

SISTEM PAKAR DIAGNOSA PENYAKIT REFRAKSI MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS WEB

Edha Jenny Cahyani¹⁾, Sandy Kosasi²⁾

1. Teknik Informatika, STMIK Pontianak
email: jennytasya686@gmail.com
2. Teknik Informatika, STMIK Pontianak
email: sandykosasi@gmail.com

Abstract

Refractive diseases often occur in the community. Many people are not aware of the symptoms, some are aware that they are reluctant to consult directly with a doctor for reason that they are afraid of spending a large amount of money just to diagnose refractive disease. Based on some these problems, system that can adopt ability of an expert is an expert system. With this expert system, it can be a solution for community in diagnosing refractive diseases. Expert system diagnose refractive disease was built using the forward chaining method. Software designed using prototype method by using Design Science Research method. Software testing uses black-box testing method, program code units with boundary value testing techniques. Results of this software design in the form of software that's named "Expert system for diagnosing refractive diseases". This software is designed using web-based programming language, HTML, CSS, PHP with MySQL database. Results displayed by system in the form of diagnostic results containing the disease diagnosed. These results are based on symptoms that have been filled in by user. This expert system software can provide references and assist users in providing information about their illness, initial treatment solutions and prevention of refractive diseases.

Kata Kunci : *Refractive, Forward Chaining, Expert System.*

A. PENDAHULUAN

Kelainan refraksi merupakan kelainan pembiasan sinar pada mata sehingga pembiasan sinar tidak difokuskan pada retina. Pada kelainan refraksi terjadi ketidak seimbangan sistem optik pada mata sehingga menghasilkan bayangan yang kabur. Kelainan refraksi merupakan salah satu penyebab terbanyak gangguan penglihatan di seluruh dunia dan menjadi penyebab kebutaan kedua yang dapat diatasi setelah katarak [1]. Kelainan refraksi atau gangguan rabun mata ini terjadi ketika mata tidak dapat melihat

atau fokus dengan jelas pada suatu area terbuka sehingga pandangan menjadi kabur dan untuk kasus yang parah, gangguan ini dapat menjadikan *visual impairment* atau melemahkan penglihatan. Kelainan refraksi yang umum terjadi antara lain *myopia*, *hypermetropia*, *presbyopia*, dan *astigmatisme* [2]. Sebagian besar orang yang menderita kelainan refraksi masih kesulitan mengetahui ukuran normal penglihatan tanpa dokter atau ahli mata. Bahkan Sebagian masih kesulitan untuk memeriksa gangguan rabun mata yang terus bertambah parah. Oleh sebab itu

diperlukan sebuah sistem pakar berbasis komputer yang dapat membantu penderita kelainan refraksi.

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut [3]. Sampai saat ini sudah ada beberapa hasil perkembangan sistem pakar dalam berbagai bidang sesuai dengan kepakaran seseorang misalnya bidang Pendidikan, kedokteran maupun bidang yang menyangkut pendeteksi kerusakan pada alat khususnya jaringan. Pemanfaatan teknologi memudahkan manusia untuk mengakses informasi tanpa terbatas ruang dan waktu [4].

Pemanfaatan sistem pakar yaitu sebagai alat bantu yang secara otomatis dan cepat dapat melakukan diagnosa dan memberikan informasi mengenai cara penanganannya. Sistem pakar memiliki fungsi untuk menirukan pengetahuan dan kemampuan dari seorang pakar [5]. Tujuan pengembangan sistem pakar tidak dimaksudkan untuk menggantikan peran para pakar, namun untuk mengimplementasikan pengetahuan para pakar kedalam bentuk aplikasi sehingga dapat digunakan oleh banyak orang [6]. Terdapat penelitian terdahulu yang dilakukan oleh peneliti lain dengan judul "Sistem Pakar Diagnosa Penyakit Gangguan Jiwa Dengan Demster Shafer". Pada penelitian ini menuliskan bahwa sistem pakar diagnosa dapat membantu pengguna untuk mengetahui jenis penyakit yang diderita. Program sistem pakar diagnosa penyakit dapat menghemat biaya pengguna dalam mendiagnosa penyakit. Dengan adanya

sistem pakar diharapkan dapat menghemat waktu dalam menyelesaikan masalah, penyerdehanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang yang penyampaian informasinya berdasarkan request dari pengguna [7].

