

# PENERAPAN CASE-BASED REASONING PADA SISTEM CERDAS UNTUK PENDETEKSIAN DAN PENANGANAN DINI PENYAKIT SAPI

Irlando Moggi Prakoso<sup>1)</sup>, Wiwik Anggraeni<sup>2)</sup> Ahmad Mukhlason<sup>3)</sup>

<sup>1)</sup>Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Jl. Raya ITS, Surabaya, 60111  
Telp : (031) 5964965, Fax : (031) 5999944  
E-mail : wiwik@its-sby.edu<sup>1)</sup>

---

## Abstract

*Cow disease causes significant impact on meat production degradation for livestock breeders. Independent early detection and treatment using intelligent system is necessary to minimize the bad impact caused by cow disease. The former research has produced intelligent system based on Backpropagation Artificial Neural Network(ANN). But unfortunately ANN works in a blackbox because the groundwork information which lead to the diagnostic result is unshown. Intelligent system based on Cased-Based Reasoning(CBR) can solve the problem that happened in ANN based intelligent system because CBR suggests the diagnosis result based on past problems which can be revised to solve new problem accurately. Besides that, CBR can show the groundwork information from its diagnostic result. From the three experiments, using cases from case memory (scenario 1), outside case memory (scenario 2), and partial symptoms from case memory (scenario 3) gives good result with 100% and 95.83% precision value for scenario 1 and 3. And precision value which is actually true with only 59.31% for scenario 2. The poor result in scenario 2 caused by the cases used in experiment are cases that never been faced by intelligent system, so there is no cases in case memory has the same symptoms with the symptom entered in intelligent system. So it can be concluded from the experiments that this intelligent system can suggests diagnostic result accurately and eases the breeders to diagnose cow disease independently.*

## Abstrak

Penyakit sapi memberikan dampak signifikan terhadap penurunan produksi daging bagi para peternak sapi. Untuk meminimalisir dampak penyakit perlu dilakukan pendeteksian dan penanganan sedini mungkin secara mandiri menggunakan sistem cerdas. Penelitian sebelumnya menghasilkan sistem cerdas menggunakan algoritma Backpropagation Artificial neural Network(ANN). Namun ANN bersifat black-box karena kita tidak dapat melihat informasi yang mendasari hasil diagnosa. Sedangkan sistem cerdas berbasis Cased-Based Reasoning(CBR) mampu menyelesaikan permasalahan tersebut karena CBR memberikan hasil diagnosa berdasarkan permasalahan terdahulu yang dapat direvisi untuk memecahkan permasalahan terbaru. Selain itu, ciri-ciri gejala yang mendasari hasil diagnosa juga dapat diamati. 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%. Hasil yang kurang baik pada skenario 2 terjadi karena case yang digunakan belum pernah ditangani oleh sistem cerdas sehingga belum ada case di case memory yang memiliki ciri-ciri gejala yang sama dengan gejala yang diinputkan. Dari hasil uji coba dapat disimpulkan bahwa 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.

## 1. PENDAHULUAN

Daging sapi merupakan salah satu 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, menurut Direktorat Jenderal Peternakan, pertumbuhan penyediaan daging lokal lebih rendah daripada kebutuhan yang kemudian mengakibatkan peningkatan impor untuk memenuhi kebutuhan konsumsi daging sapi.

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 (Purwanto, 2011). Namun ANN memiliki kelemahan karena memerlukan jumlah data yang besar untuk tahap *training* dan dalam memberikan hasil diagnosis bergantung sepenuhnya terhadap data yang diolah dalam fase *training* (Lees, 1999). ANN juga memiliki masalah dengan *overfitting* karena ANN menggunakan pola dari hasil perhitungan bobot data *training* untuk menghasilkan hasil diagnosa (Ahn, 2009). 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 (Ellenius, 2008). *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.

## 2. METODOLOGI

Pada bab ini akan dijelaskan mengenai dasar teori yang digunakan untuk melakukan penelitian. Pustaka yang digunakan adalah *Case-based Reasoning*, *NN Matching* dan *SWING*.

