

METODE CASE BASED REASONING (CBR) PADA SISTEM DIAGNOSA PENYAKIT KULIT

Fitria¹, Sri Lestari², Wawan Diantoro³

¹²³Fakultas Ilmu Komputer, Institut Informatika Dan Bisnis Darmajaya
Jl. Z.A. Pagar Alam No. 93, Bandar Lampung - Indonesia 35142
Telp. (0721) 787214 Fax. (0721) 700261
e-mail : fitria@darmajaya.ac.id

ABSTRACT

Health is a very important thing since it affects daily activities. If we have health disturbance, we certainly do not like it. One of these disturbances is skin disease. When we want to identify skin disease we usually visit a doctor. However, we sometimes find a problem in handling this skin disease identification which may be influenced by the high cost of visiting doctor. Because of this reason, an expert application system is needed as consultation assistance to replace doctor's role in giving solution toward problems related to skin disease.

This study uses Case-Based Reasoning (CBR) method, of which CBR is one of knowledge based approach methods to study and solve problems according to problem experience in the past to be stored in a system called case-based. In this case, there are 44 different symptoms and 14 kinds of skin diseases. The management of data similarity uses attributive value with the closest margin. These values are calculated by using Nearest Neighbor method in order to find out the highest similarity value among existing 14 old cases. In accordance with the input result from the existing case example it shows that the result of manual calculation and system calculation are the same. This system can assist in diagnosing kinds of skin disease by taking conclusion obtained from the highest similarity value from the existing cases based on some symptoms which are input by the users (patient), and the system can also give solution in handling the problems.

Keyword: Skin Disease, CBR, Nearest Neighbor

ABSTRAK

Kesehatan adalah hal yang sangat penting, karena sangat mempengaruhi aktivitas yang dilakukan sehari-hari. Jika kita mengalami gangguan kesehatan sekecil apapun tentunya menjadi hal yang sangat tidak disukai, salah satunya gangguan penyakit kulit. Mengidentifikasi penyakit kulit biasanya harus ke dokter, namun masih mengalami kendala dalam menangani pengidentifikasi penyakit kulit, hal itu biasanya dipengaruhi oleh perlunya biaya yang mahal untuk pergi ke dokter. Oleh sebab itu diperlukan suatu aplikasi sistem pakar sebagai alat bantu konsultasi sebagai pengganti dokter dalam memberikan solusi pemecahan masalah yang berkaitan dengan penyakit kulit.

Penelitian ini menggunakan Metode Case Based Reasoning (CBR), dimana CBR merupakan salah satu metode pendekatan berbasis pengetahuan untuk mempelajari dan memecahkan

masalah berdasarkan masalah pengalaman pada masa lalu yang disimpan didalam sistem yang dinamakan basis kasus. Pada kasus ini terdapat 44 gejala yang berbeda dan 14 jenis penyakit kulit. Cara pengelolaan data similarity menggunakan nilai atribut dengan nilai kedekatan, nilai – nilai tersebut dihitung dengan menggunakan metode Nearest Neighbor untuk mencari nilai kemiripan yang paling tinggi dari 14 kasus lama yang ada. Berdasarkan hasil input dari contoh kasus yang ada menunjukkan bahwa hasil perhitungan manual dengan perhitungan sistem sama. Sistem ini dapat membantu mendiagnosa jenis penyakit kulit dengan pengambilan kesimpulan yang didapat dari hasil nilai similarity tertinggi dari kasus yang ada berdasarkan gejala-gejala yang diinputkan oleh user (pasien), dan sistem juga dapat memberikan solusi cara penanganannya.

Keyword : Penyakit kulit, CBR, Nearest Neighbor

1. PENDAHULUAN

Mengidentifikasi penyakit kulit biasanya kita harus ke dokter, namun masih mengalami kendala dalam menangani pengidentifikasi penyakit hal itu biasanya dipengaruhi oleh jumlah pasien yang terlalu banyak dan perlunya biaya yang mahal untuk pergi kedokter, oleh karena itu perlu di implementasikan dalam bentuk Sistem Pakar yang dapat menggantikan fungsi dari ahli. Sistem pakar menawarkan hasil yang lebih khusus untuk dimanfaatkan, karena sistem pakar berfungsi konsisten seperti seorang pakar manusia yang menawarkan nasihat kepada pemakai dan menemukan solusi terhadap berbagai masalah khusus, termasuk menangani masalah penyakit kulit. Sistem pakar tersebut mempunyai kelebihan menangani diagnosa penyakit selayaknya seorang dokter dan pasien juga bisa berkonsultasi dengan sistem pakar, dengan demikian

sistem pakar dapat membantu layanan cepat dalam diagnosa penyakit. Mewujudkan hal di atas dibutuhkan suatu rancangan sistem pakar untuk mendiagnosa penyakit kulit. Pengguna dapat mendiagnosa gejala yang dialami sehingga *output* sistem adalah jawaban apakah pasien menderita *impetigo krustosa*, *impetigo bulosa*, dan sebagainya dengan nilai rekomendasi yang terbesar dan memberikan solusi dan pengobatan.

