

# Penerapan *Case-Based Reasoning* pada Sistem Cerdas untuk Pendeteksian dan Penanganan Dini Penyakit Sapi

Irlando Moggi Prakoso, Wiwik Anggraeni, Ahmad Mukhlason

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

*E-mail:* wiwik@its-sby.edu

**Abstrak**— Penyakit sapi memberikan dampak yang signifikan terhadap penurunan produksi daging bagi para peternak sapi. Untuk meminimalisir dampak dari penyakit perlu dilakukan pendeteksian dan penanganan dini untuk mencegah tingginya kerugian yang terjadi. Sistem cerdas dapat memudahkan peternak sapi untuk mendiagnosa secara mandiri. Penelitian sebelumnya menghasilkan sistem cerdas untuk mendiagnosa penyakit sapi menggunakan algoritma *Backpropagation Artificial neural Network*(ANN). Namun ANN bersifat *black-box* karena kita tidak dapat melihat informasi yang mendasari hasil diagnosa. Tugas akhir ini memiliki tujuan untuk menjawab permasalahan tersebut, yakni dengan membuat sistem cerdas berbasis *Case-Based Reasoning*(CBR) untuk menyempurnakan sistem cerdas yang sebelumnya dibuat menggunakan ANN. CBR memberikan hasil diagnosa berdasarkan permasalahan terdahulu yang dapat direvisi untuk memecahkan permasalahan terbaru. Dari ketiga uji coba dengan case didalam case memory(skenario 1), diluar case memory(skenario 2), dan gejala parsial dari case memory(skenario 3) mendapatkan hasil yang baik dengan nilai *precision* 100% dan 95.83% untuk skenario 1 dan 3. Serta nilai *precision* yang memang kurang baik untuk skenario 2 sebesar 59.31%. Dengan demikian, sistem cerdas ini dapat memberikan hasil diagnosa yang akurat dan memudahkan peternak sapi dalam mendiagnosa secara mandiri.

**Kata Kunci**— *Case-Based Reasoning*, *pendeteksian penyakit sapi*.

## I. PENDAHULUAN

DAGING sapi merupakan salah satu unsur sumber daya penting yang mampu menjamin ketahanan pangan nasional. Namun produksi daging sapi lokal mengalami fluktuasi. Tahun 2005 sampai dengan 2006 terjadi peningkatan 19,2%, lalu terjadi penurunan 18,8% pada tahun 2007. Pada tahun 2009 produksi sapi lokal kembali mengalami peningkatan sebesar 9,1%. Namun pada tahun 2007 hingga 2009 pertumbuhan penyediaan daging lokal lebih rendah daripada kebutuhan yang kemudian mengakibatkan peningkatan impor untuk memenuhi kebutuhan konsumsi daging sapi [1].

Dari fluktuasi produksi daging sapi tersebut, salah satu faktor penyebab penurunan produksi daging sapi disebabkan oleh faktor penyakit. Hingga saat ini peternak sapi memiliki ketergantungan yang tinggi terhadap dokter hewan untuk mendiagnosa dan melakukan penanganan penyakit sapi. Sistem cerdas merupakan salah satu alternatif yang memudahkan peternak sapi dalam mendiagnosa jenis penyakit

sapi secara mandiri serta melakukan penanganan sesuai dengan anjuran yang diberikan oleh sistem cerdas.

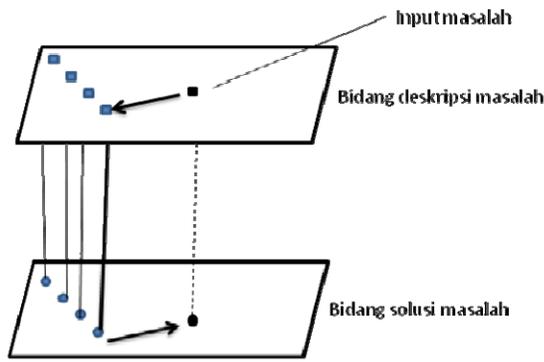
Dalam penelitian yang telah dilakukan oleh Gusviantoko pada tahun 2011 membangun sistem cerdas menggunakan *Backpropagation Artificial Neural Network* (ANN) dikarenakan memiliki kemampuan untuk belajar dan dapat mengetahui hubungan tiap atribut data [2]. Namun ANN memiliki kelemahan karena memerlukan jumlah data yang besar yang diperlukan dalam fase *training* dan dalam memberikan hasil diagnosa bergantung sepenuhnya terhadap data yang diolah dalam fase *training*. [3], ANN juga memiliki masalah dengan *overfitting* karena ANN menggunakan pola dari hasil perhitungan bobot data *training* untuk menghasilkan hasil diagnosa [4]. ANN juga memiliki keterbatasan yang disebabkan adanya *hidden layer* sehingga ANN disebut model *black-box*, sehingga representasi *knowledge* tidak terlihat. Padahal dalam proses diagnosa penyakit diperlukan penjabaran ciri-ciri gejala yang menyertai hasil diagnosa dan dapat melakukan evaluasi apakah hasil diagnosa tersebut perlu dilakukan modifikasi atau tidak [5]. *Case-based Reasoning* (CBR) memiliki kemampuan diagnosa berbasis kasus dan memberikan informasi secara otomatis berdasarkan pengetahuan terdahulu yang dapat direvisi untuk menyesuaikan dengan permasalahan terbaru. Sehingga pengetahuan CBR akan terus berkembang. Pemecahan masalah baru pada CBR dilakukan dengan cara mencari permasalahan sejenis di masa lampau dan memberikan solusi berdasarkan permasalahan yang paling mirip yang ada di dalam *case memory*. permasalahan yang dapat digunakan untuk memecahkan masalah tersimpan di dalam *case memory* yang dapat direvisi untuk memecahkan permasalahan di masa datang.

