# **SISTEM PAKAR UNTUK MENDIAGNOSIS PENYAKIT ANJING MENGGUNAKAN METODE CERTAINTY FACTOR**

**SKRIPSI**

****

**I GUSTI AGUNG NGURAH PANJI PALGUNA**

**NIM. 1608561051**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**2020**

# **SURAT PERNYATAAN KEASLIAN KARYA ILMIAH**

Yang bertanda tangan di bawah ini menyatakan bahwa naskah Skripsi dengan judul :

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Nama : I Gusti Agung Ngurah Panji Palguna

NIM : 1608561051

Program Studi : Informatika

Email : [panji\_palguna@student.unud.ac.id](mailto:panji_palguna@student.unud.ac.id)

Nomor HP : 082146149373

Alamat : Jl. Gunung Sari IV No.9 Desa Tegal Harum, Denpasar

Belum pernah dipublikasikan dalam dokumen skripsi, jurnal nasional maupun internasional atau dalam pro siding mana pun, dan tidak sedang atau akan diajukan untuk publikasi di jurnal atau pro siding mana pun. Apabila di kemudian hari terbukti terdapat pelanggaran kaidah-kaidah akademik pada karya ilmiah saya, maka saya bersedia menanggung sanksi-sanksi yang dijatuhkan karena kesalahan tersebut, sebagaimana diatur oleh Peraturan Menteri Pendidikan Nasional Nomor 17 Tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi.

Demikian Surat Pernyataan ini saya buat dengan sesungguhnya untuk dapat dipergunakan bilamana diperlukan.

Denpasar, Juni 2020

Yang membuat pernyataan,

I Gusti Agung Ngurah Panji Palguna

# **LEMBAR PENGESAHAN TUGAS AKHIR**

|  |  |  |
| --- | --- | --- |
| Judul | : | Sistem Pakar Untuk Mendiagnosis Penyakit Anjing Menggunakan Metode *Certainty Factor* |
| Kompetensi | : | Komputasi Cerdas |
| Nama | : | I Gusti Agung Ngurah Panji Palguna |
| NIM | : | 1608561051 |
| Tanggal Seminar | : |  |

Disetujui oleh :

|  |
| --- |
| Pembimbing I |
|  |

Penguji I

|  |
| --- |
| Dra. Luh Gde Astuti, M.Kom. |
| NIP. 196401141994022001 |

|  |
| --- |
| Komang Ari Mogi, S.Kom, M.Kom |
| NIP. 196401141994022001 |

|  |
| --- |
| Pembimbing II |
|  |

Penguji II

|  |
| --- |
| Agus Muliantara, S.Kom, M.Kom. |
| NIP. 198006162005011001 |
|  |

|  |
| --- |
| I Wayan Supriana, S.Si, M.Cs. |
| NIP. 1984082920181113001 |

Penguji III

|  |
| --- |
|  |
| Gusti Ayu Vida Mastrika Giri, S.Kom, M.Cs |
| NIP. 1990060620181123001 |

Mengetahui

Koordinator Program Studi Informatika FMIPA

Universitas Udayana

|  |
| --- |
|  |
| Dr. I Ketut Gede Suhartana, S.Kom, M.Kom. |
| NIP. 197201102008121001 |

|  |  |  |
| --- | --- | --- |
| Judul | : | Sistem Pakar Untuk Mendiagnosis Penyakit Anjing Menggunakan Metode *Certainty Factor* |
| Nama | : | I Gusti Agung Ngurah Panji Palguna |
| NIM | : | 1608561051 |
| Pembimbing I | : | Dra. Luh Gde Astuti, M.Kom. |
| Pembimbing II | : | Agus Muliantara, S.Kom., M.Kom. |

# **ABSTRAK**

Penyakit anjing banyak ragamnya dengan gejala-gejala yang berbeda-beda. Dari banyaknya penyakit anjing secara umum dapat dibagi menjadi 3 penyakit yaitu disebabkan oleh virus/bakteri, cacing, dan kulit. Namun pada penelitian kali ini dilakukan hanya fokus pada penyakit virus/bakteri, dan cacing. Dikarenakan sangat mudah untuk mengetahui karena penyakit cacing mudah dari jenis cacing apa yang menyebabkan anjing tersebut sakit.

Pada penelitian ini akan dilakukan perancangan dan implementasi Sistem Pakar Untuk Mendiagnosis Penyakit Anjing Menggunakan Metode *Certainty Factor*(CF) . *Certainty Factor* berasal dari 2 sumber yaitu pakar dan dokter yang sedang memeriksa anjing. Pakar yang dimaksud adalah dokter hewan yang memberikan penilaian *Certainty Factor* pengaruh gejala terhadap penyakit yang mungkin diderita.

Sistem sudah melakukan proses analisis, desain, implementasi dan pengujian pada Sistem Pakar Mendiagnosis Penyakit Anjing. Dari pertanyaan yang diberikan akan menuju pada gejala selanjutnya sesuai dengan aturan yang diberikan. Sistem juga sudah mampu untuk menghitung nilai CF pada setiap gejala dengan jawaban Ya.

Adapun pada implementasinya terdapat 8 bagian yaitu *Login,* Manajemen Gejala, Manajemen Pengguna, Manajemen Diagnosa. Manajemen Aturan, Manajemen Konsultasi, dan Manajemen Hasil Konsultasi. Adapun nilai rata-rata evaluasi adalah 3.58 menandakan sistem mudah dan berguna bagi calon dokter hewan.

|  |  |
| --- | --- |
| Title : | Expert System to Diagnose Dog Diseases Using the Certainty Factor Method |
| Name : | I Gusti Agung Ngurah Panji Palguna |
| Student Number : | 1608561051 |
| Main Supervisor : | Dra. Luh Gde Astuti, M.Kom. |
| Co-Supervisor : | Agus Muliantara, S.Kom., M.Kom |

# **ABSTRACT**

There are many kinds of dog diseases with different symptoms. Of the many diseases of dogs in general can be divided into 3 diseases that are caused by viruses / bacteria, worms and skin. However, this research is only focused on viral / bacterial diseases and worms. Because it is very easy to know because worm disease is easy from what type of worm is causing the dog to get sick.

In this research, an expert system design and implementation will be carried out to diagnose dog diseases using the Certainty Factor (CF) Method. Certainty Factor comes from 2 sources, namely experts and doctors who are examining dogs. The expert in question is a veterinarian who provides a Certainty Factor assessment of the effect of symptoms on a disease that may be suffered.

The system has carried out the process of analysis, design, implementation and testing in the Expert System to Diagnose Dog Diseases. From the questions given will lead to the next symptoms in accordance with the rules given. The system has also been able to calculate the value of CF for each symptom with the answer Yes.

In the implementation, there are 8 parts namely Login, symptom management, user management, diagnosis management. Management of rules, consultation management, and consulting results. The average value of evaluation is 3.58 signifying an easy and useful system for prospective veterinarians.

# **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadapan Tuhan Yang Maha Esa, karena berkat rahmat dan karunianya, Tugas Akhir yang berjudul “Sistem Pakar Untuk Mendiagnosis Penyakit Anjing Menggunakan Metode *Certainty Factor*” ini dapat diselesaikan tepat pada waktunya.

Secara khusus penulis mengucapkan terima kasih dan penghargaan kepada berbagai pihak yang telah membantu penelitian ini, yaitu

1. Ibu Dra. Luh Gde Astuti, M.Kom. sebagai pembimbing I yang telah bersedia mengkritisi, membantu, dan memeriksa serta menyempurnakan Laporan Tugas Akhir ini.
2. Bapak Agus Muliantara, S.Kom., M.Kom. sebagai pembimbing II yang telah bersedia mengkritisi, membantu, dan memeriksa serta menyempurnakan Laporan Tugas Akhir ini.
3. Seluruh Dosen Program Studi Informatika atas bimbingannya selama kuliah
4. Orang tua dan keluarga yang sudah memberikan semangat dan doanya
5. Ni Wayan Wiantari, I Gusti Ngurah Alit Indrawan, Kiki Dwi Pebriana, Frisca Olivia Gorianto, dan Kevin Joel sebagai sahabat sejalur atas semangat, doa dan ilmunya selama menempuh jalur Komputasi Cerdas
6. Seluruh teman Program Studi Informatika atas ilmu, pengalaman, semangat doanya selama kuliah di Program Studi Informatika
7. Seluruh teman KKN Unud Periode XIX di Desa Bungbungan, Kecamatan Banjarangkan, Kabupaten Klungkung atas semangat dan doanya
8. Wahidatul Nur Azizah sudah membantu dalam dokumentasi
9. Mahasiswa Kedokteran Hewan dan Bapak drh. Ari Sapto Nugroho sudah membantu dalam penyusunan skripsi ini
10. Semua pihak yang telah memberi dukungan sehingga laporan ini dapat diselesaikan sesuai dengan waktu yang ditentukan

Pada akhirnya penulis berharap agar adanya perbaikan pada Laporan Tugas Akhir ini mengingat keterbatasan penulis, sehingga sangat diharapkan untuk adanya kritik dan saran yang membangun untuk pencapaian yang lebih baik

Bukit Jimbaran, Juni 2020

Penulis

# **DAFTAR ISI**

[HALAMAN SAMPUL i](#_Toc40300007)

[SURAT PERNYATAAN KEASLIAN KARYA ILMIAH ii](#_Toc40300008)

[LEMBAR PENGESAHAN TUGAS AKHIR iii](#_Toc40300009)

[ABSTRAK iv](#_Toc40300010)

[ABSTRACT v](#_Toc40300011)

[KATA PENGANTAR vi](#_Toc40300012)

[DAFTAR ISI vii](#_Toc40300013)

[DAFTAR TABEL x](#_Toc40300014)

[DAFTAR LAMPIRAN xii](#_Toc40300015)

[BAB I PENDAHULUAN 1](#_Toc40300016)

[1.1 Latar Belakang 1](#_Toc40300017)

[1.2. Rumusan Masalah 3](#_Toc40300018)

[1.3. Tujuan Penelitian 3](#_Toc40300019)

[1.4. Batasan Masalah 3](#_Toc40300020)

[1.5. Manfaat Penelitian 3](#_Toc40300021)

[BAB II 8](#_Toc40300022)

[TINJAUAN PUSTAKA 8](#_Toc40300023)

[2.1. Sistem Pakar 8](#_Toc40300024)

[2.2. Basis Pengetahuan (Knowledge Base) 10](#_Toc40300025)

[2.3. Representasi Pengetahuan 11](#_Toc40300026)

[2.3.1. Kaidah Produksi 11](#_Toc40300027)

[2.3.2. Tabel Keputusan 12](#_Toc40300028)

[2.4. Forward Chaining 14](#_Toc40300029)

[2.5. Metode Pengembangan Waterfall 16](#_Toc40300030)

[2.5.1. Pendefinisian Kebutuhan 17](#_Toc40300031)

[2.5.2. Desain Software dan Sistem 18](#_Toc40300032)

[2.5.3. Implementasi Sistem 18](#_Toc40300033)

[2.5.4. Integrasi dan Pengujian Sistem 19](#_Toc40300034)

[2.6. Pemrograman Web 19](#_Toc40300035)

[2.7. Framework Yii 20](#_Toc40300036)

[2.8. *Model* View-Controller 21](#_Toc40300037)

[2.9 Tinjauan Studi 21](#_Toc40300038)

[BAB III 25](#_Toc40300039)

[ANALISIS DAN PERANCANGAN 25](#_Toc40300040)

[3.1 Analisis Kebutuhan 25](#_Toc40300041)

[3.1.1. Kebutuhan Pemakai 25](#_Toc40300042)

[3.1.2. Kebutuhan Fungsional 25](#_Toc40300043)

[3.2 Perancangan Sistem Pakar 29](#_Toc40300044)

[3.2.1. Perancangan Akusisi Pengetahuan 29](#_Toc40300045)

[3.2.2. Perancangan Basis Pengetahuan 29](#_Toc40300046)

[3.2.3 Perancangan Representasi Pengetahuan 31](#_Toc40300047)

[3.2.4. Perancangan Mesin Inferensi 33](#_Toc40300048)

[3.2.5. Analisis Pembobotan 34](#_Toc40300049)

[3.2.6. Workplace 35](#_Toc40300050)

[3.2.7. Perbaikan Pengetahuan 36](#_Toc40300051)

[3.3. Desain Sistem 36](#_Toc40300052)

[3.3.1. Use Case Diagram 36](#_Toc40300053)

[3.3.2. Class Diagram 37](#_Toc40300054)

[3.3.3. Physical Data Model 38](#_Toc40300055)

[3.3.4. Alur Program Konsultasi 39](#_Toc40300056)

[BAB IV 40](#_Toc40300057)

[HASIL DAN PEMBAHASAN 40](#_Toc40300058)

[4.1 Implementasi Sistem 40](#_Toc40300059)

[4.1.1. Implementasi *Database* 40](#_Toc40300060)

[4.1.2. Implementasi Login dan Manajemen *User* 43](#_Toc40300061)

[4.1.3. Implementasi Manajemen Diagnosa 45](#_Toc40300062)

[4.1.4. Implementasi Manajemen Gejala 45](#_Toc40300063)

[4.1.5. Implementasi Manajemen Aturan 46](#_Toc40300064)

[4.1.6. Implementasi Konsultasi 47](#_Toc40300065)

[4.1.7. Implementasi Manajemen Hasil Konsultasi 48](#_Toc40300066)

[4.1.8. Implementasi Beranda/*Dashboard* 49](#_Toc40300067)

[4.1.9. Implementasi Pengetahuan Pakar 50](#_Toc40300068)

[4.1.10. Implementasi Model-View Controller dalam Yii Framework 50](#_Toc40300069)

[4.1.11. Proses Perhitungan CF dalam Sistem 52](#_Toc40300070)

[4.2. Pengujian Sistem 55](#_Toc40300071)

[4.2.1. Black Box Testing 55](#_Toc40300072)

[4.2.2. Pengujian Metode *Certainty Factor* 62](#_Toc40300073)

[BAB V 68](#_Toc40300074)

[PENUTUP 68](#_Toc40300075)

[5.1. Kesimpulan 68](#_Toc40300076)

[5.2. Saran 68](#_Toc40300077)

[DAFTAR PUSTAKA 69](#_Toc40300078)

# **DAFTAR TABEL**

[Tabel 1.1. Tabel Contoh Kebutuhan Perangkat Lunak 4](#_Toc40380865)

[Tabel 3.1. Use Case Description 36](#_Toc40380874)

[Tabel 4.1. Blackbox Login 54](#_Toc40380898)

[Tabel 4.2. Blackbox Manajemen Gejala 55](#_Toc40380899)

[Tabel 4.3 Blackbox Manajemen Pengguna 56](#_Toc40380900)

[Tabel 4.5. Blackbox Manajemen Diagnosa 57](#_Toc40380901)

[Tabel 4.6. Blackbox Manajemen Aturan 58](#_Toc40380902)

[Tabel 4.7 Blackbox Manajemen Hasil Konsultasi 59](#_Toc40380903)

[Tabel 4.8. Blackbox Konsultasi 60](#_Toc40380904)

Tabel 4.9. Tabel Pengujian Kuesioner Usability Sistem........................................63

Tabel 4.10. Tabel Penilaian Kuesioner..................................................................65

# **DAFTAR GAMBAR**

[Gambar 2.1 Konsep Dasar Fungsi Sistem Pakar 8](#_Toc40381963)

[Gambar 2.2 Arsitektur Sistem Pakar (Sumber : Desiani dkk, 2006) 10](#_Toc40381964)

[Gambar 2.3. Tabel Keputusan (sumber : Hartati dan Iswanti) 13](#_Toc40381965)

[Gambar 2.4. Model Proses Waterfall 16](#_Toc40381966)

[Gambar 2. 5. Arsitektur Model View dan Controller 22](#_Toc40381967)

[Gambar 3.1 Use Case Diagram 35](#_Toc40381970)

[Gambar 3.2. *Class Diagram* 37](#_Toc40381972)

[Gambar 3.3. Physical Data Model 37](#_Toc40381973)

[Gambar 4.1. Tabel *User* 39](#_Toc40381974)

[Gambar 4.2. Tabel Konsultasi 40](#_Toc40381975)

[Gambar 4.3. Tabel Hasil Konsultasi 40](#_Toc40381976)

[Gambar 4.4. Tabel Aturan 41](#_Toc40381977)

[Gambar 4 5. Tabel Gejala 41](#_Toc40381978)

[Gambar 4 6. Tabel Diagnosa 41](#_Toc40381979)

[Gambar 4 7. Tabel Pakar 42](#_Toc40381980)

[Gambar 4 8. Manajemen *User* 43](#_Toc40381981)

[Gambar 4.9 Halaman Login 43](#_Toc40381982)

[Gambar 4 10. Halaman Manajemen Diagnosa 44](#_Toc40381983)

[Gambar 4.11. Implementasi Manajemen Gejala 45](#_Toc40381984)

[Gambar 4.12. Implementasi Manajemen Aturan 45](#_Toc40381985)

[Gambar 4.13. Implementasi Manajemen Konsultasi 46](#_Toc40381986)

[Gambar 4.14 Implementasi Manajemen Konsultasi 47](#_Toc40381987)

[Gambar 4.15. Implementasi Manajemen Hasil Konsultasi 47](#_Toc40381988)

[Gambar 4 16. Implementasi Beranda 48](#_Toc40381989)

[Gambar 4.17. Implementasi Halaman Manajemen Pakar 49](#_Toc40381990)

[Gambar 4.18. Controller dalam Yii Framework 50](#_Toc40381991)

[Gambar 4.19. Model dalam Yii Framework 50](#_Toc40381992)

[Gambar 4.20. Views dalam Aplikasi 51](#_Toc40381993)

[Gambar 4.21. Source Code Proses Perhitungan *Certainty Factor* dalam Sistem 54](file:///D:\1608561051_Ujian%20Skripsi.docx#_Toc40381994)

# **DAFTAR LAMPIRAN**

Dokumentasi Sistem 71

Dokumentasi Pencarian Data dan Pengujian Data…………………..…………...76

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Perkembangan teknologi informasi khususnya komputer sudah mengalami perkembangan yang sangat pesat, seiring dengan kebutuhan manusia yang semakin banyak dan kompleks. Kecerdasan buatan atau *artificial intelligence* merupakan bagian dari ilmu komputer yang membuat agar mesin (computer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Sistem cerdas (Intelligent sistem) adalah sistem yang dibangun dengan menggunakan teknik-teknik *artificial intelligence*. Sistem Pakar *(Expert System)* merupakan sistem cerdas. Sistem Pakar merupakan program komputer yang meniru proses pemikiran dan pengetahuan pakar dalam menyelesaikan suatu masalah tertentu (Fauzijah dan Rohman, 2008).