1.1 Tujuan

Tujuan Rancang bangun sistem ini adalah untuk memberikan kemudahan dan referensi pada pemerintah dan masyarakat yang ingin mendapatkan informasi tentang jenis penyakit kulit beserta solusinya.

1.3 Manfaat

Manfaat yang akan diperoleh dengan adanya sistem ini yaitu:

- a. Bagi masyarakat sebagai sarana informasi dalam penentuan jenis

penyakit kulit beserta solusi penanganannya.

- b. Bagi penulis adanya sistem pendiagnosa penyakit kulit yang dibangun dapat memberikan acuan untuk langkah pembangunan sistem yang lebih baik kedepannya.

2. LANDASAN TEORI

2.1 Sistem Penalaran Komputer Berbasis Kasus

Sistem penalaran komputer berbasis kasus (*case based reasoning – CBR*) merupakan sistem yang bertujuan untuk menyelesaikan suatu kasus baru dengan cara mengadaptasi solusi-solusi yang terdapat kasus-kasus sebelumnya yang mirip dengan kasus baru tersebut [3].

2.2 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama yaitu lingkungan pengembangan (*development environtment*) dan lingkungan konsultasi (*consultant environtment*) [15]. Lingkungan pengembangan sistem pakar digunakan untuk memasukan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh penggunaan yang bukan pakar guna memperoleh pengetahuan pakar [12].

2.6.1 Komponen Utama Sistem Pakar

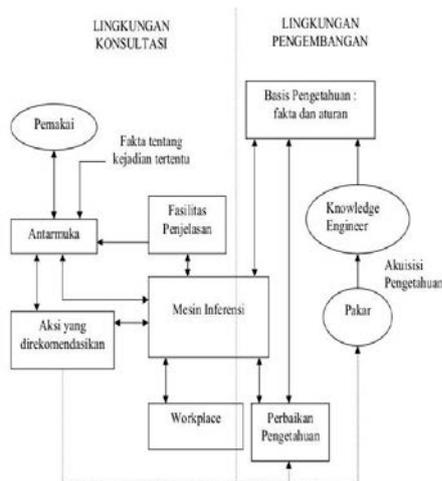
Sistem pakar terdiri dari beberapa bagian utama:

- 1) Lingkungan pengembangan yang digunakan dalam sistem pakar untuk membangun komponen-komponennya dan menempatkan pengetahuan dalam basisnya.
- 2) Lingkungan konsultasi yang digunakan oleh pemakai untuk mendapatkan informasi atau pengetahuan dari pakar.
- 3) Lingkungan pengembangan digunakan sebagai sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi.

2.6.2 Fasilitas Akuisisi pengetahuan

Akuisisi pengetahuan dilakukan proses akumulasi, transfer dan transformasi kepakaran. Pemecahan persoalan dari sumber pengetahuan ke perangkat lunak untuk membantu atau mengembangkan basis pengetahuan-pengetahuan tentang dasar tentang domain meliputi istilah dan konsep dasar. Pengetahuan pakar tersebut terdapat dalam jurnal, buku dan sebagainya. Namun, tidak semua kepakaran dapat didokumentasikan. Prosedur interaktif diperlukan untuk

mendapatkan informasi tambahan dari pakar dalam mengembangkan pengetahuan dasar. Proses ini cukup kompleks dan biasanya membutuhkan bantuan rekayasa pengetahuan. Komponen sistem pakar ditunjukkan pada gambar 1. dibawah ini



Gambar 1. Komponen Sistem Pakar

2.6.3 Mesin Inferensi

Mesin inferensi berfungsi untuk melakukan penelusuran pengetahuan yang terdapat dalam basis pengetahuan untuk mencapai kesimpulan tertentu. Mesin Inferensi menyediakan arahan tentang bagaimana menggunakan pengetahuan sistem dalam membangun agenda yang mengorganisasikan dan mengontrol langkah yang diambil untuk memecahkan persoalan saat konsultasi berlangsung.

