
PERANCANGAN SISTEM PAKAR UNTUK DIAGNOSAPENYAKIT PADA BALITA MENGUNAKAN METODE *DEMPSTER-SHAFER*

Adil Wahyudi¹, Rusdi Efendi², Yudi Setiawan³

^{1,2,3}Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(tel: 0736-341022; fax: 0736-341022)

¹adil.choedil@gmail.com

²rusdi.efendi@unib.ac.id

³ysetiawan@unib.ac.id

Abstrak: Tujuan dari penelitian ini adalah untuk membangun sebuah sistem berbasis pengetahuan kedokteran dalam mendiagnosa penyakit pada balita yang ditampilkan dalam bentuk *website*. Metode sistem pakar yang digunakan adalah metode *Dempster Shafer*. Metode *Dempster-Shafer* adalah salah satu metode matematika untuk mencari nilai kepastian dari potongan-potongan gejala yang dikombinasikan menjadi satu sehingga didapat nilai kepastian dari setiap masalah yang ada. *Output* yang dihasilkan berupa nama penyakit, nilai densitas dan solusi berupa pengobatan dan pencegahan dari hasil diagnosa penyakit. Pada penelitian ini pengujian yang digunakan yaitu pengujian akurasi sistem pakar dengan data dari pakar dengan nilai akurasi sebesar 100% dari 30 data uji. Hasil pengujian menunjukkan bahwa sistem pakar ini mampu mendiagnosa penyakit balita disertai dengan nilai keyakinan *Dempster-Shafer*.

Kata Kunci: Sistem Pakar, *Dempster-Shafer*, Penyakit Balita, PHP

Abstract: The purpose of this thesis is to build a system that based on medical knowledge that can diagnose babies disease which is dipla. The method used in expert system is Dempster-Shafer method. Dempster-Shafer is one of math method to find value of certainty of symptoms pieces combined to one till get value of certainty for each problem. Output generated is name of disease, value of certainty and solution form treatment and prevention of disease. In this research, testing used accuracy testing between expert system and data from expert . Test result show this expert system able to diagnose babies disease whitin value of certainty of Dempster Shafer

Keywords: expert system, Dempster-Shafer, babies disease, PHP

I. PENDAHULUAN

Balita adalah istilah umum bagi anak usia 1-3 tahun (batita) dan anak prasekolah (3-5 tahun). Saat usia batita, anak masih tergantung penuh kepada orang tua untuk melakukan kegiatan penting, seperti mandi, buang air dan makan. Perkembangan berbicara dan berjalan sudah bertambah baik. Namun kemampuan lain masih terbatas [1].

Masa balita merupakan masa yang memerlukan perhatian khusus, karena pada masa ini termasuk masa yang rawan terhadap penyakit. Terbatasnya informasi mengenai penyakit yang menyerang balita membuat para orang tua kesulitan untuk memprediksi penyakit yang diderita oleh anak mereka. Untuk orang tua yang

baru memiliki anak, hal ini merupakan hal yang baru bagi mereka. Para orang tua biasanya lebih memilih untuk mempercayakan hal diagnosis penyakit kepada dokter yang ahli tentang kesehatan. Untuk beberapa orang, biaya berkonsultasi kepada dokter ahli tidaklah murah sehingga banyak orang tua memutuskan untuk merawat sendiri anak mereka. Akibat keterlambatan penanganan terhadap penyakit membuat penyakit tersebut menjadi semakin parah dan semakin susah disembuhkan. Oleh sebab itu orang tua membutuhkan bantuan suatu sistem yang memiliki kemampuan diagnosis penyakit menular seperti halnya seorang dokter.

Pada penelitian ini metode yang digunakan adalah metode *Dempster-shafer*, dimana metode ini menggunakan nilai kepastian untuk mendiagnosa sesuatu. Metode ini akan melakukan perhitungan terhadap nilai kepastian dari setiap gejala-gejala penyakit sehingga mendapatkan hasil *output* berdasarkan nilai kepastian tertinggi dari setiap penyakit. Metode ini merupakan cara paling tepat dalam mendeteksi suatu penyakit, sehingga *user* tidak perlu lagi menebak penyakit yang diderita oleh anak mereka karena sistem akan memberikan jawaban berdasarkan gejala yang ada. Dengan adanya sistem ini diharapkan dapat memberikan informasi diagnosis dan cara penanganan secara tepat dan informatif kepada orang tua tentang penyakit yang diderita oleh balita.