Penerapan sistem pakar pada bidang kesehatan dan kedokteran sudah banyak dilakukan. Di aantaranya adalah MYCIN yang digunakan untuk diagnosis dan pengobatan penyakit infeksi yang disebabkan oleh bakteri, VM yang berfungsi untuk memonitor pasien *intensive-care*, dan GUIDON yang digunakan sebagai instruksi/tutorial tentang penyakit infeksi yang disebabkan oleh bakteri. Beberapa alat kedokteran saat ini memanfaatkan sistem pakar. Ada yang sebagai pendukung keputusan dan ada juga yang bekerja untuk menyembuhkan suatu penyakit mulai yang sederhana hingga kompleks. Bidang kedokteran saat ini juga telah manfaatkan sistem pakar untuk membantu peningkatan pelayanan yang lebih baik kepada masyarakat dengan menggunakan sistem pakar untuk membantu peningkatan pelayanan yang lebih baik kepada masyarakat dengan menggunakan sistem pakar untuk membantu seorang dokter dalam mendiagnosis berbagai macam penyakit (Hamdani, 2010).

Anjing adalah mamalia atau hewan peliharaan yang cukup menarik karena memberikan beberapa manfaat. Kita, terutama yang mencintai dan memelihara anjing juga harus memperhatikan kondisi kesehatan anjing, karena tidak menutup

kemungkinan penyakit yang diderita anjing dapat mempengaruhi kita atau bahkan menular ke kita. Salah satu pencegahannya adalah dengan mengetahui gejala penyakitnya, salah satunya adalah memeriksakan anjing kita ke dokter hewan secara teratur. Namun, keterbatasan waktu dan biaya saat ini untuk pergi ke dokter menyebabkan sulitnya mendapatkan informasi tentang penyakit yang diderita anjing. Tentu dari masalah tersebut, akan mencoba membangun sistem pakar rekayasa perangkat lunak untuk mendiagnosis penyakit pada anjing. Metode inferensi yang digunakan aplikasi ini dilengkapi dengan *forward chaining* dan sistem manajemen pengetahuan yang mudah digunakan dan dinamis. Ini berarti bahwa para ahli dapat menambah, memodifikasi, dan menghapus pengetahuan atau aturan baru tanpa harus memulai dari awal. Aplikasi ini dapat membantu pemilik anjing untuk mendapatkan beberapa informasi tentang gejala yang dialami anjing. Semakin cepat penyakit diketahui, maka semakin cepat mereka dapat mencegahnya.

Dalam melakukan diagnosis, seorang pakar terkadang mendasarkan pemikirannya pada validitas data yang tidak pasti. Agar sistem pakar dapat melakukan penalaran sebagaimana seorang pakar meskipun data yang diperoleh kurang valid atau kurang pasti, dapat digunakan metode *Certainty Factor* (CF). Kebanyakan sistem pakar yang ada, untuk mendapatkan kepercayaan pengguna terhadap gejala yang terjadi adalah dengan meminta nilai kepercayaan (CF) langsung dari pengguna.

Pada penelitian ini akan dilakukan perancangan dan implementasi Sistem Pakar Untuk Mendiagnosis Penyakit Anjing Menggunakan *Metode Certainty Factor*(CF) . *Certainty Facto*r berasal dari 2 sumber yaitu pakar dan dokter yang sedang memeriksa anjing. Pakar yang dimaksud adalah dokter hewan yang memberikan penilaian *Certainty Factor* pengaruh gejala terhadap penyakit yang mungkin dialami

## **1.2. Rumusan Masalah**

Sesuai dengan permasalahan dari objek yang ada pada latar belakang yang telah disampaikan di atas, dapat dirumuskan permasalahan dalam penelitian ini adalah bagaimana merancang dan mengimplementasikan aplikasi sistem pakar untuk mendiagnosis penyakit anjing menggunakan *certainty factor*

## **1.3. Tujuan Penelitian**

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah mendapatkan hasil perancangan dan implementasi berupa sistem pakar untuk mendiagnosis penyakit anjing menggunakan metode *certainty factor.*

## **1.4. Batasan Masalah**

Batasan masalah pada penelitian tugas akhir ini adalah sebagai berikut :

1. Aplikasi sistem pakar dalam penelitian ini memberikan hasil berupa kesimpulan awal yang merupakan penyakit pada anjing yang diderita oleh anjing berupa penyakit kulit, cacing, dan virus/bakteri.
2. Sistem dibuat menggunakan Bahasa pemrograman php dan *database* mysql dengan menggunakan *framework* yii
3. Metode yang digunakan adalah *Forward Chaining* dan *Certainty Factor*

## **1.5. Manfaat Penelitian**

Manfaat yang diperoleh dari penelitian ini adalah :

1. Bagi masyarakat umum dapat digunakan sebagai alat mendiagnosis penyakit anjing sehingga dapat dilakukan penanganan awal terhadap penderita penyakit anjing tanpa perlu berkonsultasi dengan dokter hewan
2. Bagi dokter hewan yang tidak terlalu memahami penyakit pada anjing dapat dipakai sebagai alat bantu diagnosis awal penyakit sehingga dapat dilakukan penanganan awal terhadap anjing

## **1.6. Metodelogi Penelitian**

Metode penelitian ini menggunakan metode pengembangan system yaitu metode pengembangan sistem *waterfall,* yang mempunyai tahapan analisis kebutuhan, desain system, implementasi sistem, dam pengujian sistem.

### **1.6.1. Analisis Kebutuhan**

Tahap ini adalah tahap untuk merancang atau mengidentifikasi apa saja kebutuhan sistem yang diperlukan oleh pengguna sehingga sistem dapat digunakan secara maksimal. Pada tahap ini langkah yang dilakukan adalah :

1. Mengidentifikasi kebutuhan pengguna sistem pakar penyakit pada anjing
2. Mengidentifikasi kebutuhan fungsional sistem pakar penyakit pada anjing

Kebutuhan fungsional akan dideskripsikan pada table berikut :

Tabel 1.1. Tabel Contoh Kebutuhan Perangkat Lunak

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kode Fungsi | Nama Fungsi | Deskripsi |
| 1 | FA | Fungsi A | Deskripsi A.1  Deskripsi A.2 |
| 2 | FB | Fungsi B | Deskripsi B.1  Deskripsi B.2 |

1. Mengidentifikasi kebutuhan non fungsional sistem pakar penyakit pada anjing dari aspek kebutuhan produk yaitu aspek keakurasian sistem dalam mendiagnosis

Sumber data dalam penelitian ini berasal dari data primer dan sekunder. Data primer adalah data yang diambil langsung dari objek penelitian atau merupakan data yang berasal dari sumber asli atau pertama (Zainal Hasibuan, 2007). Jadi data didapatkan langsung dari sumber aslinya atau sumber pertama, sedangkan data sekunder merupakan data yang didapat dari orang lain atau instansi lain. Jadi data sekunder tidak didapatkan langsung dari objek yang diteliti, melainkan didapatkan dari orang atau kelompok yang mengumpulkan data-data tersebut dari objeknya.

Metode pengumpulan data yang digunakan disini adalah metode wawancara dan metode studi pustaka. Wawancara ini dilakukan dengan mencari narasumber yang ahli untuk mendapatkan data-data dari objek penelitian, sedangkan metode studi pustaka dengan mengumpulkan buku-buku terkait penyakit anjing

### **1.6.2. Perancangan**

Proses desain akan mengimplementasikan syarat kebutuhan dari sistem ke dalam sebuah perancangan perangkat lunak. Tahap ini adalah tahap dimana proses untuk merancang atau merencanakan sistem yang akan dibangun. Langkah yang dilakukan adalah sebagai berikut :

1. Perancangan Sistem Pakar
2. Membangun akusisi pengetahuan

Pada tahap perancangan sistem pakar, untuk membangun akusisi pengetahuan pada sistem pakar dilakukan dengan menyerap pengetahuan dari seorang ahli, yang pada penelitian ini bersumber dari pakar yaitu dokter hewan

1. Membangun basis pengetahuan

Pada tahapan ini merancang basis pengetahuan yang diperlukan untuk memahami, merumuskan dan memecahkan masalah. Basis pengetahuan terdiri dari pengetahuan dari seorang pakar yang terdiri dari gejala-gejala untuk melakukan diagnosis

1. Membangun representasi pengetahuan

Untuk menangkap sifat-sifat penting dari suatu masalah dan mempermudah prosedur dalam pemecahan masalah, yang dimana dilakukan untuk pemecahan diagnosis penyakit, maka pengetahuan harus direpresentasikan dalam bentuk frame jika maka (IF-THEN)

1. Membangun mesin inferensi

Pada tahapan ini merancang program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarakan aturan yang ada. Sistem pakar ini menggunakan teknik penalaran maju (Forward Chaining) dimana pelacakan maju dimulai dari informasi, masukan dan selanjutnya menggambarkan kesimpulan

1. Pembobotan pada gejala penyakit

Pada tahapan ini, gejala diberikan sebuah bobot yang dimana bobot didapatkan dari wawancara dengan pakar, bobot tersebut akan digunakan untuk menghitung nilai kepercayaan dari kesimpulan penyakit

1. Perancangan peran pengguna menggunakan *Use Case Diagram*
2. Perancangan *database* dimodelkan menggunakan *Class Diagram*
3. Perancangan alur dari komponen-komponen sistem

### **1.6.3. Implementasi Sistem**

Pada tahapan ini adalah merupakan hasil dari tahapan perancangan yang akan diimplementasikan menjadi sebuah sistem. Dalam pengimplementasiannya perancangan ke dalam sebuah sistem web. Dalam pengimplementasiannya menggunakan bahasa pemrograman php dan *database* mysql.

Adapun langkah-langkah dalam implementasi sistem adalah sebagai berikut :

1. Identifikasi Gejala Pengguna

Sistem mengidentifikasi gejala-gejala yang disarankan *user* dengan memberikan pertanyaan mengenai gejala dari penyakit anjing dengan pilihan jawaban ya atau tidak, selanjutnya *user* akan memilih salah satu dari jawaban yang diberikan oleh sistem, jawaban akan menentukan alur dari *Forward Chaining* yang telah sesuai dengan *rule* yang diberikan oleh pakar dan diberikan dengan sebuah kesimpulan

1. Identifikasi gejala dengan *rule*

Setelah sistem mengidentifikasi gejala pengguna, selanjutnya sistem akan mengidentifikasi gejala pengguna dengan *rule* dari penyakit pada anjing

1. Perhitungan Nilai Keyakinan

Setelah selesai mendapat *rule* yang sesuai dengan memasukkan gejala pengguna maka sistem selanjutnya menghitung nilai keyakinan yang sesuai dengan gejala yang diinputkan dan dihitung berdasarkan gejala apa saja yang dipilih oleh pengguna.

1. Menampilkan Diagnosis

Sistem akan menampilkan hasil diagnosis penyakit dan juga nilai persentase keyakinan dari penyakit tersebut, yang kemudian pengguna dapat melihat hasil diagnosis serta bisa mengidentifikasi diagnosis sesuai dengan persentase yang dihasilkan

### **1.6.4 Pengujian Sistem**

Pengujian sistem dilakukan untuk mengetahui apakah sistem pakar sudah berjalan dengan baik dari sisi pengambilan kesimpulan dari diagnosis gejala pengguna maupun dari segi kebutuhan fungsional dari sistem. Tahap pengujian terdiri dari

1. Pengujian Algoritma

Apakah sistem sudah berjalan sesuai dengan algoritma yang digunakan dalam hal ini menggunakan metode *certainty factor*

1. Pengujian Kebutuhan Fungsional

Tujuan dari pengujian fungsional adalah untuk memvalidasi perilaku perangkat lunak terhadap kebutuhan fungsional didokumentasikan dalam persyaratan perangkat lunak

# **BAB II**

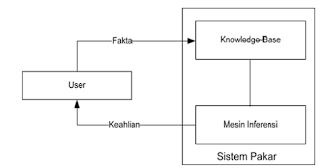
# **TINJAUAN PUSTAKA**

## **2.1. Sistem Pakar**

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 (Kusrini, 2007).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain : pembuatan keputusan *(decision making)*, perencanaan *(planning)*, peramalan *(forecasting)*, pengaturan *(regulating)*, pengendalian *(controlling)* dan diagnosis.

Fungsi Sistem Pakar berkaitan dengan tujuan adalah memecahkan suatu permasalahan pada tingkat pakar dengan cara di mana seorang pakar/*user* mentransfer kepakaran yang berupa fakta ke dalam sistem pakar dan menghasilkan keahlian atau saran pakar sebagai respons. Secara internal, sistem pakar terdiri dari dua komponen utama. Basis pengetahuan *(knowledge base)* yang berisi pengetahuan pakar kemudian dengan mesin inferensi *(inference engine)* untuk menarik kesimpulan. Kesimpulan yang dihasilkan akan menjadi *output* untuk pengguna



Gambar 2.1 Konsep Dasar Fungsi Sistem Pakar

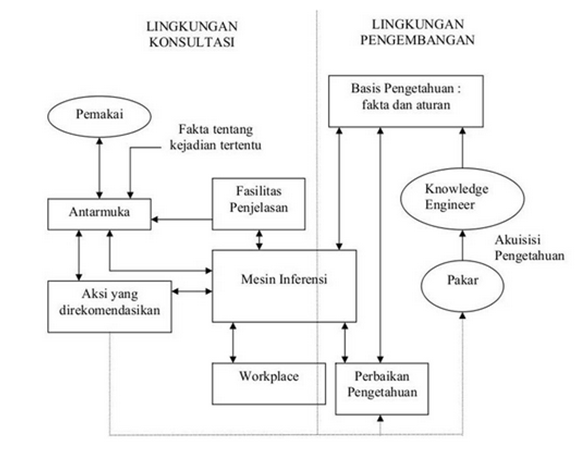
(sumber : Rahmadi Wijaya, 2007)

### **2.1.1. Komponen Sistem Pakar**

Secara umum, Sistem Pakar biasanya terdiri atas beberapa komponen yang masing-masing berhubungan di antaranya adalah :

1. Basis Pengetahuan *(knowledge base)*, berisi pengetahuan yang dibutuhkan. Seorang pakar akan melakukan akuisisi pengetahuan ke dalam basis pengetahuan. Basis pengetahuan tersusun atas 2 elemen dasar :
2. Fakta, Situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu.
3. Aturan, yang mengarahkan penggunaan pengetahuan untuk memecahkan masalah yang spesifik dalam bidang khusus
4. Mesin Inferensi *(Inference Engine)*. Komponen ini berupa program komputer yang menyediakan suatu metodologi untuk memikirkan *(reasoning)* dan memformulasi kesimpulan. Kerja mesin inferensi meliputi|:
5. Menentukan kaidah mana yang akan dipakai
6. Menyajikan pernyataan kepada pemakai, Ketika diperlukan
7. Menambahkan jawaban ke dalam memori Sistem Pakar
8. Menyimpulkan fakta baru dari sebuah aturan
9. Menambahkan fakta tadi ke dalam memori
10. *Workplace (Blackboard*), adalah memori/lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data
11. Antarmuka Pemakai *(User Interface)* . Sistem Pakar mengatur komunikasi antara pengguna dan komputer
12. Fasilitas penjelasan *(Explanation Facility).* Kemampuan untuk menjejak bagaimana suatu kesimpulan bisa diambil
13. Sistem Perbaikan Pengetahuan. Seorang pakar mempunyai sistem penghalusan pengetahuan. Pada sistem pakar, evaluasi sendiri ini penting sehingga dapat menganalisis sistem keberhasilan atau kegagalan pengambilan kesimpulan, serta memperbaiki basis pengetahuannya.

Hubungan dari semua komponen di atas digambarkan dalam sebuah arsitektur sistem pakar seperti pada gambar 2.2



Gambar 2.2 Arsitektur Sistem Pakar (Sumber : Desiani dkk, 2006)

Arsitektur sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan *(development environment)* dan lingkungan konsultasi *(consultation environment)*. Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar., sedangkan lingkungan konsultasi digunakan oleh pengguna untuk memasukkan fakta dan memperoleh pengetahuan pakar.

## **2.2. Basis Pengetahuan *(Knowledge Base)***

Basis pengetahuan berisi tentang pengetahuan-pengetahuan dalam penyelesaian masalah. Pendekatan basis pengetahuan yang umum digunakan yaitu penalaran berbasis aturan *(Rule-Based Reasoning)*. Pada penalaran berbasis aturan pengetahuan direpresentasikan dengan menggunakan aturan berbentuk : IF-THEN. Bentuk ini digunakan apabila telah memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan pakar dapat menyelesaikan permasalahan secara berurutan.

## **2.3. Representasi Pengetahuan**

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah.

Adapun karakteristik dari metode representasi pengetahuan adalah :

1. Harus bisa diprogram dengan Bahasa pemrograman dan hasilnya disimpan ke dalam sebuah memori
2. Dirancang sedemikian sehingga isinya dapat digunakan untuk proses penalaran.
3. Model representasi pengetahuan merupakan sebuah struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencarian untuk aktivitas pencocokan pola.

Dalam sistem pakar ada beberapa metode representasi pengetahuan. Sistem pakar mendiagnosis penyakit menggunakan pengetahuan prosedural yang merepresentasikan aksi dan prosedur. Metode representasi yang cocok digunakan adalah kaidah produksi.

### **2.3.1. Kaidah Produksi**

Pengetahuan dalam kaidah produksi direpresentasikan dalam bentuk sebuah aturan :

JIKA <gejala> MAKA <hipotesis>

JKA <kondisi> MAKA <aksi>

JKA <premis> MAKA <konklusi>

Contoh sebuah aturan :

JIKA bersin-bersin MAKA terserang *influenza*

Aturan yang menggunakan operator logikan AND atau OR. Misalnya:

JIKA bersin-bersin DAN demam MAKA terserang *influenza*

Aturan dalam kaidah produksi diklasifikasikan menjadi kaidah derajat pertama dan kaidah meta. Kaidah derajat pertama adalah aturan yang bagian konklusinya tidak menjadi premis bagi kaidah lain. Sebaliknya kaidah meta merupakan kaidah yang konklusinya merupakan premis bagi kaidah lain. Contoh :

Aturan 1 :

JIKA Pusing

DAN Cepat Lelah

DAN Sering Kesemutan

MAKA Anemia

Aturan 2 :

JKA Anemia

DAN Batuk Kronis

MAKA TBC

Sebelum sampai pada bentuk kaidah produksi, terdapat langkah-langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut adalah menyajikan pengetahuan yang berhasil didapatkan dalam bentuk tabel keputusan *(decision table)* kemudian dari tabel keputusan dibuat pada pohon keputusan *(decision tree)*

### **2.3.2. Tabel Keputusan**

Tabel Keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matriks kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Gambar 2.4 merupakan suatu bentuk tabel keputusan.

