

# Sistem Pakar Pendeteksian Dini Jenis dan Perawatan Kulit Wajah dengan Menggunakan Metode *Forward Chaining*

Ari Puspita<sup>1</sup>, Ade Fitria Lestari<sup>2</sup>, Hilda Amalia<sup>3</sup>

**Abstract**— *Every human being, especially women, want to have healthy skin. That is a very important way to treat our skin. Consider the many aspects that can cause skin problems, therefore we need to choose the right products and treatments. This application is based on expert system, it is the way experts think, analyze and make conclusions or decisions. In this application program, will analyze your skin type and problem yang dihadapi to determine the right products and treatments. Care provided comprising face care, body care, eye care, hair care, hand care and foot.*

**Intisari** – Setiap manusia khususnya perempuan, ingin memiliki kulit yang sehat. Itu adalah cara yang sangat penting untuk merawat kulit kita. Pertimbangkan banyak aspek yang dapat menyebabkan masalah kulit, karena itu kita perlu memilih produk kosmetik yang cocok dengan kulit wajah. Aplikasi ini didasarkan pada sistem pakar, itu adalah cara ahli berpikir, menganalisa dan membuat kesimpulan atau keputusan. Pada program aplikasi ini, akan menganalisis jenis kulit dan masalah yang dihadapi untuk menentukan produk yang tepat dan pengobatan. Pengembangan sistem pakar menggunakan metode inferensi maju *chaining*, yang merupakan data-driven proses inferensi untuk menemukan kesimpulan. Berdasarkan kuesioner, kita dapat menyimpulkan bahwa sistem pakar ini sangat cocok untuk membantu para wanita dan pria untuk menentukan perawatan kulit nya

**Kata Kunci:** Sistem Pakar, *Forward Chaining*, Kulit

## I. PENDAHULUAN

Dewasa ini banyak para wanita yang mendambakan kulit yang putih, sehat, dan juga wajah yang cantik, bersih dan terawat. Bahkan tidak hanya kalangan wanita dewasa saja tapi juga gadis - gadis ABG, bahkan pria juga ingin memiliki kulit yang sehat dan terawat. Sekarang ini dapat dilihat para wanita dan juga pria banyak termakan pula oleh iklan – Iklan yang menjanjikan kulit mulus dan tampak lebih putih. Tak bisa terelakkan lagi jika pengetahuan tentang kulit dan wajah pada jaman sekarang ini sangat diperlukan mengingat hampir semua bahkan dapat

dipastikan semua orang yang ada didunia ini menggunakan produk atau kosmetik untuk kulit dan wajah. Bahkan bayi pun menggunakannya. Banyak sekali produk bayi yang menawarkan kulit bayi lebih halus dan juga untuk perawatan rambut seperti *baby oil*, *hair lotion*, *moisturiser*, dan masih banyak lagi. Kulit bisa menjadi cermin keadaan tubuh seseorang. Orang yang tidak sehat, kulitnya kurang cerah, kisut dan tidak elastis, karena kekurangan gizi dan nutrisi. Sementara itu, untuk menangkal pengaruh buruk akibat paparan sinar matahari, debu, gesekan, dan perubahan cuaca, kulit memerlukan makanan seimbang yang mengandung protein, kalori, dan lemak. Selain itu, membutuhkan vitamin C yang berguna sebagai kolagen (penunjang kulit), vitamin E dan A yang berfungsi sebagai antioksidan (melindungi kulit dari berbagai pengaruh luar). Lemak tak jenuh juga akan membuat penampakan kulit menjadi lebih muda. Karenanya, ketika melakukan diet, asam tak jenuh jangan sampai ditinggalkan. Asam lemak tak jenuh ini berperan untuk menjaga elastisitas kulit

