APLIKASI UNTUK MENDIAGNOSA PENYAKIT EPILEPSI MENGGUNAKAN METODE *CASE BASE REASONING* BERBASIS WEB

Realdo Meyantoro[1], Selvia Lorena Br. Ginting[2], Sri Nurhayati[3]
[123]Jurusan Teknik Komputer Universitas Komputer Indonesia, Bandung
rea7do@gmail.com

ABSTRAK

Sering kali setiap orang selalu menganggap remeh gejala yang dialami tanpa mencari tahu sebab dari gejala yang dirasakan. Gejala-gejala yang dirasakan seringkali membuat kepanikan orang disekitar seperti serangan penyakit epilepsi ini. Penyakit epilepsi ini terdiri dari berbagai jenis serangan, hal ini belum diketahui secara luas oleh masyarakat. Masalah ini mendorong pembuatan aplikasi diagnosa penyakit epilepsi berbasis web. Aplikasi ini menggunakan penalaran berbasis kasus/Case Based Reasoning (CBR) yang merupakan metode yang dipergunakan untuk membangun sebuah sistem berbasis pengetahuan. Sumber pengetahuan sistem diperoleh dengan mengumpulkan penanganan kasus-kasus oleh seorang pakar. Kasus-kasus yang dipergunakan dalam sistem diperoleh dari catatan penanganan kasus diagnosa serangan penyakit epilepsi dari seorang dokter spesialis saraf. Aplikasi ini akan memberikan keluaran berupa kemungkinan penyakit dan solusi yang harus dilakukan yang didasarkan pada kemiripan kasus baru dengan pengetahuan yang dimiliki sistem.

Kata kunci : Sistem Berbasis Pengetahuan, Case Base Reasoning, web, penyakit epilepsi

1. PENDAHULUAN

Gejala yang mucul secara tiba-tiba seringkali membuat seseorang panik dan bingung untuk melakukan tindakan yang tepat dalam menanganinya. Terlebih jika benar penyakit itu adalah serangan dari epilepsi.Memang banyak sekali gejala yang baru muncul saat serangan epilepsi mulai menyerang seseorang. Epilepsi atau penyakit ayan ini juga memerlukan pengobatan yang cukup lama bahkan bisa seumur hidup, akan tetapi dengan tindakan yang pengobatan yang tepat penderita dapat dibebaskan dari epilepsi tersebut, namun dokter spesialis saraf yang bertindak sebagai pakar tidak selamanya konsisten dalam melakukan diagnosa, sehingga berdampak penanganan yang kurang tepat. Metode yang digunakan untuk membangun sebuah sistem berbasis pengetahuan yang pengetahuannya bersumber dari catatan kasus-kasus lampau dikenal dengan *Case Base* Reasoning (CBR). Pada penelitian ini penulis menerapkan metode *CBR* untuk membangun sebuah sistem yang memiliki kemampuan mendiagnosa serangan penyakit epilepsi berbasis web. Diharapkan dengan aplikasi ini dapat menyelesaikan masalah gejala yang dialami sekalipun tanpa bantuan dokter dan juga mendapatkan solusi dalam pengambilan tindakan akibat dari serangan penyakit epilepsi ini. Bagi dokter spesialis penyakit saraf, diharapkan sistem ini dapat digunakan sebagai asisten yang berpengalaman dan meringankan beban pekerjaan. Penulis menggunakan metodologi Rational Unified Process (RUP) dan UML (Unified Modelling Language) sebagai bahasa pemodelan selama periode pengembangan dan juga interative incremental yang mana sebagai model siklus pengembangan perangkat lunak.

