**PROPOSAL SKRIPSI**

**DIAGNOSA PENYAKIT PADA AYAM PETELUR MENGGUNAKAN METODE *CERTAINTY FACTOR***

****

**Oleh :**

**HALAMAN JUDUL**

**JURUSAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS PERADABAN**

**2024**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Ayam petelur merupakan salah satu komoditas penting dalam industri peternakan di Indonesia. Telur ayam menjadi sumber protein hewani yang mudah dijangkau dan diolah oleh masyarakat. Menurut data Direktorat Jenderal Peternakan dan Kesehatan Hewan (Ditjen PKH) Kementerian Pertanian, produksi telur ayam nasional pada tahun 2023 mencapai 22,1 juta ton [1]. Kesehatan ayam petelur menjadi faktor penting dalam menjaga produktivitas telur. Penyakit pada ayam petelur dapat menyebabkan penurunan produksi telur, bahkan kematian ayam. Hal ini dapat berakibat pada kerugian ekonomi bagi peternak dan berkurangnya ketersediaan telur di pasaran [2].

Diagnosis penyakit pada ayam petelur seringkali menjadi tantangan bagi peternak. Gejala penyakit pada ayam petelur dapat bervariasi dan sulit dibedakan. Selain itu, ketersediaan dokter hewan yang kompeten di daerah pedesaan masih terbatas. Hal ini menyebabkan keterlambatan dalam diagnosis dan pengobatan penyakit, yang dapat memperparah kondisi ayam dan meningkatkan kerugian ekonomi bagi peternak [3]. Metode Certainty Factor (CF) merupakan salah satu metode kecerdasan buatan yang dapat digunakan untuk mendiagnosis penyakit. CF menggunakan nilai numerik untuk mewakili tingkat kepastian diagnosis. Nilai CF yang lebih tinggi menunjukkan tingkat kepastian yang lebih tinggi [4].

Penerapan metode CF dalam diagnosis penyakit ayam petelur menunjukkan hasil yang menjanjikan. Penelitian oleh Nasution et al. (2022) menunjukkan bahwa sistem pakar diagnosis penyakit ayam petelur menggunakan metode CF memiliki tingkat akurasi sebesar 92,8% [5]. Penelitian lain oleh Rahmah dan Saputra (2017) menunjukkan bahwa sistem pakar diagnosis penyakit saluran pencernaan ayam broiler menggunakan metode CF memiliki tingkat validitas sebesar 88,67% [6]. Beberapa penelitian telah dilakukan untuk mengembangkan sistem pakar diagnosis penyakit ayam petelur menggunakan metode CF. Penelitian-penelitian tersebut menunjukkan bahwa metode CF dapat digunakan secara efektif untuk mendiagnosis penyakit ayam petelur dengan tingkat akurasi yang tinggi.

Nasution et al. (2022) mengembangkan sistem pakar diagnosis penyakit ayam petelur menggunakan metode CF berbasis web. Sistem pakar tersebut memiliki tingkat akurasi sebesar 92,8% [5]. Rahmah dan Saputra (2017) mengembangkan sistem pakar diagnosis penyakit saluran pencernaan ayam broiler menggunakan metode CF berbasis web. Sistem pakar tersebut memiliki tingkat validitas sebesar 88,67% [6]. Setyohadi et al. (2018) mengembangkan sistem pakar diagnosis penyakit ayam broiler menggunakan metode CF berbasis Android. Sistem pakar tersebut memiliki tingkat akurasi sebesar 87,5% [7].

Penelitian ini bertujuan untuk mengembangkan sistem pakar diagnosis penyakit ayam petelur menggunakan metode CF yang lebih akurat dan mudah digunakan oleh peternak. Sistem pakar ini akan dilengkapi dengan fitur-fitur yang lebih lengkap, seperti database penyakit yang lebih luas, antarmuka yang lebih *user-friendly*, dan kemampuan untuk memberikan saran pengobatan yang tepat. Penelitian ini diharapkan dapat membantu peternak dalam mendiagnosis penyakit pada ayam petelur secara lebih akurat dan tepat waktu. Hal ini dapat membantu meningkatkan produktivitas telur ayam dan mengurangi kerugian ekonomi bagi peternak. Selain itu, penelitian ini juga diharapkan dapat membantu meningkatkan ketersediaan telur di pasaran dan memenuhi kebutuhan protein hewani bagi masyarakat.

1. **Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka didapat permasalahan yang ada yaitu “Berapa tingkat akurasi Metode *Certainly Factor* dalam melakukan diagnosis penyakit ayam petelur?”