## II. METODOLOGI PENELITIAN

### A. Definisi Sistem Pakar

Komputer telah berkembang sebagai alat pengolahan data, penghasil informasi. bahkan komputer juga turut berperan dalam pengambilan keputusan .tidak puas hanya dengan fungsi tersebut ,para ahli komputer masih terus mengembangkan kecanggihan komputer agar dapat memiliki kemampuan seperti manusia. Kecerdasan buatan adalah membuat bagaimana komputer bertindak seperti manusia dan memiliki kecerdasan seperti manusia. bidang – bidang yang termasuk dalam kecerdasan buatan adalah Penglihatan komputer ,pengolahan bahasa alami (*natural language*) Robotika (*robotic*), sistem syaraf buatan (*Artificial Natural System*), dan Sistem Pakar (*Expert System*). Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat di pecahkan oleh seorang pakar dalam bidang tersebut [1]. Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah, beberapa aktivitas pemecahan yang dimaksud antara lain : pembuatan keputusan (*decision making*), pemanduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberi nasehat (*advising*) dan pelatihan

<sup>1</sup> Jurusan Teknik Informatika, STMIK Antar Bangsa, Kawasan Bisnis CBD Ciledug, Blok A5 No.29-36, Jl. HOS Cokroaminoto Karang Tengah , Tangerang; Telp.021-73453000; e-mail [ari.puspita@gmail.com](mailto:ari.puspita@gmail.com)

<sup>2</sup> Program Studi Komputerisasi Akuntansi, AMIK BSI Jakarta Jl.RS. Fatmawati No.24, Pondok Labu, Jakarta Selatan; e-mail [ade.afr@bsi.ac.id](mailto:ade.afr@bsi.ac.id)

<sup>3</sup> Program Studi Komputerisasi Akuntansi, AMIK BSI Jakarta Jl.RS. Fatmawati No.24, Pondok Labu, Jakarta Selatan e-mail : [hilda.ham@bsi.ac.id](mailto:hilda.ham@bsi.ac.id)

(*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar [1].

#### B. Teknik Representasi Pengetahuan

Representasi pengetahuan adalah teknik untuk merepresentasikan basis pengetahuan yang diperoleh ke dalam suatu skema atau diagram tertentu sehingga dapat diketahui relasi atau keterhubungan antara suatu data dengan data yang lain. Teknik ini membantu kowledge engineer dalam memahami struktur pengetahuan yang akan dibuat sistem pakarnya. Terdapat beberapa teknik representasi pengetahuan yang biasa digunakan dalam pengembangan suatu pakar, yaitu:

##### 1. Rule-Based Knowledge.

Pengetahuan direpresentasikan dalam suatu bentuk fakta (facts) dan aturan (rules). Bentuk representasi ini terdiri atas premise dan kesimpulan.

##### 2. Frame-Based Knowledge.

Pengetahuan direpresentasikan dalam suatu bentuk hirarki atau jaringan frame.

##### 3. Object-Based Knowledge.

Pengetahuan direpresentasikan sebagai obyek-obyek. Obyek adalah elemen data yang terdiri dari data dan metoda atau proses.

#### C. Komponen Sistem Pakar

Merupakan sebuah program yang difungsikan untuk menirukan seorang pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan seorang pakar (Feri Fahrur Rohman). Untuk membangun sistem seperti itu maka komponen-komponen dasar yang harus dimilikinya paling sedikit adalah sebagai berikut:

##### 1. Antar muka pemakai

Komponen ini 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. Syarat utama membangun antarmuka pemakai adalah kemudahan dalam menjalankan sistem, tampilan yang interaktif, komunikatif dan mudah bagi pemakai.

##### 2. Basis pengetahuan

Ada beberapa cara merepresentasikan data menjadi basis pengetahuan, seperti dalam bentuk atribut, aturan-aturan, jaringan semantik, frame dan logika. Semua bentuk representasi data tsb bertujuan untuk menyederhanakan data sehingga mudah dimengerti dan mengefektifkan proses pengembangan program. Dalam pemrograman visual umumnya disediakan sarana untuk mengembangkan tabel-tabel penyimpanan data yang terangkum dalam sebuah database.

##### 3. Mesin inferensi

Adalah bagian sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antara sistem dan pemakai, mekanisme inferensi menguji aturan satu persatu sampai kondisi aturan itu benar. Secara umum ada dua teknik utama yang

digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu penalaran maju (*forward reasoning*) dan penalaran mundur (*reverse reasoning*).