## II. KAJIAN PUSTAKA

### A. *Case-based Reasoning* (CBR)

*Case-based Reasoning* (CBR) adalah cara penyelesaian permasalahan baru dengan cara mempergunakan kembali pengetahuan paling relevan yang telah dimiliki saat ini yang selanjutnya melakukan proses adaptasi terhadap pengetahuan tersebut untuk menyesuaikan dengan permasalahan baru. CBR merefleksikan cara kerja penyelesaian masalah manusia yang menggunakan pengetahuan yang ia miliki dalam pemecahan

masalah terdahulu yang kemudian dipergunakan sebagai titik awal proses penyelesaian permasalahan baru yang mirip dengan permasalahan yang ia selesaikan sebelumnya. [6]. Ilustrasi dari proses mendapatkan solusi *case based reasoning* dapat dilihat pada gambar 1.

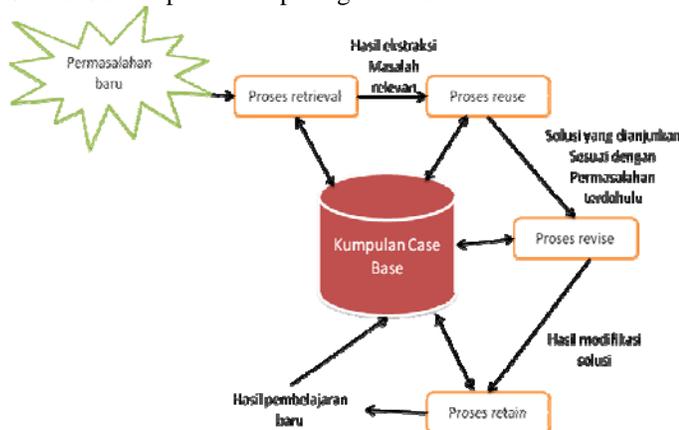


- Deskripsi dari permasalahan baru yang perlu diselesaikan
- Deskripsi dari permasalahan yang sudah terselesaikan.
- Solusi yang tersimpan
- Solusi baru yang didapatkan dari adaptasi solusi yang tersimpan

Gambar. 1. Proses Menghasilkan Solusi Baru Pada CBR.

**B. Case-based Reasoning Cycle**

*Case-based Reasoning* melakukan proses mengingat penyelesaian masalah sebelumnya. Kemudian ketika ada permasalahan baru, *Case-based Reasoning* melakukan perbandingan antara karakteristik permasalahan baru dengan permasalahan yang pernah diselesaikan sebelumnya, ketika permasalahan terbaru mirip dengan permasalahan sebelumnya, CBR melakukan proses ekstraksi solusi dari permasalahan yang relevan dengan permasalahan baru yang dihadapi, apabila solusi tersebut sesuai maka solusi tersebut dipergunakan untuk memecahkan permasalahan baru. Setelah itu, dilanjutkan dengan proses adaptasi, yakni memperbaiki pengetahuan lama agar sesuai untuk menyelesaikan permasalahan baru. Setelah melalui proses adaptasi, pengetahuan baru akan disimpan sebagai salah satu *case base*. Siklus CBR dapat dilihat pada gambar 2.



Gambar. 2. Siklus CBR.