Ada 3 elemen utama dalam mesin inferensi:

a. *Interpreter*

Mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai.

b. *Scheduler*

Akan mengontrol agenda.

c. *Consistency enforce*

Bertujuan memelihara konsistensi dalam mempresentasikan solusi yang bersifat darurat.

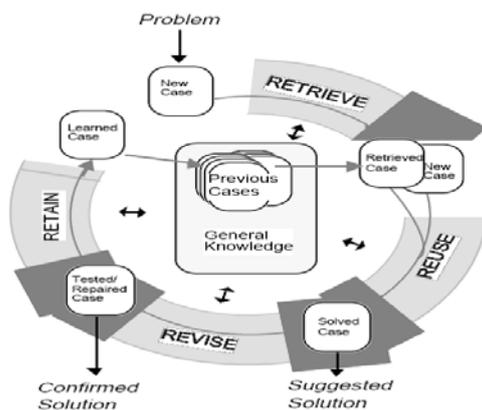
2.3 Cased Based Reasoning

Case-Based Reasoning (CBR) merupakan metode yang digunakan untuk mengimplementasikan sistem diagnosa komputer ke dalam aplikasi di dunia nyata. CBR juga dapat digunakan untuk menganalisa suatu masalah sesuai dengan kasus yang dihadapi dan untuk selanjutnya mengklasifikasikan kasus tersebut berdasarkan pada pengalaman masa lalu pengklasifikasian. Kelebihan dari CBR yaitu memungkinkan penggunaan contoh kasus masa lalu untuk mengakuisisi pengetahuan dan akhirnya diketahui pokok permasalahannya. Selain itu CBR juga dapat mencari solusi dari permasalahan tersebut berdasarkan dari pengalaman kasus masa lalu sehingga segala permasalahan dapat diselesaikan untuk selanjutnya kasus serta solusinya disimpan untuk kemudian dapat digunakan kembali

untuk memecahkan kasus baru, jika kasus tersebut hampir sama atau mungkin sama dengan kasus terdahulu.

2.7.2 Struktur CBR

Menurut Aamont & Plaza, selain CBR cycle, gambaran dari struktur CBR dapat lebih memperjelas proses dari CBR secara detail [1] seperti tampak pada keempat proses yang telah dilakukan pada CBR Cycle, semuanya masih dapat dipecah kembali menjadi beberapa bagian pekerjaan yang harus dilakukan, seperti keterangan berikut : Proses ini terdiri dari memilih informasi apa dari kasus yang akan disimpan, cara menyusun kasus agar mudah untuk menentukan masalah yang mirip, dan bagaimana mengintegrasikan kasus baru pada struktur memori. *Cbr cyle* ditunjukkan pada gambar 2. dibawah ini



Gambar 2. CBR Cycle

Pada saat terjadi permasalahan baru, pertama-tama sistem akan melakukan

proses *Retrieve*. Proses *Retrieve* akan melakukan dua langkah pemrosesan, yaitu pengenalan masalah dan pencarian persamaan masalah pada database. Setelah proses *Retrieve* selesai dilakukan, selanjutnya sistem akan melakukan proses *Reuse*. Di dalam proses *Reuse*, sistem akan menggunakan informasi permasalahan sebelumnya yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru. Pada proses *Reuse* akan menyalin, menyeleksi, dan melengkapi informasi yang akan digunakan. Selanjutnya pada proses *Revise*, informasi tersebut akan dikalkulasi, dievaluasi, dan diperbaiki kembali untuk mengatasi kesalahan-kesalahan yang terjadi pada permasalahan baru. Pada proses terakhir, sistem akan melakukan proses *Retain*. Proses *Retain* akan mengindeks, mengintegrasikan, dan mengekstrak solusi yang baru. Selanjutnya, solusi baru itu akan disimpan ke dalam *knowledge-base* untuk menyelesaikan permasalahan yang akan datang. Tentunya, permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan dengannya.

2.7.3 Algoritma Nearest Neighbor

Proses retrieve (penelusuran kasus) pada sistem penalaran komputer berbasis kasus untuk sistem pendiagnosa penyakit kulit

peoderma ini menggunakan *algoritma nearest neighbor*. *Algoritma nearest neighbor* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur-fitur yang ada [14].