Sedangkan untuk menjadikan sistem pakar menjadikan lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dapat dilengkapi dengan fasilitas berikut:

##### a) Fasilitas penjelasan (*Explanation*)

Merupakan bagian komponen sistem pakar yang memberikan penjelasan tentang bagaimana program dijalankan, apa yang harus dijelaskan kepada pemakai tentang suatu masalah, memberikan rekomendasi kepada pemakai, mengakomodasi kesalahan pemakai dan menjelaskan bagaimana suatu masalah terjadi. Dalam sistem pakar, fasilitas penjelasan sistem sebaiknya diintegrasikan ke dalam tabel basis pengetahuan dan basis aturan karena hal ini lebih memudahkan perancangan sistem.

##### b) Fasilitas akuisisi pengetahuan

Merupakan suatu proses untuk mengumpulkan data-data pengetahuan tentang suatu masalah dari pakar. Bahan pengetahuan dapat diperoleh dengan berbagai cara, seperti dari buku, jurnal ilmiah, pakar di bidangnya, laporan, literatur dsb. Sumber pengetahuan tsb dijadikan dokumentasi untuk dipelajari, diolah dan diorganisir secara terstruktur menjadi basis pengetahuan.

##### c) Fasilitas swa-pelatihan (*self-training*)

Merupakan proses yang paling akhir dalam pembuatan sistem pakar, sebuah pelatihan atau training sangatlah penting agar kita dapat menguji aplikasi yang sudah dibuat dan kita dapat melihat sudah sesuai kah dengan apa yang diinginkan user agar aplikasi yang kita buat mudah digunakan oleh user.

#### D. Mesin Inferensi

Metode inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk memformulasikan kesimpulan (Feri Fahrur Rohman). Mesin inferensi mempunyai dua fungsi yaitu inferensi dan kendali. Inferensi adalah proses menalar, sedangkan kendali berfungsi mengendalikan eksekusi. Inferensi melibatkan proses *watching* (pencocokan dan *unifaction* (penggabungan). Proses tersebut berdasarkan pada suatu basis data yang berisi fakta-fakta, biasanya tersimpan dalam berkas khusus dan dapat juga diperoleh dari konsultasi dan dipakai dalam proses pengujian aturan-aturan yang diisyaratkan dari basis pengetahuan. Dua teknik inferensi yaitu :

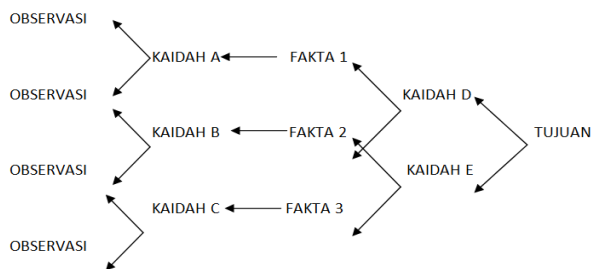
##### a. Pelacakan ke belakang (*backward chaining*)

##### b. Pelacakan ke depan (*forward chaining*)

##### a. Pelacakan ke belakang (*Backward Chaining*)

Pelacakan ke belakang adalah pendekatan yang dimotori oleh tujuan (*goal driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses

pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru

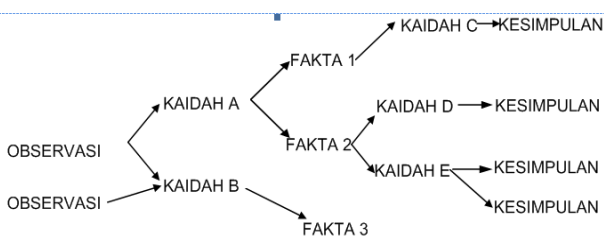


Sumber : [3]

Gbr 1. Diagram Pelacakan Kebelakang (*backward chaining*)

b. Pelacakan ke depan (*forward chaining*)

Pelacakan kedepan adalah pendekatan yang dimotori data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan menurut Kusri (2006:25). Pelacakan ke depan, mencari fakta yang sesuai dengan bagian *IF* dari aturan *IF-THEN*. 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 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.