**C. NN Matching**

*NN matching* dipergunakan dalam mengukur kemiripan antar *case* (Ahn & Kim, 2009). Penghitungan kemiripan

dilakukan dengan membandingkan kemiripan atribut ke *j* pada *case* (*I*), dan atribut ke *j* pada *case* (*i*) pada *case memory* (*R*). Formula untuk menghitung kemiripan *case input* dengan *case* di dalam *case memory* terdapat pada persamaan 1 :

$$Similarity(f_j^I, f_j^R) = \frac{\sum_{i=1}^n w_i \cdot s(f_j^I, f_j^R)}{\sum_{i=1}^n w_i}$$

(1)

- J* = atribut ke *j*
- n* = atribut ke *n*
- W<sub>j</sub>* = bobot atribut ke *j*.
- f<sub>j</sub><sup>I</sup>* = nilai dari atribut *j* pada *case* inputan *I*.
- f<sub>j</sub><sup>R</sup>* = atribut *j* pada *case* *i* yang tersimpan dalam *case memory* (*R*).
- S(f<sub>j</sub><sup>I</sup>, f<sub>j</sub><sup>R</sup>)* = hasil perbandingan *f<sub>j</sub><sup>I</sup>* dengan *f<sub>j</sub><sup>R</sup>*, hasil ∈ [0,1].

**D. Pembobotan SWING**

Pembobotan SWING merupakan metode yang memiliki kemampuan menggabungkan rentang atribut dari masing-masing input dari bobot dimana semua atribut didasarkan pada nilai terburuk dan nilai terbaik. Penentuan nilai dilakukan melalui penilaian yang dilakukan oleh para ahli yang diberikan dengan cara memberikan nilai maksimal pada atribut paling penting, lalu memberikan nilai dibawah nilai maksimal secara berurutan untuk atribut berikutnya. Nilai yang diberikan kepada seluruh atribut dinormalisasikan untuk menjadi nilai bobot yang memiliki rentang antara 0 hingga 1 [7]. Persamaan yang dipergunakan untuk menormalisasi nilai untuk dijadikan bobot dapat dilihat pada persamaan 2:

$$w_i = \frac{r_i}{\sum_{k=1, m}^n r_i}, i = 1, 2, \dots, m,$$

(2)

- r<sub>i</sub>* = nilai bagi atribut ke *i*
- w<sub>i</sub>* = bobot ke *i*

**E. Adaptasi Case Pada CBR**

Proses adaptasi *case* merupakan proses penyesuaian dari solusi yang diambil dimana solusi tersebut telah memiliki kemiripan paling tinggi. Penyesuaian dilakukan agar solusi yang diambil tersebut sesuai dengan permasalahan yang sedang dialami dan membuat pengetahuan sistem cerdas menjadi berkembang. Langkah yang dapat diambil dalam proses adaptasi sebagai berikut [8]:

- Solusi yang dihasilkan dari *case* yang diambil dapat dipergunakan sebagai solusi dari permasalahan yang sedang dialami baik tanpa ataupun dengan melalui modifikasi.
- Ketika ternyata dari proses *retrieval* mendapatkan lebih dari satu kasus yang paling relevan, solusi bisa didapatkan dari salah satu kasus diantara kasus yang paling relevan.

**F. Pembelajaran Pada CBR**

Cara Kerja CBR sama dengan proses penalaran masalah pada otak manusia. Ketika suatu masalah atau *case* yang

ditangani ternyata gagal, maka kegagalan akan disimpan kedalam *case memory* sehingga kesalahan yang sama tidak terulang. Dan ketika *case* baru yang ditangani ternyata sukses CBR menyimpannya kedalam *case memory* untuk memecahkan permasalahan yang sama di kemudian hari. Untuk mengetahui *case* yang sukses atau gagal perlu dilakukan penilaian dalam dunia nyata untuk memastikan solusi dari CBR benar-benar sukses atau gagal [9].

G. *Pemeliharaan Case-based*

Pemeliharaan dengan mengurangi duplikasi *case* dan menghapus atau memperbaiki *case* yang salah merupakan hal yang penting untuk mencegah level error terlalu tinggi. Pemeliharaan *case-based* dapat memberikan pengaruh positif pada kualitas *case* tersimpan untuk memberikan solusi pada permasalahan yang akan datang. Karena semakin banyaknya jumlah *case* yang tersimpan, performa pencarian *case* juga semakin menurun [10].