### 2.1 Case-based Reasoning (CBR)

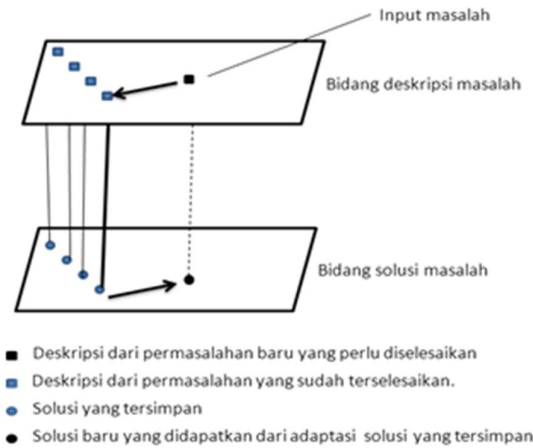
*Case-based Reasoning* (CBR) adalah cara penyelesaian permasalahan baru dengan cara menggunakan kembali pengetahuan paling relevan yang telah dimiliki saat ini yang selanjutnya melakukan proses adaptasi terhadap pengetahuan tersebut untuk menyesuaikan dengan permasalahan baru. CBR merefleksikan cara kerja penyelesaian masalah manusia yang menggunakan pengetahuan yang ia miliki dalam pemecahan masalah terdahulu yang kemudian digunakan sebagai titik awal proses penyelesaian permasalahan baru yang mirip dengan permasalahan yang ia selesaikan sebelumnya (Salamó, 2010). Ilustrasi dari proses mendapatkan solusi CBR dapat dilihat pada gambar 1.

### 2.2 Case-based Reasoning Cycle

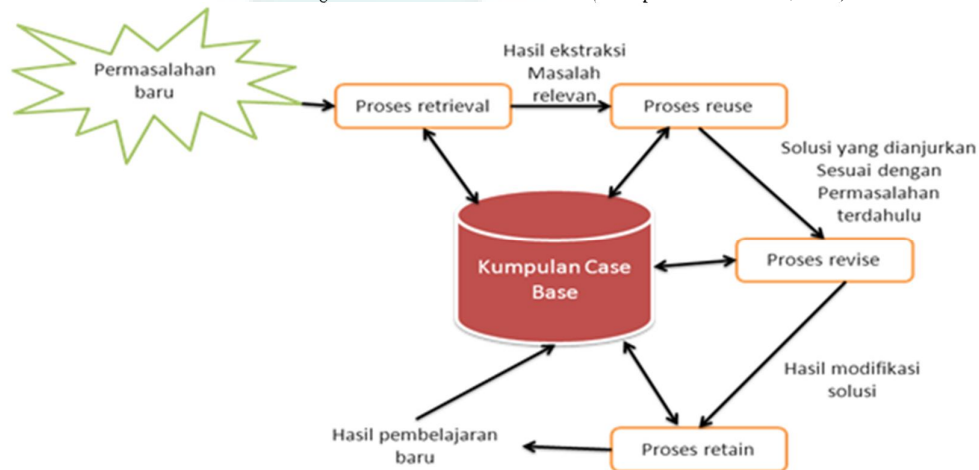
CBR melakukan proses mengingat penyelesaian masalah sebelumnya. Kemudian ketika ada permasalahan baru, CBR melakukan perbandingan antara karakteristik permasalahan baru dengan permasalahan yang pernah diselesaikan sebelumnya, ketika permasalahan terbaru mirip dengan permasalahan sebelumnya, CBR melakukan 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.

### 2.3 NN Matching

*NN matching* dipergunakan dalam mengukur kemiripan antar *case* (Ahn, 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:



Gambar 2 Proses Menghasilkan Solusi Baru Pada CBR (diadaptasi dari Subroto, 2007)



Gambar 3 Siklus CBR (diadaptasi dari Salamó & Sánchez, 2010)

(1)

$j$  = atribut ke  $j$   
 $n$  = atribut ke  $n$   
 $W_j$  = bobot atribut ke  $j$ .  
 = nilai dari atribut  $j$  pada *case* inputan  $I$ .  
 = atribut  $j$  pada *case*  $i$  yang tersimpan dalam *case memory* (R).  
 = hasil perbandingan

## 2.4 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 (Balasubramaniam, 2007). Persamaan yang dipergunakan untuk menormalisasi nilai untuk dijadikan bobot dapat dilihat pada persamaan 2:

$$r_i = \text{nilai bagi atribut ke } i$$

$$w_i = \text{bobot ke } i$$

## 2.5 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 (Pal, 2004):

- 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.

