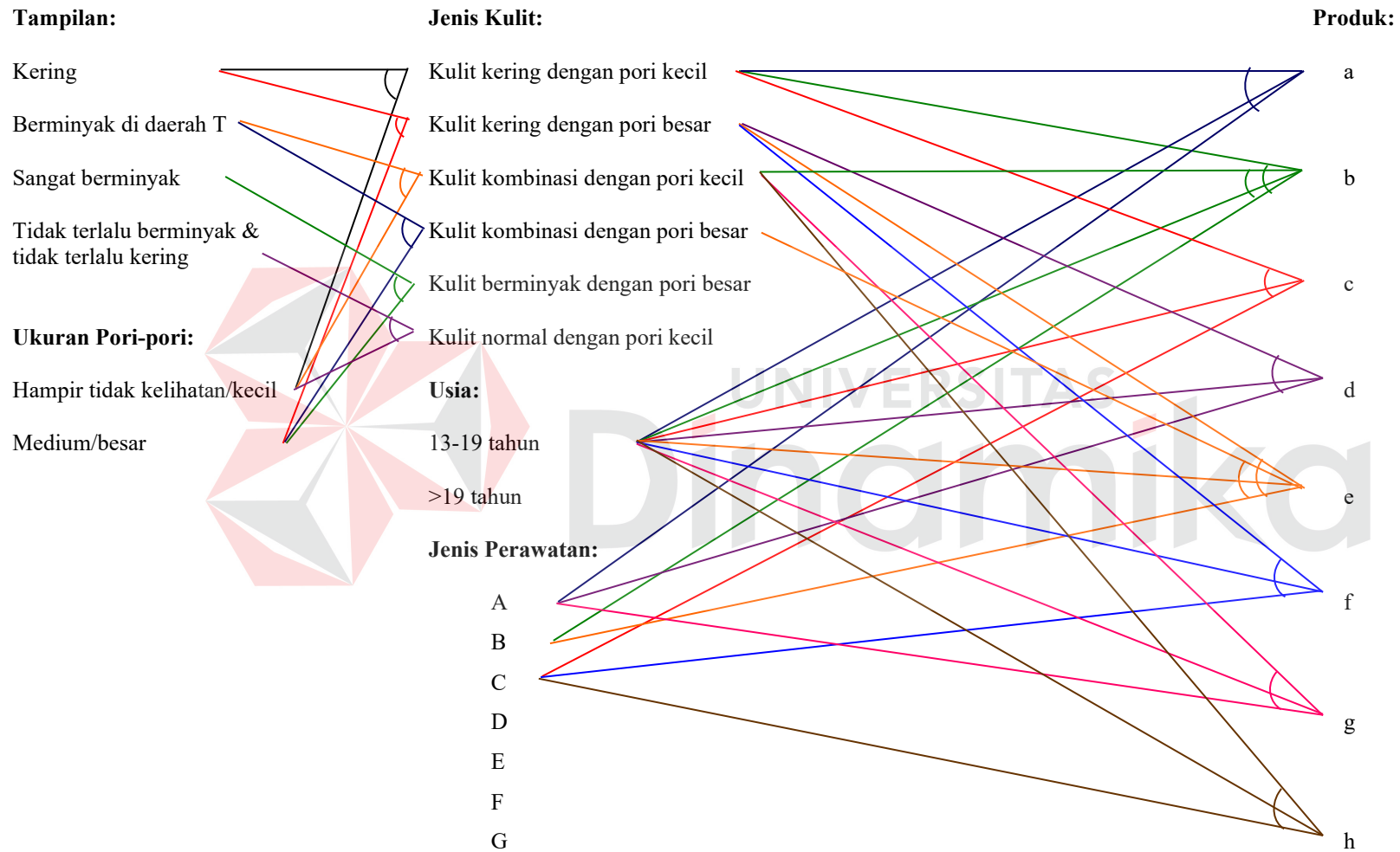
Derma A forte (seminggu sekali)

8 : Gentle foaming cleanser

Gentle day cream

Gentle night cream





Gambar 3.26 Graphical representation untuk produk perawatan wajah

Tampilan:

- Kering
- Berminyak di daerah T
- Sangat berminyak
- Tidak terlalu berminyak & tidak terlalu kering

Ukuran Pori-pori:

- Hampir tidak kelihatan/kecil
- Medium/besar

Jenis Kulit:

- Kulit kering dengan pori kecil
- Kulit kering dengan pori besar
- Kulit kombinasi dengan pori kecil
- Kulit kombinasi dengan pori besar
- Kulit berminyak dengan pori besar
- Kulit normal dengan pori kecil

Usia:

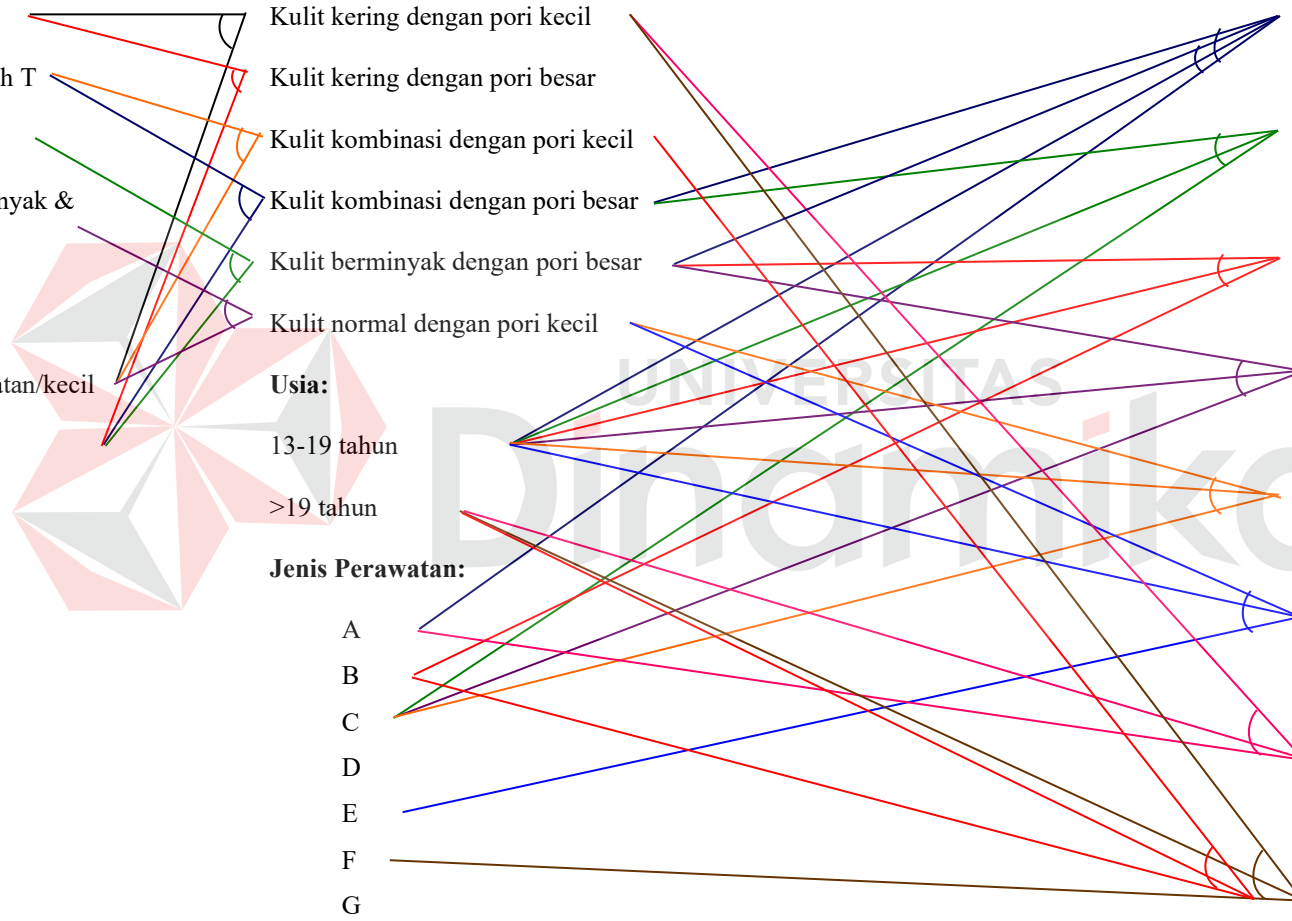
- 13-19 tahun
- >19 tahun

Jenis Perawatan:

- A
- B
- C
- D
- E
- F
- G

Produk:

- i
- j
- k
- l
- m
- n
- o
- p



Tampilan:

Kering

Berminyak di daerah T

Sangat berminyak

Tidak terlalu berminyak &
tidak terlalu kering**Ukuran Pori-pori:**

Hampir tidak kelihatan/kecil

Medium/besar

Jenis Kulit:

Kulit kering dengan pori kecil

Kulit kering dengan pori besar

Kulit kombinasi dengan pori kecil

Kulit kombinasi dengan pori besar

Kulit berminyak dengan pori besar

Kulit normal dengan pori kecil

Usia:

13-19 tahun

>19 tahun

Jenis Perawatan:

A

B

C

D

E

F

G

Produk:

q

r

s

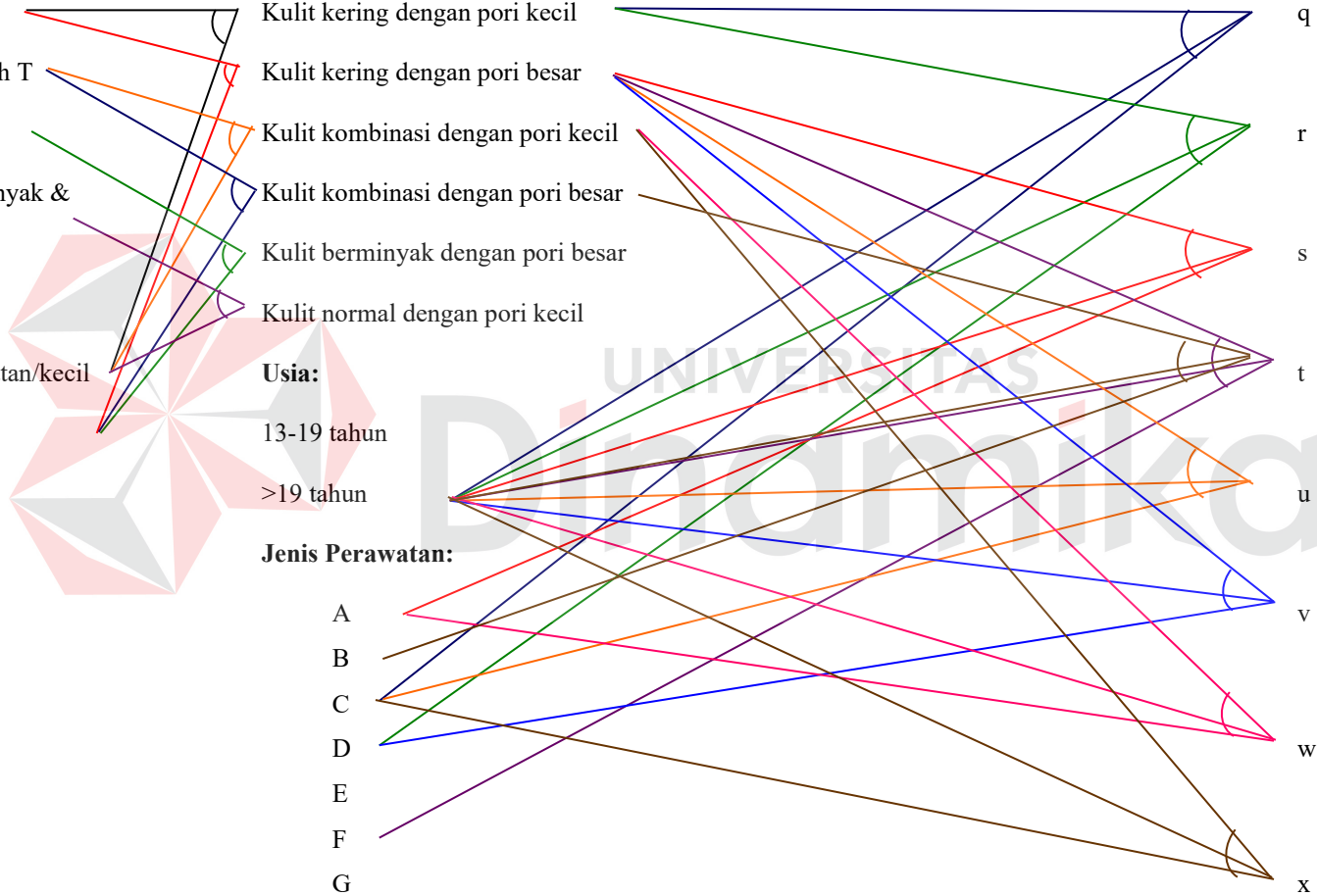
t

u

v

w

x



Tampilan:

- Kering
- Berminyak di daerah T
- Sangat berminyak
- Tidak terlalu berminyak & tidak terlalu kering

Ukuran Pori-pori:

- Hampir tidak kelihatan/kecil
- Medium/besar

Jenis Kulit:

- Kulit kering dengan pori kecil
- Kulit kering dengan pori besar
- Kulit kombinasi dengan pori kecil
- Kulit kombinasi dengan pori besar
- Kulit berminyak dengan pori besar
- Kulit normal dengan pori kecil

Usia:

- 13-19 tahun
- >19 tahun

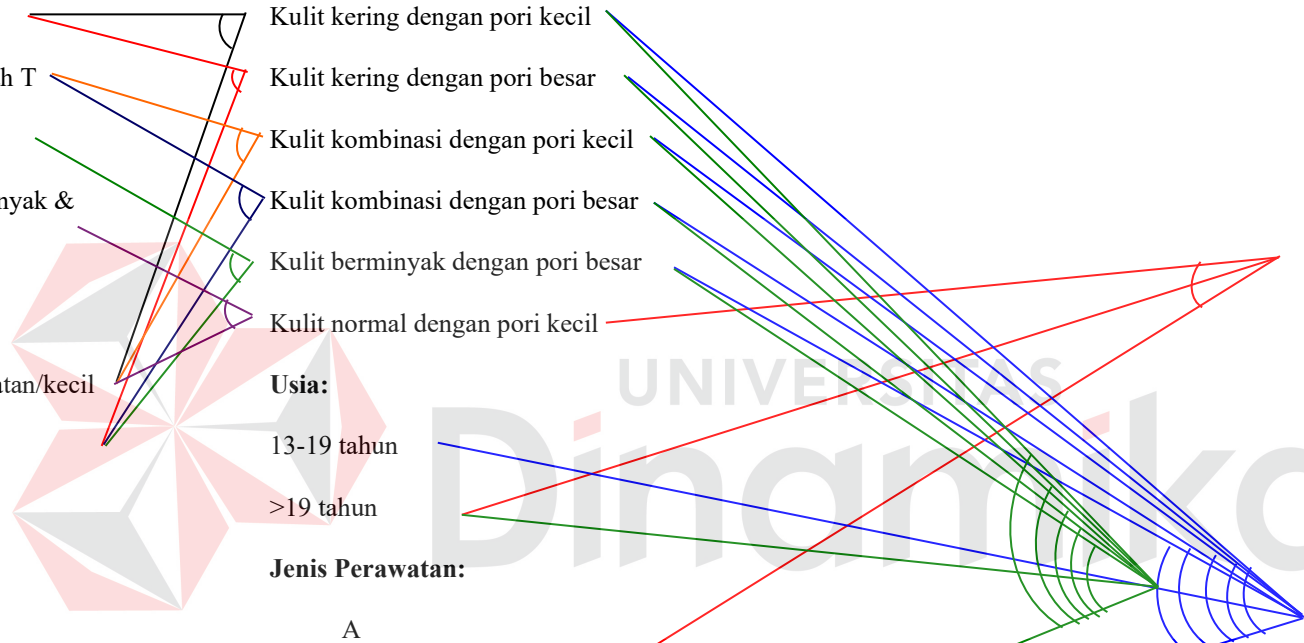
Jenis Perawatan:

- A
- B
- C
- D
- E
- F
- G

Produk:

7

8



3.3.5 IF-THEN rules

Setelah dibuat *Graphical Representation*, langkah berikutnya dalam perancangan aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini, yaitu mengkonversikan *Graphical Representation* tersebut ke dalam *IF-THEN Rules*. Pada aplikasi sistem pakar ini, *IF-THEN Rules* disusun dengan menggunakan operator *AND*. Sehingga *rule-rule* yang seharusnya menggunakan operator *OR* atau *ELSE* akan dibuat menjadi dua buah *rule* dengan menggunakan operator *AND*. Telah ditemukan 213 *rules* pada sistem pakar ini yaitu sebagai berikut:

RULE 1

IF Set=Tubuh and Ciri_kulit_tubuh=mengkilat/kasar/tebal/lengket

THEN Jenis_kulit=Kulit_berminyak(*CF_rule*=0,9)

RULE 2

IF Set=Tubuh and Ciri_kulit_tubuh=kenyal_dan_tidak_bersisik

THEN Jenis_kulit=Kulit_normal(*CF_rule*=0,9)

RULE 3

IF Set=Tubuh and Ciri_kulit_tubuh=agak_kering

THEN Jenis_kulit=Kulit_normal_cenderung_kering(*CF_rule*=0,9)

RULE 4

IF Set=Tubuh and Ciri_kulit_tubuh=bersisik_halus_dan_keriput

THEN Jenis_kulit=Kulit_kering(*CF_rule*=0,9)

RULE 5

IF Set=Tubuh and Jenis_kulit_tubuh=berminyak and Kondisi_tubuh=Payudara_kendur

THEN Jenis_perawatan=Bust_treatment(*CF_rule*=0,9)

Sebagian *rules* sistem pakar yang lain, yaitu *rule* 6 sampai dengan *rule* 213 terdapat pada lampiran.