II. LANDASAN TEORI

A. Balita

Balita adalah istilah umum bagi anak usia 1-3 tahun (batita) dan anak prasekolah (3-5 tahun). Saat usia batita, anak masih tergantung penuh

kepada orang tua untuk melakukan kegiatan penting, seperti mandi, buang air dan makan. Perkembangan berbicara dan berjalan sudah bertambah baik. Namun kemampuan lain masih terbatas [1].

B. Penyakit Balita

Masa Balita merupakan masa yang memerlukan perhatian khusus, karena pada masa ini termasuk masa yang rawan terhadap penyakit. Penyakit itu sendiri merupakan jejalan mekanisme adaptasi suatu organisme untuk bereaksi secara tepat terhadap setiap tekanan ataupun ransangan yang menimbulkan gangguan pada fungsi ataupun struktur organ dan sistem di dalam tubuh [2].

C. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada HTML (*Hyper Text Ha*). PHP dibuat pertama kali oleh satu orang yaitu Rasmus Lerdorf, yang pada awalnya dibuat untuk menghitung jumlah pengunjung pada halaman utamanya. Diawal Januari 2001, PHP telah dipakai lebih dari 5 juta *domain* diseluruh dunia dan akan terus bertambah karena kemudahan aplikasi PHP ini dibandingkan dengan bahasa *server side* yang lain.

Saat ini PHP sangat populer dan menggantikan Perl yang sebelumnya juga populer sebagai bahasa *scripting web*. PHP telah menjadi modul Apache terpopuler melebihi *FrontPage* dan *mod_perl* [3].

D. Codeigniter

Codeigniter adalah salah satu dari sekian banyak kerangka kerja siap pakai (*framework*) berbasis PHP, jadi dengan menggunakan Codeigniter hanya tinggal memikirkan proses bisnisnya saja tanpa perlu memikirkan yang lain, karena fitur-fitur standar sudah tersedia di

Codeigniter seperti *pagging*, *validation*, *routing* dan masih banyak lagi, hanya cukup dengan memanggilnya saja.

Codeigniter menerapkan lingkungan pengembangan dengan metode MVC (*Model - View - Controller*). MVC memisahkan antara logika proses bisnis, dengan tampilan atau *template*. Dengan MVC, kode program menjadi terstruktur, terorganisir dan akan mudah dikembangkan kembali oleh selain pengembang awalnya

E. Unified Modelling Language (UML)

Secara umum *Unified Modeling Language* (UML) merupakan “bahasa” untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. Dalam kerangka visualisasi, para pengembang menggunakan UML sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem/perangkat lunak. Dengan adanya “bahasa” yang bersifat standar, komunikasi perancang dengan pemrogram (lebih tepat lagi komunikasi antar anggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus [4].

III. METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang akan dilakukan adalah penelitian terapan (*applied research*). Penelitian ini bertujuan merancang dan membangun aplikasi yang dapat menganalisa penyakit pada balita algoritma *Dempster-Shafer* berbasis PHP

IV. ANALISIS DAN PERANCANGAN

A. Analisis Sistem

Desain sistem yang akan dibuat memerlukan beberapa kebutuhan data masukan, kebutuhan data keluaran dan kebutuhan antar muka. Tujuan

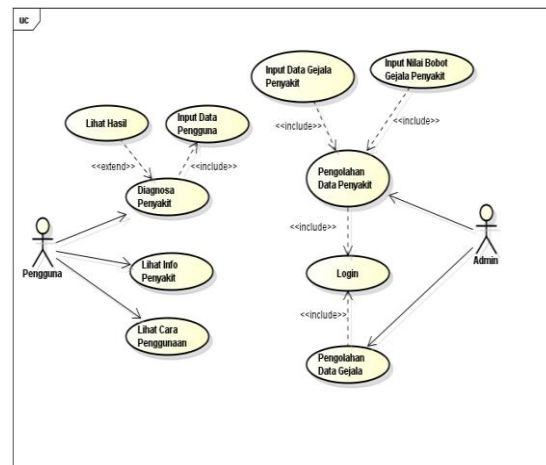
analisis kebutuhan adalah untuk menentukan spesifikasi fungsi, kemampuan serta fasilitas dari program. Analisis kebutuhan juga bermanfaat sebagai dasar evaluasi setelah program selesai disusun.