## 2.6 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 (Iqbal, 2006).

## 2.7 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* 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 (Subroto, 2007).

## 3. HASIL dan PEMBAHASAN

### 3.1 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.

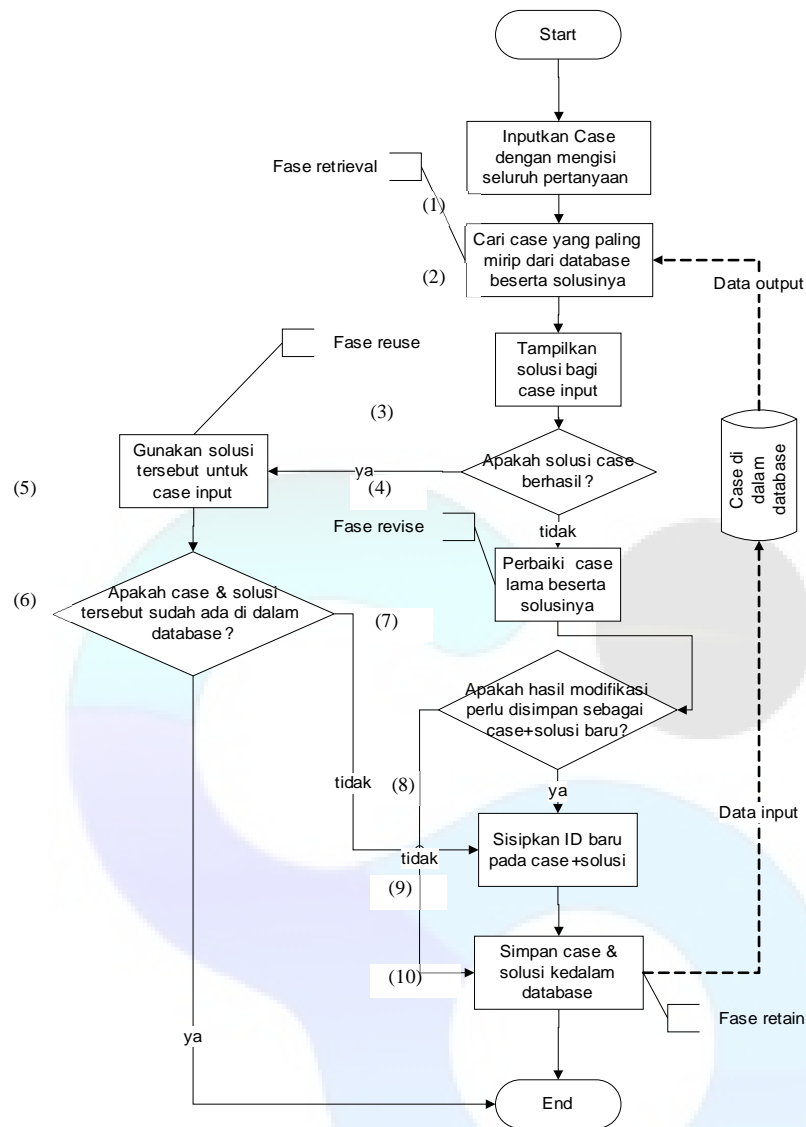
Pengguna menginputkan ciri-ciri gejala pada *form input*. Ciri-ciri yang dimasukkan tersebut akan dibandingkan dengan ciri-ciri gejala yang tersimpan di dalam *case memory*. Dari hasil perbandingan tersebut akan dihitung nilai kemiripan dan ditampilkan *case* yang memiliki kemiripan tertinggi. Gambar 3 menampilkan algoritma CBR untuk diagnosa penyakit sapi.

Langkah-langkah pada algoritma CBR adalah sebagai berikut:

1. Pengguna menginputkan gejala penyakit pada sistem cerdas.
2. Sistem cerdas menghitung kemiripan antara *case* yang diinputkan dengan *case* pada *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 mengonfirmasi 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.