3.3.6 Database

Pembuatan tabel-tabel pada *database* aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini menggunakan program *Microsoft Access*. *Database* digunakan untuk menyimpan *rule*, premis, produk dan jenis perawatan. *Database* juga digunakan untuk pencapaian kesimpulan, melalui sejumlah *rule* yang diaplikasikan ke dalam *database*. Keuntungan yang dapat diperoleh dengan menggunakan *database* yaitu setiap *rule* akan lebih dinamis, dimana mudah untuk diubah dan digantikan. Tabel-tabel yang digunakan oleh sistem adalah sebagai berikut:

a. Tabel jika

Nama Tabel	: Tabel jika
Primary Key	: noRule, noPremis
Foreign Key	: noRule; References maka (noRule)
Fungsi	: Menyimpan premis-premis dari <i>rule</i> yang ada

Tabel 3.1 Tabel jika

Nama Kolom	Tipe Data	Keterangan
noRule	Number(Long Integer)	Primary Key; Foreign Key – references maka (noRule)
noPremis	Number(Long Integer)	Primary Key
Indikator	Text(30)	Merupakan variabel dari tiap premis
premis value	Text(255)	Merupakan nilai dari tiap premis

b. Tabel maka

Nama Tabel : Tabel maka

Primary Key : noRule

Foreign Key : noRule; References penanganan (rule_value)

Fungsi : Menyimpan kesimpulan dari *rule* yang ada

Tabel 3.2 Tabel maka

Nama Kolom	Tipe Data	Keterangan
noRule	Number(Long Integer)	Primary Key; Foreign Key – references penanganan (rule_value)
Indikator	Text(30)	Merupakan variabel dari tiap kesimpulan
Perawatan	Text(50)	Menyatakan suatu perawatan
rule_value	Text(255)	Merupakan nilai dari tiap kesimpulan
cf_rule	Number(Decimal)	Menyimpan faktor keyakinan dari <i>knowledge engineer</i>
Goal	Yes/No	Untuk menandai, apakah konklusi dari <i>rule</i> tersebut merupakan konklusi akhir atau tidak.

c. Tabel detil_penanganan

Nama Tabel : Tabel detil_penanganan

Primary Key : detilpenanganan

Foreign Key : -

Fungsi : Menyimpan keterangan dari produk dan perawatan yang ada pada klinik kecantikan “London Beauty Centre” jika konklusi dari *rule* terpenuhi

Tabel 3.3 Tabel detil_penanganan

Nama Kolom	Type Data	Keterangan
detilpenanganan	Text(100)	Primary Key
keterangan	Text(255)	Merupakan keterangan dari detilpenanganan yang berupa penjelasan tentang produk-produk dan perawatan yang ada pada klinik kecantikan “London Beauty Centre”

d. Tabel penanganan

Nama Tabel : Tabel penanganan

Primary Key : rule_value, detilPenanganan

Foreign Key : detilPenanganan; References detil_penanganan (detilpenanganan)

Fungsi : Menyimpan jenis penanganan yang berupa produk atau perawatan yang akan dilakukan berdasarkan dari rule_value dan nama produk atau perawatan jika konklusi dari *rule* terpenuhi

Tabel 3.4 Tabel penanganan

Nama Kolom	Tipe Data	Keterangan
rule_value	Text(100)	Primary Key
detilPenanganan	Text(100)	Primary Key; Foreign Key – references detil_penanganan (detilpenanganan)
jenis	Text(50)	Merupakan penjelasan dari detilPenanganan, apakah termasuk dalam jenis produk atau perawatan

e. Tabel login

Nama Tabel : Tabel login

Primary Key :-

Foreign Key :-

Fungsi : Menyimpan *user* dan *password* yang berhak untuk melakukan login dan menggunakan sistem

Tabel 3.5 Tabel login

Nama Kolom	Tipe Data	Keterangan
level	Text(50)	Menyimpan level dari pengguna sistem, apakah pengguna termasuk user atau admin (yang berhak melakukan <i>edit</i> terhadap sistem)
user	Text(50)	Menyimpan nama dari <i>user</i> yang menggunakan sistem
pwd	Text(50)	Menyimpan <i>password</i> dari <i>user</i> yang menggunakan sistem

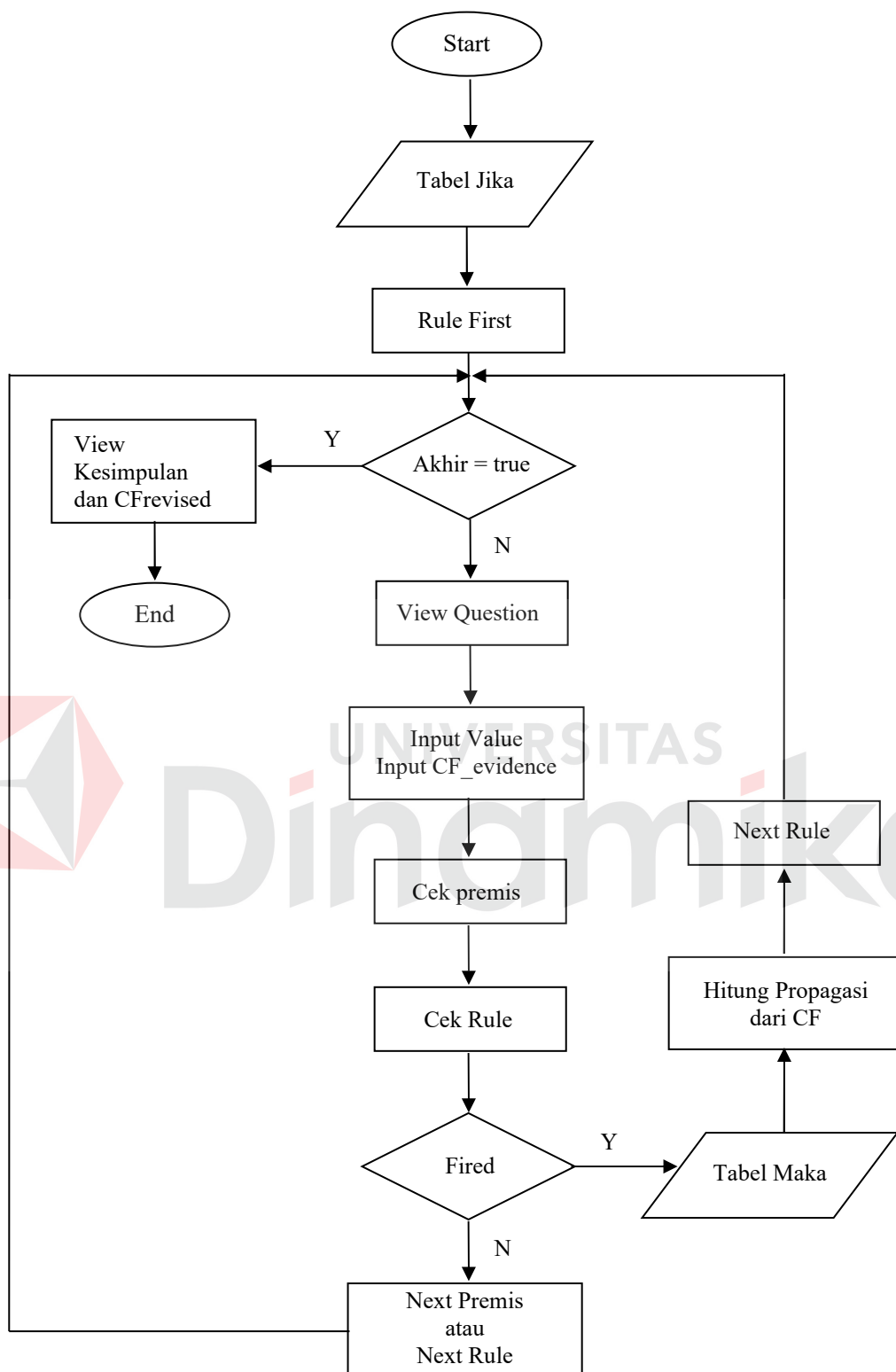
3.3.7 Inference engine

Inference engine untuk aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini menggunakan metode *Forward Chaining*. Dengan metode *Forward Chaining*, sistem ini akan mencari hasil kesimpulan dari hasil suatu sesi analisa tanya jawab dengan *user* secara menyeluruh. Maka dengan menggunakan metode *Forward Chaining* pada *inference engine*, efisiensi kerja *inference engine* akan lebih meningkat.

Secara umum, proses kerja *inference engine* dapat digambarkan menjadi seperti diagram alir sebagai berikut:



UNIVERSITAS
Dinamika



Gambar 3.27 Diagram alir proses kerja inference engine

3.3.8 Faktor kepastian (CF)

Pengetahuan dalam sebuah sistem pakar menggunakan formalisme *Certainty Factor* (CF) yang dinyatakan sebagai kumpulan dari *rules* dengan format:

IF EVIDENCE
THEN HYPOTHESIS (CF)

dimana *EVIDENCE* adalah satu atau beberapa fakta yang diketahui untuk mendukung suatu *HYPOTHESIS*. Nilai dari CF menunjukkan kepercayaan dalam suatu *HYPOTHESIS* yang berdasarkan pada beberapa fakta.