Sumber : [3]

Gambar 2. Diagram pelacakan kedepan (*forward chaining*)

E. UML (*Unified Modelling Language*)

*Unified Modelling Language (UML)* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek (Munawar). *Unified Modelling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standart dalam industri untuk visualisasi dalam merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standard untuk merancang model sebuah sistem. Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntax/semantik*. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari 3 notasi yang telah ada sebelumnya : Grady Booch *OOD (Object Oriented Design)*, Jim Rumbaugh *OMT (Object Modeling Technique)*, dan Ivar Jacobson *OOSE (Object Oriented Software Engineering)*.

Abstraksi konsep dasar *UML* terdiri dari *structural classification, dynamic behavior* dan *model management*. *UML* mendefinisikan diagram-diagram sebagai berikut [5]:

1. *Use Case Diagram*

*Use Case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar informasi, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu. *Use Case diagram* dapat sangat membantu apabila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

2. *Class Diagram*

*Class* adalah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek (Munawar). *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class Diagram* menggambarkan struktur dan *descriptsi class, package* dan objek beserta hubungan satu sama lain seperti *containment, pewarisan, asosiasi, dan lain-lain*.

3. *Statechart Diagram*

*Statechart Diagram* menggambarkan *transisi* dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam *UML, state* digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. *Transisi antar state* umumnya memiliki

kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

#### 4. Activity Diagram

*Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

#### 5. Sequence Diagram

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri antar dimensi vertikal (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

#### 6. Collaboration Diagram

*Collaboration Diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1 (satu). *Messages* dari level yang sama memiliki *prefix* yang sama pula.

#### 7. Component Diagram

*Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

#### 8. Deployment Diagram

*Deployment/physical Diagram* menggambarkan detail bagaimana komponen di-*deploy* dalam *infrastruktur sistem*, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, *spesifikasi server*, dan hal-hal lain yang bersifat *fisikal* [4]. Sebuah *node* adalah *server*, *workstation* atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya.

#### 9. Package Diagram

Sebuah bentuk pengelompokan yang memungkinkan untuk mengambil sebuah bentuk di *UML* dan mengelompokkan elemen-elemennya dalam tingkatan unit yang lebih tinggi [4]. Kemampuan *package* yang paling umum adalah untuk mengelompokkan *class*.

#### F. Aplikasi Web

Pada awalnya situs web (*website*) merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink* atau menghubungkan satu halaman ke halaman lainnya. Informasi yang disajikan dengan web menggunakan konsep multimedia dari mulai teks, gambar, animasi, suara (*audio*) dan film (*video*). Pada perkembangannya sebuah situs web selalu dituntut agar menyajikan informasi yang *up to date*. Apabila situs tidak dirancang dengan sistematis maka programmer akan mengalami kesulitan dalam proses pemeliharannya. Salah satu solusinya adalah dengan melakukan penambahan atau perubahan dalam database.

Saat ini pemanfaatan aplikasi database *client server* sudah banyak diterapkan oleh kalangan bisnis untuk mengefisienkan pekerjaan mereka untuk memperoleh data secara cepat dan akurat. Informasi yang disajikan dari data tersebut sangat menunjang dalam proses pengambilan keputusan yang tepat. Sebab dengan menggunakan arsitektur *client server*, semua proses pengolahan data dapat dilakukan pada satu tempat saja. Sehingga proses perolehan informasi terhindar dari perolehan data kadaluwarsa akibat terpisahnya tempat untuk menyimpan data.