**Aksi 1**

**Aksi 2**

|  |  |  |
| --- | --- | --- |
| Kondisi 1 | x |  |
| Kondisi 2 | x | X |
| Kondisi 3 |  | X |

Gambar 2.3. Tabel Keputusan (sumber : Hartati dan Iswanti)

Kaidah yang disajikan dalam bentuk kaidah produksi disusun dari tabel keputusan (dibentuk dari pengubahan tabel keputusan). Pembuatan suatu kaidah dilakukan dengan beberapa tahapan. Sebagai contoh perhatikan Kaidah 1. Pertama Aksi 1 merupakan konklusi dari kaidah 1. Kondisi ini akan dapat dicapai bila kondisi-kondisi yang mendukungnya terpenuhi. Kedua, tanda silang (x) pada kolom dibawah Aksi 1 menunjukkan kondisi mana yang harus dipenuhi untuk mencapai konklusi tersebut. Pada Aksi 1, terlihat tanda silang berada pada kondisi 1 dan kondisi 2. Ketiga, pembuatan kaidah 1 dan 2 menggunakan Aksi dan kondisi yang diperoleh dari langkah 1 dan 2, seperti berikut ini :

Kaidah 1: Aksi 1 IF

Kondisi 1 AND

Kondisi 2

Kaidah 2 : Aksi 2 IF

Kondisi 2 AND

Kondisi 3

Untuk mempermudah biasanya tabel keputusan digambarkan dengan pohon keputusan.

## **2.4. *Forward Chaining***

*Forward chaining* adalah metode pencarian atau penarikan kesimpulan yang berdasarkan pada fakta yang ada menuju ke kesimpulan, penelusuran dimulai dari fakta yang ada lalu bergerak maju melalui beberapa premis untuk menuju ke kesimpulan *(bottom up reasoning)*. *Forward chaining* adalah *data-driven*, karena inferensi dimulai dengan informasi atau fakta-fakta yang ada baru kesimpulan diperoleh. Dalam melakukan proses *Forward Chaining*, perlu suatu kumpulan aturan *(rule),* aturan yang ada ditelusuri satu persatu sehingga penelurusan dihentikan karena kondisi terakhir telah terpenuhi (Durkin J, 1994). *Forward chaining* memiliki aturan-aturan untuk diuji satu demi satu dalam urutan tertentu. Urutan itu berupa urutan pemasukan aturan ke dalam basis aturan atau juga aturan lain yang ditentukan oleh pemakai. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisinya benar, maka menghasilkan solusi kemudian aturan berikutnya diuji. Proses ini akan diulang sampai seluruh basis aturan teruji dengan berbagai kondisi.

Contoh :

Sistem Pakar : Penasihat Keuangan

Kasus : Seorang *user* ingin berkonsultasi apakah tepat jika dia berinvestasi pada saham IBM?

Variabel-variabel yang digunakan :

A = memiliki uang $10.000 untuk berinvestasi

B= berusia <30 tahun

C= tingkat Pendidikan pada level college

D= pendapatan minimum pertahun $40.000

E=investasi pada bidang Sekuritas(Asuransi)

F=investasi pada saham pertumbuhan (growth stock)

G=investasi pada saham IBM

Setiap variable dapat bernilai TRUE atau FALSE

Fakta:

* Diasumsikan si investor memiliki data :

1. Memiliki Uanng $10.000 (A TRUE)
2. Berusia 25 tahun (B TRUE)

Dia ingin meminta nasihat apakah tepat jika berinvestasi pada saham IBM?

*RULE*S

R1 : IF seseorang memiliki uang $10.0000 untuk berinvestasi AND dia berpendidikan pada level college THEN dia harus berinvestasi pada bidang sekuritas

R2 : IF seseorang memiliki pendapatan per tahun minimal $40.000 AND dia berpendidikan pada level college THEN dia harus berinvestasi pada saham pertumbuhan

R3 : IF seorang berusia <30 tahun AND dia berinvestasi pada bidang sekuritas THEN dia sebaiknya berinvestasi pada saham pertumbuhan

R4 : IF seseorang berusia <22 tahun THEN dia berpendidikan college

R5 : IF seseorang ingin berinvestasi pada saham pertumbuhan THEN saham yang dipilih adalah saham IBM.

*Rules simplification*

R1 : IF A AND C, THEN E

R2 : IF D AND C, THEN F

R3 : IF B AND E, THEN F

R4 : IF B, THEN C

R5: IF F then G, G->True

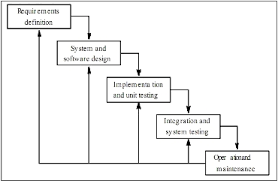
Kesimpulan : Cocok untuk investasi saham IBM

## **­­Metode Pengembangan *Waterfall***

Model proses mencakup kegiatan yang merupakan bagian dari proses perangkat lunak dan peran orang yang terlibat dalam rekayasa perangkat lunak.

Model proses *Waterfall* merupakan suatu model proses klasik yang bersifat sistematis, berurutan dari satu tahap ke tahap lain dalam membangun perangkat lunak *(Summerville, 2011).*

Model proses *Waterfall* memiliki tahapan-tahapan dalam prosesnya, setiap tahapan tersebut harus diselesaikan sebelum berlanjut ke tahap berikutnya. Berikut tahapan yang ada dalam *Waterfall* adalah (Sommerville, 2011).



Gambar 2.4. Model Proses Waterfall

Pada prinsipnya, hasil dari tahapan demi tahapan pada model proses SDLC harus memiliki dokumentasi yang jelas. Jika tahap berikutnya dimulai sebelum tahap sebelumnya selesai, maka akan terjadi berbagai kendala misalkan pada saat desain sistem, spesifikasi kebutuhan belum sepenuhnya selesai maka desain akan melenceng dari kebutuhan.

Keuntungan dari model proses SDLC adalah dokumentasi dihasilkan pada setiap tahapan, hal tersebut dapat berguna untuk model proses perangkat lunak yang lain namun masalah utama dari model proses ini adalah tidak fleksibel, pada tahap awal semua kebutuhan harus diketahui secara jelas dan rinci, model proses ini akan sulit untuk merespons perubahan dari kebutuhan pelanggan, model proses SDLC baik digunakan ketika persyaratan dipahami dengan baik oleh pengembang sistem (Summervile, 2011)

Adapun tahapan dalam pengembangan sistem adalah sebagai berikut :

### **Pendefinisian Kebutuhan**

Tahap pendefinisian kebutuhan adalah tahap untuk mempelajari dan memahami masalah yang akan dibuat perangkat lunaknya, menetapkan ranah informasi, fungsi, perilaku, unjuk kerja dan antarmuka perangkat lunak untuk didefinisikan sebagai kebutuhan perangkat lunak. Kebutuhan perangkat lunak sering diklasifikan menjadi dua yaitu kebutuhan fungsional dan non-fungsional (Summervile, 2011)

1. Kebutuhan Fungsional

Kebutuhan fungsional menyatakan layanan apa yang harus disediakan oleh sistem, bagaimana sistem harus bereaksi terhadap input tertentu dari bagaimana sistem harus berperilaku dalam situasi tertentu. Dalam beberapa kasus, kebutuhan fungsional dapat juga secara eksplisit menyatakan apa yang harus tidak dilakukan oleh sistem

1. Kebutuhan non-fungsional

Kebutuhan non-fungsional adalah batasan suatu layanan atau fungsi yang ditawarkan sistem. Di antaranya termasuk Batasan waktu, Batasan pada proses pengembangan, dan kendala yang dikarenakan oleh standar tertentu. Kebutuhan non-fungsional sering berlaku untuk sistem secara keseluruhan dibandingkan layanan atau fitur sistem secara individual.

### **Desain *Software* dan Sistem**

Desain *software* dan sistem adalah proses untuk merencanakan atau mengatur sistem yang akan dibangun menurut tahapan tertentu sebelum sistem tersebut diwujudkan atau dengan kata lain, Desain sistem adalah aktivitas untuk merancang struktur dan keterkaitan antar komponen-komponen yang sesuai dengan kriteria yang sudah ditetapkan., termasuk antarmuka dengan lingkungan operasionalnya. Sementara objek desain pada umumnya meliputi :

1. Prosedur

Rancangan yang berkaitan dengan cara bagaimana sistem nanti akan beroperasi dilihat dari sudut pandang pemakai seperti bagaimana sistem menerima masukan dan menghasilkan keluaran, termasuk interaksi dengan pemakai dan sistem lainnya. Yang kedua adalah bagaimana perilaku sistem untuk menanggapi semua masukan atau kondisi-kondisi. Yang ketiga, bagaimana sistem menyajikan informasi dari basis data atau *file* data kepada pemakai. Dan yang terakhir adalah bagaimana pendekatan yang dipilih untuk memenuhi keselamatan, keamanan, dan kerahasiaan pribadi.

1. Antarmuka

Rancangan karakteristik antarmuka dari komponen-komponen sistem misalnya modul-modul penghubung sistem dengan komponen eksternal, data antar aplikasi, media dan perangkat komunikasi

### **Implementasi Sistem**

Pada dasarnya, implementasi adalah menyalin alur data dan alur sistem ke dalam Bahasa pemrograman yang kita inginkan. Pada implementasi dilakukan dua tahap yaitu pengkodean alur sistem dan data, pengkodean tampilan. Adapun metode pengerjaannya bisa mendahulukan pengkodean sistem kemudian pengkodean tampilan atau sebaliknya atau bahkan keduanya dilakukan bersamaan tergantung perangkat lunak yang akan dibangun.

### **Integrasi dan Pengujian Sistem**

Pengujian ini dimaksud untuk menunjukkan bahwa program melakukan apa yang seharusnya dilakukannya dan menemukan cacat yang ada pada program sebelum mulai digunakan. Ketika menguji perangkat lunak, yang dilakukan adalah menjalankan sebuah program dengan menggunakan data buatan dan memeriksa hasil uji coba untuk menemukan kesalahan, anomali, atau informasi yang berkaitan dengan non fungsional program (Sommerville, 2011)

Pendekatan pengujian yang ada salah satunya adalah pengujian *blackbox.* Pengujian *blackbox* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *blackbox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi memasukkan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *blackbox* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

## **Pemrograman Web**

Dalam [rekayasa perangkat lunak](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak), suatu aplikasi web ([bahasa Inggris](https://id.wikipedia.org/wiki/Bahasa_Inggris): *web application* atau sering disingkat *webapp*) adalah suatu [aplikasi](https://id.wikipedia.org/wiki/Aplikasi) yang diakses menggunakan [penjelajah web](https://id.wikipedia.org/wiki/Penjelajah_web) melalui suatu [jaringan](https://id.wikipedia.org/wiki/Jaringan_(komputer)) seperti [Internet](https://id.wikipedia.org/wiki/Internet) atau [intranet](https://id.wikipedia.org/wiki/Intranet). Ia juga merupakan suatu aplikasi [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) komputer yang dikodekan dalam [bahasa](https://id.wikipedia.org/wiki/Bahasa_pemrograman) yang didukung penjelajah web (seperti [ASP](https://id.wikipedia.org/wiki/Active_Server_Pages_.NET), [Perl](https://id.wikipedia.org/wiki/Perl" \o "Perl), [Java](https://id.wikipedia.org/wiki/Java), [Java Script](https://id.wikipedia.org/wiki/JavaScript), [PHP](https://id.wikipedia.org/wiki/PHP), [Python](https://id.wikipedia.org/wiki/Python_(bahasa_pemrograman)" \o "Python (bahasa pemrograman)), [Ruby](https://id.wikipedia.org/wiki/Ruby_(bahasa_pemrograman)), dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu *thin client* (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya *[webmail](https://id.wikipedia.org/w/index.php?title=Webmail&action=edit&redlink=1" \o "Webmail (halaman belum tersedia))*, toko [ritel](https://id.wikipedia.org/wiki/Ritel), [lelang](https://id.wikipedia.org/w/index.php?title=Lelang_daring&action=edit&redlink=1) *online,* [wiki](https://id.wikipedia.org/wiki/Wiki" \o "Wiki), [papan diskusi](https://id.wikipedia.org/w/index.php?title=Papan_diskusi&action=edit&redlink=1), [weblog](https://id.wikipedia.org/wiki/Weblog" \o "Weblog).

Banyak keuntungan yang diberikan oleh Aplikasi berbasis Web daripada aplikasi berbasis *desktop*, sehingga aplikasi berbasis web telah diadopsi oleh perusahaan sebagai bagian dari strategi teknologi informasinya, karena beberapa alasan:

1. Akses informasi mudah,
2. Setup *server* lebih mudah
3. Informasi mudah didistribusikan
4. Bebas *platform*, informasi dapat disajikan oleh browser web pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan

## **2.7. *Framework* Yii**

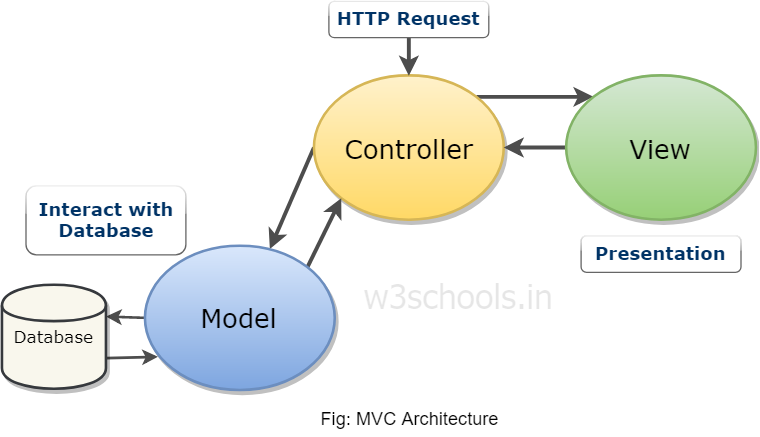
Yii adalah *framework* (kerangka kerja) PHP berbasis komponen, dan kinerja terbaik untuk pengembangan aplikasi web berskala besar. Yii menyediakan *reusability* maksimum dalam pengembangan web dan mampu meningkatkan kecepatan pengembangan secara signifikan. Nama Yii diambil dari singkatan “Yes It Is!”(Qiang Xue, 2014). Yii juga merupakan *framework* yang merespon paling tepat dan akurat. Sehubungan dengan itu, penggunaan *framework* membutuhkan pemahaman dasar pemrograman secara procedural maupun pemrograman berorientasi objek.

*Framework* Yii menggunakan dua konsep paling penting, yaitu *Object-Oriented Programming (OOP)* dan *Model-View-Controller (MVC)*

## **2.8*. Model View-Controller***

*Model View Controller (MVC)* adalah model pembuatan program yang menerapkan arsitektur yang memisahkan proses, tampilan, dan bagian yang menghubungkan antara proses dan tampilan. Pembuatan program yang menggunakan MVC ini biasanya dikemas dalam kerangka kerja *(framework),* sehingga pengembang aplikasi tinggal menggunakan kerangka kerja yang sudah disediakan.

*Framework* Yii mengimplementasikan pola desain *Model-View-Controller(MVC)*, yang diadopsi secara luas dalam pemrograman web. MVC bertujuan untuk memisahkan proses bisnis dan pertimbangan antar muka *user* agar para pengembang bisa lebih mudah mengubah setiap bagian tanpa mempengaruhi yang lain. Dalam MVC, model menggambarkan informasi (data) dan proses bisnis, *view* (tampilan) berisi elemen antarmuka *user* seperti teks, formulir masukan, sementara *controller* mengatur komunikasi antar *model* dan *view.*



Gambar 2. 5. Arsitektur Model View dan Controller

**2.9.** **Metode Certainty Factor**

Metode *certainty factor* digunakan ketika menghadapi suatu masalah yang jawabannya tidak pasti. Ketidakpastian ini bisa merupakan probabilitas. Metode ini diperkenalkan oleh Shortlife Buchanan pada tahun 1970-an. Beliau menggunakan metode ini saat melakukan diagnosis dan terapi terhadap penyakit meningitis dan infeksi darah

Tim pengembang dari metode ini mencatat bahwa, dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti “mungkin”, “hampir pasti”. Metode ini mirip dengan *fuzzy logic,* karena ketidakpastian direpresentasikan dengan derajat kepercayaan sedangkan perbedaannya adalah pada fuzzy logic saat perhitungan untuk *rule* yang premisnya lebih dari satu, fuzzy logic tidak memiliki nilai keyakinan untuk *rule* tersebut sehingga perhitungannya hanya melihat nilai terkecil untuk operator AND atau nilai terbesar untuk operator OR dari setiap premis yang pada *rule* tersebut berbeda dengan *certainty factor* yaitu setiap *rule* memiiki nilai keyakinannya sendiri tidak hanya premis-premisnya saja yang memiliki nilai keyakinan.

*Certainty factor* menunjukkan ukuran kepastian terhadap suatu fakta atau aturan . CF[h,e] = MB[h,e] – MD[h,e] …(1)

(1) Keterangan : CF[h,e] = faktor kepastian

MB[h,e] = *measure of belief*, ukuran kepercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan *evidence* (e) antara 0 dan 1

MD[h,e] = *measure of disbelief*, ukuran ketidakpercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan *evidence* (e) antara 0 dan 1.

Adapun beberapa kombinasi *certainty factor* terhadap premis tertentu:

1. *Certainty factor* dengan satu premis. CF[h,e] = CF[e] \* CF[rule]=CF[user] \* CF[pakar] …(2)

1. *Certainty factor* dengan lebih dari satu premis.

CF[A ˄ B] = Min(CF[a],CF[b]) \* CF[rule] …(3)

CF[A ˅ B] = Max(CF[a],CF[b]) \* CF[rule]…(4)

*Certainty factor* dengan kesimpulan yang serupa.

CFgabungan[CF1, CF2] = CF1 + CF2 \* (1 – CF1) …(5)

Kelebihan dari metode ini adalah cocok digunakan pada sistem pakar yang mengukur sesuatu yang pasti atau tidak pasti seperti mendiagnosis penyakit dan perhitungan dari metode ini hanya berlaku untuk sekali hitung, serta hanya dapat mengolah dua data sehingga keakuratannya terjaga (Turban E., 2001)

## **2.10 Tinjauan Studi**

**a. Sistem Pakar Pendiagnosa Penyakit Menular Pada Anjing (Kadek Risna Witari , I.G.K. Gandhiadi , I Putu Eka Nila Kencana, 2013)**

Pada jurnal ini dibahas mengenai pembuatan aplikasi android untuk Sistem Pakar Pendiagnosaan Penyakit Menular pada Anjing, pada penelitian ini menggunakan metode *certainty factor* dimana sistem akan memberi pertanyaan mengenai beberapa gejala kemudian pengguna memilih gejala dari setiap pertanyaan dan sistem akan menampilkan kemungkinan penyakit apa yang diderita oleh anjing. Namun yang kurang dari jurnal ini adalah tidak adanya halaman admin . Sehingga hanya *programmer* saja yang mengetahui atau mengubah *source code* untuk mengedit basis pengetahuan yang ada dalam sistem pakar tersebut.