B. Perancangan Sistem

Perancangan sistem yang digunakan dalam pembuatan sistem ini adalah Perancangan UML

1. Use case Diagram

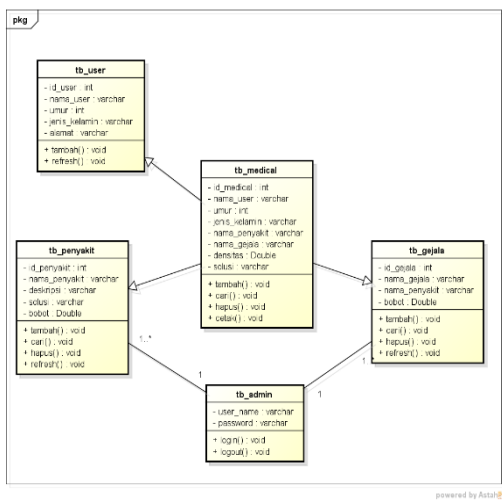
Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Interaksi aktor dengan sistem dapat dilihat pada Gambar 1.



Gambar 1. Use Case Diagram

2. Class Diagram

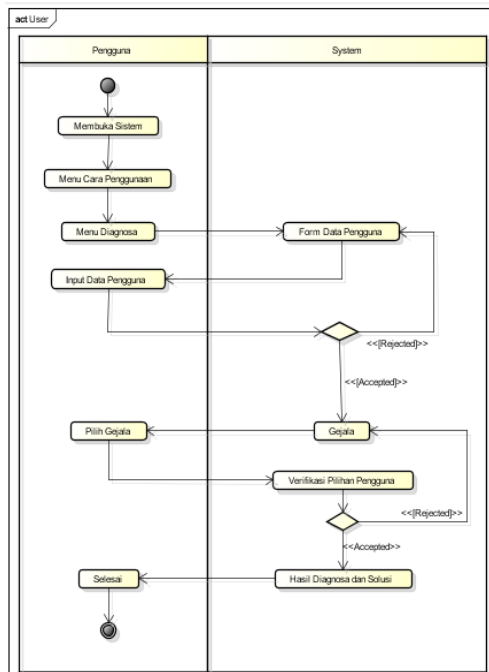
Class Diagram adalah jenis diagram struktur statis yang menjelaskan struktur dari sebuah sistem dengan menunjukkan kelas-kelas, interface, atribut-atribut dan hubungan antar kelas dari struktur sistem tersebut. *Class* diagram akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Adapun *class* diagram dapat dilihat pada Gambar 2.



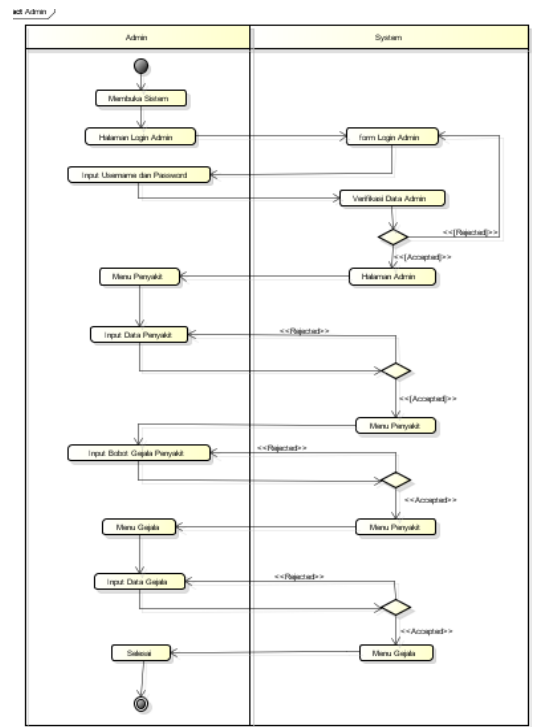
Gambar 2. Tampilan Class Diagram

3. Activity diagram

Activity Diagram menggambarkan aliran aktivitas dalam sistem yang dibangun, bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi dan bagaimana mereka berakhir. Pada sistem ini terdapat dua activity diagram, yaitu administrator dan pengunjung. Untuk activity diagram user terdapat pada Gambar 3, untuk admin pada Gambar 4.