#### G. Pengertian Web Browser dan HTML

*Web browser* adalah program untuk menampilkan halaman yang berbentuk kode HTML. Semua halaman web ditulis dengan bahasa HTML (*Hypertext Mark Up Language*). Walaupun beberapa file mempunyai ekstensi yang berbeda (contoh .html, .php, .php3), output file-file tersebut tetap HTML. HTML adalah medium yang selalu dikirimkan ke *web browser* baik halaman itu berupa halaman statis, sebuah *script* (seperti PHP), ataupun yang dibuat oleh program CGI (*Common Gateway Interface*). HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. HTML adalah versi yang sederhana dari SGML (*Standardized Generalized Markup Language*), yaitu bahasa untuk pertukaran data. Dikatakan *markup language* karena HTML berfungsi untuk memformat file teks biasa untuk bisa ditampilkan pada web browser dengan bantuan tanda-tanda yang sudah ditentukan [6].

### III. HASIL DAN PEMBAHASAN

#### 1. Prosedur Sistem Berjalan

Prosedur kerja sistem yang ada pada klinik keluarga kita dalam pemeriksaan kulit wajah dan perawatannya adalah sebagai berikut:

##### a. Prosedur Pendaftaran

Setiap pasien yang ingin berobat harus mendaftarkan diri terlebih dahulu ke petugas administrasi. Lalu petugas administrasi akan mencatat data-data pasien

kedalam data periksa pasien. Setelah didata pasien akan mendapat Kartu berobat beserta no urut registrasi. Kemudian petugas administrasi menyerahkan data periksa pasien ke perawat. Apabila pasien telah menerima kartu berobat beserta no urut registrasi, maka bisa menunggu ruang di ruang tunggu pasien.

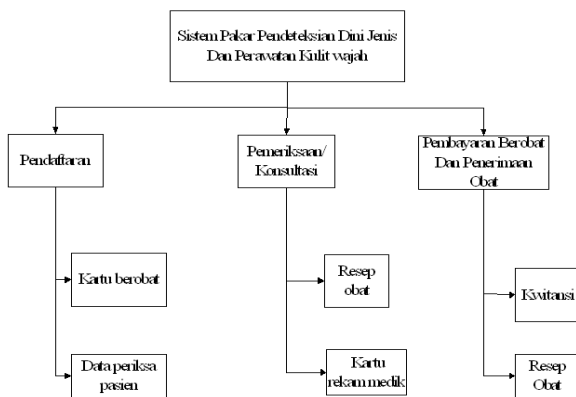
b. Prosedur Pemeriksaan Pasien

Perawat akan memanggil pasien sesuai dengan nomor registrasi pasiennya dan pasien dapat langsung masuk ke ruang pemeriksaan. Kemudian Dokter memeriksa pasien dengan cara melakukan wawancara dengan pasien seputar keluhan keluhan pasien terhadap masalah kulit wajah yang di alami pasien. Kemudian dokter melakukan tindakan terhadap wajah pasien sesuai dengan masalah pasien dan mencatat diagnosis pasien dalam kartu rekam medik. Setelah selesai dokter memberikan resep obat pada pasien.

c. Prosedur Pembayaran Berobat Dan Penerimaan Obat

Pasein menyerahkan resep obat kepada petugas administrasi dan melakukan pembarayan berobat sesuai dengan biaya pengobatan yang di lakukan. Setelah itu petugas administrasi membuat kwitansi sebagai bukti pembayaran pasien. Kemudian pasien menerima kwitansi dan obat sesuai dengan resep yang di berikan oleh dokter. Penulisan singkatan dan akronim