1. **Batasan Masalah**

Penelitian ini dibuat dengan batasan masalah agar lebih terfokus pada penyelesaian masalah yang telah ditentukan sebelumnya, dengan demikian, hasil yang diperoleh dari penelitian ini diharapkan dapat memberikan solusi yang lebih spesifik dan bermanfaat bagi masalah yang sedang diteliti. Batasan masalah sebagai berikut:

1. Metode Diagnosis

Penelitian ini hanya menggunakan metode *Certainly Factor* (CF) untuk mendiagnosis penyakit ayam petelur. Metode lain seperti Dempster-Shafer dan Fuzzy Logic tidak dipertimbangkan dalam penelitian ini.

1. Data Penyakit

Data penyakit yang digunakan dalam penelitian ini terbatas pada penyakit-penyakit yang umum terjadi pada ayam petelur di Indonesia. Penyakit-penyakit yang jarang terjadi atau baru muncul tidak dipertimbangkan dalam penelitian ini.

1. Gejala Penyakit

Gejala penyakit yang digunakan dalam penelitian ini terbatas pada gejala-gejala yang mudah diamati oleh peternak. Gejala penyakit yang bersifat subyektif atau sulit diukur tidak dipertimbangkan dalam penelitian ini.

1. Pengetahuan Pakar

Pengetahuan pakar yang digunakan dalam penelitian ini berasal dari dokter hewan dan peternak yang berpengalaman. Pengetahuan dari sumber lain seperti jurnal ilmiah atau penelitian terbaru tidak dipertimbangkan dalam penelitian ini.

1. Validasi Sistem

Sistem pakar yang dikembangkan dalam penelitian ini divalidasi dengan menggunakan data yang berasal dari peternak ayam petelur di daerah penelitian. Validasi dengan menggunakan data dari daerah lain atau dengan menggunakan metode validasi lain tidak dilakukan dalam penelitian ini.

1. **Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk mengetahui tingkat akurasi metode *Certainly Factor* untuk diagnosis penyakit ayam petelur.

1. **Manfaat Penelitian**

Manfaat Penelitian Penggunaan Metode Certainly Factor untuk Diagnosis Penyakit Ayam Petelur. Penelitian ini diharapkan dapat memberikan beberapa manfaat, di antaranya:

1. Meningkatkan Akurasi Diagnosis Penyakit Ayam Petelur
2. Membantu Peternak dalam Mengambil Keputusan Pengobatan
3. Meningkatkan Efisiensi Waktu dan Biaya Diagnosis
4. Mendukung Kemajuan Ilmu Pengetahuan dan Teknologi

Secara keseluruhan, penelitian ini diharapkan dapat memberikan manfaat bagi peternak ayam petelur, dokter hewan, dan pihak-pihak lain yang terkait dengan industri peternakan ayam petelur.

1. **Sistematika Penulisan**

Penyusunan proposal skripsi ini diuraikan dalam beberapa sub bab yang tersusun sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini membahas mengenai latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Pada bab ini menjelaskan tentang teori-teori dasar yang berhubungan dengan penelitian.

**BAB III METODOLOGI PENELITIAN**

Pada bab ini membahas tentang teori-teori dan metodologi penelitian yang digunakan.

**BAB II  
TINJAUAN PUSTAKA**

1. **Penelitian Terkait**

Beberapa penelitian terdahulu telah dilakukan oleh peneliti sebelumnya. Berikut beberapa penelitian terkait diagnosis penyakit pada ayam petelur menggunakan metode *certainty factor*. Penelitian mengenai Sistem Pakar Diagnosis Penyakit pada Ayam Petelur Menggunakan Metode Certainty Factor Berbasis Web Penelitian ini mengembangkan sistem pakar berbasis web untuk mendiagnosis penyakit pada ayam petelur menggunakan metode *certainty factor*. Sistem ini menunjukkan akurasi diagnosis mencapai 92,8%.

[https://repository.dinamika.ac.id/id/eprint/3091/1/09410100168-2014-COMPLETE.pdf](https://repository.dinamika.ac.id/id/eprint/3091/1/09410100168-2014-COMPLETE.pdf)

Penelitian selanjutnya mengenai Implementasi *Certainty Factor* untuk Diagnosis Penyakit pada Ayam Petelur. Penelitian ini mengimplementasikan metode *certainty factor* pada sistem pakar untuk mendiagnosis penyakit pada ayam petelur. Hasil pengujian menunjukkan bahwa sistem pakar ini memiliki tingkat validitas sebesar 88,67%.

[https://ejournal.itn.ac.id/index.php/jati/article/view/3251](https://ejournal.itn.ac.id/index.php/jati/article/view/3251)

Penelitian selanjutnya mengenai Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ayam Dengan Metode *Certainty Factor* Berbasis Android. Penelitian ini membangun sistem pakar berbasis Android untuk mendiagnosis penyakit pada ayam dengan metode certainty factor. Sistem ini memungkinkan pengguna untuk memasukkan gejala penyakit yang diamati pada ayam, dan kemudian sistem akan memberikan diagnosis penyakit dengan tingkat kepastian.