**b. Sistem Pakar Diagnosa Penyakit Menular pada Anjing Menggunakan Metode Dempster Shafer (Niken Candra Ningrum, Hengky Anra, Helfi Nasution, 2016)**

Pada jurnal ini dibahas mengenai pembuatan aplikasi web untuk Sistem Pakar Diagnosa Penyakit Anjing menggunakan Metode Dempster Shafer Sistem dapat mendiagnosa penyakit menular pada anjing berdasarkan gejala-gejala yang tampak pada anjing dengan menerapkan metode Dempster Shafer dengan tingkat keakuratan sebesar 100% berdasarkan 45 data sampel kasus yang digunakan dalam menentukan nilai belief pada metode Dempster Shafer.Namun yang kurang dari jurnal ini adalah waktu penelitian yang dibutuhkan sangat lama, karena membutuhkan banyak dan bervariasi data sampel kasus yang digunakan dalam menentukan nilai belief pada perhitungan dempster shafer.

**c. Sistem Pakar: Diagnosis Penyakit Unggas Dengan Metode *Certainty Factor* (Siti Rohajawati, Rina Supriyati, 2010)**

Pada jurnal ini dibahas mengenai penyakit unggas . Pengguna akan memilih gejala-gejala yang mungkin. Kemudian akan muncul kemungkinan penyakit unggas yang diderita dan muncul juga penyakit kemungkinan yang terbesar melalui bobot *certainty factor.* Namun yang kurang dari jurnal ini adalah belum adanya halaman admin. Sehingga hanya programmer saja yang mengetahui atau mengubah source code untuk mengedit basis pengetahuan yang ada dalam sistem pakar tersebut

# **BAB III**

# **ANALISIS DAN PERANCANGAN**

## **3.1 Analisis Kebutuhan**

Pada analisis kebutuhan ini akan dijelaskan kebutuhan sistem yang terdiri dari kebutuhan pemakai, kebutuhan fungsional dan kebutuhan non fungsional.

### **Kebutuhan Pemakai**

Kebutuhan pemakai sistem pakar penyakit anjing diperoleh dari hasil observasi terhadap kebutuhan masyarakat untuk mendapatkan informasi kesehatan yang tepat dan cepat menggunakan sistem pakar. Daftar kebutuhan pemakai dapat dipaparkan dalam penjelasan berikut ini :

1. Sistem dapat berjalan pada web sehingga memudahkan pengguna dalam menggunakan sistem pakar
2. Dapat mendiagnosis penyakit pada anjing sesuai dengan penalaran pakar terhadap gejala-gejala yang dialami anjing
3. Dapat memberikan nilai keyakinan dari pakar terhadap diagnosis yang diperoleh dari penggunaan sistem pakar
4. Dapat menyimpan hasil diagnosis dari pemeriksaan gejala yang dirasakan pengguna sistem sebagai rekam medis, yang bisa dilihat kapan pun dan di mana saja.
5. Dapat memberikan informasi mengenai penyakit pada anjing

### **Kebutuhan Fungsional**

Kebutuhan fungsional sistem pakar penyakit pada anjing merupakan hasil analisis terhadap kebutuhan pemakai. Kebutuhan fungsional akan dijelaskan pada tabel berikut.

Tabel 3 1. Tabel Kebutuhan Fungsional

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kode Fungsi | Nama Fungsi | Kode Deskripsi Fungsi | Deskripsi |
| 1 | F1 | Login | F1A | Sistem dapat menyimpan data pengguna yaitu berupa *username* dan password |
|  |  |  | F1B | Sistem dapat menampilkan *username* saat telah melakukan login pada sistem |
| 2 | F2 | Diagnosis Penyakit | F2A | Sistem dapat memunculkan pertanyaan satu persatu setelah dijawab oleh pengguna |
|  |  |  | F2B | Sistem menanyakan pertanyaan sesuai dengan rule |
|  |  |  | F2C | Sistem dapat memberikan opsi jawaban yaitu ya dan tidak |
|  |  |  | F2D | Pengguna dapat memilih jawaban pertanyaan |
|  |  |  | F2E | Pengguna dapat mengulangi diagnosis jika merasakan keraguan dalam menjawab pertanyaan yang telah diajukan sistem |
|  |  |  | F2F | Sistem dapat mendiagnosis berdasarkan gejala yang dirasakan sesuai dengan rule |
|  |  |  | F2G | Sistem dapat memberikan nilai keyakinan pakar terhadap diagnosis yang diperoleh |
|  |  |  | F2H | Sistem dapat memberikan informasi mengenai diagnosis yang dihasilkan |
| 3 | F3 | Hasil Diagnosis | F3A | Sistem dapat menyimpan hasil diagnosis pengguna |
|  |  |  | F3B | Pengguna dapat melihat rekam medis dari hasil penyimpanan data hasil diagnosis |
|  |  |  | F3C | Pengguna dapat menghapus semua rekam medis yang sudah dilakukan |
| 4 | F4 | Informasi Penyakit pada Anjing | F4A | Sistem dapat memberikan pilihan jenis penyakit |
|  |  |  | F4B | Sistem dapat memperlihatkan informasi sesuai jenis penyakit yang dipilih |

## **3.2 Perancangan Sistem Pakar**

Untuk membuat sistem pakar. Ada tahapan dalam merancang sistem pakar, terdiri dari akuisisi pakar, perancangan basis pengetahuan, perancangan representasi pengetahuan, perancangan mesin inferensi dan analisis pembobotan.

### **3.2.1. Perancangan Akuisisi Pengetahuan**

Dalam penelitian menggunakan data primer dan sekunder yang telah dikumpulkan dari hasil wawancara dengan pakar. Data yang diperoleh dari pakar adalah gejala-gejala penyakit pada anjing dari pengalaman pakar selama ini menangani pasien dan juga penalaran pakar dalam mendiagnosis penyakit anjing yang sering dialami oleh anjing tersebut selama pakar mengalami pemeriksaan.

### **3.2.2. Perancangan Basis Pengetahuan**

Pengetahuan yang telah dikumpulkan akan dibangun basis pengetahuannya yang diperlukan untuk memahami, merumuskan, dan memecahkan masalah. Basis tersebut mencakup dua elemen dasar yaitu fakta dan aturan khusus yang mengarahkan pengguna pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu. Pengetahuan yang diperoleh adalah sebagai berikut :

Tabel 3 2. Tabel Penyakit pada Anjing

|  |  |
| --- | --- |
| ID Penyakit | Nama Penyakit |
| P01 | Distemper |
| P02 | Parvovirus |
| P03 | Hepatitis |
| P04 | Batuk Kennel |
| P05 | Rabies |
| P06 | Sarcoptes Sp (scabies) |
| P07 | Demodekosis |

Tabel 3 3. Tabel Gejala Penyakit

|  |  |
| --- | --- |
| ID Gejala | Gejala Penyakit |
| G01 | Disebabkan oleh virus/bakteri |
| G02 | Penyakit kulit |
| G03 | Suhu tubuh meningkat atau demam |
| G04 | Nafsu makan berkurang |
| G05 | Diare |
| G06 | Gangguan Pernafasan |
| G07 | Bantalan kaki mengalami penebalan |
| G08 | Muntah |
| G09 | Diare berdarah |
| G10 | Pemeriksaan partikel virus melalui feses anjing |
| G11 | Pendarahan mendadak |
| G12 | Timbul warna kekuningan pada kulit, mata, dan gusi |
| G13 | Batuk Kering |
| G14 | Kerusakan epitel saluran nafas anjing |
| G15 | Leleran hijau atau kuning pada hidung |
| G16 | Air Liur berlebihan |
| G17 | Bergetar |
| G18 | Kadang Agresif |
| G19 | Tidak mampu menelan makanan |
| G20 | Dilakukan tes negeri bodies pada otak anjing |
| G21 | Disebabkan oleh tungau |
| G22 | Mengalami iritasi kulit |
| G23 | Rambut rontok |
| G24 | Terdapat lesi bersisik dan berkeropeng yang berbau seperti air kencing tikus |
| G25 | Terjadi kegundulan tempat di mana tungau hidup |
| G26 | Lesi sifatnya kering dan bersisik |

### **3.2.3 Perancangan Representasi Pengetahuan**

Setelah basis pengetahuan dibangun maka dibutuhkan representasi pengetahuan untuk mempermudah prosedur pemecahan masalah dalam mengakses informasi. Dalam penelitian ini representasi basis pengetahuan yang digunakan dibentuk dalam frame dan aturan produksi yang dituliskan dalam bentuk jika-maka (IF-THEN). Pada susunan pertanyaan yang diajukan pertanyaan kepada pengguna dibentuk dengan pohon keputusan

|  |  |  |  |
| --- | --- | --- | --- |
| ID Gejala | Gejala Penyakit | Ya | Tidak |
| G01 | Apakah penyakit disebabkan oleh virus/bakteri? | 3 | 2 |
| G02 | Apakah penyakit disebakan oleh kulit? | 21 | Tidak Terdiagnosis |
| G03 | Apakah Suhu tubuh meningkat atau demam? | 4 | Tidak Terdiagnosis |
| G04 | Apakah Nafsu makan berkurang? | 5 | Tidak Terdiagnosis |
| G05 | Apakah Diare? | 6 | 11 |
| G06 | Apakah Gangguan Pernafasan? | 7 | 10 |
| G07 | Apakah Bantalan kaki mengalami penebalan? | Distemper | 8 |
| G08 | Apakah Muntah? | 9 | Tidak Terdiagnosis |
| G09 | Apakah Diare berdarah? | 10 | Tidak Terdiagnosis |
| G10 | Apakah Pemeriksaan partikel virus melalui feses anjing? | Parvovirus | Tidak Terdiagnosis |
| G11 | Apakah Pendarahan mendadak? | 12 | 16 |
| G12 | Timbul warna kekuningan pada kulit, mata, dan gusi | Hepatitis | 13 |
| G13 | Batuk Kering | 14 | Tidak Terdiagnosis |
| G14 | Kerusakan epitel saluran nafas anjing | 15 | Tidak Terdiagnosis |
| G15 | Leleran hijau atau kuning pada hidung | Batuk Kennel | Tidak Terdiagnosis |
| G16 | Air Liur berlebihan | 17 | Tidak Terdiagnosis |
| G17 | Bergetar | 18 | Tidak Terdiagnosis |
| G18 | Kadang Agresif | 19 | Tidak Terdiagnosis |
| G19 | Tidak mampu menelan makanan | 20 | Tidak Terdiagnosis |
| G20 | Dilakukan tes negri bodies pada otak anjing | Rabies | Tidak Terdiagnosis |
| G21 | Disebabkan oleh tungau | 22 | Tidak Terdiagnosis |
| G22 | Mengalami iritasi kulit | 23 | Tidak Terdiagnosis |
| G23 | Rambut rontok | 24 | Tidak Terdiagnosis |
| G24 | Terdapat lesi bersisik dan berkeropeng yang berbau seperti air kencing tikus | Skabies | 25 |
| G25 | Terjadi kegundulan tempat dimana tungau hidup | 26 | Tidak Terdiagnosis |
| G26 | Lesi sifatnya kering dan bersisik | Demodekosis | Tidak Terdiagnosis |

### **3.2.4. Perancangan Mesin Inferensi**

Setelah membuat representasi pengetahuan maka dibutuhkan mesin inferensi yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada untuk mencapai suatu kesimpulan atau solusi. Pada penelitian ini menggunakan teknik penalaran maju atau dengan kata lain yaitu Forward Chaining yang berguna untuk mengontrol inferensi atau alur dari pertanyaan yang diajukan oleh system yang diberikan kepada pengguna, sehingga pertanyaan yang diberikan sesuai dengan alur untuk mencapai kesimpulan atau solusi. Pertanyaan pertama yang diberikan adalah pertanyaan dengan kode G01 kemudian system akan menelusuri pertanyaan demi pertanyaan sampai kesimpulan sesuai dengan jawaban yang diberikan oleh pengguna.

### **3.2.5. Analisis Pembobotan**

Setelah mendapatkan aturan penyakit pada anjing menggunakan wawancara dengan pakar maka pakar juga menentukan keyakinan terhadap gejala penyakit anjing. Adapun bobot masing-masing gejala adalah sebagai berikut :

|  |  |  |
| --- | --- | --- |
| ID Gejala | Gejala Penyakit | Bobot |
| G01 | Disebabkan oleh virus/bakteri | 0,5 |
| G02 | Penyakit kulit | 0,5 |
| G03 | Suhu tubuh meningkat atau demam | 0,75 |
| G04 | Nafsu makan berkurang | 0,75 |
| G05 | Diare | 0,75 |
| G06 | Gangguan Pernafasan | 0,75 |
| G07 | Bantalan kaki mengalami penebalan | 0,9 |
| G08 | Muntah | 0,75 |
| G09 | Diare berdarah | 0,75 |
| G10 | Pemeriksaan partikel virus melalui feses anjing | 0,9 |
| G11 | Pendarahan mendadak | 0,75 |
| G12 | Timbul warna kekuningan pada kulit, mata, dan gusi | 0,9 |
| G13 | Batuk Kering | 0,75 |
| G14 | Kerusakan epitel saluran nafas anjing | 0,9 |
| G15 | Leleran hijau atau kuning pada hidung | 0,75 |
| G16 | Air Liur berlebihan | 0,75 |
| G17 | Bergetar | 0,75 |
| G18 | Kadang Agresif | 0,75 |
| G19 | Tidak mampu menelan makanan | 0,75 |
| G20 | Dilakukan tes negeri bodies pada otak anjing | 0,9 |
| G21 | Disebabkan oleh tungau | 0,75 |
| G22 | Mengalami iritasi kulit | 0,75 |
| G23 | Rambut rontok | 0,75 |
| G24 | Terdapat lesi bersisik dan berkeropeng yang berbau seperti air kencing tikus | 0,9 |
| G25 | Terjadi kegundulan tempat dimana tungau hidup | 0,75 |
| G26 | Lesi sifatnya kering dan bersisik | 0,9 |

### **3.2.6. Workplace**

Pada bagian perancangan system pakar workplace merupakan bagian dari sistem pakar yang berfungsi sebagai halaman pada aplikasi yang bertujuan untuk melakukan proses-proses yang ada dan dijalankan sesuai dengan pilihan pengguna, Dimana pada aplikasi ini proses yang menjadi workplace sebagai contoh adalah proses diagnosis, pada aplikasi terdapat halaman diagnosis yang bertujuan agar pengguna dapat memeriksa gejala yang dialaminya dan menjawab pertanyaan yang diajukan oleh system sehingga mendapat hasil diagnosis.

Pada bagian workplace diagnosis, terdapat bagian pertanyaan dan bagian jawaban, dimana pada bagian jawaban terdapat jawaban yes dan no, dimana pengguna dapat memilih salah satu dari jawaban tersebut.

### **3.2.7. Perbaikan Pengetahuan**

Pada bagian perbaikan pengetahuan atau pembaruan pengetahuan, pada aplikasi ini terdapat fasilitas *login* sebagai pakar, yang di mana pada *login* pakar dapat mengedit pertanyaan, mengedit bobot, mengedit *rule* yang sudah tersedia pada *database*, sehingga jika dilakukan sebuah pembaharuan, pertanyaan bobot dan *rule* dapat diperbarui secara sekaligus untuk aplikasi sistem pakar yang sudah digunakan.

## **3.3. Desain Sistem**

Perancangan dari sistem pakar penyakit anjing ini menggunakan Use Case Diagram, *Class Diagram* dan Alur Program.

### **3.3.1. *Use Case Diagram***

*Use Case Diagram* adalah pemodelan untuk menggambarkan kelakukan sistem yang akan dibuat. *Use Case Diagram* menggambarkan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. Secara sederhana, diagram *Use Case* digunakan untuk memahami fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.



Gambar 3.1 Use Case Diagram

Keterangan :

<<extend>. Relasi *use case* tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri meski tanpa use case tambahan itu

<<include>> . Relasi *use case* tambahan ke sebuah use case dimana use case yang ditambahkan membutuhkan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

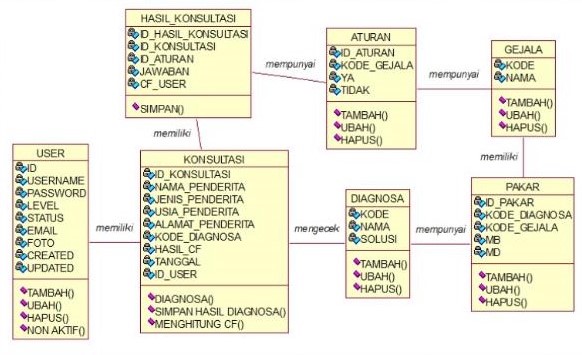
Tabel 3.1. Use Case Description

|  |  |  |
| --- | --- | --- |
| No | *User* | Deskripsi |
| 1 | Pakar | 1. Dapat melakukan login  2. Dapat memasukkan data pakar |
| 2 | Petugas | 1. Dapat Melakukan Login  2. Dapat memasukkan data gejala  3. Dapat melihat hasil diagnosis  4. Dapat melakukan konsultasi  5. Dapat melihat hasil diagnosis  6. Dapat mencetak hasil diagnosis |

### **3.3.2. *Class Diagram***

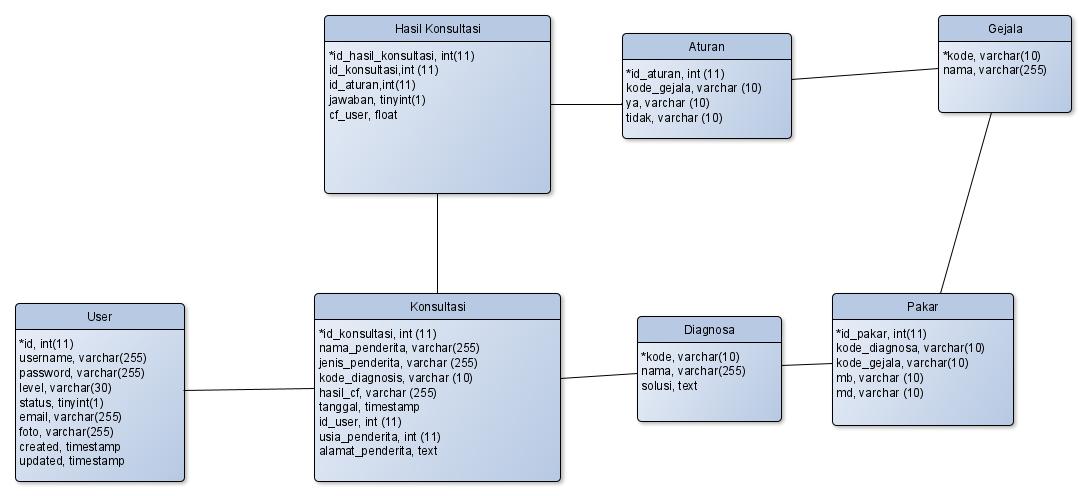
Berikut adalah *class diagram* menunjukkan model statis yang menggambarkan struktur dan deskripsi kelas serta hubungannya antara class. Class-diagram terdapat operasi/metode , bukan hanya atribut. Class terdiri dari nama kelas, atribut dan operasi/metode.