Adapun rumus untuk melakukan perhitungan kedekatan antara dua kasus adalah sebagai berikut :

$$\text{Similarity (T,S)} = \frac{\sum_{i=1}^n f(T_i, S_i) \times W_i}{W_i}$$

Keterangan :

T = kasus baru

S = kasus yang ada dalam penyimpanan

N = jumlah atribut dalam setiap kasus

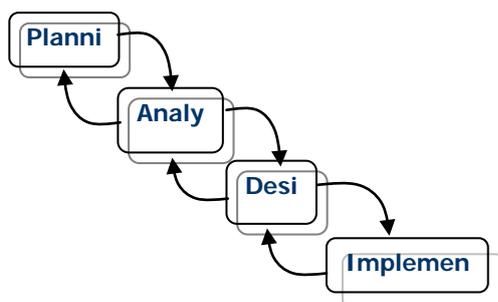
I = atribut individu antara 1 s.d. n

f = fungsi *similarity* atribut i antara kasus T dan kasus S

w = bobot yang diberikan pada atribut ke-i

2.7 Metode Pengembangan Sistem

Pada fase SDLC suatu proyek pengembangan sistem dapat dibagi kedalam empat fase dan teknik-teknik sistem yang digunakan adalah model waterfall. [5] model water fall dapat dilihat pada gambar 3. dibawah ini



Gambar 3. model waterfall.

2.13 Penyakit Kulit

Penyebab penyakit kulit dapat di sebabkan oleh *microorganism*, contohnya penyakit kulit jenis *peoderma*. *Pioderma* adalah penyakit kulit yang disebabkan oleh *microorganism*, *Staphylococcus* dan *Streptococcus* atau oleh keduanya. Penyakit kulit *peoderma* terdiri dari *Impetigo*, *Folikulitis*, *Furunkel* (bisul), *Ektima*, *Pionikia*, *Flegmon*, *Hidraadenitis Suprativa*, *Abses Multipel* (kelenjar keringat), *Erisipelas*, *selulitis* dan *Staphylococcal scalded skin syndrome* [6].

3. METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini sebagai berikut.

3.2 Metode Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan metode sebagai berikut.

- a) Observasi
- b) Studi Pustaka
- c) Wawancara

3.3 Tahapan Pengembangan Perangkat Lunak

a. Perencanaan

Pada tahapan ini, penulis mengumpulkan semua kebutuhan elemen sistem kemudian dialokasikan pada sistem yang ada, dan pada tahap ini berkaitan dengan penentuan kebutuhan pengguna dan perencanaan proyek. Perencanaan yang harus dipersiapkan meliputi menyiapkan metode sistem pakar yaitu *case based reasoning* untuk melakukan penelusuran dari basis pengetahuan. Metode *case based reasoning* merupakan sistem yang bertujuan untuk menyelesaikan suatu kasus baru dengan cara mengadaptasi solusi-solusi yang terdapat kasus-kasus sebelumnya yang mirip dengan kasus baru tersebut.

b. Proses Reuse

Dari perhitungan diatas kasus yang memiliki bobot kemiripan paling rendah adalah kasus lama no 4 yaitu sebesar 0.174737. Dan kasus yang menghasilkan bobot kemiripan paling tinggi adalah kasus lama no 3 yaitu sebesar 0.526316. Pada proses *reuse*, solusi yang diberikan adalah solusi dengan bobot kemiripan kasus lama dengan kasus baru yang paling tinggi. Hasil perhitungan dengan bobot paling tinggi menjadi solusi kasus no 3 yang direkomendasikan.

c. Proses Revise

Proses *revise* adalah proses peninjauan kembali kasus dan solusi yang diberikan jika pada proses *retrieve* sistem tidak dapat memberikan hasil diagnosa yang tepat. Pada kasus lama no 3 diatas sudah menghasilkan solusi. Informasi berupa masukan ciri pada kasus baru yang tidak ditemukan kemiripannya dengan basis pengetahuan (*rule*) tersebut akan ditampung pada suatu tabel khusus (tabel *revise*) yang selanjutnya akan dievaluasi dan diperbaiki kembali oleh pakar untuk menemukan solusi yang tepat.

d. Proses Retain

Setelah proses *revise* selesai dan sudah ditemukan solusi yang benar-benar tepat oleh pakar barulah admin mulai menambah aturan dengan memasukkan data kasus baru yang sudah ditemukan solusinya tersebut ke dalam basis pengetahuan yang nantinya dapat digunakan untuk kasus berikutnya yang memiliki permasalahan yang sama. Proses inilah yang disebut dengan proses *retain*.