Metode *Forward Chaining* merupakan salah satu dari metode yang ada dalam sistem pakar [8]. Metode forward chaining digunakan pada penelitian ini karena metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan/menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut. Metode ini mampu menyediakan banyak sekali informasi dari hanya sejumlah kecil data. Dengan adanya sistem pakar diharapkan dapat menghemat waktu dalam menyelesaikan masalah, penyerdehanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang yang penyampaian informasinya berdasarkan request dari pengguna. Metode *forward chaining* sangat banyak digunakan dalam sistem pakar bidang kesehatan beberapa diantaranya adalah sistem pakar untuk diagnosa penyakit jantung berbasis android [9], sistem pakar untuk diagnosa penyakit kulit [10], sistem pakar diagnosa penyakit ikan komet [11], sistem pakar tes psikometri kepribadian manusia [12]. Sistem pakar untuk mendiagnosa penyakit sakit kepala [13]. Dari beberapa penelitian terdahulu tersebut dapat disimpulkan sistem pakar menggunakan metode *forward chaining* dalam bidang kesehatan sangat efektif penggunaannya, ini dapat menjadi solusi awal dalam hal diagnosa penyakit karena tingkat kesuksesan aplikasi sistem pakar menggunakan metode *forward chaining*

yang cukup tinggi dalam berbagai bidang salah satunya bidang kesehatan.

B. METODE PENELITIAN

Bentuk penelitian yang digunakan oleh penulis pada penelitian ini adalah survei. Penelitian survei ini bertujuan untuk mengumpulkan data yang pokok. Metode penelitian yang digunakan oleh penulis pada penelitian ini adalah metode *Design Science Research* (DSR). Pemilihan metode ini tujuannya untuk menjawab kebutuhan dari masyarakat yang dipadukan dengan pengetahuan sehingga bisa menjadi solusi untuk memenuhi kebutuhan masyarakat. Dengan digunakan metode DSR ini pula diharapkan penelitian ini akan menjadi penelitian yang berkualitas dan bisa bermanfaat untuk seluruh kalangan.

Metode pengumpulan data yang digunakan dalam penelitian ini adalah data primer yang diperoleh secara langsung dari sumber yang berkaitan dengan penyakit refraksi, yang dilakukan melalui wawancara dengan dr. Bobby Kristianto, Sp.M. peneliti juga melakukan kuesioner guna mendukung dalam pengumpulan data yang akan digunakan dalam aplikasi, dan juga melakukan studi dokumentasi yang berkaitan dengan kebutuhan perancangan aplikasi ini.

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *Prototype* yang terdiri dari enam tahapan yaitu Pengumpulan kebutuhan pelanggan, Membangun *prototype*, Evaluasi *prototype*, Pengkodean sistem, Pengujian Sistem, dan Evaluasi sistem pengguna [14]. Metode pengujian perangkat lunak yang digunakan dalam penelitian ini adalah

Black-box testing dengan metode Teknik *Boundary Value Analysis*. Pengujian yang dilakukan menguji perangkat lunak berdasarkan nilai yang mendekati data valid dan tidak valid. Teknik *Boundary Value Analysis* digunakan untuk memilih menentukan jumlah digit maksimal dan minimal yang ingin diuji [15].

C. HASIL DAN PEMBAHASAN

Aplikasi sistem pakar diagnosa penyakit refraksi ini dibangun dengan metode *forward chaining* yang telah memberikan manfaat sangat berarti bagi pengembangan sistem. Perancangan aplikasi berbasis *web* agar memudahkan masyarakat dalam pengaksesan aplikasi sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining* dengan bahasa pemrograman PHP, HTML, CSS dan dengan *database mysql*. Gejala dalam aplikasi ini didapatkan berdasarkan wawancara kepada dokter dan didapatkan sebanyak 20 gejala dengan 4 penyakit. Berdasarkan gejala yang ada maka dibuatlah sebuah implementasi rumus *IF-THEN forward chaining* kedalam aplikasi sistem pakar diagnosa penyakit refraksi dan diolah dengan sedemikian rupa sehingga bisa dapat melakukan pertimbangan berdasarkan dengan gejala yang telah diinputkan oleh *user*. Perancangan aplikasi sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining* berbasis *web* ini mengacu kepada metode perancangan *prototype* yang terdiri atas fase, sebagai berikut:

3.1 Identifikasi Kebutuhan *User*

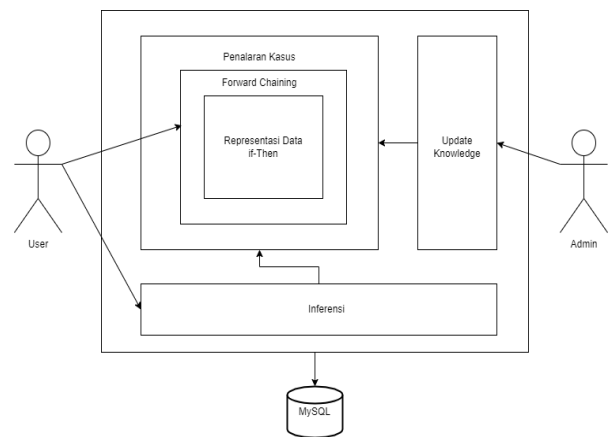
Tahap pertama pada metode *prototype* adalah identifikasi kebutuhan *user*. Ditahap awal ini dibutuhkan analisis terkait kebutuhan *user* mulai dari

kebutuhan fungsional yaitu kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang mampu dikerjakan oleh perangkat lunak, perangkat lunak sistem pakar ini memiliki fungsi form diagnosa penyakit berdasarkan gejala yang dipilih dengan menjawab “Ya” atau “Tidak” dan sistem juga harus dapat memberikan solusi berdasarkan pertanyaan yang diberikan. Kebutuhan antarmuka adalah kebutuhan yang menghubungkan perangkat lunak dengan elemen perangkat keras, perangkat lunak, atau basis data. Kerja Kebutuhan unjuk kerja (*performance requirement*) adalah kebutuhan yang menetapkan karakteristik unjuk kerja yang harus dimiliki oleh perangkat lunak, misalnya kecepatan, ketepatan, dan frekuensi. Sehubungan dengan ini yaitu berupa sistem harus mampu menampilkan basis data secara cepat dalam waktu kurang dari 5 detik. Contohnya ketika menampilkan hasil diagnosa kelainan *refraksi* beserta dengan solusi penanganannya, penyebab penyakit kelainan refraksi dan berbagai macam informasi lainnya mengenai penyakit *refraksi* yang dialami oleh *user*.

3.2 Pembuatan *Prototype*

Tahap kedua ini adalah tahap dimana semua hal yang berkaitan dengan kebutuhan perancangan perangkat lunak dikumpulkan dan dipertimbangkan sebelum melakukan perancangan perangkat lunak agar tidak salah dalam pengimplementasian pengetahuan kedalam sistem nantinya. Ada enam tahapan pembuatan *prototype* ini, yaitu Pembangunan arsitektur perangkat lunak sistem pakar diagnosa penyakit refraksi, Perancangan basis pengetahuan, Perancangan basis data, Perancangan *prototype* berupa algoritma yang diimplementasikan kedalam gambar *flowchart*, Perancangan UML (*Unified Modelling Language*), dan Pembuatan hasil perancangan aplikasi. Pembangunan *prototype* pertama adalah dengan

membuat arsitektur perangkat lunak seperti pada Gambar 1.



Gambar 1. Arsitektur Perancangan Perangkat Lunak

Arsitektur perancangan perangkat lunak merupakan struktur atau gambaran fitur-fitur dari perangkat lunak yang dibangun.