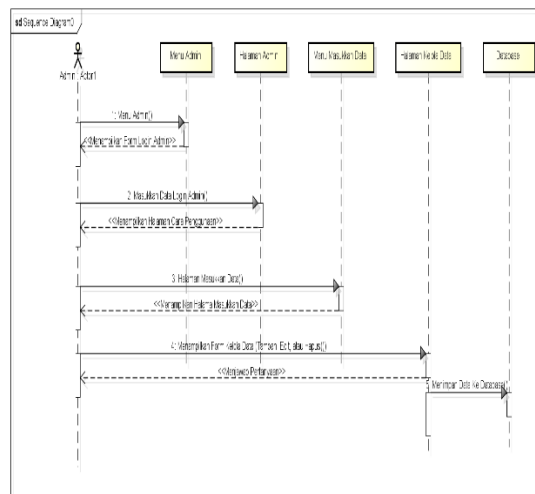
Gambar 3. Tampilan Activity Diagram User



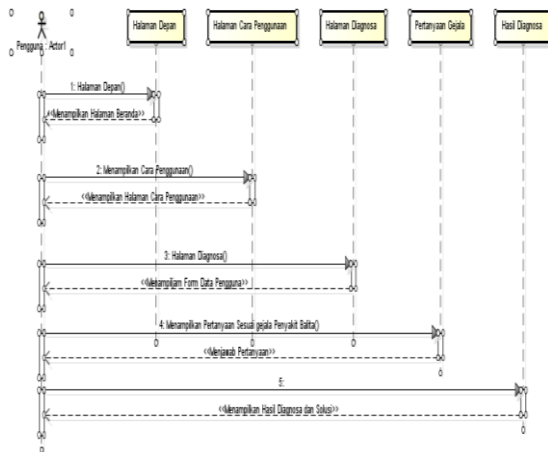
Gambar 4. Tampilan Activity Diagram Admin

4. Sequence Diagram

Sequence diagram merupakan diagram yang digunakan untuk menggambarkan urutan-urutan kejadian atau proses yang terjadi pada sistem. Pada sistem ini terdapat dua sequence diagram administrator dan user. Untuk sequence diagram administrator terdapat pada Gambar 5, sequence diagram user terdapat pada Gambar 6.



Gambar 5. Sequence Diagram Administrator



Gambar 6. Sequence Diagram User

V. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan mengenai hasil dan pembahasan dari sistem yang telah dibangun, yaitu sistem pakar diagnosa penyakit balita dengan menggunakan metode *Dempster-Shafer*. Penjelasan pada bab ini antara lain terdiri dari implementasi antar muka, pengujian *black box* serta uji kelayakan system.

A. Implemenasi Antar Muka

Hasil dari analisis dan perancangan sistem yang telah dilakukan sebelumnya akan mempengaruhi hasil dari implementasi sistem. Berikut tampilan hasil implementasi sistem :

1. Menu Beranda

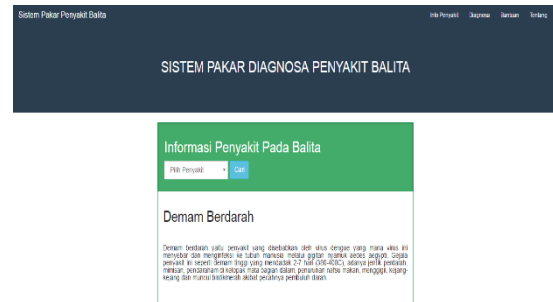
pada sistem ini adalah halaman yang pertama kali dijalankan. Menu ini dibuat berdasarkan perancangan antarmuka. Berikut adalah tampilan dari menu beranda yang ditunjukkan pada Gambar 7.



Gambar 7. Tampilan Halaman Utama

2. Menu Info Penyakit

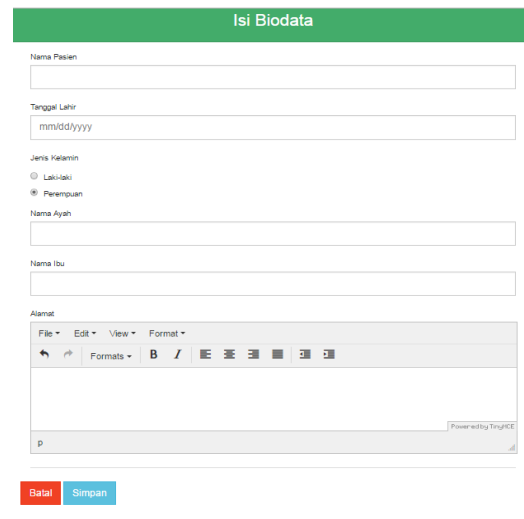
Menu info penyakit pada sistem ini dibuat berdasarkan perancangan antarmuka. Berikut adalah tampilan dari menu info penyakit yang ditunjukkan pada Gambar 8.