Sehingga *Certainty Factor* (CF) dalam sistem pakar ini terbagi menjadi

tiga, yaitu:

1. CF *EVIDENCE*, yaitu merupakan faktor kepastian yang didapat dari jawaban user. Pada aplikasi sistem pakar penentuan produk dan jenis perawatan di klinik kecantikan “London Beauty Centre” ini, hanya dibatasi sebanyak 6 faktor kepastian *EVIDENCE*, yaitu: (Durkin, 1994)
 - a. Tahu sedikit, dengan tingkat kepastian 0,2.
 - b. Barangkali, dengan tingkat kepastian 0,4.
 - c. Mungkin, dengan tingkat kepastian 0,6.
 - d. Kemungkinan besar, dengan tingkat kepastian 0,7.
 - e. Hampir pasti, dengan tingkat kepastian 0,8.
 - f. Pasti, dengan tingkat kepastian 1.

2. CF *CONCLUSION*, yaitu merupakan faktor kepastian dari *knowledge engineer* akan *knowledge*-nya. Pada aplikasi sistem pakar penentuan produk dan jenis perawatan di klinik kecantikan “London Beauty Centre” ini, faktor kepastian *CONCLUSION* memiliki tingkat kepastian 0,9.
3. Propagasi dari *Certainty Factor* (CF), selama eksekusi dari basis pengetahuan jika ada kaidah lain termasuk dalam hipotesis yang sama tetapi berbeda dalam faktor kepastian maka perhitungan faktor kepastian dari kaidah yang sama dihitung dari penggabungan fungsi untuk faktor keyakinan yang didefinisikan sebagai berikut: (Gonzales, 1993)

$$\begin{aligned}
 CF_{\text{revised}}(CF_{\text{old}}, CF_{\text{new}}) &= \\
 &= CF_{\text{old}} + CF_{\text{new}} (1 - CF_{\text{old}}) \quad , \text{ jika kedua } CF_{\text{old}} \text{ dan } CF_{\text{new}} > 0 \\
 &= -CF_{\text{revised}}(-CF_{\text{old}}, -CF_{\text{new}}) \quad , \text{ jika kedua } CF_{\text{old}} \text{ dan } CF_{\text{new}} < 0 \\
 &= \frac{CF_{\text{old}} + CF_{\text{new}}}{1 - \min(|CF_{\text{old}}|, |CF_{\text{new}}|)} \quad , \text{ jika satu dari } CF_{\text{old}} \text{ dan } CF_{\text{new}} < 0
 \end{aligned}$$

dimana:

$$CF_{\text{new}} = CF_{\text{conclusion}} * CF_{\text{evidence}}$$

3.3.9 Fasilitas edit

Fasilitas *edit* ini dimaksudkan untuk memudahkan *user* untuk menambah, mengurangi, atau mengedit *rules* dan premis, nama, jenis, dan keterangan dari penanganan yang berupa produk dan perawatan yang disediakan oleh klinik kecantikan “London Beauty Centre”, serta menambah atau menghapus *account* dari *user* yang menggunakan sistem. Selain itu, dengan adanya fasilitas ini, maka aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh

pada klinik kecantikan "London Beauty Centre" yang dibuat menjadi lebih fleksibel dan dinamis mengikuti perkembangan yang ada. Namun tentu saja yang dapat menggunakan fasilitas ini, haruslah *user* yang benar-benar mengerti, agar tidak terjadi hal-hal yang dapat membuat aplikasi sistem pakar ini tidak dapat berjalan sebagaimana mestinya, ataupun yang membuat hasil yang diperoleh tidak akurat. Oleh karena itu, untuk dapat menggunakan fasilitas *edit* ini, seorang *admin* haruslah *login* terlebih dahulu.

A. Edit rule

Fasilitas *edit rule* ini digunakan untuk menginputkan *rule* baru, menghapus *rule*, ataupun mengubah *rule* yang telah ada, yaitu yang terdapat pada tabel maka. Ini dimaksudkan supaya aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini dapat dinamis, berkembang mengikuti perkembangan yang ada. Selain itu, juga dimaksudkan agar dapat memperbaiki kesalahan penginputan pada tabel maka.

B. Edit premis

Fasilitas *edit* premis ini digunakan untuk menginputkan premis baru, menghapus premis, ataupun mengubah premis yang telah ada, yaitu yang terdapat pada tabel jika. Ini dimaksudkan supaya aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini dapat dinamis, berkembang mengikuti perkembangan yang ada. Selain itu, juga dimaksudkan agar dapat memperbaiki kesalahan penginputan pada tabel jika.

C. Edit detail penanganan

Fasilitas *edit* detail penanganan ini digunakan untuk menginputkan detail penanganan yang meliputi nama produk dan perawatan, serta keterangan penanganan yang baru, menghapus data penanganan dan keterangan yang ada di *database*, ataupun mengubah keterangan dari perawatan yang ada di *database*, yaitu yang terdapat pada tabel *detail_penanganan*. Ini dimaksudkan supaya aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini dapat dinamis, berkembang mengikuti perkembangan yang ada. Selain itu, juga dimaksudkan agar dapat memperbaiki kesalahan penginputan pada tabel *detail_penanganan*.

D. Edit penanganan kondisi

Fasilitas *edit* penanganan kondisi ini digunakan untuk menginputkan keterangan dari penanganan yang terdiri dari produk dan perawatan, jika konklusi dari *rule* terpenuhi, menghapus data-data yang meliputi *rule value*, penanganan, dan keterangan dari penanganan yang ada di *database*, yaitu yang terdapat pada tabel *penanganan*. Ini dimaksudkan supaya aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini dapat dinamis, berkembang mengikuti perkembangan yang ada. Selain itu, juga dimaksudkan agar dapat memperbaiki kesalahan penginputan pada tabel *penanganan*.

E. Edit manajemen user

Fasilitas *edit* manajemen *user* ini digunakan untuk menginputkan *account user* baru yang meliputi *User ID* dan *password*, serta menghapus *account* dari *user* yang tersimpan pada tabel *login*. Ini dimaksudkan untuk memperbaiki kesalahan penginputan pada tabel *login*.

3.3.10 Verifikasi

Verifikasi digunakan untuk melakukan validasi terhadap parameter dan data. Proses ini dijalankan apabila terdapat penambahan ataupun perubahan data yang digunakan sebagai basis pengetahuan sistem, sehingga tidak akan terjadi redundansi data. Pada sistem pakar penentuan produk dan jenis perawatan ini akan dilakukan tiga pengecekan, yang meliputi pengecekan *Redundant Rules*, *Circular Rules*, dan *Dead-End Rules*.

3.3.11 User interface

User interface yang baik adalah *interface* yang *user friendly*. Yang menjadi patokan suatu *interface* dikatakan *user friendly* yaitu *interface* tersebut dapat memberikan kemudahan dan kenyamanan bagi *user* dalam mengoperasikannya. *User interface* yang dirancang harus jelas dan lazim, baik format teks, ukuran *font*, maupun komponen-komponen yang digunakan, agar tidak membingungkan *user*.

Pada perancangan *form* perawatan, yaitu *form* yang membutuhkan inputan dari *user* untuk memperoleh *value* bagi *inference engine*, akan digunakan komponen *combobox*. Hal ini dimaksudkan agar inputan yang dimasukkan *user* adalah valid, karena *user* hanya akan memilih salah satu atau beberapa dari *value*

yang telah disediakan. Selain itu, akan dibuat pengecekan-pengecekan agar tidak ada kesalahan atau kekurangan dalam memperoleh *value* yang dibutuhkan *inference engine* untuk melakukan pemrosesan.

User interface aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan "London Beauty Centre" ini juga akan dibuat sedemikian atraktif dengan disertai keterangan-keterangan sehingga *user* benar-benar jelas, dan tercipta adanya suatu interaksi antara *user* dengan sistem aplikasi.



UNIVERSITAS
Dinamika

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Implementasi

Tahap implementasi program merupakan suatu tahap penerapan dari analisa dan perancangan sistem yang telah dibuat sebelumnya. Perangkat lunak yang dibangun, dikembangkan dengan bahasa pemrograman Visual Basic 6 dan penggunaan Microsoft Access sebagai *database*-nya.

Adapun kebutuhan yang harus dipersiapkan agar program dapat diterapkan adalah kebutuhan perangkat keras maupun kebutuhan perangkat lunak.

4.1.1 Kebutuhan perangkat lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah:

- a. Sistem Operasi Microsoft Windows XP Professional, Service Pack 1.
- b. Database Microsoft Access.
- c. Microsoft Visual Basic 6.0.

4.1.2 Kebutuhan perangkat keras

Kebutuhan minimum perangkat keras yang diperlukan untuk menjalankan aplikasi ini adalah:

- a. Processor Intel Pentium III 696 MHz, disarankan lebih.
- b. Memori 256 MB, disarankan lebih.
- c. Harddisk minimal 20 GB.
- d. VGA 16 MB atau lebih.
- e. Monitor VGA.

f. Keyboard dan mouse.

4.1.3 Instalasi program dan pengaturan sistem

Pengembangan aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan “London Beauty Centre” dengan menggunakan metode *Forward Chaining* di dalam aplikasi sistem pakar dan juga menggunakan *Certainty Factor* untuk menghitung tingkat kepastian ini membutuhkan perangkat lunak yang telah terinstalasi, adapun tahapan-tahapan instalasi dan pengaturan sistem adalah:

- a. Install Sistem Operasi Microsoft Windows XP Professional, Service Pack 1.
- b. Buat database baru dengan menggunakan Microsoft Access.
- c. Install aplikasi program Microsoft Visual Basic 6.0.