2. TEORI PENUNJANG

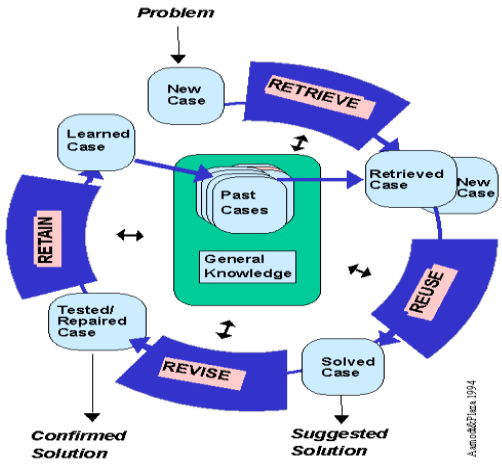
2.1 *Case-Based Reasoning*

Case-Base Reasoning (CBR) adalah metode untuk menyelesaikan masalah dengan mengingat kejadian-kejadian yang sama (*similar*) yang pernah terjadi di masa lalu kemudian menggunakan pengetahuan/informasi tersebut untuk menyelesaikan masalah yang baru, atau dengan kata lain menyelesaikan masalah dengan mengadaptasi solusi-solusi yang pernah digunakan di masa lalu. Perbedaan lain dari CBR yang tidak kalah penting adalah CBR juga merupakan suatu pendekatan ke arah *incremental* yaitu pembelajaran yang terus menerus. Dalam *Case-Based Reasoning* ada empat tahapan yang meliputi:

a. Retrieve, yaitu mendapatkan kembali kasus yang paling menyerupai (*similar*) dengan kasus yang baru.

b. Reuse, yaitu menggunakan kembali pengetahuan dan informasi kasus lama berdasarkan bobot kemiripan yang paling relevan ke dalam kasus yang baru.
c. Revise, yaitu meninjau kembali solusi yang diusulkan kemudian mengetesnya pada kasus nyata (simulasi) dan jika diperlukan memperbaiki solusi tersubut agar cocok dengan kasus yang baru.
d. Retain, yaitu menyimpan kasus baru yang telah berhasil mendapatkan solusi agar dapat digunakan oleh kasus-kasus selanjutnya yang mirip dengan kasus tersebut.

Empat proses masing-masing melibatkan sejumlah langkah-langkah spesifik, yang akan dijelaskan pada gambar 1 berikut ini [14]



Gambar 1. Siklus Metode *Case-Based Reasoning* (Aamodt & Plaza, 1994)

2.2 Penerapan Algoritma *Nearest Neighbor Retrieval* Pada Metode *Case-Based Reasoning*

Algoritma *Nearest Neighbor Retrieva* adalah sebuah algoritma untuk melakukan klasifikasi pendekatan untuk mencari kasus dengan menghitung kedakatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada.
Rumus untuk menghitung bobot kemiripan (*similarity*) dengan *nearest neighbor retrieval* adalah:

$$Similarity\ (problem,case) = \frac{s_1 * w_1 + s_2 * w_2 \ldots\ldots\ldots\ldots + s_n * w_n}{w_1 + w_2 + \cdots\ldots\ldots\ldots + wn}$$

Keterangan:
S = *similarity* (nilai kemiripan)
W = *weight* (bobot yang diberikan)

2.3 PHP

PHP atau *Hypertext Preprocessor* adaah bahasa *server-side programming* yang *powerfull* untuk membuat halaman web yang dinamis dan interaktif. Sintak PHP mirip dengan bahasa Perl dan C. PHP biasanya sering digunakan bersama web server apache di beragam sistem operasi. Dengan begitu kita dapat menggunakan *database* MySQL untuk menyimpan data dan mengambilnya kembali pada saat diperlukan.

3. PERANCANGAN

3.1 Deskripsi Sistem

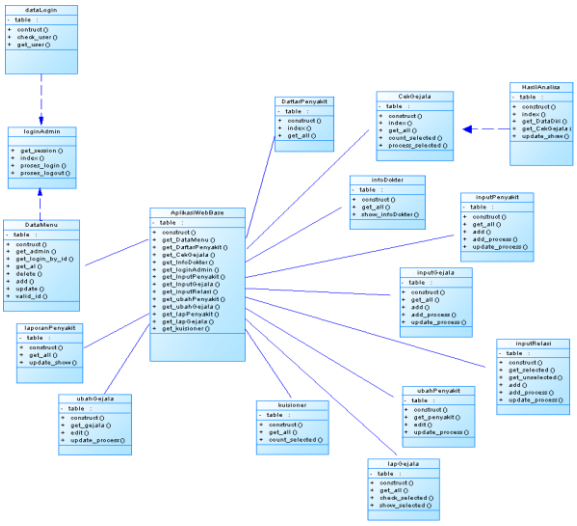
Aplikasi diagnosa penyakit epilepsi berbasis web ini berfungsi untuk memberikan penanganan dan informasi seputar penyakit epilepsi. Informasi yang didapat berdasarkan jenis keluhan berupa gejala yang dirasakan *user* berdasarkan pertanyaan yang diajukan oleh aplikasi ini. Data-data yang terdapat dalam aplikasi ini di dapatkan dari hasil wawancara langsung terhadap dokter spesialis saraf dan buku pintar keperawatan epilepsi yang ditulis oleh dr. Rudi Hantoro, Sp.S(K)