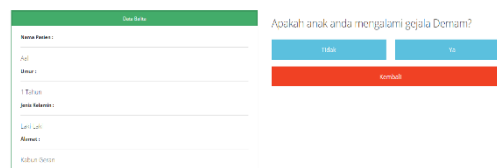
Gambar 8. Menu Info Penyakit

3. Menu Diagnosa

Menu Biodata pasien pada sistem ini dibuat berdasarkan perancangan antarmuka. Menu diagnosa dibagi menjadi 2 tahap, tahap pengisian biodata dan tahap diagnosa. Berikut adalah tampilan dari menu pengisian biodata yang ditunjukkan pada Gambar 9 dan menu diagnosa pada Gambar 10.

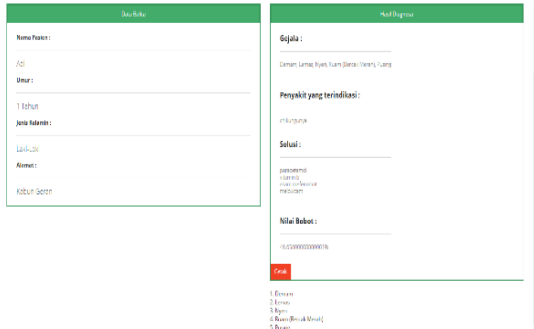


Gambar 9. Menu Pengisian Biodata



Gambar 10. Menu Diagnosa

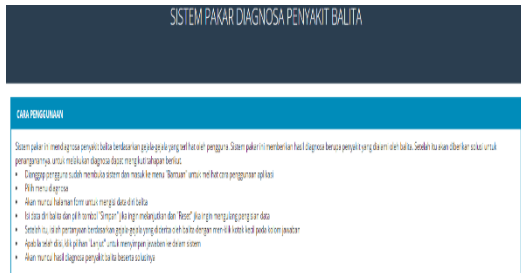
Setelah menjawab pertanyaan pada menu diagnose, maka akan tampil hasil diagnosa seperti pada gambar 11.



Gambar 11. Hasil Diagnosa

4. Menu Bantuan

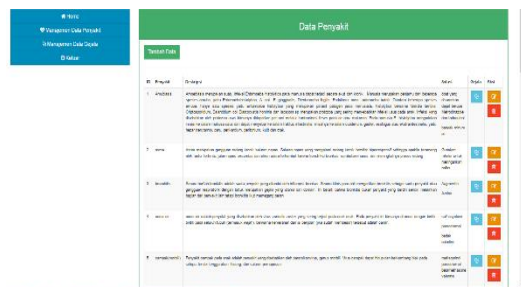
Menu bantuan pada sistem ini dibuat berdasarkan perancangan antarmuka. Berikut adalah tampilan dari menu bantuan yang ditunjukkan pada Gambar 12.



Gambar 12. Menu Bantuan

5. Menu Manajemen Data Penyakit

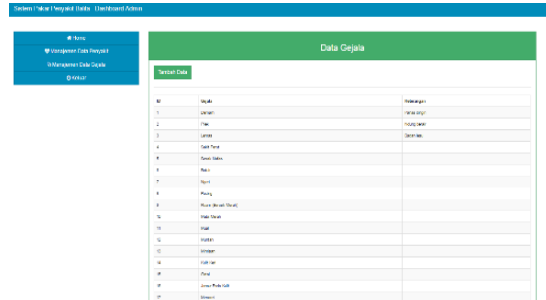
Menu Admin pada sistem ini dibuat berdasarkan perancangan antarmuka. Berikut adalah tampilan dari menu manajemen data penyakit yang ditunjukkan pada Gambar 13.



Gambar 13. Menu Manajemen Data Penyakit

6. Menu Manajemen Data Gejala

Menu Admin pada sistem ini dibuat berdasarkan perancangan antarmuka. Berikut adalah tampilan dari menu manajemen data gejala yang ditunjukkan pada Gambar 14.



Gambar 14. Menu Manajemen Data Gejala

B. Pengujian Black Box

Berikut adalah kasus untuk menguji perangkat lunak yang telah dibangun menggunakan metode *black box*.