Pembangunan *prototype* yang kedua adalah perancangan basis pengetahuan, Basis pengetahuan (*knowledge base*) adalah komponen utama dari sebuah aplikasi sistem pakar. Basis pengetahuan ini diperoleh dari hasil proses akuisisi pengetahuan dan representasi pengetahuan, yaitu berupa fakta dan aturan. Dalam basis pengetahuan ini terdapat pengetahuan dalam perancangan aplikasi sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining* ini seperti *rule* yang berisi aturan atau rumus *forward chaining*, tabel keputusan yang berisi *rule* yang telah dibuat dan diimplementasikan kedalam sebuah tabel, pohon keputusan yang berisi implementasi *rule* kedalam sebuah pohon keputusan, kaidah produksi yang berisi informasi mengenai penyakit, gejala, solusi atau sebagai basis pengetahuan, perhitungan *forward chaining* yang berisi *rule* dari *forward*

chaining ini sendiri. Implementasi *rule forward chaining* ini adalah sebagai berikut:

a. Rule 1
 IF Mata Lelah
 AND Sering Menggosok Mata
 AND Pandangan Kabur Saat Melihat Objek
 AND Frekuensi Mengedipkan Mata yang Berlebihan
 THEN Miopia

b. Rule 2
 IF Kesulitan Membaca
 AND Melihat Objek Jauh Terlihat Jelas
 AND Melihat Objek Dekat Terlihat Buram
 AND Mata Terasa Panas dan Gatal
 AND Menggerlingkan Mata untuk Melihat Objek Jelas
 THEN Hipermetropia

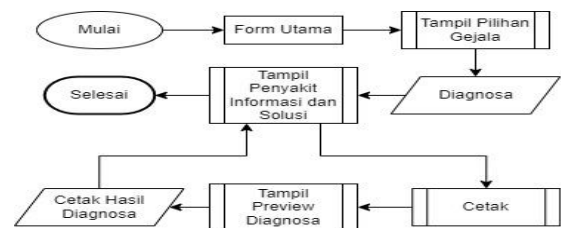
c. Rule 3
 IF Distorsi Penglihatan
 AND Pandangan Samar
 AND Penglihatan Ganda
 AND Sulit Melihat Saat Malam Hari
 AND Mata Sering Tegang dan Mudah Lelah
 AND Sensitif Pada Sorotan Cahaya
 THEN Astigmatisma

d. Rule 4
 IF Sering Menyipitkan Mata
 AND Pandangan Kabur Saat Melihat Objek
 AND Sulit Membaca Huruf Berukuran Kecil
 AND Kesulitan Membdekan Warna yang Mirip
 AND Membutuhkan Penerangan Lebih Saat Membaca
 THEN Presbiopi

Tahap ketiga perancangan *prototype* adalah perancangan basis data. Basis data dirancang menggunakan *MySql*. Terdapat 6 tabel dalam perancangan basis data aplikasi sistem pakar diagnosa penyakit refraksi ini yaitu

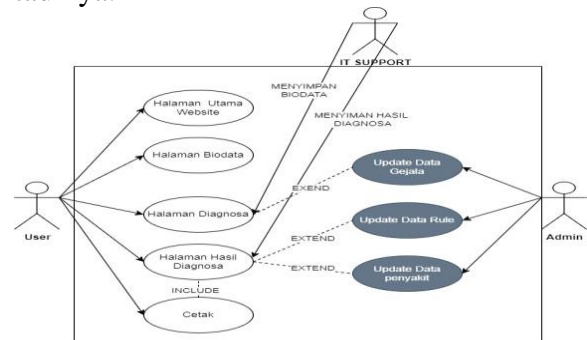
tabel gejala, tabel diagnosa, tabel hasil, tabel konsultasi, tabel relasi, dan tabel admin.

Tahap keempat perancangan *prototype* adalah perancangan Algoritma. Perancangan algoritma berisi alur aplikasi sistem pakar ini yang diimplementasikan kedalam sebuah *flowchart* agar lebih mudah dipahami algoritma perangkat lunaknya. *Flowchart* aplikasi ini dapat dilihat pada Gambar 2.