3.2 Perancangan sistem

Pada perancangan aplikasi diagnosa penyakit epilepsi ini menggunakan bahasa pemodelan UML (*Unified Modelling Language*). UML adalah metode pemodelan (tools/model) secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek dan memberikan standar penulisan sebuah sistem untuk pengembangan sebuah software yang dapat menyampaikan beberapa informasi untuk proses implementasi pengembangan software. Ada empat diagram yang digunakan untuk perancangan aplikasi ini, diantaranya : *class* diagram, *use case*, *suquence* dan *activity* diagram.

3.2.1 *Class* Diagram

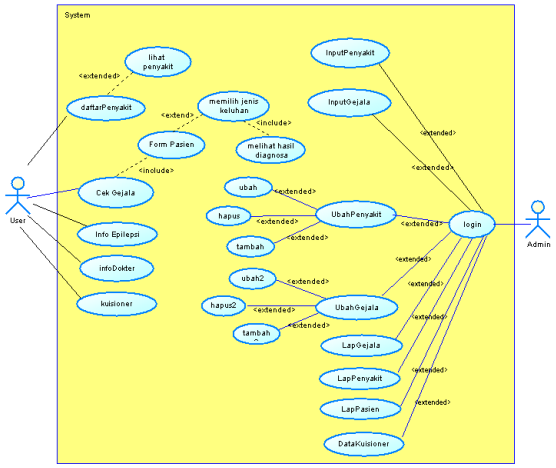
Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Rancangan kelas diagram untuk aplikasi ini terlihat pada gambar 1 berikut ini:



Gambar 1. *Class* Diagram

3.2.2 *Use Case*

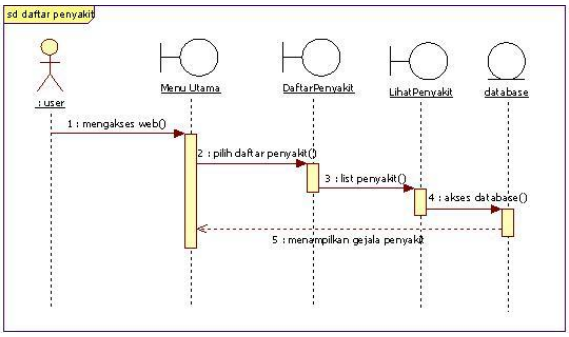
Use case adalah merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* untuk aplikasi ini ditunjukan pada gambar 2



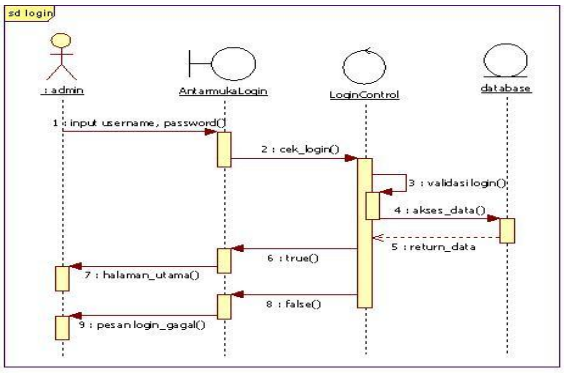
Gambar 2. *Use Case* Diagram

3.2.3 *Sequence* Diagram

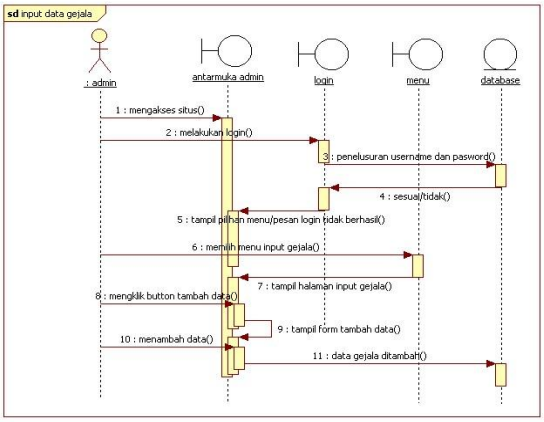
Diagram sekuence menggambarkan perilaku objek pada *use case* dengan mendeskripsikan objek dengan *message* yang dikirimkan dan diterima antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Berikut *Sequence* diagram yang dirancang pada aplikasi diagnosa penyakit epilepsi ini:



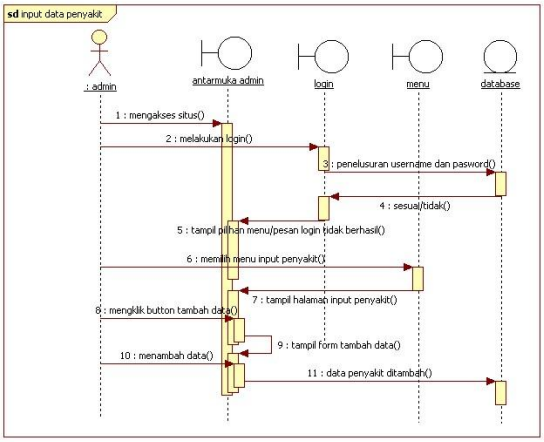
Gambar 3. *Sequence* Diagram Daftar Penyakit



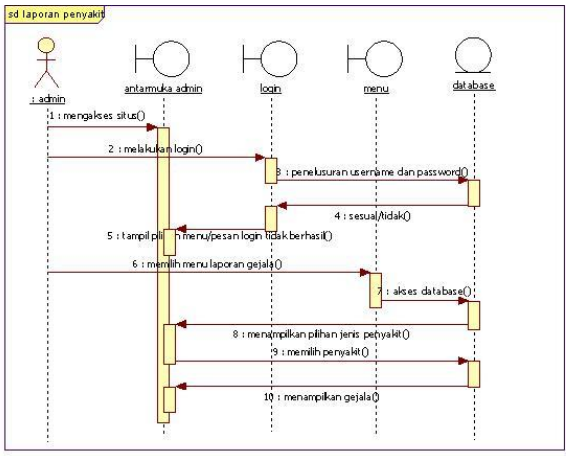
Gambar 4. *Sequence* Diagram Login



Gambar 5. *Sequence* Diagram Input Gejala



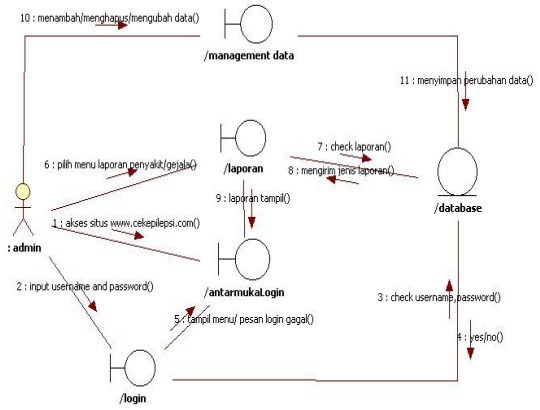
Gambar 6. *Sequence* diagram input penyakit



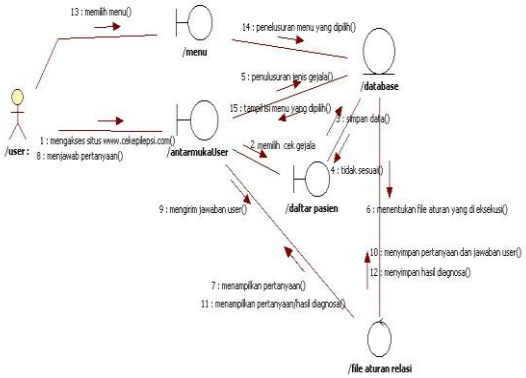
Gambar 7. *Sequence* diagram laporan gejala

3.2.4 *Collaboration* Diagram

Collaboration diagram digunakan untuk memodelkan interaksi antar objek didalam sistem. Diagram ini lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan objek. Ada dua *collaboration* yang dirancang pada aplikasi ini yaitu *collaboration* admin dan *collaboration* user. Berikut adalah *collaboration* yang dirancang:



Gambar 8. *Collaboration* diagram daftar admin



Gambar 9. *Collaboration* diagram user

4. IMPLEMENTASI SISTEM

Pada implementasi aplikasi ini terdapat halaman web yang berbeda, perbedaan ini dibedakan menjadi dua, yaitu halaman web sebagai pengguna/pasien yang ingin melakukan diagnosa penyakit dan halaman web sebagai pakar/admin yang mana memanajemen tampilan web dan manajemen sistem aplikasi ini. Halaman web utama ditunjukan seperti pada gambar 10.