2. Dekomposisi Fungsi Sistem Berjalan

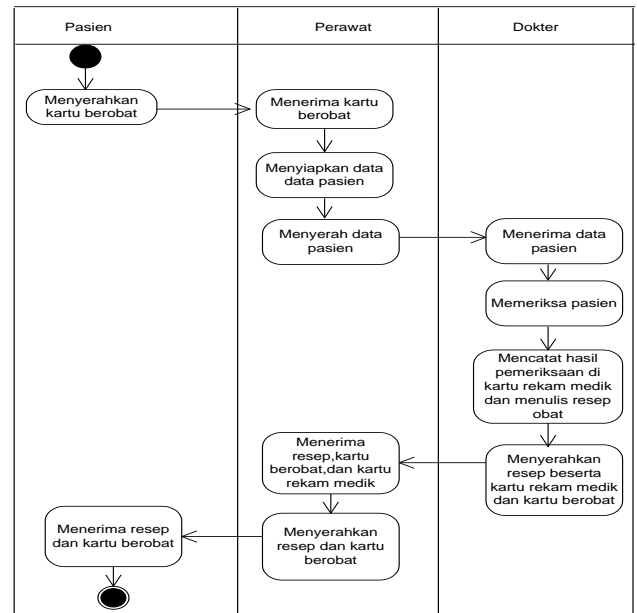


Sumber : Hasil Analisa

Gbr 3. Dekomposisi Fungsi Berjalan

3. Activity Diagram

Activity Diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus [4]. Activity Diagram menggambarkan berbagai alir aktivitas sistem yang sedang dirancang.



Sumber: Hasil Analisa

Gbr 4. Activity Diagram Pemeriksaan

4. Tabel Pakar

Tabel pakar merupakan fakta-fakta yang diperoleh dari pakar, ilmu pengetahuan, penelitian dan pengalaman - pengalaman mereka dalam mengidentifikasi ciri- ciri jenis kulit . Adapun data-data yang telah di kumpulkan adalah sebagai berikut:

TABEL I.  
CIRI CIRI KULIT

id	Ciri-ciri kulit
C01	Tidak banyak mengeluarkan sisa minyak
C02	Mengeluarkan banyak minyak di seluruh bagian wajah
C03	Pori-pori tampak besar
C04	Pori-pori kecil atau tidak terlihat sama sekali
C05	Mudah berjerawat
C06	Sering gatal-gatal
C07	Wajah tampak kemerahan
C08	Mengeluarkan banyak minyak di daerah T-Zone (hidung, dahi, dagu)
C09	Kulit terasa kencang
C10	Kulit mengelupas
C11	Terasa lembut dan kenyal
C12	Kulit terlihat kasar dan kusam
C13	Kulit terlihat lengket
C14	Kulit tampak lembab
C15	Cenderung kering dibagian pipi
C16	Alergi dengan kotoran/debu
C17	Tidak banyak mengeluarkan minyak

TABEL II  
JENIS KULIT

id	Jenis Kulit
J1	Kulit Normal
J2	Kulit Kering
J3	Kulit Berminyak
J4	Kulit Kombinasi
J5	Kulit Sensitif

TABEL III  
RELASI CIRI-CIRI

id	nama ciri- ciri	J1	J2	J3	J4	J5
C01	pori pori kecil atau tidak terlihat sama sekali	x	x		x	x
C02	banyak mengeluarkan minyak dari seluruh wajah			x		
C03	terasa kencang		x			x
C04	mengeluarkan minyak di daerah T-Zone (hidung,dahi,dagu)				x	
C05	pori pori wajah anda besar			x		
C06	mudah berjerawat			x		x
C07	sering gatal gatal jika menggunakan kosmetik yang baru di coba					x
C08	tampak kemerahan					x
C09	terasa lembut dan kenyal	x				
C10	kulit mengelupas		x			
C11	tampak kusam dan kasar		x			
C12	kulit terlihat lengket			x		
C13	tidak mudah berjerawat	x	x		x	
C14	kulit tampak lembab	x				
C15	cenderung kering di bagian pipi				x	
C16	alergi dengan kotoran/debu					x
C17	tidak banyak mengeluarkan minyak	x	x		x	x
ket :						
J1	Kulit normal					
J2	Kulit kering					
J3	Kulit berminyak					
J4	Kulit kombinasi					
J5	Kulit sensitif					

Rule-rule pada Pakar

Rule 1

**JIKA** pori pori kecil **DAN**  
Terasa lembut dan kenyal **DAN**  
Kulit tampak lembab **DAN**  
Kulit tidak mudah berjerawat **DAN**  
Tidak banyak mengeluarkan minyak **MAKA**  
Jenis kulit anda “kulit normal “