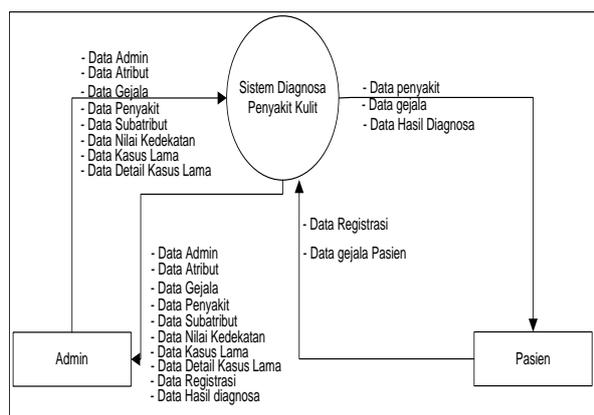
e. Perancangan

Pada tahapan ini proses desain sistem dibagi berdasarkan kebutuhan-kebutuhan berdasarkan analisis ke dalam bentuk yang

mudah dimengerti pengguna. Perancangan dimulai dari penyusunan konteks diagram dan *data flow diagram* (DFD).

1. Diagram Konteks

Admin menginputkan data admin, data atribut, data gejala, data penyakit, data Sub atribut, data nilai kedekatan, data kasus lama, data detail nilai kasus lama. Kedalam sistem diagnosa penyakit kulit kemudian dari sistem diagnosa penyakit kulit data-data tersebut ditampilkan kembali ke admin. Pasien menginputkan data registrasi dan data gejala pasien kedalam sistem diagnosa penyakit kulit kemudian dari sistem diagnosa penyakit kulit, data penyakit kulit, data gejala, dan data hasil diagnosa ditampilkan keadmin sebagai hasil kesimpulan. Seperti ditunjukkan gambar 4. diagram konteks



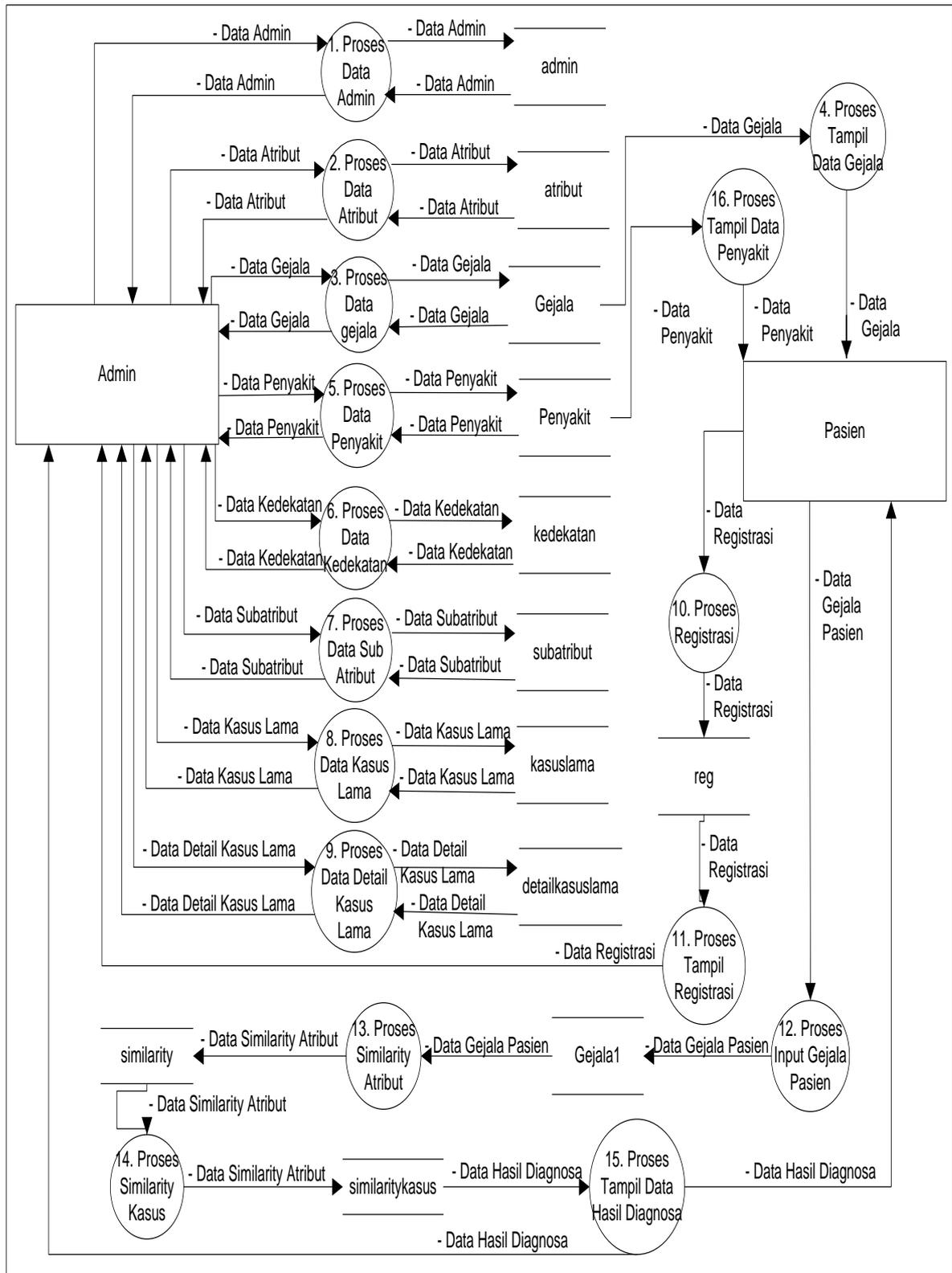
Gambar 4. Diagram Konteks.