Gambar 10. Halaman web utama

Saat seorang pakar melakukan login maka tampilan antarmuka untuk pakar akan tampil berbeda seperti halaman web penggguna.Seperti yang ditunjukan pada gambar 11.



Gambar 11. Halaman web setelah login

5. HASIL PENGUJIAN

Hasil pengujian aplikasi diagnosa penyakit epilepsi ini dilakukan secara *alpha* dengan metode *black box* dan secara *betha* dengan metode kuisioner.

5.1 Pengujian secara *alpha*

Pengujian Alpha merupakan pengujian dimana fungsional yang digunakan untuk menguji system yang telah dibuat dengan metode pengujian *black box.* Pengujian *black box* terfokus pada persyaratan fungsionalitas aplikasi dengan memasukan data ke dalam *form-form* yang terdapat pada aplikasi ini. Perencanaan penguji *alpha* ditunjukkan pada tabel 1

Tabel 1. Perencanaan pengujian *alpha*

No	Item pengujian	Detail pengujian	Jenis pengujian
1	Login Pakar	*username*	*Blackbox*
		password	*Blackbox*
2.	Cek gejala	Data diri	*Blackbox*
		Jenis gejala	
3.	Input penyakit	Nama penyakit	*Blackbox*
		Nama latin	
		Definisi	
		Solusi	
4.	Ubah gejala	Jenis gejala	*Blackbox*

5.1.1 Hasil pengujian *alpha*

1. Login pakar

Tabel 2. Pengujian Login Pakar

Kasus data hasil pengujian (data benar)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
Username: admin	Text box *username* terisi dengan benar	Isi text box *username* sesuai dengan masukkan.	*black box*	Diterima
Password: qwerty	Text box *password* terisi dengan benar.	Isi text box*password* sesuai dengan *password* yang diinput.	*black box*	Diterima

Kasus data hasil pengujian (data salah)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
Username: kosong *Password*: kosong	Ketika menekan tombol login, menampilkan pesan "*username* tidak ada"	Tampil pesan "Username tidak ada"	*black box*	Diterima
Username: admin *Password*: 123	Ketika menekan tombol login, menampilkan pesan "*username* dan *passwordi* anda tidak sesuai anda belum	Tampil pesan "User dan Password tidak sesuai"	*black box*	Diterima

2. Cek Gejala

Tabel 3. Pengujian cek gejala

Kasus data hasil pengujian (data benar)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
input data diri	Memasukan data diri dengan lengkap	Isi data diri pada form telah dilakukan dengan benar	black box	Diterima
Pilih gejala	Ketika simpan masuk, menampilkan cek gejala	Tampilan pertanyaan seputar gejala yang dialami	black box	Diterima
	Ketika memilih gejala masuk halaman analisa	Tampilan hasil analisa diagnosa penyakit yang dialami beserta solusinya	black box	Diterima
Kasus data hasil pengujian (data salah)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
input data diri	Memasukan data diri dengan tidak lengkap	Menampilkan pesan "data diri masih kosong, silakan ulangi kembali"	Black box	Berhasil

3. Input Penyakit

Tabel 4. Pengujian input penyakit

Kasus data hasil pengujian (data salah)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
Nama Penyakit:kosong Nama Latin:kosong Defini:kosong Solusi:kosong	Ketika klik simpan penyakit, tidak menampilkan pesan "nama penyakit masih kosong"	Tidak menampilkan pesan "nama penyakit masih kosong"	*blackbox*	Diterima
Nama Penyakit:*roman eps* Nama Latin:kosong Defini:kosong Solusi:kosong	Ketika klik simpan penyakit, tidak menampilkan pesan "nama latin masih kosong"	Tidak menampilkan pesan "nama latin masih kosong"	*blackbox*	Diterima
Nama Penyakit:*roman eps* Nama Latin:*Cirtilius* Defini:kosong Solusi:kosong	Ketika klik simpan penyakit, tidak menampilkan pesan "definisi masih kosong"	Tidak menampilkan pesan"definisi masih kosong"	*blackbox*	Diterima
Nama Penyakit:*roman eps* Nama Latin:*Cirtilius* Definisi: penyakit berbahaya Solusi:kosong	Ketika klik simpan penyakit, tidak menampilkan pesan "solusi masih kosong"	Tidak menampilkan pesan "solusi masih kosong"	*blackbox*	Diterima
Kasus data hasil pengujian (data benar)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
Nama Penyakit: Epilepsi Parsial	Text box nama penyakit sesuai dengan yang diinputkan dan berhasil disimpan	Isi text box nama penyakit sesuai dengan yang diinputkan dan berhasil disimpan	*blackbox*	Diterima
Nama Latin: *Simple Partial*	Text box nama latin sesuai dengan yang diinputkan dan berhasil disimpan	Isi text box nama latin sesuai dengan yang diinputkan dan berhasil disimpan	*blackbox*	Diterima
Definisi : Jenis ini tidak disertai ganggua n	Text box definisi sesuai dengan yang diinputkan dan berhasil disimpan	Isi text box definisi sesuai dengan yang diinputkan dan berhasil disimpan	*blackbox*	Diterima
Solusi: obat antikonvulstan	Text box solusi sesuai dengan yang diinputkan dan berhasil disimpan	Isi text box solusi sesuai dengan yang diinputkan dan berhasil disimpan	*blackbox*	Diterima