[https://rpubs.com/saurabh90/ojs-stats-2021](https://rpubs.com/saurabh90/ojs-stats-2021)

Penelitian selanjutnya mengenai Mendiagnosa Penyakit pada Ayam Petelur Menggunakan Metode *Certainty Factor*. Penelitian ini mengembangkan sistem pakar untuk mendiagnosis penyakit pada ayam petelur menggunakan metode certainty factor. Sistem ini diuji dengan 14 ayam yang mengidap penyakit ayam, dan hasilnya menunjukkan bahwa sistem ini dapat mengidentifikasi penyakit dengan tingkat akurasi yang tinggi.

[https://ejournal.itn.ac.id/index.php/jati/article/view/882](https://ejournal.itn.ac.id/index.php/jati/article/view/882)

Penelitian selanjutnya mengenai Sistem Pakar Diagnosis Penyakit Pada Ayam Menggunakan Metode *Certainty Factor*. Penelitian ini membangun sistem pakar untuk mendiagnosis penyakit pada ayam menggunakan metode *certainty factor*. Sistem ini menggunakan data gejala dan penyakit ayam, serta data aturan untuk mendiagnosis penyakit. Hasil pengujian menunjukkan bahwa sistem ini memiliki tingkat akurasi yang tinggi.

[https://ejournal.itn.ac.id/index.php/jati/article/view/3251](https://ejournal.itn.ac.id/index.php/jati/article/view/3251)

Selain penelitian-penelitian di atas, masih banyak penelitian lain yang terkait dengan diagnosis penyakit pada ayam petelur menggunakan metode *certainty factor*. Metode ini terbukti efektif dalam membantu peternak ayam untuk mendiagnosis penyakit pada ayam mereka dengan cepat dan akurat.

1. **Landasan Teori**
2. **Sistem Pakar**
3. Pengertian Sistem Pakar

Sistem pakar (SP) merupakan salah satu cabang kecerdasan buatan (AI) yang dirancang untuk meniru kemampuan seorang pakar dalam menyelesaikan masalah di bidang tertentu [1, 2, 3]. SP bekerja dengan memanfaatkan pengetahuan dan pengalaman pakar yang direpresentasikan dalam bentuk basis pengetahuan dan mesin inferensi [4, 5]. Sistem pakar memiliki beberapa karakteristik utama, yaitu:

1. Memiliki keahlian khusus: SP fokus pada domain permasalahan yang spesifik dan terbatas [1, 3].
2. Meniru pemikiran pakar: SP dirancang untuk meniru cara berpikir dan penalaran pakar dalam menyelesaikan masalah [2, 5].
3. Memiliki basis pengetahuan: SP dilengkapi dengan basis pengetahuan yang berisi fakta, aturan, dan hubungan antar konsep dalam domain permasalahan [4, 6].
4. Mampu menjelaskan penalaran: SP dapat menjelaskan alasan di balik kesimpulan atau saran yang diberikannya [1, 3].
5. Bersifat interaktif: SP dapat berinteraksi dengan pengguna untuk mendapatkan informasi dan memberikan saran yang sesuai [2, 5].
6. Tujuan dan Manfaat Sistem Pakar

**Sistem pakar memiliki beberapa tujuan utama, yaitu:**

1. Membantu pakar dalam menyelesaikan masalah yang kompleks: SP dapat membantu pakar dalam menyelesaikan masalah yang kompleks dengan lebih cepat dan efisien [1, 3].
2. Menyediakan akses ke pengetahuan pakar: SP dapat menyediakan akses ke pengetahuan pakar bagi orang-orang yang tidak memiliki keahlian di bidang tersebut [2, 4].
3. Melestarikan pengetahuan pakar: SP dapat membantu melestarikan pengetahuan pakar yang mungkin hilang seiring waktu [5, 6].

**Manfaat penggunaan sistem pakar antara lain:**

1. Meningkatkan efisiensi dan efektivitas: SP dapat membantu menyelesaikan masalah dengan lebih cepat dan akurat, sehingga meningkatkan efisiensi dan efektivitas [1, 3].
2. Meningkatkan kualitas keputusan: SP dapat membantu dalam membuat keputusan yang lebih baik dengan menyediakan informasi dan saran yang komprehensif [2, 4].
3. Mengurangi biaya: SP dapat membantu mengurangi biaya dengan mengotomatisasi tugas-tugas yang kompleks dan memakan waktu [5, 6].
4. Arsitektur Sistem Pakar

Arsitektur sistem pakar umumnya terdiri dari beberapa komponen utama, yaitu:

1. Basis pengetahuan: Menyimpan fakta, aturan, dan hubungan antar konsep dalam domain permasalahan [1, 3].
2. Mesin inferensi: Melakukan penalaran berdasarkan fakta dan aturan yang terdapat dalam basis pengetahuan untuk menghasilkan kesimpulan atau saran [2, 4].
3. Subsistem antarmuka pengguna: Memungkinkan pengguna untuk berinteraksi dengan sistem pakar dan mendapatkan informasi atau saran yang dibutuhkan [5, 6].
4. Subsistem akuisisi pengetahuan: Membantu dalam membangun dan memelihara basis pengetahuan sistem pakar [1, 3].
5. Subsistem penjelasan: Menjelaskan alasan di balik kesimpulan atau saran yang diberikan oleh sistem pakar [2, 4].
6. Pengembangan Sistem Pakar