III. ANALISIS DAN PERANCANGAN

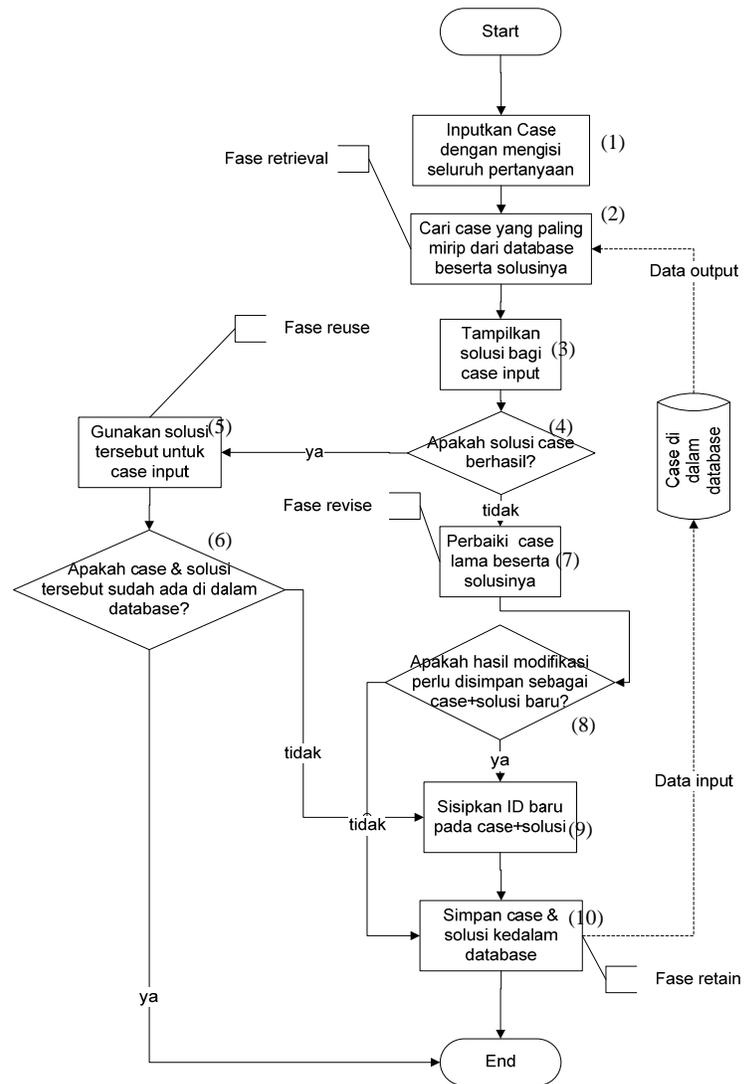
A. *Algoritma CBR Untuk Diagnosa Penyakit Sapi*

Sistem cerdas berbasis CBR ini merupakan sebuah aplikasi berbasis web yang berfungsi untuk melakukan diagnosa penyakit sapi berdasarkan ciri-ciri yang diinputkan oleh pengguna.

Pengguna dapat menginputkan ciri-ciri gejala pada *form input*. Ciri-ciri yang diinputkan tersebut akan dibandingkan dengan ciri-ciri gejala yang tersimpan di dalam *case memory* untuk tiap-tiap atribut. Dari hasil perbandingan tersebut akan dihitung nilai kemiripan dan ditampilkan *case* yang memiliki kemiripan tertinggi. Gambar 3 menampilkan algoritma CBR untuk diagnosa penyakit sapi. Langkah-langkah pada algoritma tersebut adalah sebagai berikut :

1. Pengguna menginputkan gejala penyakit pada sistem cerdas.
2. Sistem cerdas menghitung kemiripan antara *case* yang diinputkan dengan *case* yang ada di *case memory* menggunakan *NN matching*.
3. Sistem cerdas menampilkan *case* paling mirip beserta hasil diagnosa berdasarkan tingkat kemiripan tertinggi.
4. Pengguna mengecek kebenaran hasil diagnosa di dunia nyata dan mengkonfirmasi kebenaran hasil diagnosa. Jika hasil benar, proses berlanjut pada langkah (5). Jika hasil diagnosa salah maka *case* dan hasil diagnosa tersebut akan ditandai dan dilanjutkan pada proses (7)
5. Jika hasil diagnosa benar, hasil diagnosa yang diusulkan tersebut dipasangkan dengan *case* yang baru diinputkan pengguna. Namun belum disimpan kedalam *case memory*.
6. Sistem cerdas memeriksa apakah kombinasi *case* yang diinputkan, dan diagnosa yang diusulkan sudah ada di dalam *case memory*. Jika sudah ada, kombinasi *case* dan solusi tersebut tidak disimpan. Namun jika belum ada lanjut pada proses (9) & (10).
7. *Case* beserta hasil diagnosa yang dilaporkan salah dan telah ditandai akan diperbaiki oleh pakar.

8. Setelah *case* dan usulan solusi diperbaiki, apabila *case* tersebut memiliki kemiripan identik maka langsung masuk proses (10), namun jika tidak identik maka masuk pada proses(9) dilanjutkan dengan proses (10).
9. Menyisipkan ID baru pada *case* yang tidak terdapat dalam *case memory*.
10. Melakukan proses penyimpanan *case* & solusi kedalam database.