4.2 Tampilan Aplikasi

Tampilan aplikasi merupakan pengenalan *interface* program baik berupa tampilan input maupun tampilan output.

4.2.1 Tampilan form login

Form ini adalah form yang digunakan oleh pengguna sistem untuk melakukan login ke dalam sistem. Apabila login berhasil, maka berdasarkan tingkat hak aksesnya, maka pengguna sistem yang bersangkutan dapat mulai menggunakan sistem. Khusus untuk pengguna sistem dengan *level* admin dapat melakukan penambahan dan perubahan terhadap *rule*, penambahan dan perubahan terhadap data produk dan jenis perawatan, serta penambahan dan perubahan terhadap data pengguna sistem.

Gambar 4.1 Tampilan form login

4.2.2 Tampilan form utama

Form ini adalah MDI Form yang berfungsi sebagai kontainer untuk menampung form-form lainnya sehingga akses ke setiap form menjadi lebih mudah. Menu-menu tertentu hanya dapat diakses apabila pengguna sistem memiliki hak akses untuk mengakses menu tersebut, yaitu pengguna sistem dengan *level* admin.

Menu-menu yang disediakan adalah:

a. Menu Utama

Menu ini digunakan untuk melakukan proses penentuan produk dan jenis perawatan yang tepat bagi pasien “London Beauty Centre”, serta melakukan penambahan dan perubahan terhadap *rule*. Menu Utama ini terdiri dari Sub Menu Perawatan dan Sub Menu Forward Chaining. Sub Menu Perawatan terdiri dari Subsub Menu Wajah, Subsub Menu Mata, Subsub Menu Bibir, Subsub Menu Tangan, Subsub Menu Tubuh, dan Subsub Menu Kaki. Sub Menu Forward Chaining terdiri dari Subsub Menu Rule, Subsub Menu Premis, dan Subsub Menu Verifikasi.

b. Menu Penanganan

Menu ini digunakan untuk mengelola data penanganan terhadap produk dan jenis perawatan yang ada di klinik kecantikan “London Beauty Centre”. Menu ini terdiri dari Sub Menu Detil Penanganan dan Sub Menu Penanganan Kondisi.

c. Menu User

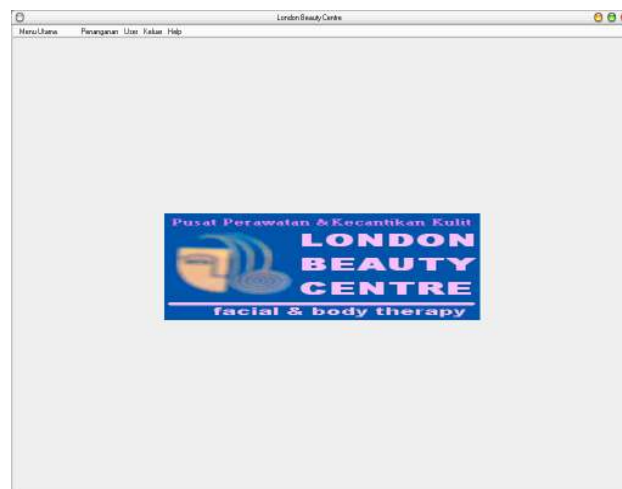
Menu ini digunakan untuk melakukan penambahan dan perubahan terhadap data pengguna sistem, serta melakukan perubahan *password* pengguna sistem. Menu ini terdiri dari Sub Menu Manajemen User dan Sub Menu Ganti Password.

d. Menu Keluar

Menu ini digunakan untuk keluar dari aplikasi dan menutup semua koneksi database. Menu ini terdiri dari Sub Menu Logout dan Sub Menu Exit.

e. Menu Help

Menu ini menjelaskan fungsi form-form dan tombol-tombol yang ada untuk membantu pengguna sistem apabila mengalami kesulitan dalam mengaplikasikan sistem tersebut.



Gambar 4.2 Tampilan form utama

4.2.3 Tampilan form perawatan wajah

Form ini berfungsi untuk melakukan pemeriksaan terhadap wajah pasien dengan menggunakan metode *Forward Chaining* yang berdasarkan dari Tampilan Wajah, Ukuran Pori-pori, Usia, Kondisi Wajah, yang diinputkan melalui *ComboBox* oleh *user* untuk menghasilkan produk dan jenis perawatan wajah yang tepat bagi pasien. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Proses

Digunakan untuk memproses data-data yang sudah diinputkan oleh *user*. Dan untuk memproses data-data tersebut, semua kriteria wajah harus berisi paling sedikit satu kondisi.

b. Batal

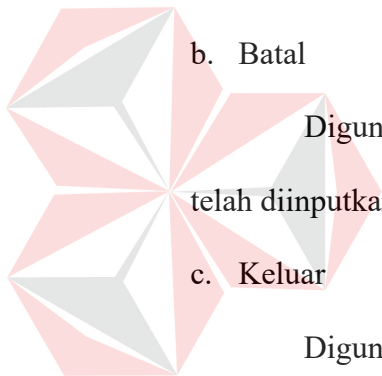
Digunakan untuk mengosongkan kriteria-kriteria wajah dari pasien yang telah diinputkan, dan untuk mengosongkan hasil dari proses.

c. Keluar

Digunakan untuk keluar dari form perawatan wajah.

d. Cetak Hasil

Digunakan untuk mencetak hasil dari konsultasi perawatan wajah, yang berupa jenis perawatan wajah dan produk yang digunakan oleh pasien pada klinik kecantikan “London Beauty Centre”.



Gambar 4.3 Tampilan form perawatan wajah

4.2.4 Tampilan form perawatan mata

Form ini berfungsi untuk melakukan pemeriksaan terhadap mata pasien dengan menggunakan metode *Forward Chaining* yang berdasarkan dari Kondisi Mata pasien, yang diinputkan melalui *ComboBox* oleh *user* untuk menghasilkan produk dan jenis perawatan mata yang tepat bagi pasien. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Proses

Digunakan untuk memproses data-data yang sudah diinputkan oleh *user*.

Dan untuk memproses data-data tersebut, semua kriteria mata harus berisi paling sedikit satu kondisi.

b. Batal

Digunakan untuk mengosongkan kriteria-kriteria mata dari pasien yang telah diinputkan, dan untuk mengosongkan hasil dari proses.

c. Keluar

Digunakan untuk keluar dari form perawatan mata.

b. Batal

Digunakan untuk mengosongkan kriteria-kriteria bibir dari pasien yang telah diinputkan, dan untuk mengosongkan hasil dari proses.

c. Keluar

Digunakan untuk keluar dari form perawatan bibir.

d. Cetak Hasil

Digunakan untuk mencetak hasil dari konsultasi perawatan bibir, yang berupa produk yang digunakan oleh pasien pada klinik kecantikan “London Beauty Centre”.



Gambar 4.5 Tampilan form perawatan bibir

4.2.6 Tampilan form perawatan tangan

Form ini berfungsi untuk melakukan pemeriksaan terhadap tangan pasien dengan menggunakan metode *Forward Chaining* yang berdasarkan dari Ciri Kulit Tangan dan Kondisi Tangan pasien, yang diinputkan melalui *ComboBox* oleh *user*

untuk menghasilkan jenis perawatan tangan yang tepat bagi pasien. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Proses

Digunakan untuk memproses data-data yang sudah diinputkan oleh *user*.

Dan untuk memproses data-data tersebut, semua kriteria mata harus berisi paling sedikit satu kondisi.

b. Batal

Digunakan untuk mengosongkan kriteria-kriteria bibir dari pasien yang telah diinputkan, dan untuk mengosongkan hasil dari proses.

c. Keluar

Digunakan untuk keluar dari form perawatan tangan.

d. Cetak Hasil

Digunakan untuk mencetak hasil dari konsultasi perawatan tangan, yang berupa jenis perawatan tangan yang dilakukan oleh pasien pada klinik kecantikan “London Beauty Centre”.

Gambar 4.6 Tampilan form perawatan tangan

4.2.7 Tampilan form perawatan tubuh

Form ini berfungsi untuk melakukan pemeriksaan terhadap tubuh pasien dengan menggunakan metode *Forward Chaining* yang berdasarkan dari Ciri Kulit Tubuh, Kondisi Tubuh, dan Kondisi Tubuh 2, yang diinputkan melalui *ComboBox* oleh *user* untuk menghasilkan produk dan jenis perawatan tubuh yang tepat bagi pasien. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Proses

Digunakan untuk memproses data-data yang sudah diinputkan oleh *user*. Dan untuk memproses data-data tersebut, semua kriteria mata harus berisi paling sedikit satu kondisi.

b. Batal

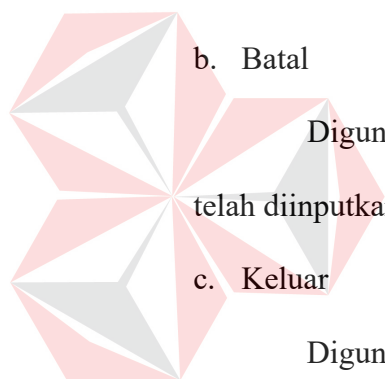
Digunakan untuk mengosongkan kriteria-kriteria tubuh dari pasien yang telah diinputkan, dan untuk mengosongkan hasil dari proses.

c. Keluar

Digunakan untuk keluar dari form perawatan tubuh.

d. Cetak Hasil

Digunakan untuk mencetak hasil dari konsultasi perawatan tubuh, yang berupa jenis perawatan dan produk yang digunakan oleh pasien pada klinik kecantikan “London Beauty Centre”.