4. Ubah Gejala

Tabel 5. Pengujian ubah gejala

Kasus data hasil pengujian (data salah)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
Ubah Gejala	Saat akan merubah gejala , tidak masuk ke form ubah gejala	Gejala tidak berubah dan gagal masuk form ubah data	*black box*	Diterima
Hapus Gejala	Saat memilih hapus gejala, gejala yang sudah dihapus masih tersimpan	Nama gejala yang ingi dihapus masih tampil di database	*black box*	Diterima
Kasus data hasil pengujian (data benar)				
Data Inputan	Diharapkan	Pengamatan	Jenis Uji	Keterangan
Ubah Gejala	Saat akan merubah gejala , maka tampilan form ubah gejala akan muncul	Tampilan form ubah gejala muncul dan data gejala berhasil diubah	*blackbox*	Diterima
Hapus Gejala	Saat memilih hapus gejala, gejala berhasil dihapus	Nama gejala yang ingi dihapus berhasil dihapus dari database	*blackbox*	Diterima

5.1.2 Analisa pengujian *alpha*

Berdasarkan hasil pengujian aplikasi yang dilakukan maka dapat disimpulkan bahwa aplikasi diagnosa penyakit epilepsi ini sudah cukup baik. Hal ini dapat dilihat dari seluruh form yang diujikan pada pengujian *alpha* seluruhnya dapat berjalan dengan baik.

5.2 Analisa pengujian *betha*

Pengujian beta dilakukan dengan cara melakukan pengujian melalui kuisioner yang dibagikan ke beberapa responden sebagai target pengguna aplikasi (*user*) yang sedang dibangun. Pengujian dilakukan agar mengetahui sejauh mana aplikasi diagnosa penyakit epilepsi ini dapat membantu menyelesaikan permasalahan yang ada pada user, yaitu masyarakat umum yang telah menggunakan aplikasi ini. Untuk melakukan pengujian ini penulis melakukan pengujian kepada 30 *user*. Kuesioner terdiri dari 5 pertannyaan dengan skala 1 sampai 5 (skala *likert*).

Untuk mencari presentase dari masing-masing jawaban kuesioner digunakan rumus skala *likert* sebagai berikut:

$$P = \frac{S}{Skorideal} \times 100 \text{ .....(4.1)}$$

Keterangan:

P = Nilai presentase yang dicari

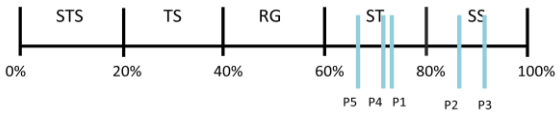
S = Jumlah frekuensi dikalikan dengan nilai skala jawaban

Skorideal = Skala tertinggi jawaban dikalikan dengan jumlah *sample*

5.2.1 Hasil pengujian *betha*

Jawaban atas user atas kuisioner yang diajukan adalah sebagai berikut

Kuisioner Aplikasi Web Diagnosa Penyakit Epilepsi		
No	**Pertanyaan**	**Hasil Pengujian**
1	Apakah anda setuju bahwa tampilan website ini sudah *user friendly*?	$P = \frac{116}{150} \times 100 = 7$
2.	Apakah dengan adanya website ini dapat memberikan informasi tentang epilepsi?	$P = \frac{137}{150} \times 100 = 8$
3.	Apakah aplikasi website diagnosa penyakit epilepsi ini mudah digunakan?	$P = \frac{132}{150} \times 100 = 88\%$
4.	Apakah aplikasi website epilepsi ini sudah dapat membantu dalam mendiagnosa penyakit?	$P = \frac{115}{150} \times 100 = 76.6\%$
5.	Apakah aplikasi website epilepsi ini sudah sesuai dengan manfaat yang diinginkan?	$P = \frac{112}{150} \times 100 = 7$



Gambar 12. Interprestasi Skor Pertanyaan 1-5

Dari pertanyaan 1 sampai 5 di dapatlah interprestasi skor dari masing-masing pertanyaan seperti yang terlihat pada gambar 12.