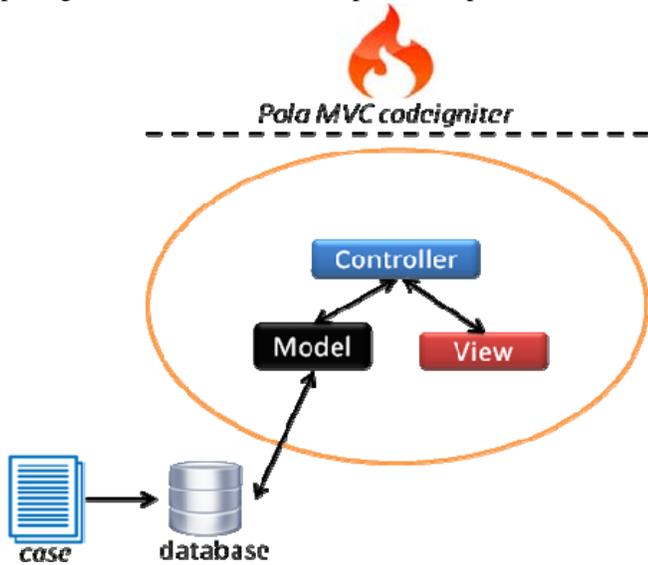


Gambar. 3. Algoritma CBR Untuk Diagnosa Penyakit Sapi.

B. *Arsitektur Perangkat Lunak Sistem Cerdas*

Sistem cerdas ini dibuat menggunakan bahasa pemrograman PHP dengan framework Codeigniter 1.7.2. Framework Codeigniter berbasis MVC (Model, View, Controller) yang memberikan kemudahan dalam manajemen konektivitas dan fleksibilitas antar fungsi. Pola MVC membuat sistem cerdas memiliki tiga lapisan yaitu *model*, yang berhubungan dengan sistem manajemen basis data yang dibutuhkan oleh sistem cerdas. *View* yang merupakan lapisan presentasi aplikasi yang menampilkan antarmuka dan masukan dari pengguna. Serta *controller* yang

merupakan lapisan logika yang menjadi penghubung antara *view* dan *controller* yang menangani semua proses logis seperti menampilkan informasi ke lapisan presentasi hingga menghubungkan lapisan manajemen basis data. Arsitektur perangkat lunak sistem cerdas dapat dilihat pada Gambar 4.



Gambar. 4. Arsitektur Perangkat Lunak Sistem Cerdas.

#### IV. IMPLEMENTASI

##### A. Implementasi Pada Fase Retrieval

Pengguna dapat memasukkan gejala penyakit sapi pada seluruh formulir input pada sistem cerdas kemudian memilih tombol Cari pada bagian bawah formulir input. Antarmuka formulir input dapat dilihat pada gambar 5.

Setelah itu hasil diagnosa dan solusi penanganan dini ditampilkan pada antarmuka hasil diagnosa dan solusi yang dapat dilihat pada gambar 6. Pada antarmuka hasil diagnosa tersebut pengguna dapat mengetahui penyakit apa yang menyerang penyakit sapi, beserta solusi penanganan dini untuk mengatasi penyakit tersebut.

##### B. Implementasi Pada Fase Reuse

Setelah sistem cerdas menampilkan antarmuka hasil diagnosa dan solusi beserta konfirmasi kelayakan hasil diagnosa (Gambar 6). Pengguna kemudian mengkonfirmasi apakah hasil dari diagnosa atau solusi penanganan dini tersebut benar kemudian memilih tombol kirim. Kemudian sistem akan menyimpan case tersebut dengan solusi yang diusulkan dan akan disimpan sebagai case baru apabila tidak ada case yang identik dengan case tersebut, sekaligus mencatat konfirmasi pengguna yang menyatakan case dan solusi tersebut benar atau salah. Setelah itu sistem cerdas memunculkan pemberitahuan bahwa konfirmasi telah dilaporkan yang ditunjukkan pada gambar 7.