Gambar 4.7 Tampilan form perawatan tubuh

4.2.8 Tampilan form perawatan kaki

Form ini berfungsi untuk melakukan pemeriksaan terhadap kaki pasien dengan menggunakan metode *Forward Chaining* yang berdasarkan dari Ciri Kulit Kaki dan Kondisi Kaki, yang diinputkan melalui *ComboBox* oleh *user* untuk menghasilkan jenis perawatan kaki yang tepat bagi pasien. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Proses

Digunakan untuk memproses data-data yang sudah diinputkan oleh *user*. Dan untuk memproses data-data tersebut, semua kriteria mata harus berisi paling sedikit satu kondisi.

b. Batal

Digunakan untuk mengosongkan kriteria-kriteria kaki dari pasien yang telah diinputkan, dan untuk mengosongkan hasil dari proses.

c. Keluar

Digunakan untuk keluar dari form perawatan kaki.

d. Cetak Hasil

Digunakan untuk mencetak hasil dari konsultasi perawatan kaki, yang berupa jenis perawatan kaki yang dilakukan oleh pasien pada klinik kecantikan “London Beauty Centre”.

Gambar 4.8 Tampilan form perawatan kaki

4.2.9 Tampilan form rule

Form ini hanya dapat di akses oleh level admin. Form ini berfungsi untuk memasukkan *rule* baru, mengubah *rule*, dan untuk menghapus *rule* yang telah tersimpan di *database*. Setelah memasukkan data *rule* yang baru, harus mengisi atau memasukkan premis yang bersangkutan dengan *rule* yang baru tersebut di form Premis. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Simpan

Digunakan untuk menyimpan data dari *rule* baru yang telah diinputkan.

b. Ubah

Digunakan untuk mengubah data dari *rule* yang telah tersimpan di *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

c. Batal

Digunakan untuk membatalkan pembuatan *rule* yang baru dan mengosongkan semua *field* Text.

d. Keluar

Digunakan untuk keluar dari form rule.

e. Hapus

Digunakan untuk menghapus data dari *rule* yang telah tersimpan di dalam *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

f. Cari

Digunakan untuk melakukan pencarian data *rule* yang ada di *database*, berdasarkan dari *field* No Rule, Perawatan, Indikator, Value, atau Goal, berdasarkan keinginan dari *user*.

g. Refresh

Digunakan untuk memperbaharui tampilan data *rule* yang ada di *MSFlexGrid* data, setelah dilakukan penambahan, perubahan, atau penghapusan data *rule*.

Rule ID	Indikator	Perawatan	Rule Value
1	JENIS_KULIT	TUBUH	KULIT_BERMINYAK
2	JENIS_KULIT	TUBUH	KULIT_NORMAL
3	JENIS_KULIT	TUBUH	KULIT_NORMAL_CENDERUNG_KERING
4	JENIS_KULIT	TUBUH	KULIT_KERING
5	JENIS_PERAWAT/	TUBUH	BUST_TREATMENT
6	JENIS_PERAWAT/	TUBUH	BACK_TREATMENT
7	JENIS_PERAWAT/	TUBUH	WHITENING_TREATMENT
8	JENIS_PERAWAT/	TUBUH	RELAXING_TREATMENT

Gambar 4.9 Tampilan form rule

4.2.10 Tampilan form premis

Form ini hanya dapat di *akses* oleh level admin. Form ini berfungsi untuk memasukkan premis baru dan untuk menghapus premis yang telah tersimpan di *database*. Sebelum menggunakan form ini, pastikan bahwa *rule* yang bersangkutan sudah tersimpan di *database*. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Simpan

Digunakan untuk menyimpan data dari premis baru yang telah diinputkan.

b. Ubah

Digunakan untuk mengubah data dari premis yang telah tersimpan di *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

c. Batal

Digunakan untuk membatalkan pembuatan premis yang baru dan mengosongkan semua *field* Text.

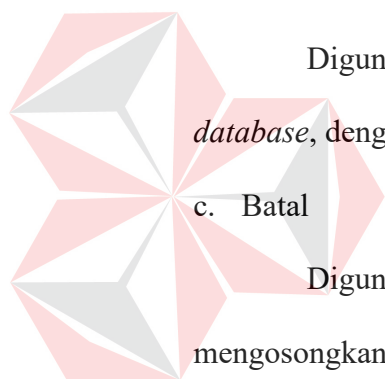
d. Keluar

Digunakan untuk keluar dari form premis.

e. Hapus

Digunakan untuk menghapus data dari premis yang telah tersimpan di dalam *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

f. Cari



Digunakan untuk melakukan pencarian data premis yang ada di *database*, berdasarkan dari *field* No Rule, No Premis, Perawatan, Indikator, atau Value, berdasarkan keinginan dari *user*.

g. Refresh

Digunakan untuk memperbaharui tampilan data premis yang ada di *MSFlexGrid* data, setelah dilakukan penambahan, perubahan, atau penghapusan data premis.

Rule ID	Premis ID	Indikator	Perawatan	Premis Value
1	1	SET	TUBUH	TUBUH
1	2	CIRI_KULIT_TUBU	TUBUH	MENGGILAT/KASAR/TEBAL/TUBUH
2	1	SET	TUBUH	TUBUH
2	2	CIRI_KULIT_TUBU	TUBUH	KENYAL_DAN_TIDAK_BERSI
3	1	SET	TUBUH	TUBUH
3	2	CIRI_KULIT_TUBU	TUBUH	AGAK_KERING
4	1	SET	TUBUH	TUBUH
4	2	CIRI_KULIT_TUBU	TUBUH	BERSISIK_HALUS_DAN_KER

Gambar 4.10 Tampilan form premis

4.2.11 Tampilan form verifikasi

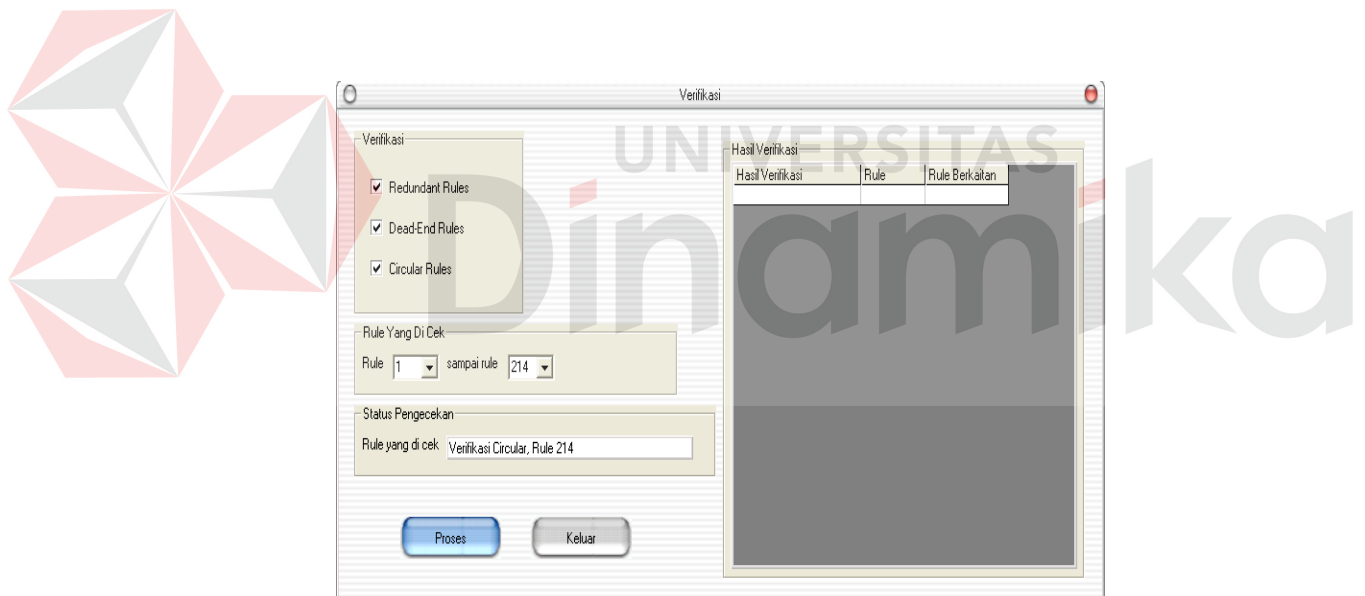
Form ini hanya dapat di *akses* oleh level admin. Form ini berfungsi untuk melakukan verifikasi *rule*, meliputi pengecekan *Redundant Rules*, *Circular Rules*, dan *Dead-End Rules*. Form ini dijalankan apabila terdapat penambahan atau perubahan pada *rule*, karena *rule* tersebut sudah ada pada sistem. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Proses

Digunakan untuk melakukan proses verifikasi, berdasarkan dari *CheckBox* yang dipilih oleh *user*. Bila terjadi *Redundant Rules*, *Circular Rules*, atau *Dead-End Rules*, hasilnya akan ditampilkan pada *MSFlexGrid* Hasil Verifikasi. Dan untuk memperbaiki *rule* tersebut, yaitu dengan melakukan klik dua kali pada *rule* yang terjadi *Redundant Rules*, *Circular Rules*, atau *Dead-End Rules*, yang ada di *MSFlexGrid* Hasil Verifikasi, dan selanjutnya akan ditampilkan Form Premis atau Form Rule untuk memperbaiki data.

b. Keluar

Digunakan untuk keluar dari form verifikasi.