*Class diagram* ini terdapat 7 entitas diantaranya user, hasil konsultasi, konsultasi, aturan, gejala, pakar, dan diagnosis. Adapun lebih jelasnya dapat dilihat pada gambar berikut .



Gambar 3.2. Class Diagram

### **3.3.3. *Physical Data Model***



Gambar 3.3. Physical Data Model

Gambar 3.3. menunjukkan *physical data model* dari sistem pakar penyakit anjing, physical data model merupakan representasi dari database yang akan dibuat, pada setiap entitas telah dilengkapi dengan atribut dan juga tipe atribut dari setiap entitas.

### **3.3.4.** **Alur Program Konsultasi**

Adapun alur program Sistem Pakar Penyakit Anjing adalah sebagai berikut :

1. Petugas memilih menu konsultasi atau menekan tombol mulai konsultasi pada *dashboard*
2. Petugas menulis data yang diperlukan seperti nama pelanggan, jenis anjing, Usia anjing, dan alamat
3. Petugas menjawab setiap pertanyaan dari sistem kemudian sistem akan menemukan anjing, sistem akan mengkaji kira-kira anjing terkena penyakit apa beserta solusi yang diberikan.

Adapun tugas dari pengguna yaitu petugas dan pakar adalah sebagai berikut:

1. Petugas

Adapun tugas dari petugas antara lain :

1. Memanajemen diagnosa
2. Memanajemen gejala
3. Memanajemen *user*
4. Memanajemen konsultasi beserta hasilnya

2. Pakar

Adapun tugas dari pakar antara lain :

1. Memanajemen pengetahuan (nilai mb dan md setiap gejala terhadap diagnosa yang terjadi

# **BAB IV**

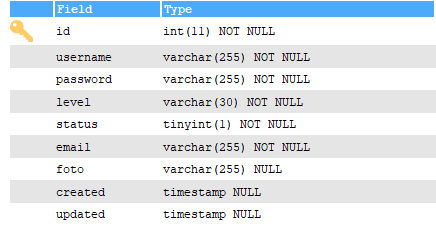
# **HASIL DAN PEMBAHASAN**

## **4.1 Implementasi Sistem**

Tahap implementasi merupakan penerjemahan desain dalam bahasa yang dimengerti oleh komputer. Dalam pengembangan aplikasi ini menggunakan IDE Visual Studio Code, pada sistem ini akan berjalan pada platform website. Pada sistem ini akan menggunakan bahasa pemrograman PHP, yang berfungsi untuk mengirimkan data atau memproses data yang diperlukan sistem. Database yang disimpan pada MySQL, yang diakses oleh PHP dan ditampilkan oleh sistem web. Pada penelitian ini akan dibagi menjadi 7, yaitu implementasi database, implementasi login, implementasi daftar diagnose, implementasi daftar gejala, implementasi halaman awal, implementasi diagnosis, dan implementasi hasil diagnosis.

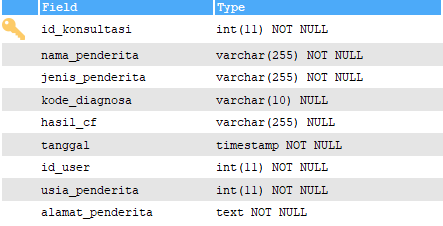
### **4.1.1****. Implementasi Database**

Database yang diperlukan oleh sistem ini guna menyimpan secara permanen data yang diperlukan sistem. Pada implementasi terdapat 7 buah tabel yang saling berhubungan sesuai dengan perancangan database sebelumnya.



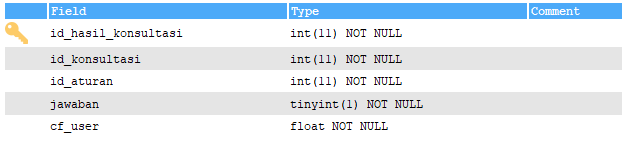
Gambar 4.1. Tabel User

Pada gambar 4.1. menunjukkan tabel user dari database sistem pakar, tabel user berguna untuk menyimpan *username* , password, level, status, email, foto, dan created dan update data oleh petugas.



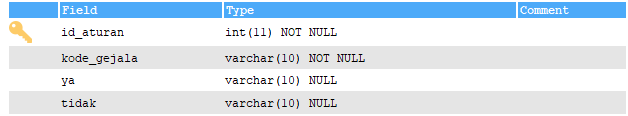
Gambar 4.2. Tabel Konsultasi

Pada gambar 4.2. menunjukkan tabel konsultasi. Tabel konsultasi berguna untuk menyimpan id konsultasi, nama penderita, jenis penderita, kode diagnosis, hasil cf, tanggal, id user, usia penderita, dan alamat penderita.



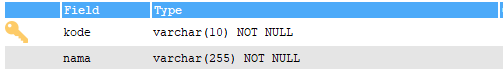
Gambar 4.3. Tabel Hasil Konsultasi

Pada gambar 4.3. menunjukkan tabel hasil konsultasi. Tabel hasil konsultasi berguna untuk menyimpan id hasil konsultasi, id konsultasi, id aturan, jawaban, dan cf user. Dapat dikatakan id hasil konsultasi berguna untuk mendapatkan rekam jejak setiap jawaban dari gejala yang diberikan pada setiap konsultasi.



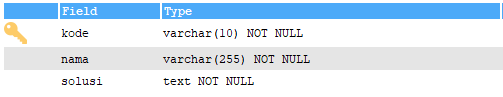
Gambar 4.4. Tabel Aturan

Pada gambar 4.4. menunjukkan tabel aturan yang berfungsi menyimpan id aturan, kode gejala, ya dan tidak. Pada tabel ini berfungsi untuk menyimpan setiap aturan yang ada pada manajemen aturan pada petugas.



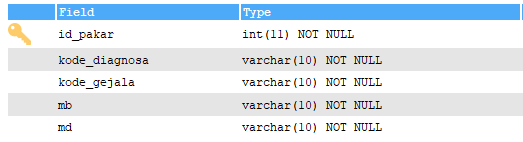
Gambar 4 5. Tabel Gejala

Pada gambar 4.5. menunjukkan tabel gejala yang berfungsi menyimpan gejala yang ada pada sistem. Pada tabel ini terdapat kode dan nama gejala



Gambar 4 6. Tabel Diagnosa

Pada gambar 4.6. menunjukkan tabel diagnosa yang berfungsi menyimpan hasil diagnosis berupa kode diagnosa, nama dan solusi yang didapatkan dari buku dan pakar.



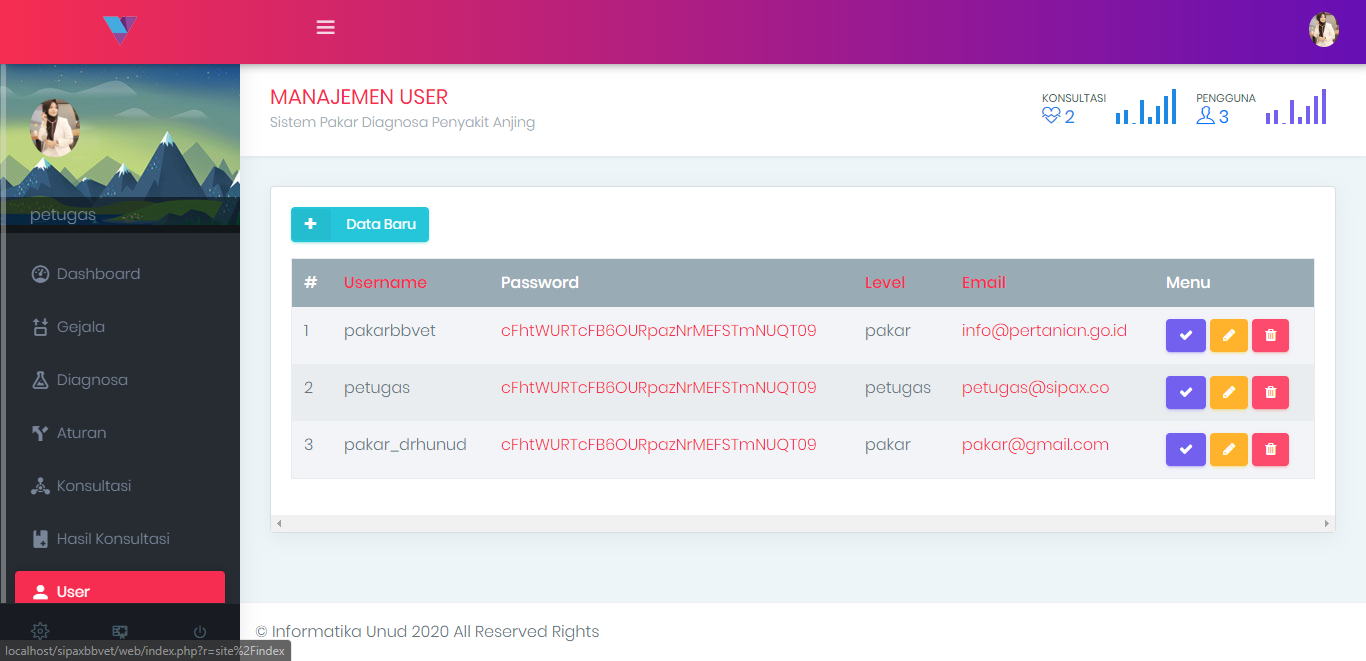
Gambar 4 7. Tabel Pakar

Tabel pakar bertugas untuk menghimpun data pakar berupa id pakar, kode diagnose, kode gejala, mb dan md dari pakar. Pakar akan memberikan mb dan md untuk setiap kasus gejala dan diagnosa yang terjadi pada sistem.

### **4.1.2. Implementasi *Login* dan Manajemen *User***

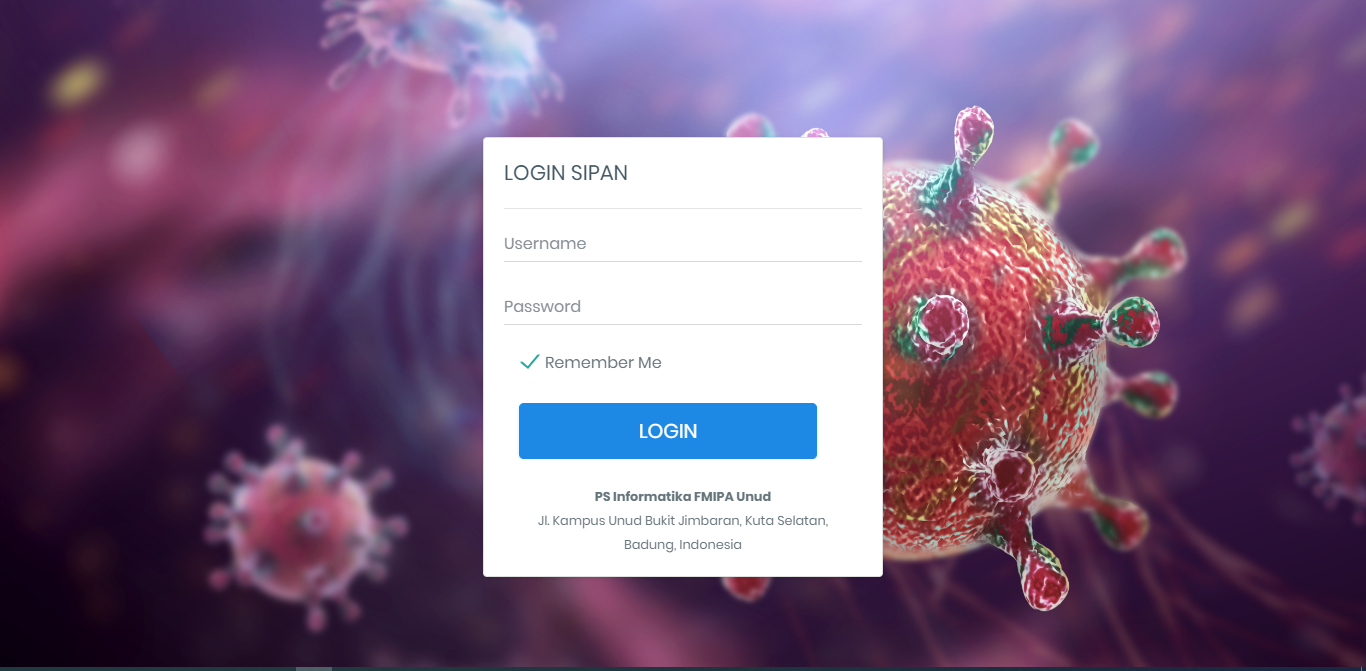
Pada Sistem Pakar Penyakit Anjing halaman yang akan pertama kali dilihat oleh *user* adalah halaman login, pada halaman login *user* akan diminta untuk memasukkan *username* dan password yang sudah dimiliki oleh *user*, jika user belum memiliki *username* dan password, petugas dapat menambahkan *username* dan password, ada 2 level *user* yang dapat ditambah yaitu petugas dan pakar.

Tahapan pada implementasi login adalah pertama mengecek *user* sudah memasukkan data dengan benar, yaitu data *username* dan data password kemudian pengecekan ke dalam database, jika inputan *username* dan *password* oleh *user* salah maka user tidak dapat melakukan *login*, jiks inputan salah satu salah, antara *username* dan password maka user juga tidak dapat melakukan *login*. Pada implementasi *login user* memiiki pilihan untuk menjadi petugas atau pakar.



Gambar 4 8. Manajemen User

Jadi untuk *login* ke dalam sistem, *username* dan *password* harus sudah didaftarkan terlebih dahulu oleh petugas. Petugas dapat menambahkan *username*, *password*, level dan email user. Nanti yang dipakai untuk *login* hanyalah *username* dan *password* saja.



Gambar 4.9 Halaman Login

Pada gambar 4.9 menunjukkan antarmuka *login*, *user* menginputkan *username* dan *password*, *user* akan mendapatkan akses *login* sesuai dengan *rule based access* yang diberikan oleh petugas.

### **4.1.3.** **Implementasi Manajemen Diagnosa**

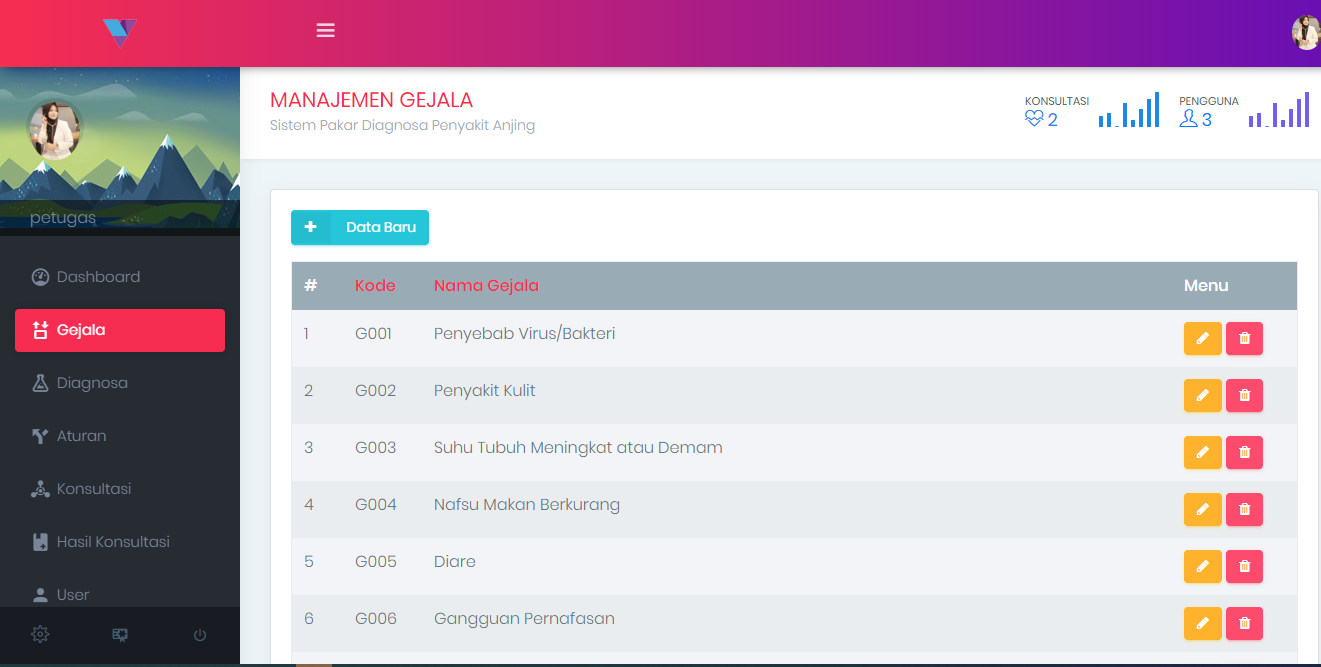
Untuk mendata penyakit yang ada menggunakan manajemen diagnosa. Pada manajemen diagnosa terdapat kode diagnosa, nama diagnosa dan solusi dari penyakit yang diderita. Petugas dapat menambahkan, mengubah dan menghapus daftar diagnosa. Jika ditemukan suatu penyakit baru, maka halaman ini yang pertama kali untuk ditambah, baru kemudian halaman gejala dan aturan



Gambar 4 10. Halaman Manajemen Diagnosa

### **4.1.4. Implementasi Manajemen Gejala**

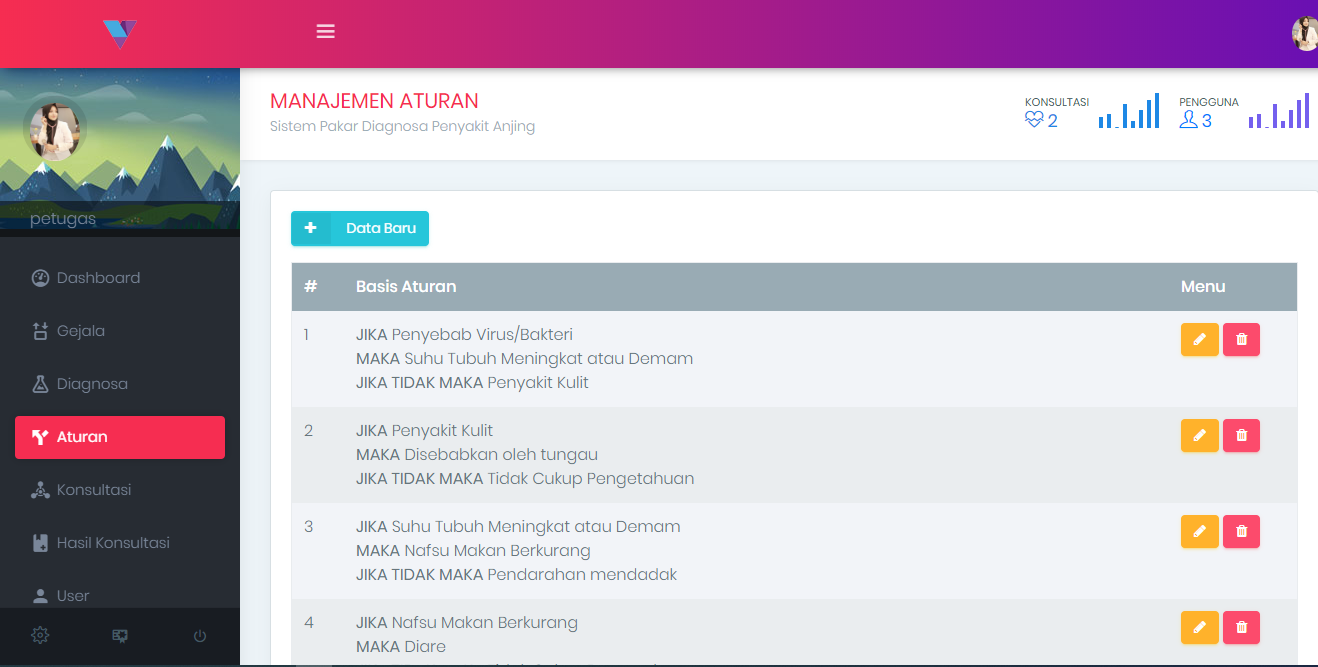
Untuk mendata daftar gejala dilakukan pada halaman manajemen gejala. Pada manajemen gejala terdapat kode dan nama gejala. Petugas dapat menambahkan, mengedit dan/atau menghapus gejala yang terjadi.