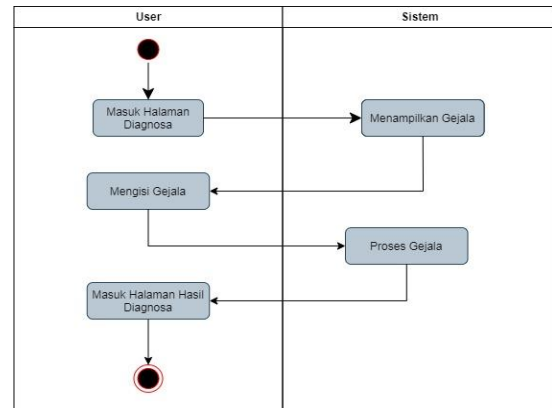
Gambar 2. Diagram Alur Diagnosa Penyakit Kelainan Refraksi

Tahap kelima perancangan *prototype* adalah perancangan UML(*Unified Modelling Language*). Dalam perancangan UML terdapat 4 diagram utama yaitu *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Perancangan UML bertujuan untuk memudahkan perencanaan dan penjelasan mengenai tahap-tahap dan alur aplikasi mulai dari awal hingga akhir. Pada aplikasi sistem pakar ini mulai dari halaman utama hingga kebagian hasil diagnosa dan menu cetak jika *user* ingin mencetak hasil diagnosa yang telah dilakukan oleh *user* tadinya.



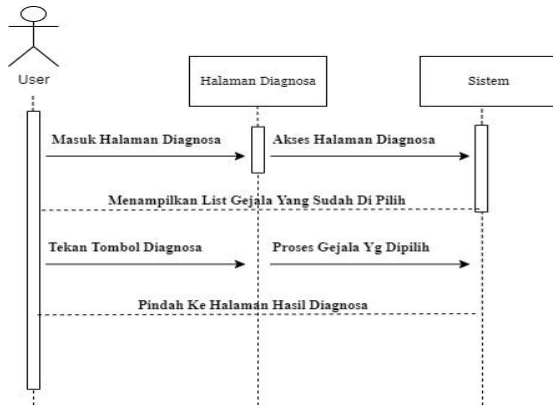
Gambar 3. Use Case Diagram Aplikasi Perangkat Lunak

Pada gambar 3 Menjelaskan *use case diagram* aplikasi sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining*. Pada aktor *user* menjelaskan halaman apa saja yang dapat diakses oleh *user* serta fitur apa saja yang dapat diakses oleh *user*, halaman yang dapat diakses oleh *user* adalah halaman utama website yang merupakan halaman awal sebelum melakukan diagnosa kemudian halaman biodata merupakan halaman pengisian biodata *user* sebagai bahan dokumentasi, halaman diagnosa merupakan halaman utama diagnosa atau halaman dimana *user* melakukan proses menjawab pertanyaan mengenai gejala yang dialami oleh *user*, kemudian setelah menjawab pertanyaan mengenai gejala maka *user* dapat mengakses halaman hasil diagnosa yang berisi hasil diagnosa berdasarkan pertanyaan gejala yang telah dijawab, halaman hasil diagnosa akan tampil penyakit yang diderita, penyebab penyakit, solusi penanganan awal. Setelah tampil hasil diagnosa maka *user* dapat mengakses menu cetak jika ingin mencetak hasil diagnosa yang telah dilakukan. IT *Support* adalah sistem yang bekerja melakukan proses-proses dalam perangkat lunak. Admin adalah orang yang bertanggung jawab dalam aplikasi ini, admin disini dapat mengakses menu tambah penyakit jika terdapat penyakit baru dikemudian hari beserta gejala-gejalanya. Admin disini juga berwenang dalam menghapus dan mengedit penyakit, gejala, solusi, dan penyebab yang tidak sesuai atau berubah dalam aplikasi ini.