Gambar 4.11 Tampilan form verifikasi

4.2.12 Tampilan form detail penanganan

Form ini hanya dapat di *akses* oleh level admin. Form ini berfungsi untuk menambahkan, mengubah, atau menghapus penanganan, yang meliputi produk dan jenis perawatan, beserta dengan keterangan dari penanganan tersebut, jika

konklusi dari *rule* terpenuhi. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Simpan

Digunakan untuk menyimpan data keterangan yang sudah diinputkan ke dalam *TextBox*.

b. Ubah

Digunakan untuk mengubah keterangan penanganan, yang meliputi keterangan produk dan jenis perawatan, yang berada di *database*.

c. Batal

Digunakan untuk membatalkan perubahan keterangan dan mengosongkan semua *field* Text.

d. Keluar

Digunakan untuk keluar dari form detail penanganan.

e. Hapus

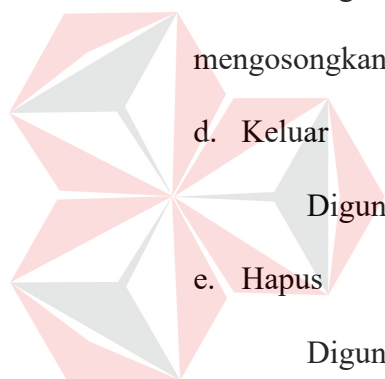
Digunakan untuk menghapus data yang telah tersimpan di dalam *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

f. Cari

Digunakan untuk melakukan pencarian data yang ada di *database*, berdasarkan dari *field* Penanganan, dan Keterangan, berdasarkan keinginan dari *user*.

g. Refresh

Digunakan untuk memperbaharui tampilan data yang ada di *MSFlexGrid* data, setelah dilakukan penambahan, perubahan, atau penghapusan data penanganan.



Penanganan	Keterangan
ACNE_LOTION	OBAT JERAWAT DENGAN PENGELUPASAN UNTUK MENI
ACNE_WASH_FDAM	SABUN PEMBERISH WAJAH UNTUK KULIT YANG BERJEF
DEEP_FACIAL_CLEANSING	SABUN PEMBERSIH WAJAH UNTUK KULIT NORMAL BER
DERMA_A_FORTE	N/A
DERMABLEND	N/A
DERMACLEAR_FORTE_GEL	N/A
DERMACLEAR_ULTRA	N/A
DERMACLEAR_ULTRA_GEL	N/A
DERMACDRTE	N/A

Gambar 4.12 Tampilan form detil penanganan

4.2.13 Tampilan form penanganan kondisi

Form ini hanya dapat di akses oleh level admin. Form ini berfungsi untuk memasukkan penanganan, yang meliputi produk atau jenis perawatan, jika konklusi dari *rule* terpenuhi. Tombol-tombol yang digunakan dalam form ini,

yaitu:

a. Simpan

Digunakan untuk menyimpan data yang sudah diinputkan.

b. Batal

Digunakan untuk membatalkan dan mengosongkan semua *field* Text.

c. Keluar

Digunakan untuk keluar dari form penanganan kondisi.

d. Hapus

Digunakan untuk menghapus data yang telah tersimpan di dalam *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

e. Cari

Digunakan untuk melakukan pencarian data yang ada di *database*, berdasarkan dari *field* Rule Value, Penanganan, dan Jenis, berdasarkan keinginan dari *user*.

f. Refresh

Digunakan untuk memperbaharui tampilan data yang ada di *MSFlexGrid* data, setelah dilakukan penambahan atau penghapusan data.

Rule Value	Penanganan	perawatan
A	FACIAL_CLASSIC_EXCLUSIVE	perawatan
A	FACIAL_CLASSIC_REGULER	perawatan
B	FACIAL_ACNE	perawatan
C	FACIAL_WHITENING	perawatan
D	FACIAL_MICROLIFT	perawatan
E	FACIAL_CLASSIC_EXCLUSIVE	perawatan
E	FACIAL_CLASSIC_REGULER	perawatan
E	FACIAL_WHITENING	perawatan
F	FACIAL_ACNE	perawatan
F	FACIAL_CLASSIC_EXCLUSIVE	perawatan

Gambar 4.13 Tampilan form penanganan kondisi

4.2.14 Tampilan form manajemen user

Form ini hanya dapat di *akses* oleh level admin. Form ini berfungsi untuk membuat *account user* baru dan untuk menghapus *account user* yang telah tersimpan di *database*. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Simpan

Digunakan untuk menyimpan *account user* baru yang dibuat.

b. Batal

Digunakan untuk membatalkan pembuatan *account user* yang baru dan mengosongkan penginputan *account user* baru tersebut.

c. Keluar

Digunakan untuk keluar dari form manajemen user.

d. Hapus

Digunakan untuk menghapus *account user* yang telah tersimpan di dalam *database*, dengan melakukan klik dua kali terlebih dahulu pada *MSFlexGrid* data.

e. Cari

Digunakan untuk melakukan pencarian data *user*, berdasarkan dari level *user*.

f. Refresh

Digunakan untuk memperbaharui tampilan data *user* yang ada di *MSFlexGrid* data.

The screenshot shows a web application window titled "User". It contains the following elements:

- Account Section:** A form with three input fields: "Level" (a dropdown menu showing "Admin"), "User ID" (a text box containing "fira"), and "Password" (a text box containing "fira").
- Action Buttons:** Four buttons labeled "SIMPAN", "BATAL", "KELUAR", and "HAPUS" are arranged horizontally below the account form.
- Data Section:** A table with three columns: "Level", "User ID", and "Password". The table contains two rows of data:

Level	User ID	Password
User	user	user
admin	admin	admin
- Pencarian Data Section:** A search area with a dropdown menu for "Field" (set to "Level"), a text input field for "Cari", and a "CARI" button. To the right of this section is a "REFRESH" button.

Gambar 4.14 Tampilan form manajemen user

4.2.15 Tampilan form ganti password

Form ini berfungsi untuk mengganti *password* dari *user*. Tombol-tombol yang digunakan dalam form ini, yaitu:

a. Ubah

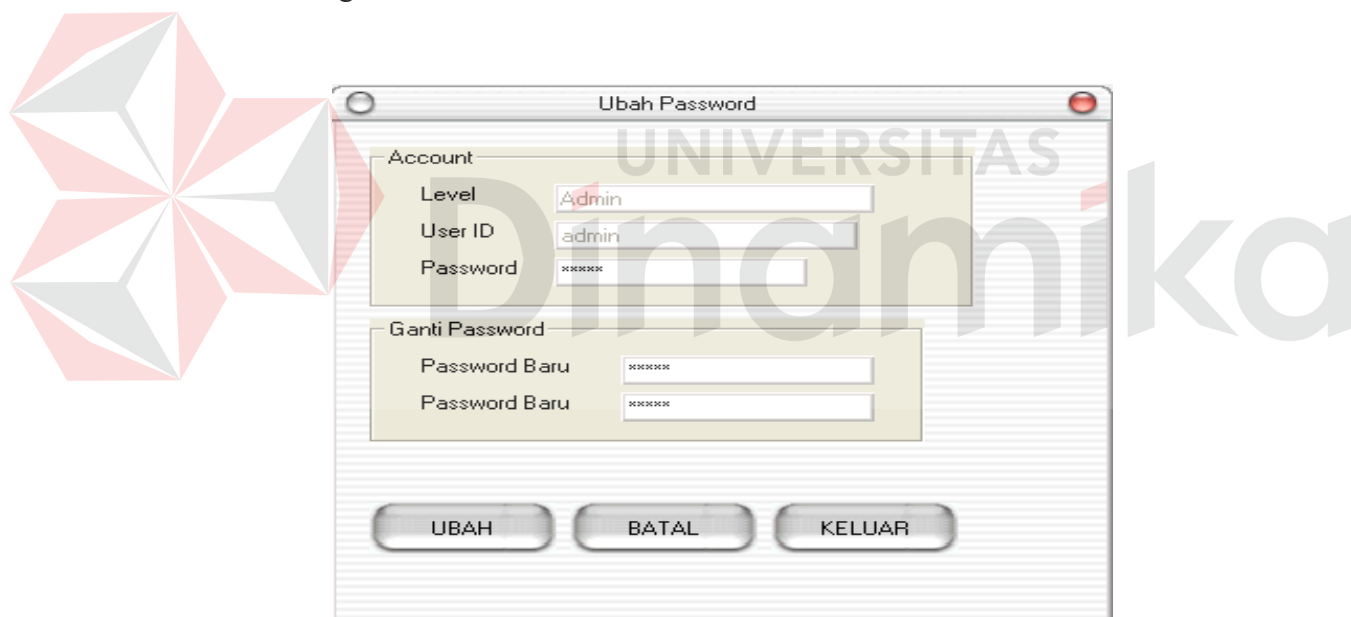
Digunakan untuk melakukan perubahan *password* dari *user*.

b. Batal

Digunakan untuk membatalkan perubahan *password* dari *user*, dan mengosongkan penginputan *password* baru oleh *user*.

c. Keluar

Digunakan untuk keluar dari Form Ganti Password.