2. Data flow diagram level 0

Pada proses 1 admin mengalirkan data admin, data admin diproses dan dialirkan ketabel admin, kemudian dari table admin data admin diproses dan dialirkan kembali keadmin. Pada proses 2 admin mengalirkan data atribut, data atribut diproses dan dialirkan ketabel atribut kemudian dari dari tabel atribut data atribut diproses dan dialirkan kembali keadmin. Pada proses 3 admin mengalirkan data gejala, data gejala diproses dan dialirkan ketabel gejala kemudian dari dari tabel gejala data gejala diproses dan dialirkan kembali keadmin. Pada proses 4 dari tabel gejala data gejala dialirkan dan diproses kemudian ditampilkan ke pasien. Pada proses 5 admin mengalirkan data penyakit, data penyakit diproses dan dialirkan ketabel penyakit kemudian dari dari tabel penyakit data penyakit diproses dan dialirkan kembali keadmin. Pada proses 6 admin mengalirkan data nilai kedekatan, data nilai kedekatan diproses dan dialirkan ketabel kedekatan kemudian dari dari kedekatan data nilai kedekatan diproses dan dialirkan kembali keadmin. Pada proses 7 admin mengalirkan data sub atribut, data sub atribut diproses dan dialirkan ketabel

sub atribut kemudian dari dari tabel sub atribut data sub atribut diproses dan dialirkan kembali keadmin.Pada proses 8 admin mengalirkan data kasus lama, data kasus lama diproses dan dialirkan ketabel kasus lama kemudian dari dari tabel kasus lama data kasus lama diproses dan dialirkan kembali keadmin.Pada proses 9 admin mengalirkan data detail kasus lama, data detail kasus lama diproses dan dialirkan ketabel detail kasus lama kemudian dari dari tabel detail kasus lama data detail kasus lama diproses dan dialirkan kembali keadmin.Pada proses 10 pasien mengalirkan data registrasi data registrasi diproses dan dialirkan ketabel reg, dari table reg data registrasi diproses tampil registrasi pada proses 11 dan kemudian data registrasi dialirkan

keadmin.Pada proses 12 pasien mengalirkan data gejala pasien kemudian dialirkan diproses 12 input gejala pasien, data gejala pasien dialirkan ketabel gejala1, kemudian data gejala pasien dialirkan keproses 13 proses similarity atribut, data similarity atribut dialirkan ketabel similarity, data similarity atribut dialirkan diproses14 similarity kasus dialirkan ketabel similarity kasus data hasil diagnosa diproses15 ditampilkan data diagnosa, kemudian data hasil diagnosa tersebut dialirkan kembali keadmin dan kepasien.Pada proses 16 data penyakit pada data penyakit dialirkan dan diproses kemudian data penyakit dialirkan kepasien. Seperti pada gambar 5. *Data Flow Diagram* dibawah ini.



Gambar 5. Data Flow Diagram Level 0

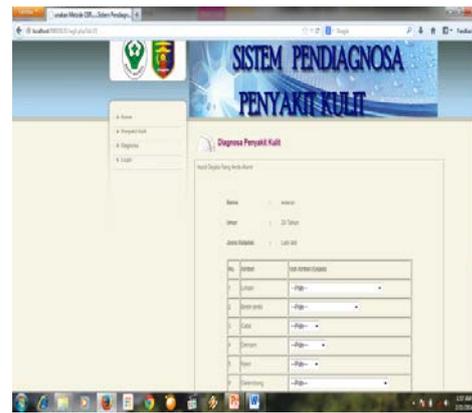
4. HASIL DAN PEMBAHASAN

4.1 Implementasi Program

Hal yang perlu diperhatikan sebelum menjalankan website pada komputer PC adalah pastikan pada komputer tersebut *Apache* telah terinstal. Untuk menjalankan program ini dapat dilakukan dengan membuka program *Mozilla Firefox* yang diambil dari menu *start* kemudian pilih menu *Mozilla Firefox*, selanjutnya jendela isian alamat browser ketikkan <http://localhost> apabila kemudian tampilan jendela informasi mengenai *appserv*, maka *appserv* berjalan dengan baik. Setelah *appserv* berjalan, lalu ketikkan alamat website dengan mengetikkan <http://localhost/09010121/>