5.2.1 Analisa hasil pengujian *betha*

Berdasarkan hasil pengujian secara *beta* dapat dianalisa bahwa aplikasi diagnosa penyakit epilepsi berbasis web ini cukup baik meskipun persentase jawaban dari *user* dari masing-masing pertanyaan tidak terlalu signifikan. Namun dari hasil kuesioner yang didapat tersebut juga terdapat pengguna yang merasakan adanya kekurang dari aplikasi ini. Seperti tampilan dan menu-menu yang dirasa perlu untuk ditambahkan.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang didapat berdasarkan pengujian uang telah dilakukan,yaitu:

1. Sistem mampu memberikan solusi dengan metode *Case Based Reasoning* yaitu setiap gejala penyakit yang dimasukkan oleh *user* akan dicocokkan dengan gejala penyakit terdahulu kemudian sistem akan menampilkan penanganan terdahulu yang akan direvisi sesuai gejala penyakit yang baru.Aplikasi diagnosa penyakit epilepsi berbasis web ini sudah berjalan dengan baik. Hal ini dapat terlihat dari hasil pengujian secara *Alpha* dan *beta* yang sesuai dengan yang diharapkan oleh penulis.
2. Penelitian ini berhasil merancang dan membuat aplikasi yang mampu mendiagnosa terhadap gejala serangan epilepsi serta memberikan solusi,

informasi dan fungsional sistem sudah dapat berjalan dan berfungsi dengan baik. Hal ini dapat ditunjukan salah satunya dari beberapa pengguna yang menyatakan 76.6% setuju aplikasi sistem ini dapat membantu mendiagnosa penyakit epilepsi.

6.2 Saran

Saran yang diajukan penulis agar dapat membantu proses pengembangan aplikasi ini antara lain:

1. Dikarenakan ilmu pengetahuan terus berkembang dan ditemukannya hal-hal baru maka basis pengetahuan pada sistem aplikasi ini perlu di *update* atau ditambah, sehingga data-data yang ada menjadi lebih lengkap dan kompleks..
2. Pada pengembangan aplikasi selanjutnya, aplikasi diagnosa penyakit epilepsi ini tidak hanya berbasis web tapi juga dirancang dengan aplikasi berbasis android ataupun mobile OS lainnya

6. DAFTAR PUSTAKA

[1]Rudi Hantoro.2013. Buku Pintar Keperawatan Epilepsi. Cakrawala Ilmu:Yogyakarta.

[2]Kusrini S. Kom. 2006. Sistem PakarTeori dan Aplikasi. Andi offset:Yogyakarta.

[3]Bunafit, Nugroho, (2006), Membuat Aplikasi Sistem Pakar dengan PHPdan My SQL dengan PHP dan MySQL dengan Editor Dreamweaver,Ardana Media, Yogyakarta.

[4]Nugroho, Bunafit. 2008. *Latihan Membuat Aplikasi Web PHP dan MySQL dengan Dreamweaver MX (6, 7, 2004) dan 8*. Yogyakarta: Gava Media.

[4] Sri Kusumadewi.2003. Artificial Intelegence (Teknik dan aplikasi.) Graha Ilmu:Yogyakarta

[5] Betha Sidik.2012. Pemrograman Web PHP.Informatika:Bandung

[6] Sunyoto, A. 2007. *AJAX Membangun Web Dengan Teknologi Asyncronouse Javascript dan XML*. Yogyakarta: ANDI.

[6] Aamodt, A & Plaza, E, 1994, Case Based Reasoning: Foundation Issues Methodological Variations, and System Approaches, AI Communication Vol 7 Nr, 1 March 1994, pp 39-

[7]Octaviani, Fransica S. 2013. *Implementasi Case Base Reasoning Untuk Sistem Diagnosis Penyakit Anjing.* Jurnal Informatika Vol.7 No2, November 2013