### 3.2 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.



Gambar 4 Algoritma CBR Untuk Diagnosa Penyakit Sapi

---

**Hasil Diagnosa & Solusi**

**Hasil Diagnosa**

Berdasarkan gejala penyakit yang anda inputkan, dapat disimpulkan bahwa sapi anda :

Penyakit Mulut Kuku

asal diagnosa

**Solusi**

Berdasarkan jenis penyakit, penanganan yang dapat dilakukan antara lain :

Sapi yang terjangkit Penyakit Mulut Kuku dibunuh, lalu dibakar atau dikubur, sebisa mungkin sapi yang terjangkit Penyakit Mulut Kuku tidak berada dekat atau berinteraksi dengan sapi yang sehat. sapi yang berada dalam radius 5 km dari sapi yang terjangkit Penyakit Mulut Kuku juga harus dimusnahkan. untuk keterangan lebih lanjut hubungi dokter hewan.

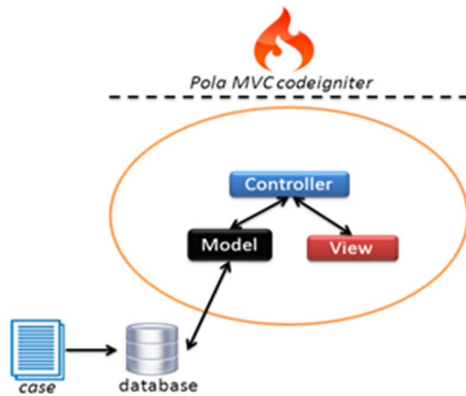
**Konfirmasi Kelayakan Hasil Diagnosa**

Apakah hasil diagnosa & solusi benar?

☒ Benar
 ☐ Salah

Gambar 4. Antarmuka Hasil Diagnosa & Solusi Beserta Konfirmasi Kelayakan Hasil Diagnosa





Gambar 5. Arsitektur Perangkat Lunak 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 5.

### 3.3 Implementasi pada Fase Retrieval

Pengguna dapat memasukkan gejala penyakit sapi pada seluruh formulir input pada sistem cerdas kemudian memilih tombol Cari. Antarmuka formulir input dapat dilihat pada gambar 6. Setelah itu hasil diagnosa dan solusi penanganan dini ditampilkan pada antarmuka hasil diagnosa dan solusi yang dapat dilihat pada gambar 4. Pada antarmuka tersebut pengguna dapat mengetahui penyakit apa yang menyerang penyakit sapi, beserta solusi penanganan dini.

### 3.4 Implementasi pada Fase Reuse

Setelah sistem cerdas menampilkan antarmuka hasil diagnosa dan solusi beserta konfirmasi kelayakan hasil diagnosa (Gambar 4). Pengguna mengkonfirmasi apakah hasil dari diagnosa atau solusi penanganan dini tersebut benar kemudian memilih tombol kirim. 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 menampilkan pemberitahuan bahwa konfirmasi telah dilaporkan yang ditunjukkan pada gambar 9.

### 3.5 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 dapat direvisi untuk memperbaharui informasi dari *case* tersebut (Gambar 7).

### 3.6 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*.

### 3.7 Pengujian dan Evaluasi

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

1. Pengujian menggunakan *case* yang telah ada di dalam *case memory*.
2. Pengujian menggunakan *case* yang belum ada *case memory*.
3. Pengujian menggunakan *case* dengan gejala parsial dari *case memory*.

Ketiga skenario pengujian tersebut 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 Epizootica*.

*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 *accuracy* pada masing-masing skenario uji coba.

Gambar 6. Antarmuka Input Gejala Penyakit

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 *accuracy* yang dapat dilihat pada tabel 1. Sedangkan perbandingan *precision*, *recall*, *accuracy* dari ketiga skenario secara grafis dapat dilihat pada gambar 8.

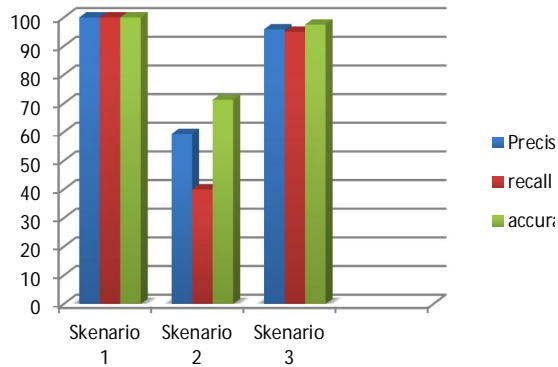
Pada gambar 8 dapat dilihat bahwa pada skenario 1 memiliki hasil *precision*, *recall*, dan *accuracy* yang paling tinggi dan skenario 3 memiliki hasil sedikit lebih rendah dibandingkan dengan skenario 1. Sedangkan skenario 2 memiliki *precision*, *recall*, dan *accuracy* yang paling rendah dan kurang baik.