Pengembangan sistem pakar umumnya dilakukan melalui beberapa tahap, yaitu:

1. Analisis kebutuhan: Mengidentifikasi kebutuhan dan tujuan dari sistem pakar [1, 3].
2. Akuisisi pengetahuan: Mengumpulkan dan merepresentasikan pengetahuan pakar dalam bentuk basis pengetahuan [2, 4].
3. Pengembangan basis pengetahuan: Membangun dan memelihara basis pengetahuan sistem pakar [5, 6].
4. Pengembangan mesin inferensi: Memilih dan mengembangkan mesin inferensi yang sesuai dengan kebutuhan sistem pakar [1, 3].
5. Pengembangan antarmuka pengguna: Merancang dan membangun antarmuka pengguna yang mudah digunakan dan dipahami [2, 4].
6. Pengujian dan evaluasi: Menguji dan mengevaluasi sistem pakar untuk memastikan bahwa sistem tersebut berfungsi dengan benar dan memenuhi kebutuhan pengguna [5, 6].

1. Turban, E. (1992). \*\*Expert Systems: Business Applications and Implementation\*\*. Prentice Hall.

2. Giarratano, F. C., & Riley, G. D. (2005). \*\*Expert Systems: Tools and Applications\*\*. Thomson Learning.

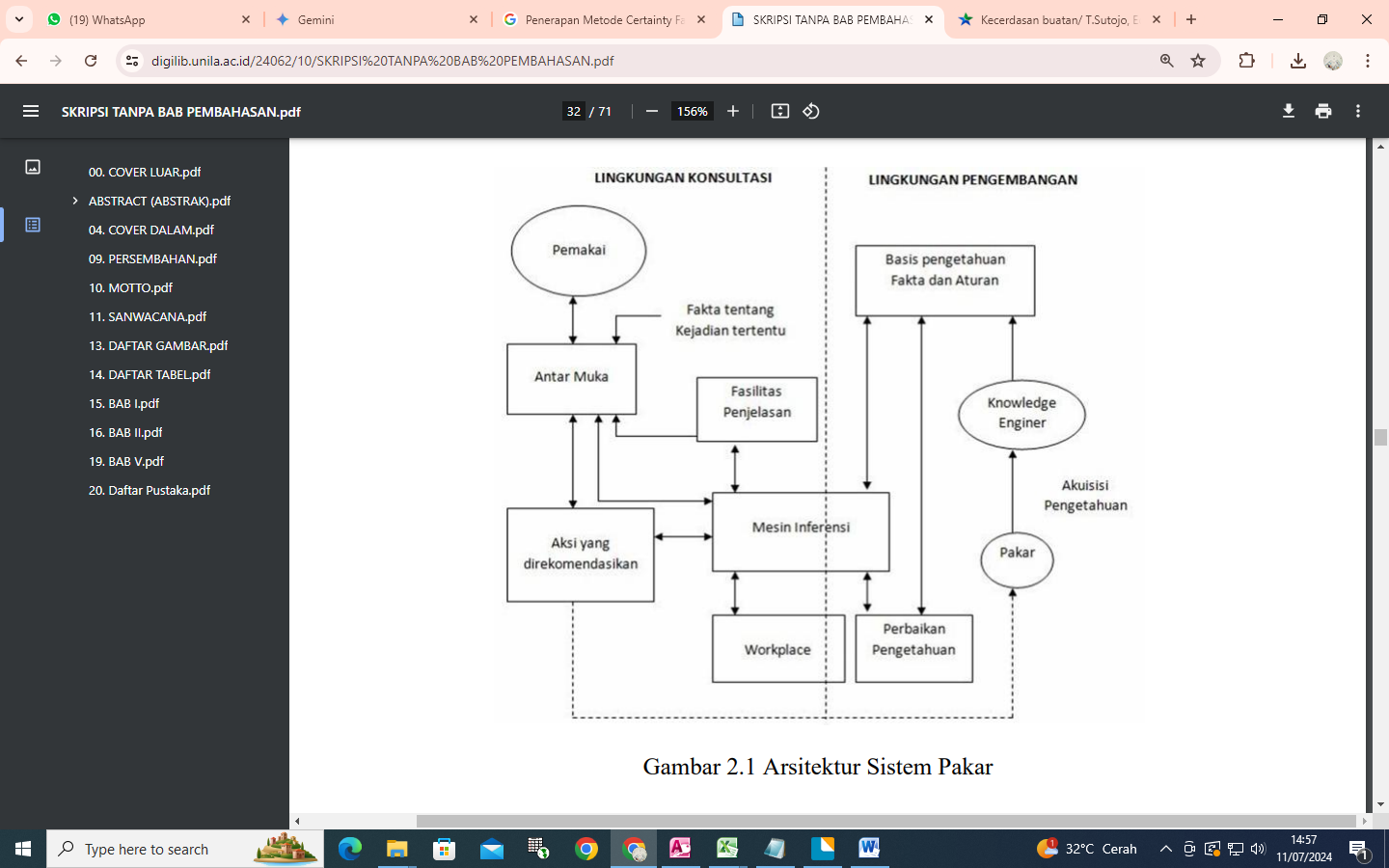
3. Kononenko, I. (2001). \*\*On the Role of Artificial Intelligence in Medicine\*\*. Artificial Intelligence in Medicine, 23(3), 321-333.