Gambar 4. Activity Diagram Halaman Diagnosa

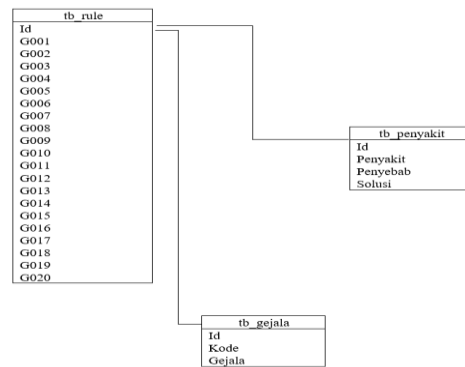
Pada gambar 4 menjelaskan *user* mengakses halaman utama diagnosa. Saat *user* masuk kehalaman diagnosa maka sistem akan menampilkan gejala-gejala yang tersedia dan *user* akan diarahkan untuk menjawab pertanyaan dengan jawaban “Ya” atau “Tidak”, pertanyaan-pertanyaan ini harus dijawab oleh *user* karena dengan menjawab pertanyaan ini sistem baru bisa merespon untuk melakukan proses perumusan *If-Then forward chaining*. Jika gejala sesuai dengan apa yang dialami oleh *user* maka *user* tinggal menekan tombol ya yang tersedia untuk melanjutkan ke pertanyaan berikutnya dan jika gejala tidak sesuai dengan yang dialami *user* maka *user* tinggal menekan tombol tidak dan sistem akan merespon dengan melanjutkan dan menyeleksi ke gejala-gejala selanjutnya. Setelah menjawab pertanyaan maka *user* akan diarahkan ke halaman hasil diagnose yang merupakan halaman tampilan hasil diagnosa yang berisi penyakit apa yang diderita oleh *user* hasil ini berdasarkan pertanyaan mengenai gejala-gejala yang telah dijawab sebelumnya, halaman hasil diagnosa juga menampilkan solusi penanganan awal penyakit refraksi serta penyebab penyakit refraksi yang terjadi pada *user*.



Gambar 5. *Sequence Diagram User* Mengakses Halaman Utama Diagnosa

Pada gambar 5 menjelaskan *sequence diagram user* mengakses halaman utama diagnosa aplikasi sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining*. Pada gambar 5. *User* masuk kehalaman diagnosa sistem pakar dan sistem akan mengarahkan *user* kehalaman biodata terlebih dahulu, *user* harus mengisi biodata terlebih dulu sebelum masuk ke halaman diagnosa, biodata yang diisikan oleh *user* tidak akan disalah gunakan dan hanya akan digunakan sebagai bahan dokumentasi aplikasi sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining* ini. Setelah selesai mengisi biodata maka *user* akan langsung diarahkan kehalaman diagnosa dan pada halaman diagnosa ini sistem akan menampilkan pertanyaan-pertanyaan mengenai gejala-gejala yang dialami oleh *user*. Kemudian *user* harus menjawab pertanyaan-pertanyaan tersebut dengan menekan tombol “Ya” atau “Tidak” pada pertanyaan tersebut. Jika *user* sudah selesai menjawab semua pertanyaan-pertanyaan yang diajukan maka *user* akan langsung diarahkan kehalaman hasil diagnosa yang merupakan bagian akhir dari diagnosa penyakit refraksi ini. Berdasarkan

pertanyaan-pertanyaan yang telah diajukan dan dijawab oleh *user* maka akan tampil hasil diagnosa pada halaman hasil diagnosa, halaman hasil diagnosa akan ditampilkan apabila *user* telah selesai menjawab pertanyaan yang ada. Halaman hasil diagnosa akan berisi informasi mengenai penyakit yang diderita, solusi penanganan awal mengenai penyakit yang diderita perlu kita garis bawahi solusi ini hanya berlaku sebagai P3K dan bukan sebagai solusi penanganan lanjutan.



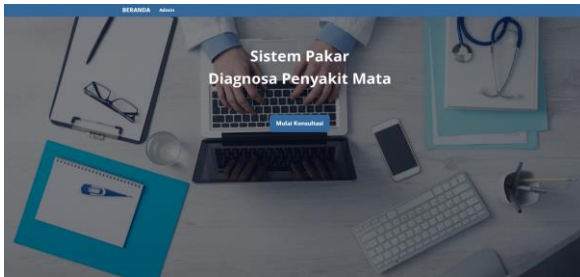
Gambar 6. *Class Diagram Relasi* DataBase

Pada gambar 6 menjelaskan *class diagram* relasi database. Pada diagram ini menjelaskan relasi gejala 1-20 dengan penyakit, ini berguna sebagai alat bantu proses perumusan *IF-THEN forward chaining* sehingga penyeleksian gejala dan penyakit dapat dilakukan dengan tepat dan tampilan hasil akhir diagnosapun akan sesuai dengan jawaban yang telah dijawab oleh *user* berdasarkan pertanyaan gejala yang diajukan oleh sistem.