Tabel 1 Tabel *Blackbox Testing*

Kelas Uji	Skenario Pengujian	Hasil yang diharapkan	Hasil dan Kesimpulan
Home	Klik menu sistem pakar penyakit balita	Menampilkan halaman <i>home</i>	[√]Berhasil []Tidak Berhasil
	Memilih halaman info penyakit	Menampilkan halaman info penyakit	[√]Berhasil []Tidak Berhasil
	Memilih halaman diagnosa	Menampilkan halaman diagnosa	[√]Berhasil []Tidak Berhasil
	Memilih halaman bantuan	Menampilkan halaman bantuan	[√]Berhasil []Tidak Berhasil
	Memilih halaman tentang	Menampilkan halaman tentang	[√]Berhasil []Tidak Berhasil
	Klik tombol mulai diagnosa	Menampilkan halaman diagnosa	[√]Berhasil []Tidak Berhasil
Info Penyakit	Klik <i>combo box</i> pada pilih penyakit	Muncul daftar penyakit	[√]Berhasil []Tidak Berhasil
	Klik tombol "cari"	Menampilkan informasi penyakit yang dipilih	[√]Berhasil []Tidak Berhasil

Biodata	Klik tombol simpan	Menampilkan halaman diagnosa	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
Diagnosa	Klik tombol Ya atau Tidak	Melanjutkan proses diagnosa	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol kembali	Kembali ke pertanyaan sebelumnya	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol proses	Menampilkan halaman hasil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
Login	Klik tombol dengan mengisi <i>username</i> dan <i>password</i> yang benar	Masuk ke halaman <i>admin</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
Penyakit	Klik tombol tambah	Menampilkan <i>form</i> tambah data penyakit	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol simpan dengan mengisi penuh <i>form</i>	Menampilkan peringatan berhasil menyimpan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol gejala	Menampilkan data gejala	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik Tombol Tambah	Menambah data gejala untuk penyakit	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol hapus	Menampilkan peringatan data berhasil di hapus	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol ubah	Menampilkan popup ubah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
Gejala	Klik tombol tambah	Menampilkan <i>form</i> tambah gejala	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Klik tombol simpan dengan mengisi penuh <i>form</i>	Menampilkan peringatan berhasil menyimpan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

C. Uji Kelayakan Sistem

Pengujian kecocokan sistem dengan data pakar sebanyak 15 dari 30 data pasien. 15 data pengujian dianggap sesuai dengan data dari pakar. Maka dapat disimpulkan bahwa persentase kecocokan data sistem dengan data dari pakar sebesar

$$\text{persentase} = \frac{15}{15} \times 100\% = 100\%.$$

VI. PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian, pengujian, implementasi serta pembahasan mengenai sistem pakar diagnosa penyakit balita ini, maka penulis mendapatkan kesimpulan sebagai berikut :

- 1) Sistem ini sudah dapat digunakan oleh pengguna untuk mendiagnosa penyakit balita.
- 2) Jenis penyakit yang dapat di diagnosa oleh sistem ini sebanyak 25 penyakit menular yang biasa diderita oleh balita.
- 3) Dari hasil pengujian *Black Box*, Sistem ini dapat mendiagnosa penyakit dan memberikan solusi untuk pengobatan dan pencegahan dengan tingkat pengujian fungsionalitas sebesar 100%.
- 4) Hasil diagnosa dari sistem ini didapat dari proses perhitungan *Dempster Shafer*. Dimana metode *Dempster Shafer* memberikan nilai kepastian pada penyakit balita sebesar 100% dari kecocokan *output* sistem jika dibandingkan dengan data dari pakar

B. Saran

Saran dari penulis untuk sistem pakar diagnosa penyakit balita ini adalah :

- 1) Untuk mendapatkan nilai kepastian yang lebih akurat lagi, bisa dilakukan dengan menerapkan beberapa metode penanganan ketidakpastian lainnya. Seperti metode CF (*Certainty Factor*), teorema Bayes. Atau juga bisa dengan membandingkan metode *Dempster Shafer* dengan metode lain. Sehingga nantinya sistem dapat dikembangkan lebih baik lagi.
- 2) Sistem yang dikembangkan disarankan dapat dirancang di *platform* lainnya, seperti

- mobile* sehingga penerapannya akan menjadi lebih luas dan bisa digunakan setiap orang
- 3) Data penyakit yang di diagnosa dapat dikembangkan dengan menambahkan data penyakit menular balita lainnya.

REFERENSI

- [1] B. Sutomo, Menu Sehat Alami Untuk Balita dan Batita, Jakarta Selatan: Demedia, 2010.
- [2] E. Budiarto, Pengantar Epidemiologi penyakit menular, Jakarta: kedokteran egc, 2003.
- [3] B. Raharjo, I. Heryanto dan E. RK, Modul Pemrograman WEB, Bandung: Modula, 2010.
- [4] A. Nugroho, Rekayasa Perangkat Lunak dengan UML dan Java, Yogyakarta: Andi, 2005.