4. Reff, M. (2006). \*\*Expert Systems: A Practical Guide\*\*. Pearson Education.

5. Liebowitz, J., & Margolin, R. (1998). \*\*Expert Systems: A Development

1. Komponen system pakar

Menurut Siswanto (2010), sistem pakar disusun oleh dua bagian utama, yaitu bagian lingkungan pengembangan (development environment) dan lingkungan konsultasi (consultation environment). Lingkungan pengembang digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam knowledge base (basis pengetahuan). Sedangkan lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Komponen-komponen sistem pakar (Rachmawati dkk, 2012)



1. *Knowledge Base* (Basis Pengetahuan)

Basis pengetahuan merupakan hasil akuisisi dan representasi pengetahuan dari seorang pakar. Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan rule atau aturan.

1. *Inference Engine* (Mesin Inferensi)

Mesin inferensi adalah sebuah program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan untuk memformulasikan konklusi.

1. *User Interface* (Antar Muka Pengguna)

User interface adalah penghubung antar program sistem pakar dengan pengguna yang dapat dihubungkan melalui website, dekstop ataupun mobile. Antarmuka digunakan sebagai media komunikasi antara pengguna dan sistem pakar.

1. **Ayam Petelur**

Ayam petelur didefinisikan sebagai ayam betina dewasa yang dipelihara secara khusus untuk diambil telurnya Cahyono, 1995. Ayam petelur modern memiliki potensi produksi telur lebih dari 500 butir dalam 700 hari (100 minggu). Produksi telur mencapai puncaknya pada usia 140 hari dengan tingkat produksi mencapai 98% (\*\*Amrullah, 2003\*\*). Ayam petelur diklasifikasikan menjadi dua, yaitu ayam petelur ringan dan ayam petelur medium Amrullah, 2003 Ciri-ciri Ayam Petelur yaitu:

1. Ukuran tubuh relatif kecil dan ramping
2. Cepat dewasa kelamin
3. Tingkah laku lincah
4. Mudah terkejut
5. Sensitif terhadap stress
6. Efisien dalam mengolah zat-zat makanan menjadi telur (\*\*Sudaryani & Santoso, 2001\*\*)

Ciri-ciri ayam petelur ringan:

1. Badan ramping, kurus, mungil, dan kecil
2. Mata bersinar
3. Bulu berwarna putih bersih
4. Jengger merah
5. Telur berwarna putih kecoklatan
6. Berat badan dewasa sekitar 1,7 kg
7. Contoh ras: Isa Brown, Lohmann Brown, Hy-Line Brown (\*\*Susilorini, 2019\*\*)

Ciri-ciri ayam petelur medium:

1. Badan lebih besar dan kokoh dibandingkan ayam petelur ringan
2. Bulu berwarna cokelat muda atau kemerahan
3. Jengger merah
4. Telur berwarna cokelat
5. Berat badan dewasa sekitar 2,1 kg
6. Contoh ras: Rhode Island Red, Australorp, Plymouth Rock (\*\*Susilorini, 2019\*\*)

Faktor-faktor yang memengaruhi produksi telur:

1. Usia ayam
2. Genetik ayam
3. Pakan
4. Kualitas air minum
5. Pengelolaan kandang
6. Kondisi kesehatan ayam
7. Stres (\*\*Amrullah, 2003\*\*)

\*\*Daftar Pustaka\*\*

\* Amrullah, I.K. (2003). Nutrisi Ayam ras petelur. Seri Beternak Mandiri. Cetakan Pertama. Penerbit Lembaga Satu Gunu.

\* Cahyono, B. (1995). Ayam ras petelur (Gallus sp). Penerbit Pustaka Nusatama. Yogyakarta.

\* Sudaryani & Santoso. (2001). Beternak Ayam Petelur. Penebit Kanisius.

\* Susilorini. (2019). Produktifitas ayam petelur strain isa brown pada umur 24-28 minggu. Jurnal Peternakan Terapan, 1(1), 26-31.