4.2 Halaman diagnosa

Pada halaman ini terdapat form pemilihan gejala yang akan diproses untuk diagnose penyakit. Form ini digunakan untuk konsultasi pasien pengguna sistem. Halaman diagnosa ditunjukkan pada gambar 6. berikutini.



Gambar 6. Halaman diagnosa

4.3 Halaman Hasil diagnosa

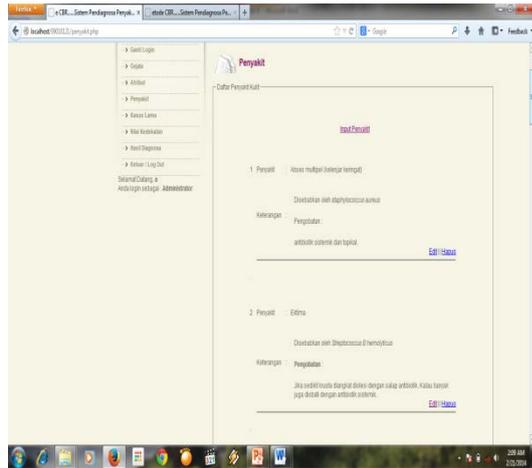
Pada halaman ini terdapat hasil diagnose penyakit kulit beserta solusi yang didapat, sehingga pasien dapat mengetahui langkah- langkah penanggulangan. Halaman hasil diagnosa ditunjukkan pada gambar 7. berikutini.



Gambar 7. Halaman Hasil diagnosa

4.4 Halaman Input data penyakit

Pada halaman ini terdapat form input data penyakit yang telah diinputkan admin berdasarkan data-data dari sang pakar. Halaman input data penyakit ditunjukkan pada gambar 8. berikutini



Gambar 8. Halaman input data penyakit

4.5 Halaman Input kasus lama

Pada halaman ini terdapat form Input kasus lama yang diinputkan oleh admin berdasarkan data – data dari sang pakar dari kasus masa lampau.Halaman input kasus lama ditunjukkan pada gambar 9. berikutini.

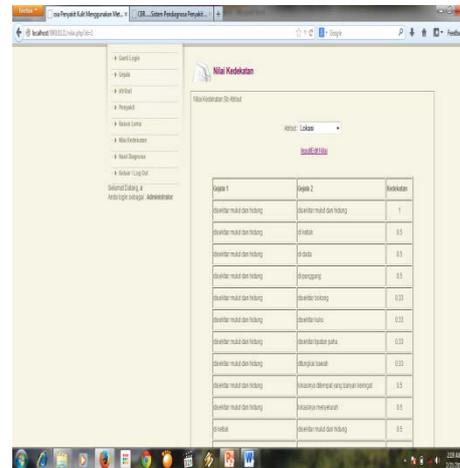


Gambar 9.Halaman Input kasus lama

4.6 Halaman Input nilai kedekatan

Pada halaman ini terdapat form input nilai kedekatan yang diinputkan oleh admin

berdasarkan data dari sang pakar.Halaman input nilai kedekatan ditunjukkan pada gambar 10. berikutini.



Gambar 10. Halaman Input nilai kedekatan

4.7 Halaman hasil diagnosa

Pada halaman ini terdapat daftar penyakit hasil diagnosa yang telah diinputkan oleh pasien dari form konsultasi pasien. Halaman hasil diagnosa ditunjukkan pada gambar 11. berikutini.



Gambar 11. Halaman hasil diagnosa

4.8 Pembahasan

Perancangan antar muka dapat dilakukan pengujian dengan mencoba menjalankan aplikasi program yang telah dirancang. Sebelum melakukan pengujian terhadap sistem yang telah dibuat, terlebih dahulu dilakukan pengujian perhitungan manual dengan menggunakan rumus similarity yang terdapat pada algoritma yang dipakai dalam proses penelusuran kasus.