Rule 2

**JIKA** pori pori kecil **DAN**  
Kult terasa kencang **DAN**  
Kulit mengelupas **DAN**  
Kulit terlihat kasar dan kusam **DAN**  
Kulit tidak mudah berjerawat **DAN**  
Kulit tidak banyak mengeluarkan minyak **MAKA**  
Jenis kulit anda “kulit kering “

Rule 3

**JIKA** pori pori besar **DAN**  
Mengeluarkan banyak minyak diseluruh wajah **DAN**  
Mudah berjerawat **DAN**  
Kulit terlihat lengket **MAKA**  
Jenis kulit anda “kulit berminyak “

Rule 4

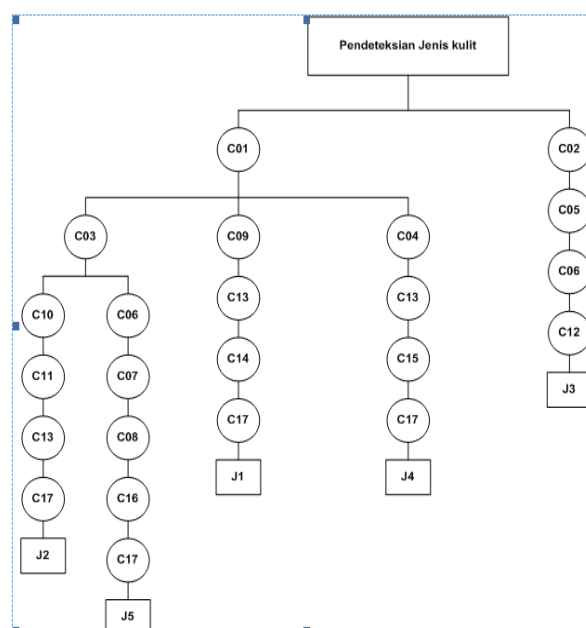
**JIKA** pori pori kecil **DAN**  
Mengeluarkan minyak di daerah T-zone **DAN**  
Cenderung kering didaerah pipi **DAN**  
Kulit tidak mudah berjerawat **DAN**  
Tidak banyak mengeluarkan minyak **MAKA**  
Jenis kulit anda “kulit kombinasi “

Rule 5

**JIKA** pori pori kecil **DAN**  
Mudah berjerawat **DAN**  
Sering gatal -gatal **DAN**  
Wajah tampak kemerahan **DAN**  
Kulit terasa kencang **DAN**  
Alergi dengan kotoran dan debu **DAN**  
Kulit tidak banyak mengeluarkan minyak **MAKA**  
Jenis kulit anda “kulit sensitif ”.

5. Pohon Keputusan Pakar

Suatu pohon adalah hierarki struktur yang terdiri dari node (simpul) yang menyimpan informasi atau pengetahuan dan cabang yang menghubungkan node. Sebuah pohon keputusan dibuat untuk memudahkan dalam pengambilan keputusan. Diagram keputusan merupakan gambaran secara sederhana permasalahan dan pemecahannya.



Sumber : Hasil Analisa

Gbr 5. Pohon keputusan pakar pendeteksian jenis kulit

6. Tampilan Web

Tampilan *web* yang dirancang adalah sebagai berikut:

- a. Tampilan *Homepage* (Halaman Utama)  
halaman homepage ini berfungsi untuk melihat isi dari skripsi yang dibuat oleh penulis melingkupi artikel-artikel kecantikan dan artikel yang berhubungan dengan landasan teori juga disediakan menu untuk membuka tampilan konsultasi, *guest book* dan *about me*.



Gbr 6. Tampilan Homepage

b. Tampilan Konsultasi

Pada Menu konsultasi terdapat tampilan input yang berfungsi untuk mengetahui jenis kulit. Data yang diinput adalah, nama, jenis kelamin, alamat dan pekerjaan. Dalam form konsultasi pengunjung dapat menjawab pertanyaan pertanyaan yang sudah di sediakan oleh sistem. Dan setelah penulis menjawab maka akan tampil solusi dan jawaban sesuai dengan jawaban yang di pilih oleh pengunjung.