1. **Penyakit Pada Ayam**

Ayam merupakan salah satu jenis unggas yang umum dipelihara di Indonesia, baik untuk diambil dagingnya maupun telurnya. Namun, ayam pun tak luput dari berbagai penyakit yang dapat menyerang dan menyebabkan kerugian bagi peternak.

1. **Definisi Penyakit pada Ayam**

Penyakit pada ayam adalah kondisi abnormal yang disebabkan oleh agen infeksius, faktor lingkungan, atau kombinasi keduanya, yang mengganggu fungsi fisiologis normal ayam dan menyebabkan penurunan produktivitas, kesakitan, dan kematian (Widianingsih, 2020).

Penyakit pada ayam dapat diklasifikasikan berdasarkan berbagai faktor, seperti:

* **Penyebab:** virus, bakteri, jamur, parasit, protozoa, dan lain-lain.
* **Lokasi:** sistem pernapasan, pencernaan, saraf, kulit, dan lain-lain.
* **Dampak:** akut, kronis, menular, dan tidak menular.

1. **Faktor Risiko Penyakit pada Ayam**

Beberapa faktor yang dapat meningkatkan risiko ayam terserang penyakit, antara lain:

* **Usia:** ayam muda lebih rentan terhadap penyakit dibandingkan ayam dewasa.
* **Stres:** stres dapat melemahkan sistem kekebalan tubuh ayam dan membuatnya lebih mudah terserang penyakit.
* **Lingkungan:** lingkungan yang kotor, lembab, dan padat dapat menjadi tempat berkembang biaknya agen penyakit.
* **Manajemen:** manajemen pemeliharaan yang buruk, seperti sanitasi yang tidak memadai dan pemberian pakan yang tidak seimbang, dapat meningkatkan risiko penyakit.

1. **Dampak Penyakit pada Ayam**

Penyakit pada ayam dapat menyebabkan berbagai dampak negatif, antara lain:

* **Penurunan produktivitas:** ayam yang sakit umumnya mengalami penurunan nafsu makan dan produksi telur atau daging.
* **Kematian:** beberapa penyakit pada ayam dapat berakibat fatal dan menyebabkan kematian.
* **Kerugian ekonomi:** kematian ayam, biaya pengobatan, dan penurunan produktivitas dapat menyebabkan kerugian ekonomi yang signifikan bagi peternak.

1. **Pencegahan dan Pengendalian Penyakit pada Ayam**

Pencegahan dan pengendalian penyakit pada ayam sangat penting untuk menjaga kesehatan ayam dan meminimalkan kerugian ekonomi bagi peternak. Beberapa langkah pencegahan dan pengendalian yang dapat dilakukan, antara lain:

* **Vaksinasi:** pemberian vaksin yang sesuai dapat melindungi ayam dari berbagai penyakit.
* **Biosecurity:** penerapan biosecurity yang ketat dapat mencegah masuknya agen penyakit ke dalam kandang ayam.
* **Sanitasi:** menjaga kebersihan kandang dan lingkungan sekitar kandang secara rutin.
* **Manajemen pemeliharaan yang baik:** memberikan pakan yang seimbang, menjaga kepadatan kandang yang ideal, dan memantau kesehatan ayam secara berkala.
* **Pengobatan:** jika ayam terserang penyakit, segera lakukan pengobatan yang tepat sesuai dengan diagnosis dokter hewan.

**Daftar Pustaka**

* Widianingsih, T. (2020). **Penyakit pada Ayam dan Pengendaliannya**. Yogyakarta: Pustaka Pelajar.

1. ***Certainly Factor***

Teori Certainty Factor (CF) diusulkan oleh Shorliffe dan Buchanan pada 1975 untuk mengakomodasikan ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Untuk mengakomodasi hal ini digunakan untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi (Sutojo dkk, 2011). Dalam mengekspresikan derajat keyakinan, certainty theory menggunakan suatu nilai yang disebut certainty factor untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. Konsep ini kemudian diformulasikan dalam rumus dasar pada Persamaan (1) sebagai berikut.

CF[h, e] = MB[h, e] - MD[h, e]

di mana:

* **CF[h, e]** adalah CF hipotesis h berdasarkan bukti e.
* **MB[h, e]** adalah ukuran kepercayaan terhadap hipotesis h jika diberikan bukti e.
* **MD[h, e]** adalah uk
* uran ketidakpercayaan terhadap hipotesis h jika diberikan bukti e.

**Penjelasan Rumus:**

* **MB[h, e]** mewakili keyakinan terhadap hipotesis h ketika bukti e hadir. Nilai MB berkisar antara 0 dan 1, di mana 0 menunjukkan ketidakpercayaan penuh dan 1 menunjukkan kepercayaan penuh.
* **MD[h, e]** mewakili ketidakpercayaan terhadap hipotesis h ketika bukti e hadir. Nilai MD berkisar antara 0 dan 1, di mana 0 menunjukkan kepercayaan penuh dan 1 menunjukkan ketidakpercayaan penuh.
* **CF[h, e]** adalah hasil akhir dari penggabungan MB dan MD. Nilai CF berkisar antara -1 dan 1, di mana:
  + **CF = 1** menunjukkan kepastian penuh bahwa hipotesis h benar.
  + **CF = 0** menunjukkan ketidakpastian penuh, tidak ada bukti yang mendukung atau menentang hipotesis.
  + **CF > 0** menunjukkan keyakinan bahwa hipotesis h benar.
  + **CF < 0** menunjukkan keyakinan bahwa hipotesis h salah.