Gambar 4.11. Implementasi Manajemen Gejala

### **4.1.5. Implementasi Manajemen Aturan**

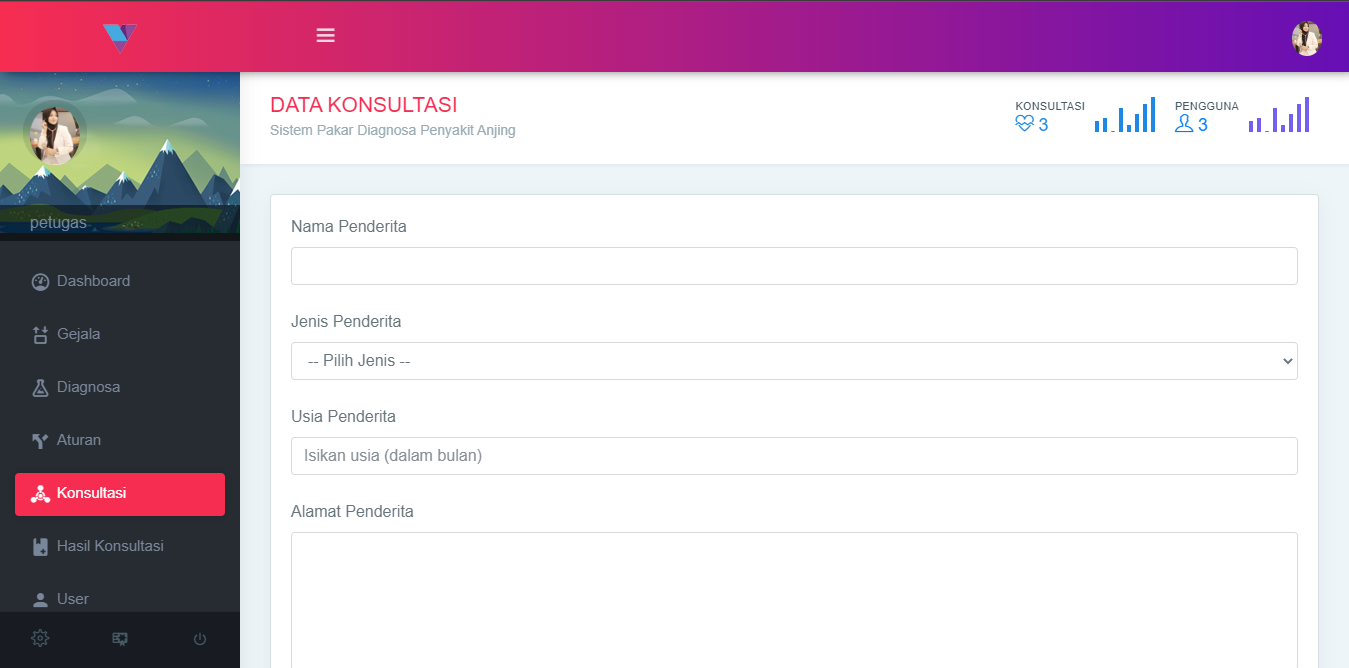
Pada manajemen aturan terdapat aturan berupa pertanyaan jika ya maka ke pertanyaan selanjutnya dan jika tidak maka ke pertanyaan selanjutnya. Manajemen aturan ini dapat ditambah, diubah dan dihapus tergantung dari perkembangan aturan yang ada.



Gambar 4.12. Implementasi Manajemen Aturan

### **4.1.6. Implementasi Konsultasi**

Pada halaman manajemen diagnosa akan dimasukkan data nama penderita, jenis anjing. usia anjing dan alamat anjing yang akan dilakukan konsultasi. Pada konsultasi sistem akan memberikan pertanyaan demi pertanyaan menuju gejala yang diderita oleh pengguna.



Gambar 4.13. Implementasi Manajemen Konsultasi

Seperti pada gambar 4.13 Pada manajemen konsultasi, petugas mendata nama penderita, jenis anjing, usia anjing, dan alamat pemilik sebelum melakukan konsultasi. Seluruh Data Konsultasi nantinya dapat dilihat pada halaman hasil konsultasi.

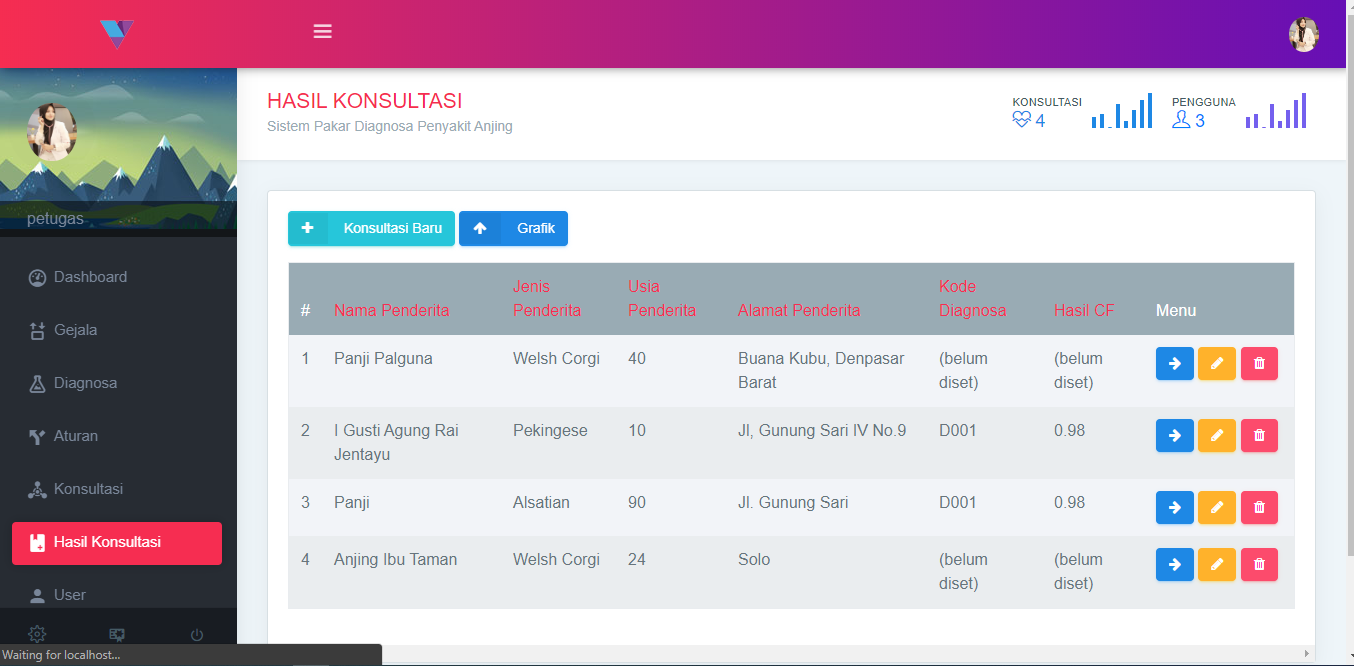


Gambar 4.14 Implementasi Manajemen Konsultasi

Seperti pada gambar 4.14 implementasi manajemen konsultasi terdapat pertanyaan-pertanyaan yang harus dijawab oleh pengguna untuk mendapatkan penyakit yang diderita anjing dengan memberikan jawaban ya atau tidak. Jika memberi jawaban Ya petugas wajib memasukkan nilai CF untuk mendapatkan nilai keyakinan pada hasil diagnosis.

### **4.1.7. Implementasi Manajemen Hasil Konsultasi**

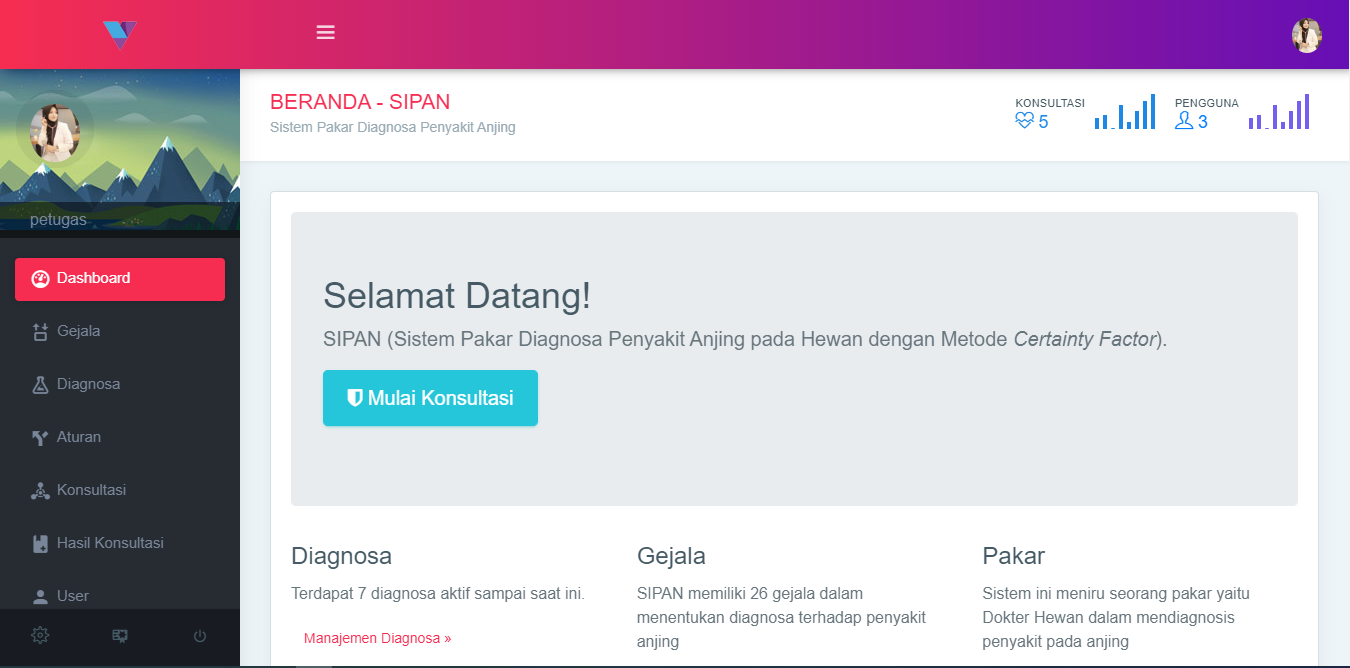
Pada halaman implementasi manajemen hasil konsultasi, terdapat nama penderita, jenis anjing, usia anjing, alamat anjing, kode diagnosa dan hasil CF.



Gambar 4.15. Implementasi Manajemen Hasil Konsultasi

Pada gambar 4.15 Implementasi Manajemen Hasil Konsultasi tercatat Nama Pemilik, Jenis Anjing, Alamat Pemilik, Kode Diagnosa, dan Hasil CF pada hasil konsultasi. Pada halaman hasil konsultasi dapat juga melihat grafik diagnosa yang terjadi melibatkan sistem ini.

### **4.1.8. Implementasi Beranda/*Dashboard***

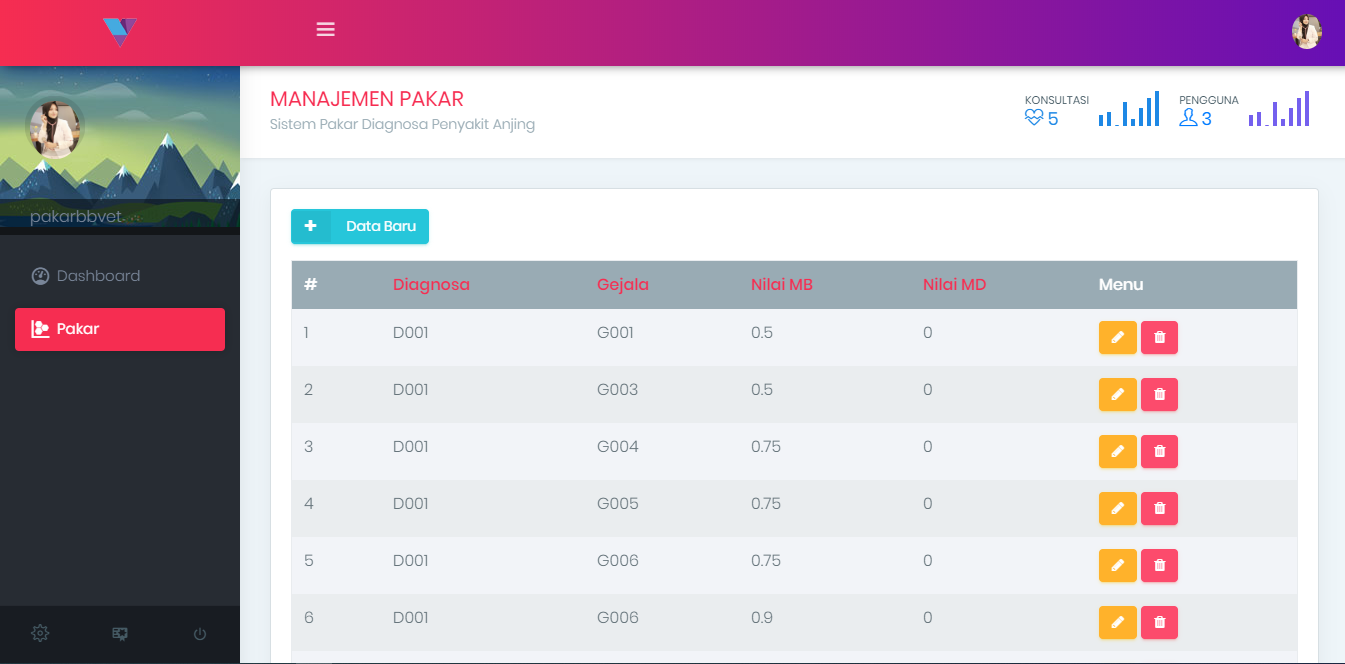


Gambar 4 16. Implementasi Beranda

Seperti pada gambar 4.16. beranda ditampilkan Halaman Selamat Datang terdapat juga jumlah konsultasi dan jumlah pengguna. Serta deskripsi aplikasi web sistem pakar. Untuk memulai konsultasi bisa juga dengan mengeklik tombol mulai konsultasi pada halaman beranda.

### **4.1.9. Implementasi Pengetahuan Pakar**

Pada halaman manajemen pakar, terdapat kode diagnosa, gejala, nilai *measure of believe (mb)* dan *measure of disbelieve(md)*. Pada halaman ini mendata seluruh kemungkinan nilai CF yang terjadi dari setiap pertanyaan yang diberikan.. Pakar dapat mengubah, mengedit atau menghapus pengetahuan yang ada dalam manajemen pakar. Untuk lebih jelasnya, dapat dilihat pada gambar 4.17.



Gambar 4.17. Implementasi Halaman Manajemen Pakar

### **4.1.10. Implementasi Model-View Controller dalam Yii *Framework***

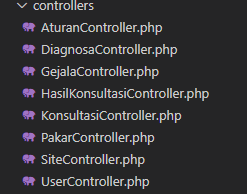
Model merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. Model adalah suatu objek yang tidak mengandung informasi tentang *user interface*. Model juga suatu kelas yang berisi metode/fungsi yang merupakan proses-proses. Model juga sebuah *instance* dari model atau sebuah *class* yang menurunkan model. Model digunakan untuk menyimpan data dan proses bisnis.

View merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam antarmuka (semua hal *user* dapat melihat dan merespons pada layar, seperti tombol, tampilan kotak, dan sebagainya).

*Controller* merupakan kelas yang menghubungkan antara model dan *view,* dan digunakan untuk komunikasi antara kelas dalam *model* dan *view.*

Penerapan MVC dalam sistem ini, dapat dijelaskan sebagai berikut :

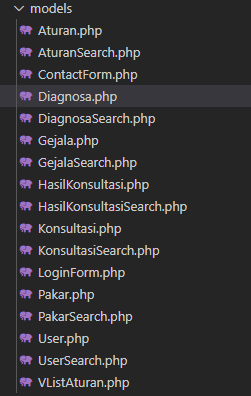
1. *Controller*



Gambar 4.18. Controller dalam Yii Framework

Karena *controller* memiliki fungsi penghubung antara model dan *view,* maka dapat dikatakan fungsi *controller* sebagai *filter* dan *action*. Serta melakukan permintaan ke model dan menerima data dari *model.*

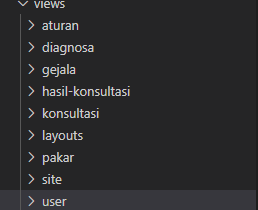
2. *Model*



Gambar 4.19. Model dalam Yii Framework

Model dalam Yii *Framework* mendasari logika dan proses dalam aplikasi. Dapat dikatakan suatu kelas yang berisi metode/fungsi yang merupakan proses-proses.