Gbr 7. Tampilan Konsultasi

Berikut tampilan output dari aplikasi setelah memasukan data yang diperlukan, berdasarkan rule yang ada maka akan diperoleh hasil dari data yang di pilih, berikut contoh output:



Gbr 8. Hasil Analisa

c. Tampilan Login Admin  
Pada Tampilan ini hanya admin yang dapat login. Admin diharuskan login terlebih dahulu untuk melakukulan modifikiasi tentang isi website.



Gbr 9. Login Admin

d. Tampilan Index Admin

Pada halaman admin ini dapat melakukan peng-upload-an data-data yang diperlukan seperti artikel, landasan teori. Memungkinkan untuk meng-upload data about me, melihat isi komentar yang dibuat oleh user. Berikut tampilannya:



Gbr 10. Index Admin

IV. KESIMPULAN

Berdasarkan beberapa proses yang dilakukan dalam pembuatan aplikasi sistem pakar ini dapat disimpulkan Pemahaman mengenai jenis kulit itu sangat penting di karenakan banyak faktor yang menyebabkan kulit kita menjadi rusak seperti faktor lingkungan, faktor cuaca, faktor iklim, atau juga faktor dari diri sendiri seperti sering mencoba coba kosmetik. Itu dapat menyebabkan kerusakan kulit dan bisa juga terjangkit penyakit kulit. Aplikasi ini dibuat agar para wanita dan pria dapat mengetahui secara dini jenis kulit nya apa yang mereka punya. Sehingga mereka dapat merawat sendiri dirumah. Pendeteksian dengan menggunakan forward chaining sangat cocok untuk menelusuri suatu masalah karena setiap masalah itu pasti ada faktor nya terlebih dahulu baru kita dapat menyimpulkan masalah apa yang dihadapi nya.

## REFERENSI

- [1] Kusni, Sistem Pakar Teori Dan Aplikasi, Yogyakarta, 2006.
- [2] A. Feri Fahrur Rohman, *Journal Teknik Informarika*, p. 6, 2008.
- [3] S. Kusumadewi, Artificial Intelligence(Teknik dan Aplikasinya, Yogyakarta: Graha Ilmu, 2003.
- [4] Munawar, Pemodelan Visual dengan UML, Yogyakarta: Graha Ilmu, 2005.
- [5] R. Presman, Software Engineering: A Practitioners Approach, 6th Edition, USA: Grow hill international , 2005.
- [6] B. Sunarfrihantono, PHP dan MySQL untuk Web, Yogyakarta: Andi, 2002.
- [7] A. Ajlan, "The Comparison Between Forward and Backward Chaining," *International Journal of Machine Learning and Computing*, pp. 106-113, 2015.
- [8] J. Conallen, Building Web Application With UML Second Edition, Addison Wesley, 2002.
- [9] M. Fowler, UML Distilled Third Edition, Addison Wesley, 2004.
- [10] D. Hunt, Artificial Intelligence & Expert System Source Book, New York: Chapman and Hall, 1986.
- [11] N. Koch and H. Baumeister, "Extending UML to Model Navigation and Presentation in Web Applications," *F.A.S.T*, 2005.
- [12] J. Rumbaugh, I. Jacobson and G. Booch, The Unified Modeling Language Reference Manual 2nd Edition, 2 ed., Boston: Pearson Education, 2005.



Ari Puspita, M.Kom. Lulus dari Magister Ilmu Komputer Program Pascasarjana STMIK Nusa Mandiri Tahun 2015.



Ade Fitria Lestari, M.Kom. Lulus dari Program Strata satu (S1) Universitas Pembangunan Veteran Jakarta . Lulus dari Magister Ilmu Komputer Program Pascasarjana Universitas Budi Luhur Tahun 2014. Saat ini mengajar di AMIK BSI Jakarta



Hilda Amalia, M.Kom adalah dosen pada program studi Manajemen Informatika AMIK BSI Jakarta. Penulis lulus dari pendidikan Pasca Sarjana STMIK Nusa Mandiri pada tahun 2012 dan aktif menulis penelitian dalam bidang data mining