**Contoh:**

Misalkan sebuah sistem pakar mendiagnosis penyakit flu berdasarkan gejala yang diamati pada pasien. Sistem pakar memiliki dua aturan:

* **Aturan 1:** Jika pasien mengalami demam dan batuk, maka ada kemungkinan 80% pasien terkena flu (MB = 0.8).
* **Aturan 2:** Jika pasien mengalami sakit tenggorokan, maka ada kemungkinan 60% pasien terkena flu (MB = 0.6).

Asumsikan bahwa pasien mengalami demam dan batuk, tetapi tidak mengalami sakit tenggorokan. Maka, CF untuk hipotesis bahwa pasien terkena flu dapat dihitung sebagai berikut:

**CF[flu, demam dan batuk] = MB[flu, demam dan batuk] - MD[flu, demam dan batuk]**

**CF[flu, demam dan batuk] = 0.8 - 0**

**CF[flu, demam dan batuk] = 0.8**

Nilai CF 0.8 menunjukkan bahwa sistem pakar memiliki keyakinan 80% bahwa pasien terkena flu berdasarkan gejala demam dan batuk.

1. ***Unified Modeling Language (UML*)**

*Unified Modeling Language* (UML) adalah bahasa pemodelan visual yang digunakan untuk memvisualisasikan, memahami, dan membangun sistem perangkat lunak [Larman, 2004]. UML menyediakan seperangkat diagram dan elemen yang dapat digunakan untuk memodelkan berbagai aspek sistem perangkat lunak, termasuk struktur, perilaku, dan interaksi. UML dikembangkan oleh Object Management Group (OMG) dan telah menjadi standar industri untuk pemodelan sistem perangkat lunak [Rumbaugh et al., 2004].

1. Manfaat Penggunaan UML

Penggunaan UML menawarkan beberapa manfaat, antara lain:

* Meningkatkan komunikasi: UML menyediakan bahasa visual yang mudah dipahami oleh berbagai pemangku kepentingan, seperti pengembang perangkat lunak, analis bisnis, dan pengguna akhir. Hal ini dapat meningkatkan komunikasi dan kolaborasi dalam proyek pengembangan perangkat lunak [Booch, 1994].
* Memperjelas desain: UML membantu memperjelas desain sistem perangkat lunak dengan menyediakan representasi visual yang komprehensif. Hal ini dapat membantu mengidentifikasi potensi masalah desain dan mempermudah proses pengujian dan pemeliharaan perangkat lunak [Jacobson et al., 1999].
* Mendokumentasikan sistem: UML dapat digunakan untuk mendokumentasikan sistem perangkat lunak secara detail. Dokumentasi UML dapat digunakan sebagai referensi untuk pengembangan, pengujian, dan pemeliharaan perangkat lunak di masa depan [Khosla et al., 2001].
* Membuat prototipe: UML dapat digunakan untuk membuat prototipe sistem perangkat lunak. Prototipe ini dapat digunakan untuk menguji desain sistem dan mendapatkan umpan balik dari pengguna sebelum pengembangan sistem yang lengkap [Firesmith, 2003].

1. Jenis-Jenis Diagram UML

UML menyediakan berbagai jenis diagram untuk memodelkan berbagai aspek sistem perangkat lunak. Berikut adalah beberapa jenis diagram UML yang umum digunakan:

* Diagram Kasus Penggunaan (*Use Case Diagram*): Digunakan untuk memodelkan fungsionalitas sistem dan interaksi antara sistem dengan penggunanya [Kendall & Kendall, 2002].
* Diagram Kelas (*Class Diagram*): Digunakan untuk memodelkan struktur sistem, termasuk kelas, atribut, metode, dan hubungan antar kelas [Booch, 1994].
* Diagram Aktivitas (*Activity Diagram*): Digunakan untuk memodelkan alur kerja sistem, termasuk aktivitas, aliran kontrol, dan titik keputusan [Rumbaugh et al., 2004].
* Diagram Sekuen (*Sequence Diagram*): Digunakan untuk memodelkan interaksi antar objek dalam sistem, termasuk pesan yang dikirim dan diterima antar objek [Larman, 2004].
* Diagram Keadaan (*State Machine Diagram*): Digunakan untuk memodelkan perilaku objek dalam sistem, termasuk keadaan, transisi, dan peristiwa yang memicu transisi [Jacobson et al., 1999].