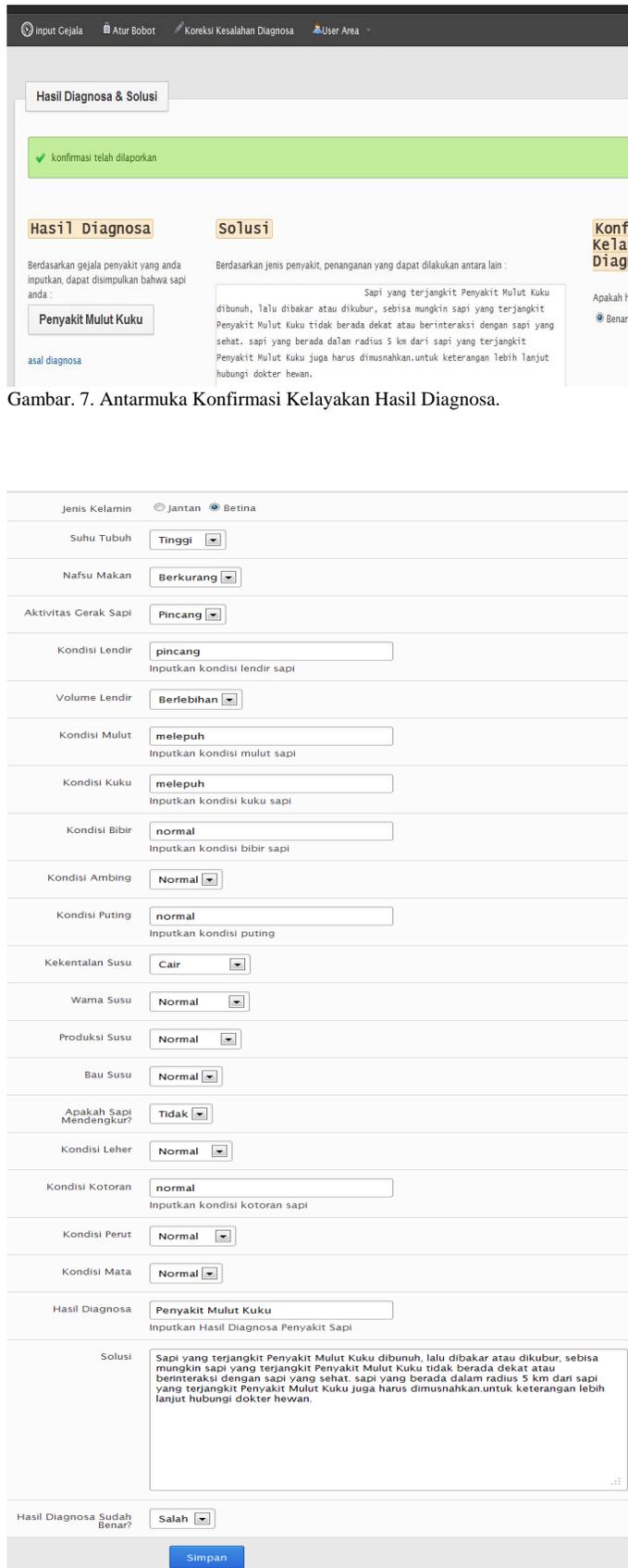
##### C. Implementasi Pada Fase Revise

Dokter hewan dapat melihat daftar case yang dilaporkan salah oleh pengguna. Pada antarmuka tersebut, dokter hewan juga dapat menghapus case yang perlu dihapus sekaligus dapat melihat detail informasi dari case yang kemudian dapat

direvisi untuk memperbaharui informasi dari case tersebut (Gambar 8).

Gambar.5. Antarmuka Input Gejala Penyakit.

Gambar 6. Antarmuka Hasil Diagnosa & Solusi Beserta Konfirmasi Kelayakan Hasil Diagnosa.



Gambar. 7. Antarmuka Konfirmasi Kelayakan Hasil Diagnosa.

D. Implementasi Pada Fase Retain

Fase *retain* merupakan fase yang berperan pada manajemen *knowledge*. Diantaranya proses penyimpanan case yang telah dikoreksi kedalam *case memory* untuk memecahkan permasalahan pada masa yang akan datang. Beberapa hal yang terjadi pada fase ini antara lain mengintegrasikan case baru kedalam *case memory*, serta mengupdate *case lama* beserta solusinya di *case memory*.

V. PENGUJIAN DAN EVALUASI

Pengujian sistem cerdas berbasis CBR dilakukan dengan tiga skenario pengujian yang dijabarkan sebagai berikut :

1. Pengujian menggunakan *case* yang terdapat di dalam *case memory*.
2. Pengujian menggunakan *case* diluar *case memory*.
3. Pengujian menggunakan *case* dengan gejala parsial dari *case memory*.

Ketiga skenario pengujian yang telah disebutkan bertujuan untuk mengetahui kemampuan sistem cerdas dalam memberikan hasil diagnosa yang akurat ketika sistem cerdas menghadapi situasi dimana *case* yang dimasukkan sudah ada di *case memory*, *case* yang dimasukkan belum ada di *case memory*, maupun *case* yang dimasukkan memiliki kesamaan secara parsial dengan *case* yang terdapat di *case memory*.