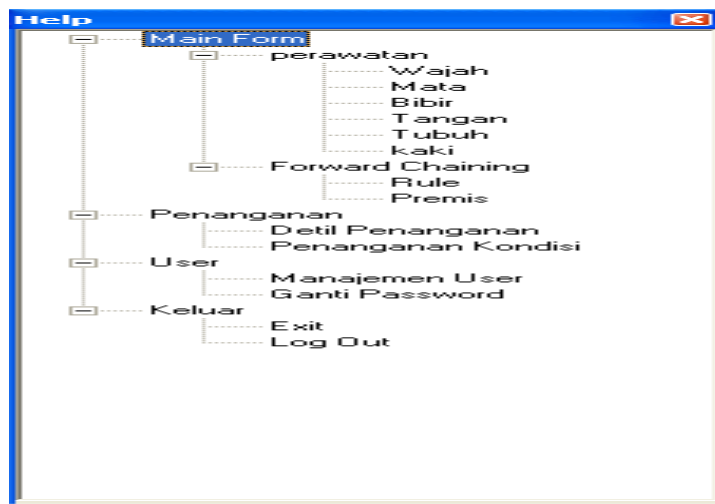


Gambar 4.15 Tampilan form ganti password

4.2.16 Tampilan Form Help

Form ini berfungsi untuk menjelaskan form-form yang ada pada aplikasi sistem pakar penentuan produk dan jenis perawatan pada klinik kecantikan “London Beauty Centre” tersebut, yang meliputi penjelasan tentang fungsi dari

masing-masing form dan penjelasan dari tombol-tombol yang digunakan di setiap form.



Gambar 4.16 Tampilan form help

4.3 Evaluasi Sistem

Pada tahap ini, dilakukan testing program untuk dievaluasi, yaitu dilakukan testing terhadap kemampuan inferensi dari sistem pakar untuk menentukan produk dan jenis perawatan bagi pasien di klinik kecantikan “London Beauty Centre” tersebut.

Dalam testing program sistem pakar penentuan produk dan jenis perawatan ini, pertama-tama *user* memasukkan kriteria-kriteria dari pasien yang di pilih dengan melakukan klik dua kali pada kriteria yang ada di *MSFlexGrid* dan juga dilakukan pemilihan kriteria yang ada di *ComboBox*. Pemilihan kriteria-kriteria tersebut berdasarkan dari perawatan yang ingin dilakukan oleh pasien, yang meliputi perawatan wajah, mata, bibir, tangan, tubuh, atau kaki.

Setelah *user* melakukan pemilihan kriteria-kriteria dari pasien tersebut, kemudian akan dilakukan pemrosesan dari kriteria-kriteria tersebut dengan

menekan tombol proses untuk menentukan jenis perawatan dan produk yang tepat bagi pasien dengan menggunakan metode *Forward Chaining*.

Pada saat tombol proses di tekan, maka sistem akan menampilkan kriteria-kriteria dan kesimpulan dengan nilai *Certainty Factor*-nya pada *List Karakteristik Pasien*. *List Karakteristik Pasien* ini merupakan *List* untuk *working memory* dari sistem, yang berisi semua informasi atau kriteria yang di pilih oleh *user* yang merupakan karakteristik dari pasien dan kesimpulan atau hasil yang diperoleh sistem dengan nilai *Certainty Factor*-nya. Selain itu *rule-rule* yang terlibat selama proses akan ditampilkan pada *List Rule Yang Terlibat*. *List Rule Yang Terlibat* ini merupakan *List* untuk *attribute queue*. Sedangkan *List Nilai Certainty Factor* merupakan *List* yang menampilkan nilai *Certainty Factor* dari setiap kriteria yang dipilih oleh *user* dan nilai *Certainty Factor* dari konklusi atau kesimpulan akhir yang di dapat dari proses inferensi.

Dalam sistem ini, *inference engine* akan mencari *rules* dimana premis dari *rules* tersebut cocok dengan informasi yang terdapat dalam *working memory*.

Setelah menemukan kecocokan, *inference engine* akan menambahkan kesimpulan *rules* tersebut ke dalam *working memory*, yang ditampilkan pada *List Karakteristik Pasien*, dan melanjutkan mencari *rules* untuk kecocokan yang baru.

Contoh evaluasi sistem:

Pemilihan atau penginputan kriteria berdasarkan karakteristik dari pasien, dapat di lihat pada Gambar 4.17., dan agar dapat memproses data-data yang sudah diinputkan tersebut maka semua kriteria harus minimal berisi satu kondisi.

Gambar 4.17. Pemilihan kriteria

Dari kriteria-kriteria yang sudah di pilih tersebut maka akan dilakukan proses untuk menghasilkan jenis perawatan dan produk yang digunakan oleh pasien dengan menggunakan metode *Forward Chaining* dan dengan menggunakan *Certainty Factor* untuk menghitung tingkat kepastian. Hasil dari proses dapat di lihat pada Gambar 4.18.

Gambar 4.18. Hasil proses inferensi

Dari proses tersebut dapat dihasilkan kesimpulan:

- a. Jenis kulit pasien, yaitu kulit kombinasi dengan pori besar dengan nilai *Certainty Factor* 0,63.
- b. Jenis perawatan, yaitu Facial Acne dengan nilai *Certainty Factor* 0,567. Jenis perawatan dan keterangannya ini dapat di lihat pada *List Jenis Perawatan Untuk Pasien*.
- c. Produk yang digunakan, yaitu Acne Lotion, Acne Wash Foam, Dermablend, Dermacorte, Dermafol, Dermatone, Retin-A 0,05%, Sunscreen I-II, dengan nilai *Certainty Factor* 0,5103. Produk-produk tersebut beserta dengan keterangannya dapat di lihat pada *List Produk Yang Digunakan*.



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1 Kesimpulan

Secara umum, aplikasi sistem pakar untuk penentuan produk dan jenis perawatan tubuh pada klinik kecantikan “London Beauty Centre” ini telah berfungsi sesuai dengan desain sistem dan memberikan hasil yang sesuai. Beberapa kesimpulan yang didapatkan dari aplikasi sistem pakar ini adalah sebagai berikut:

1. Aplikasi sistem pakar dapat membantu unit pegawai di klinik kecantikan “London Beauty Centre” tersebut dalam mendiagnosa jenis kulit dan permasalahan atau kondisi kulit yang dihadapi oleh pasien dengan lebih akurat dan cepat, serta menghasilkan atau memberikan saran kepada pasien berupa produk-produk yang harus digunakan dan jenis perawatan yang harus dilakukan oleh pasien di klinik kecantikan tersebut.
2. Aplikasi sistem pakar telah menerapkan metode *Forward Chaining* pada *inference engine* dan juga menggunakan *Certainty Factor* untuk menghitung tingkat kepastian dari tiap konklusi atau kesimpulan yang dihasilkan.

5.2 Saran

Adapun saran-saran untuk pengembangan sistem ini antara lain:

1. Aplikasi sistem pakar ini kiranya dapat dikembangkan lebih lanjut dalam bentuk *web*.
2. Sistem ini dapat dikembangkan dengan melengkapi karakteristik dari pasien dengan lebih mendetail untuk digunakan dalam proses inferensinya.

DAFTAR PUSTAKA

- Arhami, M. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi.
- Bratko, I. 1990. *Prolog Programming for Artificial Intelligence*. Singapore: Addison-Wesley Publishers Ltd.
- Durkin, J. 1994. *Expert Systems Design and Development*. London: Prentice Hall International Inc.
- Encyclopaedia Britannica. 2002. Deluxe Edition CD-ROM.
- Gonzales, A. J., and Dankel, D. D. 1993. *The Engineering of Knowledge-based Systems*. New Jersey: Prentice Hall Inc.
- Gunawan. 2000. *Kuliah Artificial Intelligence : Pengantar ke Expert System*. Surabaya.
- Ignizio, J. P. 1991. *Introduction to Expert System : The Development and Implementation of Rule-Based Expert System*. Singapore: McGraw-Hill.
- Kelompok Penyusun Buku pada Direktorat Pendidikan Masyarakat, Direktorat Jenderal Diklusepora. 2001. *Tata Kecantikan Kulit Tingkat Dasar*. Departemen Pendidikan dan Kebudayaan.
- Kelompok Penyusun Buku pada Direktorat Pendidikan Masyarakat, Direktorat Jenderal Diklusepora. 2001. *Tata Kecantikan Kulit Tingkat Terampil*. Departemen Pendidikan dan Kebudayaan.
- Tim Penerbit Andi. 2003. *Pengembangan Sistem Pakar Menggunakan Visual Basic*. Yogyakarta: Andi.