Setelah selesai melakukan perhitungan manual dan nilai total dari keseluruhan setiap indikator, nilai kedektan dan bobot maka dapat diperoleh total nilai perhitungan similarity. User dapat langsung mencoba membandingkan hasil perhitungan manual dengan sistem yang telah dibuat dengan memasukan indikator atau gejala yang ada kedalam sistem. dari proses tersebut akan memperoleh informasi nilai total keseluruhan. Perhitungan manual dengan perhitungan menggunakan sistem akan memperoleh nilai total yang sama dari setiap kasus. Berikut dapat dilihat pada tabel 4.1.

Tabel 4.1 data perhitungan manual dengan menggunakan sistem

Kasus	Manual	Sistem
1	0.297895	0.297895
2	0.350526	0.350526
3	0.526316	0.526316
4	0.174737	0.174737
5	0.174737	0.174737

6	0.245263	0.245263
7	0.192632	0.192632
8	0.245263	0.245263
9	0.227368	0.227368
10	0.332632	0.332632
11	0.332632	0.332632
12	0.297895	0.297895
13	0.227368	0.227368
14	0.324211	0.324211

Dari tabel diatas didapat hasil perhitungan *similarity* hasil dari perhitungan manual dan perhitungan sistem, dimana dari perhitungan tersebut didapat nilai 0.526316 sebagai nilai yang terbesar. Nilai 0.526316 terdapat kemiripan pada kasus lama nomor 3 yaitu penyakit *impetigo neonatorum*.

Hasil analisis perangkat lunak, sistem operasi, dan perangkat keras yang digunakan untuk menjalankan aplikasi ini, telah sesuai dengan kebutuhan sistem seperti, *Microsoft Windows XP, Dreamweaver, Appserv*.

SIMPULAN

Berdasarkan hasil analisis masalah dan pembahasan, hasil kesimpulan dari penelitian ini adalah sebagai berikut:

1. Menghasilkan sistem pendeteksian penyakit kulit yang berfungsi untuk membantu penderita penyakit kulit (pasien) dalam melakukan pendeteksian suatu penyakit tanpa

harus pergi kedokter dengan biaya yang mahal.

2. Sistem yang diusulkan metode *Case base Reasoning* (CBR) dengan proses *retrieve* yaitu pencarian atau penelusuran kemiripan antara kasus lama dengan kasus baru sehingga hasil yang diperoleh akan lebih baik dan cepat

DAFTAR PUSTAKA

- [1] Aamodt & Plaza E 1994. *Case based reasoning : Foundational issues, methodological variations, and system approaches*, IA Com-artificial intelligence communication, IOS Press
- [2] Arhami Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Penerbit Andi, Yogyakarta.
- [3] Andriana, dkk. 2008. *Sistem Penalaran Komputer Berbasis Kasus (Case Based Reasoning-CBR)*. Yogyakarta, Ardana Media.
- [4] Beheshti, 2007, *Impetigo, a brief review*, Fasa-Iran: Fasa Medical School
- [5] Dennis, A., B.H. Wixom, and D.Tegarden. 2005. *System Analysis and Design with UML version 2.0*. New Jersey: John Willey and Sons.
- [6] Djuanda, dkk. 1987. *Ilmu penyakit kulit dan kelamin*. Jakarta, Fakultas Kedokteran universitas indonesia.
- [7] Djuanda, 2005, *Ilmu penyakit kulit dan kelamin* Jakarta, Fakultas Kedokteran universitas Indonesia.
- [8] Fatta Al Hanif. 2007. *Analisis dan Perancangan Sistem Informasi*. Yogyakarta, Penerbit Andi.
- [9] Irawan, eri. 2009. *Sistem pakar untuk menganalisa error pada PC dengan metode Case Based Reasoning*. Tersedia <http://www.scribd.com/doc/50848367/CBR-indonesia>. Diakses 26 Febuari 2011.
- [10] Jogiyanto H.M. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta, Penerbit Andi.
- [11] Muzid, Syafiul. *Teknologi Penalaran Berbasis Kasus (Case Based Reasoning) untuk Diagnosa Penyakit Kehamilan* (2 Juni 2008). Tersedia: Journal.uui.ac.id/index.php/Snati/. Diakses tanggal 26 September 2011..
- [12] Muhammad Arhami. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta, Penerbit Andi Offset.
- [13] Kristanto Andi. 2003. *Konsep dan Perancangan Database*. Penerbit Andi Offset, Yogyakarta.
- [14] Kusriani, Emha Taufik Lutfi. 2009. *Algoritma Data Mining*. Yogyakarta, Penerbit Andi Offset.
- [15] Turban, E, dkk. 2005. *Decision Support Systems and Intelligent Systems*. Yogyakarta Penerbit Andi Offset.