3. *Views*



Gambar 4.20. Views dalam Aplikasi

Unsur-unsur dalam antarmuka (semua hal *user* dapat melihat dan merespons pada layar, seperti tombol, tampilan kotak, dan sebagainya) ada dalam *views.* Terdapat Halaman Antarmuka aturan, diagnosa, gejala, hasil-konsultasi, *layouts,* pakar, *site* dan *user.*

### **4.1.11. Proses Perhitungan CF dalam Sistem**

Secara sederhananya untuk perhitungan *Certainty Factor* dapat digambarkan sebagai berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| KODE GEJALA | NILAI CF | | CF(H,E) |
| PAKAR | USER |
| G002 | 0.2 | 0.7 | 0.14 |
| G021 | 1 | 0.8 | 0.8 |

CFcombine(CF1, CF2) = CF1+CF2\*(1-CF1)

CF combine (CF1, CF2) = 0.14+0.8(1-0.14)

= 0.14+0.688

CF Combine = 0.828

Persentase keyakinan =

CFcombine x 100 % = 0.83 x 100%

= 83%

public function actionCf()

{

$s\_konsultasi=Yii::$app->session['id\_konsultasi'];

IF(!empty($s\_konsultasi)){

$kode\_diagnosa=Konsultasi::find()->where('id\_konsultasi = :id\_konsultasi',

[':id\_konsultasi' => $s\_konsultasi])->one()->kode\_diagnosa;

//Array CF PAKAR

$cf\_pakar=array();

$pakar = Pakar::find()

->where('kode\_diagnosa = :kode\_diagnosa', [':kode\_diagnosa' => $kode\_diagnosa])

->orderBy('id\_pakar')

->all();

foreach($pakar as $a):

$cf\_pakar[]=$a->mb - $a->md;

endforeach;

//Array CF USER

$cf\_user=array();

$konsul = HasilKonsultasi::find()

->where('id\_konsultasi = :s\_konsultasi', [':s\_konsultasi' => $s\_konsultasi])

->andWhere('jawaban = true')

->orderBy('id\_aturan')

->all();

foreach($konsul as $b):

$cf\_user[]=$b->cf\_user;

endforeach;

$i=count($cf\_pakar);

$x=0;

$cf = 0;

$baru=array();

if(count($cf\_pakar)==count($cf\_user)){

while($x < $i){

$lama = $cf;

$baru = $cf\_pakar[$x]\*$cf\_user[$x];

$cf = $lama + ($baru \* (1 - $lama));

$x++;

}

$cf\_akhir=number\_format($cf, 2);

$model=Konsultasi::find()->where('id\_konsultasi = :id\_konsultasi', [':id\_konsultasi' => $s\_konsultasi])->one();

$model->hasil\_cf=$cf\_akhir;

if($model->save()){

return $this->redirect(['/site/no-pengetahuan', 'end'=>1]);

}

}else

echo "error";

}

}

**Gambar 4.21. Source Code Proses Perhitungan Certainty Factor dalam Sistem**

}

## **4.2. Pengujian Sistem**

Pengujian yang dilakukan pada sistem ini adalah black box testing, dan pengujian secara metode.

### **4.2.1. *Black Box Testing***

*Blackbox Testing* merupakan pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada *output* yang dihasilkan yang merespons *input* yang dipilih dan kondisi eksekusi. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

Tabel 4.1. Blackbox Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Login | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Memasukkan *username* dan *password* yang sesuai | Dapat masuk ke halaman utama sistem | Dapat masuk ke halaman utama sistem | Valid |
| 2 | Memasukkan *username* dan *password* tidak sesuai | Tidak dapat memasuki halaman utama dan menampilkan pesan kesalahan | Tidak dapat memasuki halaman utama dan menampilkan pesan kesalahan | Valid |
| 3 | Klik *button login* tanpa memasukkan *username* dan *password* | Tidak dapat memasuki halaman utama dan menampilkan pesan kesalahan | Tidak dapat memasuki halaman utama dan menampilkan pesan kesalahan | Valid |

Tabel 4.2. Blackbox Manajemen Gejala

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Manajemen Gejala | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Melihat data gejala | Menampilkan semua data gejala penyakit pada anjing | Menampilkan semua data gejala penyakit pada anjing | Valid |
| 2 | Menambah data gejala | Dapat menambahkan data gejala pada database | Dapat menambahkan data gejala pada database | Valid |
| 3 | Mengubah data gejala | Dapat mengubah data gejala dan menyimpan data perubahan pada database | Dapat mengubah data gejala dan menyimpan data perubahan pada database | Valid |
| 4 | Menghapus data gejala | Dapat menghapus data gejala di database | Dapat menghapus data gejala di database | Valid |

Tabel 4.3 Blackbox Manajemen Pengguna

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Manajemen Pengguna | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Mengaktifkan pengguna | *Username* dan *password* yang benar bisa masuk ke halaman *dashboard* sesuai level masing-masing (petugas/pakar) | *Username* dan *password* yang benar bisa masuk ke halaman *dashboard* sesuai level masing-masing (petugas/pakar) | Valid |
| 2 | Menonaktifkan pengguna | *Username* dan *password* yang salah, tidak bisa masuk ke halaman *dashboard* sesuai level masing-masing (petugas/pakar) | *Username* dan *password* yang salah, tidak bisa masuk ke halaman *dashboard* sesuai level masing-masing (petugas/pakar) | Valid |
| 3 | Mengubah *username, email, level* dan *password* pada pengguna | *Username*, *Email. Level*, dan *password* yang sudah diubah bisa masuk ke dalam sistem | *Username*, *Email. Level,* dan *password* yang sudah diubah bisa masuk ke dalam sistem | Valid |
| 4 | Menghapus pengguna | *Username* dan *password* yang sudah dihapus tidak bisa masuk ke dalam sistem | *Username* dan *password* yang sudah dihapus tidak bisa masuk ke dalam sistem | Valid |

Tabel 4.5. Blackbox Manajemen Diagnosa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Manajemen Diagnosa | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Melihat data diagnosa | Menampilkan semua data diagnosa | Menampilkan semua data diagnosa | Valid |
| 2 | Menambah data diagnosa | Dapat menambahkan data diagnosa pada *database* | Dapat menambahkan data gejala pada *database* | Valid |
| 3 | Mengubah data diagnosa | Dapat mengubah data diagnosa dan menyimpan data perubahan pada *database* | Dapat mengubah data diagnosa dan menyimpan data perubahan pada *database* | Valid |
| 4 | Menghapus data diagnosa | Dapat menghapus data diagnosa di *database* | Dapat menghapus data diagnosa di *database* | Valid |

Tabel 4.6. Blackbox Manajemen Aturan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Manajemen Aturan | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Melihat data aturan | Menampilkan semua data aturan | Menampilkan semua data aturan | Valid |
| 2 | Menambah data aturan | Dapat menambahkan data aturan pada *database* | Dapat menambahkan data aturan pada *database* | Valid |
| 3 | Mengubah data aturan | Dapat mengubah data aturan dan menyimpan data perubahan pada *database* | Dapat mengubah data aturan dan menyimpan data perubahan pada *database* | Valid |
| 4 | Menghapus data aturan | Dapat menghapus data aturan di *database* | Dapat menghapus data aturan di *database* | Valid |

Tabel 4.7 Blackbox Manajemen Hasil Konsultasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Manajemen Hasil Konsultasi | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Melihat data hasil konsultasi | Menampilkan semua data hasil konsultasi | Menampilkan semua data hasil konsultasi | Valid |
| 2 | Menambah data hasil konsultasi | Dapat menambahkan data hasil konsultasi pada *database* | Dapat menambahkan data hasil konsultasi pada *database* | Valid |
| 3 | Mengubah data hasil konsultasi | Dapat mengubah data hasil konsultasi dan menyimpan data perubahan pada *database* | Dapat mengubah data hasil konsultasi dan menyimpan data perubahan pada *database* | Valid |
| 4 | Menghapus data hasil konsultasi | Dapat menghapus data hasil konsultasi di *database* | Dapat menghapus data hasil konsultasi di *database* | Valid |

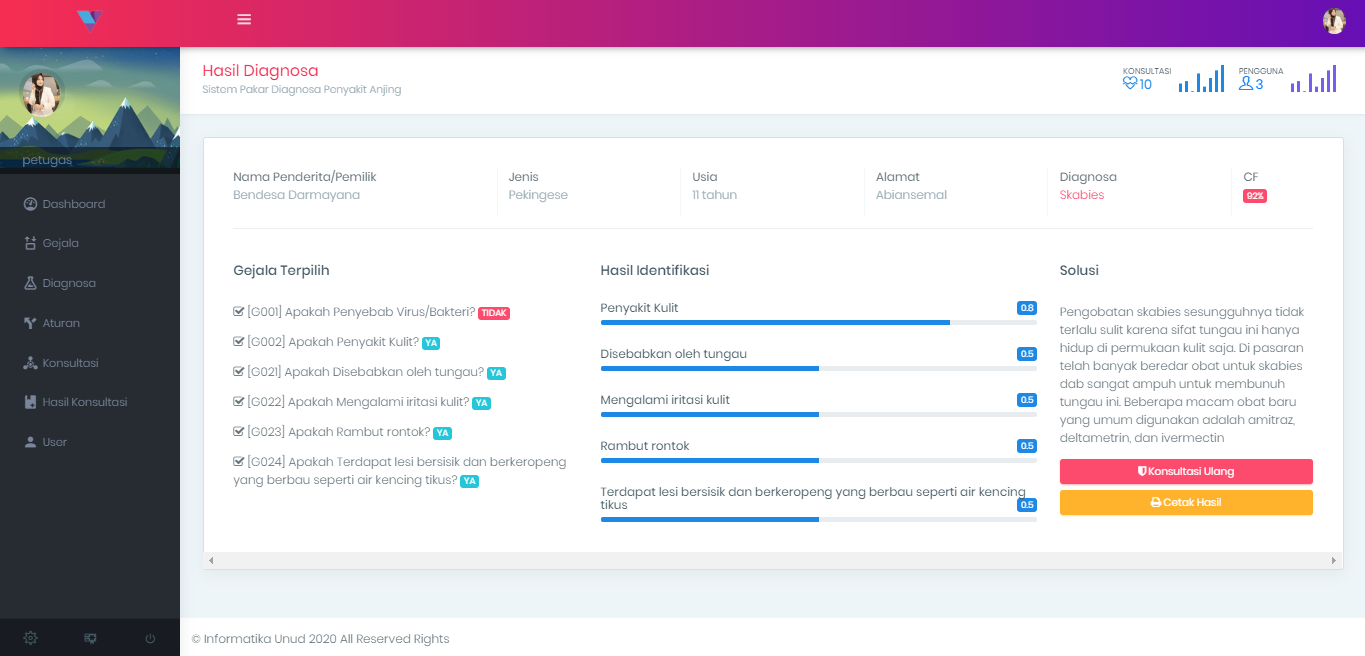
Tabel 4.8. *Blackbox* Konsultasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pengujian : Konsultasi | | | | |
| No | Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| 1 | Pertanyaan sesuai aturan | Mencoba berbagai kemungkinan jawaban yang mengarah ke hasil diagnosa dan hasil benar | Mencoba berbagai kemungkinan jawaban yang mengarah ke hasil diagnosa dan hasil benar | Valid |
| 2 | Perhitungan *Certainty Factor* sudah sesuai teori | Mencoba berbagai kemungkinan jawaban dengan memasukkan nilai CF *User* dan membandingkan dengan perhitungan manual dan memiliki hasil benar | Mencoba berbagai kemungkinan jawaban dengan memasukkan nilai CF *User* dan membandingkan dengan perhitungan manual dan memiliki hasil benar | Valid |
| 3 | Jawaban sesuai aturan | Hasil diagnosa sesuai dengan aturan | Hasil diagnosa sesuai dengan aturan | Valid |

### **4.2.2.** **Pengujian Metode *Certainty Factor***

Kasus : Penyakit Demodekosis

Misal penyakit demodekosis, petugas menulis bobot yang berbeda . Berikut bobot yang diberikan oleh petugas.



Gambar 4.22. Contoh Hasil Kobsultasi

Penyakit kulit = 0.8

Disebabkan oleh tungau = 0.5

Mengalami iritasi kulit = 0.5

Rambut rontok = 0.5

Terdapat lesi bersisik dan berkeropeng yang berbau seperti air kencing tikus = 0.5

CF=92%

Perhitungan manual

|  |  |  |  |
| --- | --- | --- | --- |
| KODE GEJALA | NILAI CF | | CF(H,E) |
| PAKAR | PETUGAS |
| G002 | 0.5 | 0.8 | 0.4 |
| G021 | 0.75 | 0.5 | 0.375 |

CFCombine = 0.4 + 0.375(1-0.4)=0.4+0.225=0.625

CFold1=0.625

|  |  |  |  |
| --- | --- | --- | --- |
| KODE GEJALA | NILAI CF | | CF(H,E) |
| PAKAR | PETUGAS |
| G022 | 0.75 | 0.5 | 0.375 |
| G023 | 0.75 | 0.5 | 0.375 |

CFCombine = 0.375+0.375(1-0,375)=0.6

CFold2=0.625+0.6(1-0.625)=0.85

|  |  |  |  |
| --- | --- | --- | --- |
| KODE GEJALA | NILAI CF | | CF(H,E) |
| PAKAR | PETUGAS |
| G024 | 0.9 | 0.5 | 0.45 |

CFCombine= 0.85+0.45(1-0.85)=0.92

Persentase keyakinan = 92 %

Jadi dapat dikatakan, berdasarkan gejala yang ditanyakan, anjing tersebut mengalami penyakit demodekosis dengan tingkat kepercayaan.

**4.3. Pengujian *Usability* Sistem (Kegunaan)**

*Usability testing* adalah sebuah metode untuk mengevaluasi *user experience* terhadap *software* ataupun *website* yang dibuat (Brooke, 1987). Umumnya, metode ini dilakukan oleh para UX developer dengan melibatkan beberapa *user* (pengguna) tertentu untuk diteliti bagaimana proses mereka selama berinteraksi dengan *website.*

**Kuesioner *Usability* Testing Sistem Pakar Pada Penyakit Anjing Menggunakan Metode *Certainty Factor***

**(Sangat Tidak Mudah = 0, Tidak Mudah = 1,Biasa Saja=2, Mudah=3, dan Sangat Mudah=4)**

Tabel 4.9. Tabel Pengujian Kuesioner Usability Sistem

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nomor** | **Pertanyaan** | **Jawaban** | | | | |
| **Sangat Tidak Mudah** | **Tidak Mudah** | **Biasa saja** | **Mudah** | **Sangat Mudah** |
| 1 | Apakah fungsi menu pada sistem pakar mudah dipelajari? |  |  |  |  |  |
| 2 | Apakah menu tampilan antarmuka) pada sistem pakar mudah dipelajari? |  |  |  |  |  |
| 3 | Apakah sistem pakar mudah digunakan atau dioperasikan? |  |  |  |  |  |
| 4 | Apakah mendapatkan informasi yang diinginkan secara mudah dan cepat? |  |  |  |  |  |
| 5 | Apakah dapat mengakses setiap menu dengan mudah? |  |  |  |  |  |
| 6 | Apakah dapat memasukkan informasi dengan mudah pada menu sistem pakar? |  |  |  |  |  |
| 7 | Apakah sistem pakar dapat mudah digunakan jika lama tidak digunakan? |  |  |  |  |  |
| 8 | Apakah dapat mudah mengingat fungsi setiap menu yang terdapat pada sistem pakar? |  |  |  |  |  |
| 9 | Apakah dapat mudah mengingat lokasi setiap menu yang terdapat pada sistem pakar? |  |  |  |  |  |
| 10 | Apakah dapat dengan mudah mengerti kesalahan yang terjadi? |  |  |  |  |  |
| 11 | Apakah dapat dengan mudah menemukan solusi dari kesalahan yang terjadi? |  |  |  |  |  |
| 12 | Apakah jika terdapat kesalahan, dapat dengan mudah diperbaiki oleh developer? |  |  |  |  |  |
| 13 | Apakah sistem pakar dapat dengan mudah sistem tidak membosankan? |  |  |  |  |  |
| 14 | Apakah setiap ikon, warna, simbol, dan gambar pada sistem pakar mudah dipahami? |  |  |  |  |  |
| 15 | Apakah setiap penempatan menu merasa nyaman? |  |  |  |  |  |

Hasil Kuesioner dari Calon Dokter Hewan (Mahasiswa Dokter Hewan) dipilih 1 dokter hewan dan 4 mahasiswa dokter hewan untuk melakukan kuesioner melalui google formulir

Tabel 4.10. Penilaian Kuesioner

|  |  |  |
| --- | --- | --- |
| **Nomor** | **Pertanyaan** | **Jawaban** |
| 1 | Apakah fungsi menu pada sistem pakar mudah dipelajari? | 3.4 |
| 2 | Apakah menu tampilan antarmuka) pada sistem pakar mudah dipelajari? | 3.4 |
| 3 | Apakah sistem pakar mudah digunakan atau dioperasikan? | 3.8 |
| 4 | Apakah mendapatkan informasi yang diinginkan secara mudah dan cepat? | 3.6 |
| 5 | Apakah dapat mengakses setiap menu dengan mudah? | 3.4 |
| 6 | Apakah dapat memasukkan informasi dengan mudah pada menu sistem pakar? | 3.6 |
| 7 | Apakah sistem pakar dapat mudah digunakan jika lama tidak digunakan? | 3.4 |
| 8 | Apakah dapat mudah mengingat fungsi setiap menu yang terdapat pada sistem pakar? | 3.8 |
| 9 | Apakah dapat mudah mengingat lokasi setiap menu yang terdapat pada sistem pakar? | 3.6 |
| 10 | Apakah dapat dengan mudah mengerti kesalahan yang terjadi? | 3.8 |
| 11 | Apakah dapat dengan mudah menemukan solusi dari kesalahan yang terjadi? | 3.6 |
| 12 | Apakah jika terdapat kesalahan, dapat dengan mudah diperbaiki oleh developer? | 3.8 |
| 13 | Apakah sistem pakar dapat dengan mudah dioperasikan dan sistem tidak membosankan? | 3.4 |
| 14 | Apakah setiap ikon, warna, simbol, dan gambar pada sistem pakar mudah dipahami? | 3.8 |
| 15 | Apakah setiap penempatan menu merasa nyaman? | 3.4 |

Fungsi dari pertanyaan adalah No. 1 -3 Menguji Mudah dipelajari, No. 4-6 Menguji keefisienan, No. 7-9 Menguji ingatan (lebih menekankan pemahaman daripada ingatan, No. 10-12 Menguji kesalahan yang terjadi pada sistem, No. 13-15 Menguji kepuasan pengguna terhadap sistem. Nilai Rata-rata evaluasi adalah 3.58, menandakan sistem mudah dan berguna bagi calon dokter hewan.