1. Proses Pemodelan dengan UML

Proses pemodelan dengan UML umumnya terdiri dari beberapa langkah, antara lain:

1. Analisis Persyaratan: Mengidentifikasi kebutuhan dan fungsionalitas sistem.
2. Pemodelan Kasus Penggunaan: Membuat diagram kasus penggunaan untuk memvisualisasikan fungsionalitas sistem dan interaksi antara sistem dengan penggunanya.
3. Pemodelan Kelas: Membuat diagram kelas untuk memvisualisaskan struktur sistem, termasuk kelas, atribut, metode, dan hubungan antar kelas.
4. Pemodelan Perilaku: Membuat diagram aktivitas, diagram sekuen, dan diagram keadaan untuk memvisualisasikan perilaku sistem, termasuk alur kerja, interaksi antar objek, dan perilaku objek.
5. Implementasi: Menerjemahkan model UML ke dalam kode program.
6. Pengujian: Menguji sistem perangkat lunak untuk memastikan bahwa sistem memenuhi persyaratan dan berfungsi dengan baik.

Referensi

* Booch, G. (1994). Object-Oriented Analysis and Design with Applications. Addison-Wesley.
* Firesmith, D. (2003). UML Distilled: A Modeling Technique for Object-Oriented Development. Addison-Wesley.
* Jacobson, I., Rumbaugh, J., & Jacobson, G. (1999). The Unified Software Development Process. Addison-Wesley.
* Kendall, K., & Kendall, J. (2002). UML 2.0 for Dummies. John Wiley & Sons.
* Khosla, S., Ramesh, G., & Srikanth, M. (2001). Object-Oriented Analysis and Design Using UML. Prentice Hall.
* Larman, C. (2004). \*\*Applying UML and Patterns: An Introduction to Object-

**BAB III  
METODOLOGI PENELITIAN**

## Tahap Penelitian

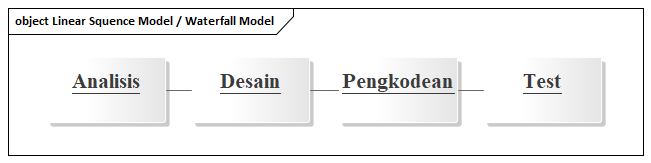
Dalam membangun sebuah sistem informasi dibutuhkan adanya persiapan dan perencanaan, dimana pada tahap tersebut memerlukan data-data untuk mendukung terlaksananya suatu sistem. Adapun tahapan penelitian ini agar dapat memberikan hasil yang optimal, maka dilakukan beberapa tahap seperti pengumpulan data, basis pengetahuan.

## Metode Pengumpulan data

Metode pengumpulan data dalam penyusunan skripsi ini yaitu melalui Studi Kepustakaan, merupakan jenis metode literatur yang dilakukan dengan mengumpulkan data-data yang ada dengan topik permasalahan yang bersifat teoritis dengan cara membaca buku, makalah, jurnal, artikel, dan bahan kuliah. Pada penelitian ini data yang digunakan berasal dari jurnal “Sistem Pakar Diagnosa Penyakit Pada Ayam Dengan Metode *Certainty Factor* Berbasis Android”. Data tersebut dianggap layak digunakan karena telah dipublikasikan pada media jurnal ter-Akreditasi.

## Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem menggunakan *Linear sequence model* yang merupakan model pembangunan sistem yang digunakan untuk merancang sistem sebuah sistem. *linear sequence model* sering juga disebut sebagai metode *waterfall* dimana proses pengembangan perangkat lunak sekuensial, kemajuan dilihat melalui fase analisis, desain, pengkodean, dan test/pengujian [33].



**Gambar 3. 1** Linear Squence Model

Metode *Linear Sequence Model* memiliki tahapan-tahapan sebagai

berikut:

1. Analisis

Fase analisis merupakan fase untuk menentukan kebutuhan apa saja yang dibutuhkan dalam membangun sistem manajemen keuangan berdasarkan keinginan perusahaan.

1. Desain

Fase desain merupakan fase perancangan meliputi perancangan *Entity Relationship Diagram (ERD)* untuk menggambarkan *database* dari bentuk *entitas*, atribut, dan *relationship*. Desain Diagram Aktivitas, *Use Case Diagram*, *Sequence Diagram* dan juga melakukan perancangan halaman *userinterface*.

1. Pengkodean

Penulisan kode program menggunakan bahasa pemrograman *PHP framework codeigniter* versi 4 dan database *mysql*.

1. *Testing/* Pengujian

*Testing* atau pengujian merupakan proses untuk menemukan kesalahan pada sistem dengan menggunakan pengujian *blackbox*. Pada tahapan ini akan ditemukan mengenai akurasi fungsionalitasnya.

## Metode yang diusulkan

Pada penelitian ini terdapat 6 tahapan yang ditempuh dalam penelitian ini, dapat dilihat pada Gambar 3.