Tahap keenam dalam pembuatan *prototype* adalah hasil perancangan aplikasi. Hasil perancangan berisi tampilan-tampilan form yang tersedia berdasarkan hasil perancangan aplikasi

sistem pakar diagnosa penyakit refraksi menggunakan metode *forward chaining*. Hasil perancangan aplikasi dilampirkan dalam bentuk gambar dibawah, sebagai berikut :

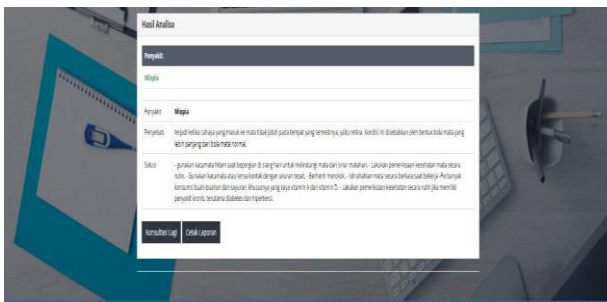
a. Halaman Utama



Gambar 7. Tampilan Halaman Utama Sistem Pakar

Pada gambar 7 merupakan tampilan halaman utama aplikasi. Sebelum masuk kemenu diagnosa *user* harus menekan tombol mulai konsultasi untuk melanjutkan ke halaman diagnosa. Pada halaman diagnosa *user* akan diberi berbagai pertanyaan mengenai gejala yang dirasakan oleh *user*.

b. Halaman Hasil Diagnosa



Gambar 8. Tampilan Halaman Hasil Diagnosa

Setelah pada halaman diagnosa *user* diminta untuk menjawab pertanyaan mengenai gejala yang terjadi pada *user* dengan menekan tombol “Ya” atau “Tidak”. Setelah selesai menjawab semua pertanyaan maka *user* akan diarahkan

kemenu hasil diagnosa. Gambar 8. merupakan tampilan halaman hasil diagnosa. Pada halaman ini akan tampil hasil diagnosa berupa penyakit yang diderita, penyebab dan solusi penanganan awal pada penyakit yang diderita. Hasil ini berdasarkan jawaban pertanyaan gejala yang telah dijawab oleh *user* pada halaman diagnosa.

c. Halaman Admin



Gambar 9. Tampilan Halaman Admin

Pada gambar 9 merupakan tampilan halaman admin. Pada halaman admin tersedia beberapa halaman informasi penyakit, gejala, pengetahuan, aturan dan laporan. Halaman admin disini memiliki wewenang dalam melakukan perubahan pada setiap menu. Halaman utama admin juga dilengkapi dengan fitur tambah, hapus, edit jika terdapat informasi baru atau terjadi kesalahan pada informasi yang tersedia.

3.3 Testing

Testing atau pengujian perangkat lunak adalah menguji semua elemen-elemen pada sistem perangkat lunak untuk mengetahui kesesuaian dari perancangan perancangan perangkat lunak yang dibangun. Pengujian ini dilaksanakan oleh *user* dengan metode pengujian menggunakan *Black-Box Testing*. Pengujian *Black-Box Testing* penulis gunakan untuk menguji masukan

serta membagi masukan kedalam kelompok-kelompok berdasarkan fungsinya. Sehingga didapatkan sebuah test case yang akurat. Adapun pengujian *Black-Box Testing* dari perancangan sistem pakar diagnosa penyakit refraksi yang dibuat yaitu :

Tabel 1. Hasil Pengujian *Black-Box Testing*

ID	Deskripsi	Hasil yang diharapkan	Hasil	Kesimpulan
1	Test Case Diagnosa	Tampil List diagnosa	1	Sukses
2	Test Case Gejala	Tampil gejala-gejala yang ada	1	Sukses
3	Test Case Penyakit	Tampil hasil dan detail penyakit	1	Sukses
4	Interface Website	Tampil semua halaman website tanpa error	1	Sukses

D. KESIMPULAN DAN SARAN

Kesimpulan pada penelitian ini adalah perancangan sistem pakar untuk mendiagnosa penyakit mata pada manusia berbasis web ini memiliki tujuan agar diagnosa penyakit refraksi dapat lebih cepat, murah serta efisien sehingga masyarakat dapat mencegah penyakit refraksi ini dengan lebih cepat. Perlu kita garis bawahi sistem pakar untuk mendiagnosa penyakit refraksi pada manusia berbasis *web* ini hanya sebatas sebagai P3K dan bukan sebagai alat pengobatan lanjutan, untuk pengobatan lanjutan tetap disarankan untuk

mengecekkkan diri kedokter sebagai pakar yang paling tepat untuk melakukan pengobatan lanjutan.