# **BAB V**

# **PENUTUP**

## **5.1. Kesimpulan**

Sistem sudah melakukan proses analisis, desain, implementasi dan pengujian pada Sistem Pakar Mendiagnosis Penyakit Anjing. Dari pertanyaan yang diberikan akan menuju pada gejala selanjutnya sesuai dengan aturan yang diberikan. Sistem juga sudah mampu untuk menghitung nilai CF pada setiap gejala dengan jawaban Ya.

Adapun pada implementasinya terdapat 8 bagian yaitu *Login,* Manajemen Gejala, Manajemen Pengguna, Manajemen Diagnosa. Manajemen Aturan, Manajemen Konsultasi, dan Manajemen Hasil Konsultasi. Adapun nilai rata-rata evaluasi adalah 3.58 menandakan sistem mudah dan berguna bagi calon dokter hewan.

## **5.2. Saran**

Untuk pengembangan sistem lebih lanjut, saran yang dapat disampaikan dalam penelitian ini adalah penambahan basis pengetahuan sehingga sistem pakar penyakit anjing dapat mengidentifikasi penyakit anjing dengan lebih baik. Untuk penelitian lebih lanjut, perlu dilakukan penelitian dengan menggunakan metode lain untuk masalah kompleks misal anjing memiliki banyak penyakit.

# **DAFTAR PUSTAKA**

Arhami, Muhammad. 2004. Konsep Dasar Sistem Pakar. Yogyakarta:Andi

Badiyanto. 2016. Mastering Framework Yii. Jakarta:Mediakom

Brooke, John. 1986. A quick and dirty usability scale. United Kingdom: Redhatch Consulting Ltd

Desiani, A dan Arhami, M., 2006, Konsep Kecerdasan Buatan, ANDI. Yogyakarta.

Daniel dan Virginia, G. 2010, Implementasi Sistem Pakar Untuk Mendiagnosis Penyakit Dengan Gejala Demam Menggunakan Metode Certainty Factor. Jurnal Informatika. 6, (1), 26-36

Durkin J, 1994. Expert Systems: Design and Development. the University of Michigan: Macmillan

Hamdani, H., 2016. Sistem Pakar Untuk Diagnosa Penyakit Mata Pada Manusia. Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer, 5(2), pp.13-21.

Kusrini. 2006. Sistem Pakar Teori dan Aplikasi. Yogyakarta : Penerbit Andi

Kusumadewi, S.2003. Artificial intelligence, Yogyakarta: Graha Ilmu

Ningrum, N. C., Anra, H., & Nasution, H. 2015. Sistem Pakar Diagnosa Penyakit Menular Pada Anjing Menggunakan Metode Dempster Shafer. Jurnal Sistem dan Teknologi Informasi (JustIN), 4(1), 122-127

Puja, I Ketut. 2011. Perawatan dan Pengembiakan Anjing. Denpasar: Udayana University Press

Rohman, F.F. and Fauzijah, A., 2008. Rancang bangun aplikasi sistem pakar untuk menentukan jenis gangguan perkembangan pada anak. *Media informatika*, *6*(1).

Qiang Xue, Alexander Makarov, Carsten Brandt,Klimov Paul. 2014. Panduan Definitif untuk Yii 2.0. Yii Software LLC.

Rohajawati, Siti.2010. Sistem Pakar: Diagnosis Penyakit Unggas Dengan Metode Certainty Factor. Bogor : Universitas Pakuan

Russel, S.,.2003. Artificial intelligence: A modern approach, 2nd ed., New

Jersey: Pearson Education

Sommerville, 2011. Software Engineering, 9th ed, New Jersey : Pearson Education

Hu, David S. 1987. Expert Systems for Software Engineers and Managers.

United States of America : Springer

Turban, E., dan Aronson, J.E. 2001. Decision Support System and Intelligent System, 6th Edition. New Jersey: Prentice Hall International Edition

Wheindrata.2012. Buku Pintar Kesehatan Anjing Ras. Yogyakarta: Lily

Publisher

Wijaya, R., 2010. Penggunaan Sistem Pakar dalam Pengembangan portal Informasi untuk Spesifikasi Jenis Penyakit Infeksi. Jurnal Informatika, 3, pp.63-88.

Witari, K. R., Gandhiadi, I. G., & Kencana, I. P. E. N. (2013). Sistem Pakar Pendiagnosa Penyakit Menular pada Anjing. E-Jurnal Matematika, 2(1), 42-48

# **LAMPIRAN**

# **Dokumentasi Sistem**

Lampiran ini berisi panduan penggunaan sistem pakar penyakit anjing dan perincian tabel-tabel dalam *database*

Panduan Penggunaan Sistem Pakar

Pertama kali aplikasi sistem pakar dijalankan sistem akan menampilkan halaman login. User harus memasukkan nama user dan password untuk bisa login ke dalam sistem. Pastikan *username* dan password sudah terdaftar oleh petugas. Langkah penggunaan aplikasi Sistem Pakar Penyakit Anjing sebagai berikut.

1. Login menggunakan *username* dan password yang telah dimiliki

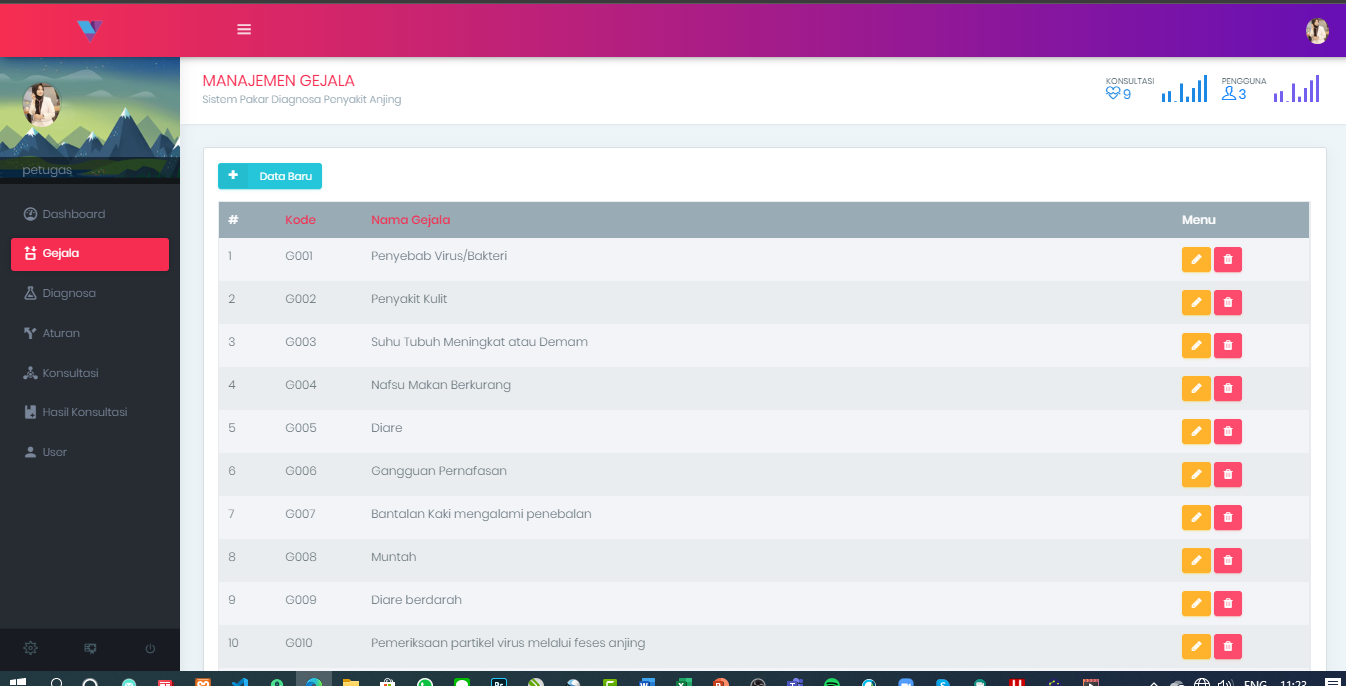
2. Terdapat beberapa tombol yang digunakan untuk mengeksekusi jalannya program.

|  |  |
| --- | --- |
| Tombol tambah data |  |
| Tombol ubah data |  |
| Tombol hapus data |  |

3. Terdapat menu master yaitu gejala, diagnosa, pakar, aturan yang harus ditentukan di awal dan menu konsultasi

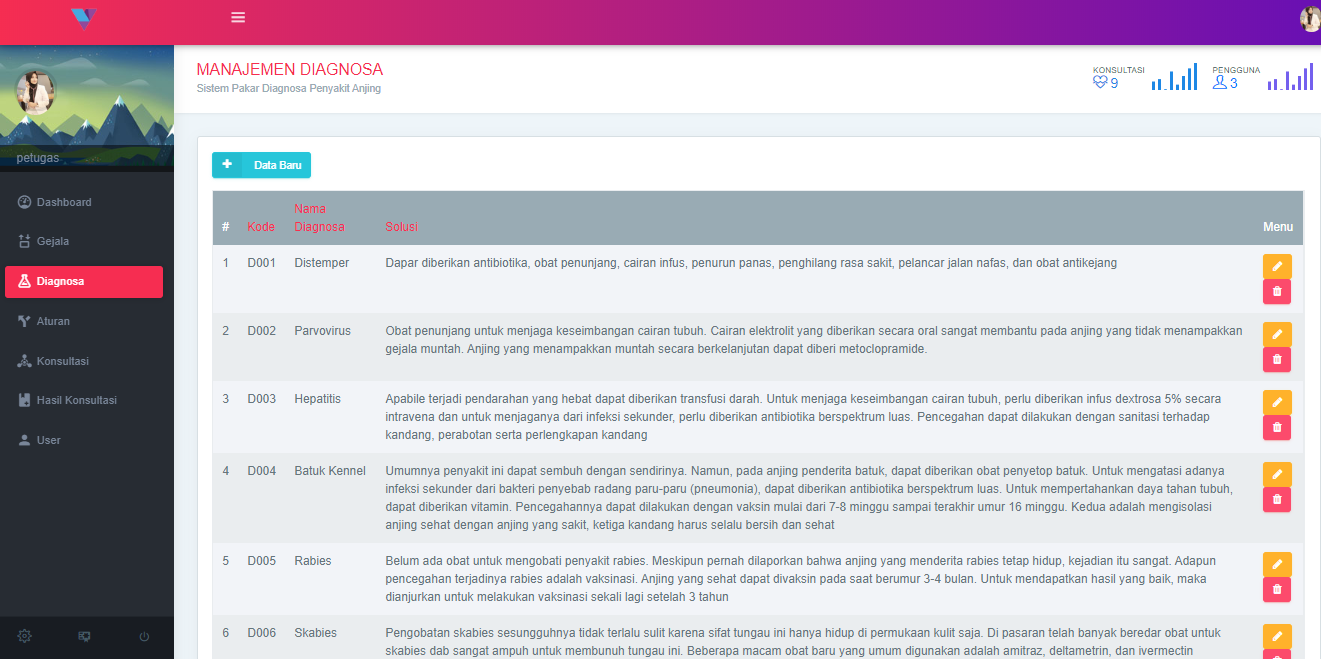
a. Menu Gejala

Menu ini digunakan untuk menambah data gejala di mana mengambil dari gejala klinis penderita yang sudah dikuasai pakar dan nantinya digunakan sebagai proses penentuan diagnosa menggunakan alur maju atau metode *forward chaining*



b. Menu Diagnosa

Menu ini digunakan untuk menambah data diagnosa di mana mengambil dari diagnosa anjing yang terkena penyakit dan nantinya digunakan sebagai hasil diagnosa untuk proses pengolahan tingkat keyakinan *(certainty factor).*



c. Menu Aturan

Menu ini dgunakan untuk proses komnsultasi menggunakan metode forward chaining untuk menentukan hasil diagnosa dari gejala-gejala yang telah ditentukan.

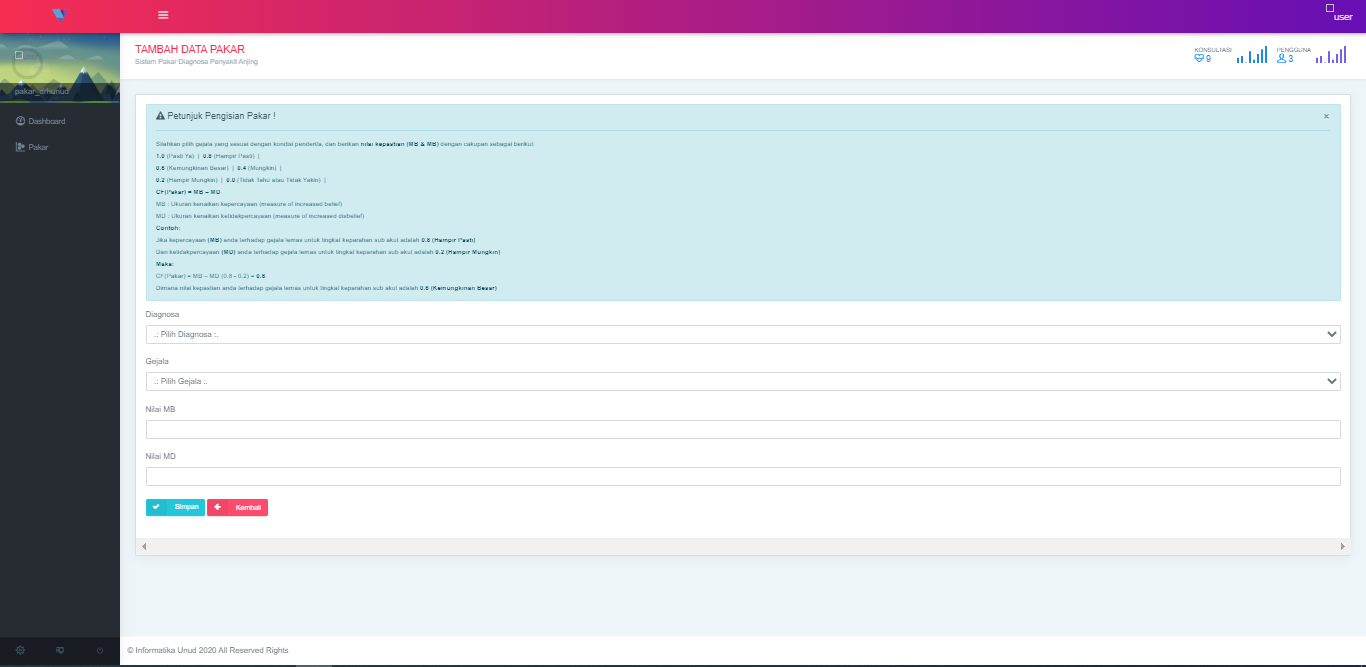


d. Menu Pakar

Menu ini digunakan untuk menambah data pakar dimana masing-masing gejala dari setiap diagnosa ditentukan nilai CF yang terdiri dari

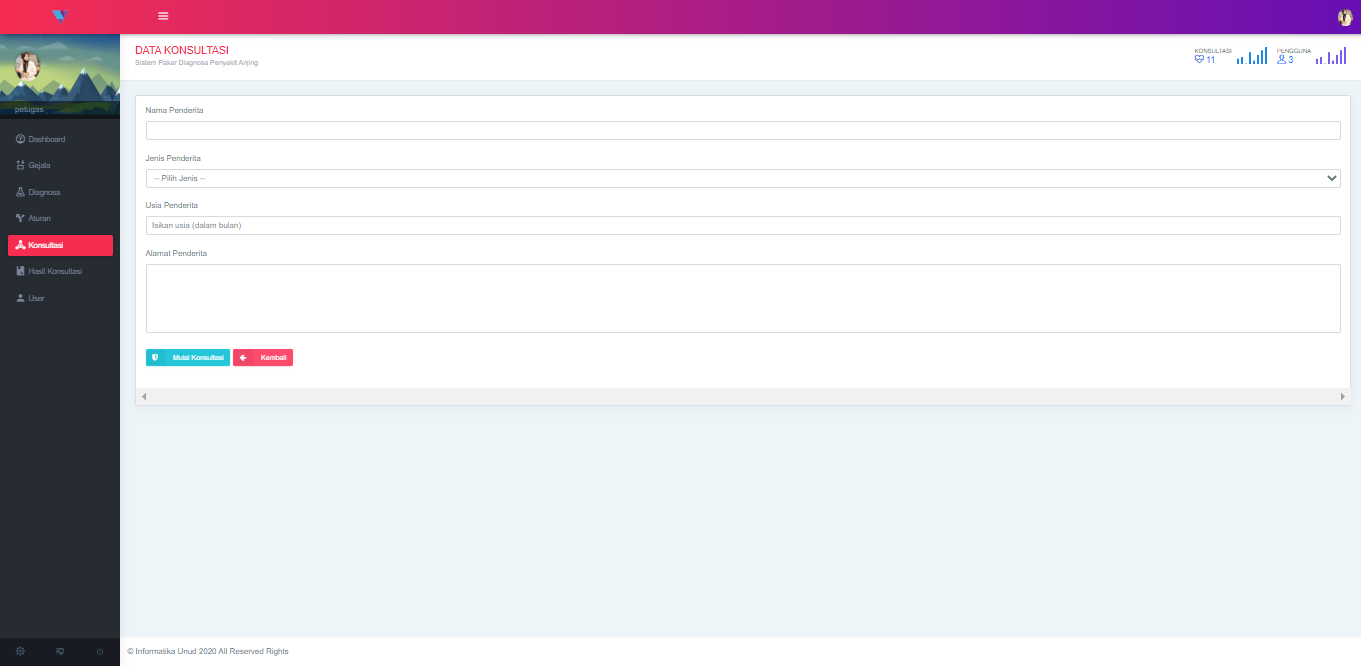
MB = ukuran kepercayaan *(measure of belief)*

MD = ukuran ketidakpercayaan *(measure of disbelief)*



e) Menu Konsultasi

Menu ini digunakan oleh petugas kesehatan dalam proses diagnosis penyakit anjing Petugas diwajibkan mengisikan data anjing berupa nama pemilik, jenis anjing, usia anjing, dan alamat.



Petugas menjawab dengan memberikan nilai tingkat keyakinan (CF) terhadap gejala-gejala yang dialami oleh penderita. Apabila telah terisi sistem akan menentukkan nilai keyakinan dari gejala yang telah dipilih menggunakan metode CF sehingga menghasilkan hasil seperti pada gambar berikut .



4. Terdapat manajemen *user* untuk mengelola hak akses dalam pengoperasian aplikasi di menu *user.*



**Dokumentasi Pencarian Data dan Pengujian Data**

Buku-buku tentang Kedokteran Hewan



Logo Klinik drh. Ari Sapto Nugroho



Pengujian Sistem dan Pengisian Kuesioner -1



Pengujian Sistem dan Pengisian Kuesioner - 2



Pengujian Sistem dan Pengisian Kuesioner -3



Pengujian Sistem dan Pengisian Kuesioner -4