Pengujian pada ketiga skenario dilakukan menggunakan 20 *case* dimana 5 *case* memiliki hasil diagnosa sehat, 5 *case* memiliki hasil diagnosa Penyakit Mulut Kuku, 5 *case* memiliki hasil diagnosa Mastitis, dan 5 *case* memiliki hasil diagnosa *Septisemia Epzootica*. *Case* uji coba tersebut dimasukkan kedalam sistem cerdas dan dicatat hasil diagnosa yang muncul. Hal ini dilakukan pada keseluruhan *case* uji coba, dilanjutkan dengan perhitungan nilai *precision*, *recall*, serta *accuration* pada masing-masing skenario uji coba. Pengujian untuk semua skenario akan diujikan sebanyak 4 kali dengan rincian sebagai berikut :

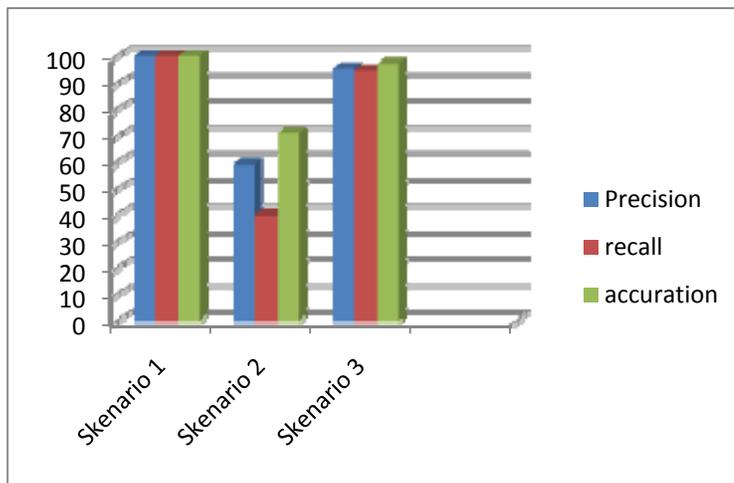
1. Melakukan uji coba dengan hasil diagnosa sehat sebagai kelas positif.
2. Melakukan uji coba dengan hasil diagnosa penyakit mulut kuku sebagai kelas positif.
3. Melakukan uji coba dengan hasil diagnosa Mastitis sebagai kelas positif.
4. Melakukan uji coba dengan hasil diagnosa *Septisemia Epzootica* sebagai kelas positif.

Setelah melakukan 4 kali pengujian pada ketiga skenario, dilanjutkan dengan menghitung nilai *precision*, *recall*, dan *accuration*. Nilai *precision*, *recall*, *accuration* dari hasil pengujian sistem cerdas pada ketiga skenario dapat dilihat pada tabel 1. Dan perbandingan nilai *precision*, *recall*, *accuration* dari ketiga skenario secara grafis dapat dilihat pada gambar 9.

Tabel 1. Hasil *Precision*, *Recall*, Dan *Accuration* Pada Ketga Skenario Uji Coba

No.	Skenario	<i>Precision</i>	<i>Recall</i>	<i>accuration</i>
1	Skenario 1	100%	100%	100%
2	Skenario 2	59.31%	40.00%	71.25%
3	Skenario 3	95.83%	95%	97.5%

Gambar. 8. Antarmuka Edit Case.



Gambar 9. Grafik Perbandingan *Precision*, *Recall*, dan *Accuraction* Dari Ketiga Skenario.

Pada gambar 9 dapat dilihat bahwa pada skenario 1, dengan melakukan uji coba dengan mengujikan kembali *case* yang sudah terdapat di dalam *case memory* memiliki hasil *precision*, *recall*, dan *accuracy* yang paling tinggi. Sedangkan skenario 2 memiliki hasil sedikit lebih rendah dibandingkan dengan skenario 1 namun dengan nilai *precision*, *recall*, dan *accuracy* yang masih sangat baik. Sedangkan skenario 3 dengan melakukan uji coba menggunakan *case* yang tidak terdapat dalam *case memory* memiliki hasil *precision*, *recall*, dan *accuracy* yang paling rendah dan kurang baik dibandingkan dengan skenario 1 dan 3. Hal ini menunjukkan bahwa sistem cerdas berbasis CBR memiliki kinerja yang baik untuk memberikan solusi yang akurat baik ketika menyelesaikan permasalahan dengan ciri-ciri yang identik dengan *case* yang sudah terdapat di dalam *case memory*, maupun menyelesaikan permasalahan dengan ciri-ciri yang sebagian sama dengan *case* yang sudah ada di dalam *case memory*, Sedangkan untuk memecahkan masalah dengan ciri-ciri yang berbeda dengan *case* yang ada di dalam *case memory* sistem cerdas kurang dapat memberikan hasil diagnosa dan solusi yang akurat. Oleh karena itu dari hasil yang didapatkan pada ketiga skenario diatas dapat disimpulkan bahwa proses pembelajaran pada CBR merupakan hal yang penting untuk menjaga agar sistem cerdas mampu memberikan hasil diagnosa dan solusi yang baik pada permasalahan baru.