Terdapat beberapa saran yang dapat diberikan berkaitan dengan pengembangan perangkat lunak sistem pakar untuk mendiagnosa penyakit refraksi pada manusia berbasis *web* menggunakan metode *forward chaining*, yaitu: 1) Penambahan fitur pada *interface* atau antarmuka supaya pengguna lebih nyaman dengan aplikasi tersebut atau *user friendly*. 2) Perangkat lunak ini bisa dikembangkan dengan bahasa pemograman lain seperti java, python, ruby, ataupun angular, sehingga perangkat dan pengembangan yang lebih mudah dan langsung diterima oleh pengguna. 3) Perangkat lunak ini masih bisa ditambahkan menu konsultasi sehingga aplikasi dapat terhubung ke dokter secara langsung.

REFERENSI

- [1] Ilyas S, 2006, *Kelainan Refraksi dan Koreksi Pengelihatan*. Jakarta. *FKUI*; 2006.
- [2] Lukman,F., Lindra, A., Heriana, C. 2016, *Skrining Kelainan Refraksi Mata Pada Siswa Sekolah Dasar Menurut Tanda dan Gejala*. *Journal of Health Education*.
- [3] Kusrini, 2004, *Sistem Pakar Teori dan Aplikasi, ANDI*.
- [4] Sihotang, H.T., Panggabean, E., Zebua, H., 2018. *Sistem Pakar Mendiagnosa Penyakit Herpes Zoster dengan Menggunakan Metode Teorema Bayes*. *Journal of Informatic Pelita Nusantara*.
- [5] Sandy Kosasi, 2015. *Perancangan Sistem Pakar Mendiagnosa Penyakit Ikan Hias Menggunakan Shell Exsys Corvid*. *CSRID*.

- [6] Suwarsito., H. Mustafidah, 2011. Diagnosa Penyakit Ikan Menggunakan Sistem Pakar. *JUITA*.
- [7] David, 2016. Sistem Pakar Diagnosa Penyakit Gangguan Jiwa Dengan *Demster Shafer*. *KNSI* 2016.
- [8] Cut Fiarnia, A.S, 2015. *Automated Scheduling System For Thesis and Project Presentation Using Forward Chaining Method With Dynamic Allocation Resource*. *Pocedia Computer Science*.
- [9] David., Eric, A., dan Taufik, S., 2021. Penerapan Metode *Forward Chaining* Untuk Mediagnosa Penyakit Jantung Berbasis Android. *Prosiding Seminar Nasional CORISINDO 2021*.
- [10] David., 2018. Penerapan *Rule Based Forward Chaining* Pada Sistem Pakar Untuk Diagnosa Penyakit Kulit. *KNSI 2018*.
- [11] Sandy Kosasi., 2014. Sistem Pakar Diagnosa Penyakit Ikan Komet Menggunakan *Forward Chaining*. *TECHSI Jurnal Penelitian Teknik Informatika Universitas Malikussaleh*.
- [12] Sandy Kosasi., 2014. Sistem Pakar Tes Psikometri Kepribadian Manusia Menggunakan Metode *Forward Chaining*. *Seminar Nasional Informatika 2013*.
- [13] Dahlan, A., Muslem., 2015. Implementasi Sistem Pakar Mendiagnosa Penyakit Sakit Kepala Dengan Menggunakan Metode *Forward Chaining*. *TECHSI Jurnal Penelitian Teknik Informatika Universitas Malikussaleh*.
- [14] Roger S. Pressman, 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. *ANDI* Yogyakarta.
- [15] Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode *BlackBox Testing Boundary Value Analysis* (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT(JPIT)*.