## VI. KESIMPULAN

Dari hasil pengamatan dari proses perancangan, dan uji coba yang dilakukan, kesimpulan yang dapat diambil adalah sebagai berikut:

- 1) Metode *Case Based Reasoning* dapat digunakan untuk melakukan diagnosa penyakit sapi beserta penanganan dini dengan tepat sehingga memudahkan peternak sapi untuk memeriksa kondisi sapi mereka secara mandiri.
- 2) Dalam studi kasus ini, saat menerapkan metode *Case Based Reasoning*, ketersediaan *case* yang lengkap dan variatif di dalam *case memory* dapat memberikan pengaruh sangat besar dalam peningkatan ketepatan hasil diagnosa karena CBR mampu mengidentifikasi *case*

berdasarkan kemiripan dengan sangat baik, baik *case* yang diinputkan memiliki kesamaan identik maupun parsial dengan *case* yang ada di dalam *case memory*.

- 3) Proses pembelajaran pada *Case Based Reasoning* dengan melakukan evaluasi dalam lingkungan nyata merupakan hal yang sangat penting karena selain ketersediaan *case* yang lengkap, CBR juga bergantung kepada kualitas *case* yang tersimpan. Hal ini penting untuk mencegah CBR mengambil *case* tersimpan dengan hasil diagnosa yang salah.
- 4) Dalam kasus ini, penentuan bobot juga merupakan hal penting dalam memberikan akurasi hasil diagnosa. Dengan penentuan bobot yang benar berdasarkan tingkat kepentingan tiap-tiap atribut gejala akan memberikan akurasi diagnosa dengan sangat baik.

## UCAPAN TERIMA KASIH

Penulis mengucapkan Puji Syukur Alhamdulillah kepada Allah SWT, Kedua orang tua, kakak kandung, dan adik kandung penulis, Ibu Wiwik Anggraeni S.Si,M.Kom dan Bapak Ahmad Mukhlason, S.Kom.,M.Sc selaku dosen pembimbing. Dr.drh Darminto selaku narasumber mengenai penyakit sapi. Serta kepada Dr.Eng. Febriliyan Samopa,S.Kom.,M.Kom selaku kepala jurusan Sistem Informasi ITS, kepada Ayu Nursepti, serta teman-teman 8IOS baik yang berada di laboratorium DSS & BI, PPSI, maupun EBIS, dan semua pihak yang telah membantu penulis dalam menyelesaikan penelitian ini.

## DAFTAR PUSTAKA

- [1] Direktorat Jenderal Peternakan. (2010). *Blue Print Program Swasembada Daging Sapi 2014*. Kementerian Pertanian.
- [2] Purwanto, G. D. (2011). *Penerapan Algoritma Artificial Neural Network Pada Sistem Cerdas Untuk Pendeteksian dan Penanganan Dini Penyakit Sapi*. Surabaya: ITS.
- [3] Lees, B., & Corchado, J. (n.d.). *Integrated Case-Based Neural Network Approach to Problem Solving*. 3.
- [4] Ahn, H., & Kim, K.-j. (2009). Bankruptcy Prediction Modeling With Hybrid Case-based Reasoning and Genetic Algorithms Approach. *Elsevier Applied Soft Computing* 9, 599.
- [5] Ellenius, J., & Groth, T. (2007). Dynamic Decision Support Graph—Visualization of ANN-Generated Diagnostic Indications of Pathological Conditions Developing Over Time. *Elsevier Artificial Intelligence in Medicine*, 190.
- [6] Salamó, M., & Sánchez, M. L. (2010). Adaptive Case-based Reasoning Using Retention And Forgetting Strategies. *Knowledge-Based Systems*, 1.
- [7] Balasubramaniam, A., Boyle, R. A., & Voulvoulis, N. (2006). Improving Petroleum Contaminated Land Remediation. *Elsevier Chemosphere* 66, 793.
- [8] Pal, S. (2004). *Foundations of Soft Case-Based Reasoning*. Wiley-Interscience publication.
- [9] Iqbal, N., & Ashraf, M. H. (2006). *Evaluation of jCOLIBRI*. Vasteras: Malardalen University.
- [10] Subroto, I. M. (2007). Case-based Retrieval On Question Items Generation. *Faculty Of Computer Science And Information System Universiti Teknologi Malaysia*, 14.