Gambar 7. Antarmuka Edit Case.

Tabel 1. Hasil *Precision*, *Recall*, Dan *Accuracy* pada Ketiga Skenario Uji Coba

Skenario	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>
Skenario 1	100%	100%	100%
Skenario 2	59.31%	40.00%	71.25%
Skenario 3	95.83%	95%	97.5%

Hal ini menunjukkan bahwa CBR memiliki kinerja yang baik dalam memberikan diagnosa yang akurat, baik ketika menyelesaikan permasalahan dengan ciri identik maupun menyelesaikan permasalahan dengan ciri yang sebagian sama dengan *case* yang telah ada di dalam *case memory*. Sedangkan untuk ciri yang berbeda dengan *case* yang ada di dalam *case memory*, sistem cerdas kurang dapat memberikan hasil diagnosa dan solusi yang akurat.



Gambar 8. Grafik Perbandingan Precision, Recall, dan Accuracy Dari Ketiga Skenario.

#### 4. SIMPULAN dan SARAN

Setelah uji coba menggunakan tiga skenario dengan menggunakan masing-masing 20 data uji coba dari *case memory*, diluar *case memory*, dan data yang memiliki kemiripan gejala parsial dari *case memory*. Dilanjutkan dengan perhitungan nilai *precision*, *recall* dan *accuracy* untuk masing-masing skenario, kesimpulan yang dapat diambil adalah sebagai berikut :

1. Berdasarkan hasil pada skenario 1, dengan mengujikan kembali *case* yang sudah terdapat di dalam *case memory* memiliki hasil *precision*, *recall*, dan *accuracy* yang sangat tinggi yakni nilai *precision* 100%, *recall* 100%, dan *accuracy* 100%. Hal ini menunjukkan bahwa sistem cerdas berbasis CBR mampu memberikan hasil diagnosa dengan sangat baik ketika sistem cerdas pernah menyelesaikan permasalahan yang sama sebelumnya, sehingga sistem cerdas hanya perlu memanggil kembali *case* dan solusi telah tersimpan di dalam *case memory*.
2. Hasil yang didapatkan dari skenario 2, dengan mengujikan *case-case* yang tidak terdapat pada *case memory* memiliki hasil *precision*, *recall*, dan *accuracy* yang kurang baik yaitu nilai *precision* 59.31%, *recall* 40.00%, dan *accuracy* 71.25%. Ternyata sistem cerdas berbasis CBR memang kurang mampu memberikan hasil diagnosa yang akurat ketika sistem cerdas sebelumnya belum pernah menangani permasalahan yang diinputkan, sehingga *case* beserta solusi yang diusulkan dari *case memory* memiliki kemungkinan tidak relevan yang lebih besar. Hal ini dapat diatasi dengan memperkaya variasi *case* di dalam *case memory*.

3. Berdasarkan hasil yang didapatkan dari skenario 3, yakni dengan mengujikan *case* dengan gejala parsial dari *case memory* memiliki nilai nilai *precision* 95.83%, *recall* 95%, dan *accuracy* 97.5%. Hal ini menunjukkan bahwa sistem cerdas berbasis CBR masih mampu memberikan hasil diagnosa dengan benar meskipun permasalahan yang diinputkan memiliki kemiripan hanya pada sebagian gejala dengan *case* yang tersimpan di dalam *case memory*.
4. Berdasarkan hasil percobaan skenario 1 & 3 menunjukkan bahwa semakin banyak jumlah atribut gejala yang sama antara *case* yang diinputkan kedalam sistem cerdas dengan *case* yang tersimpan di *case memory* dapat meningkatkan kinerja sistem cerdas dalam memberikan hasil diagnosa dan solusi dengan tepat.
5. Dalam studi kasus ini, berdasarkan hasil uji coba dari ketiga skenario menunjukkan bahwa semakin banyak jumlah *case* yang tersimpan di *case memory* dapat meningkatkan ketepatan hasil diagnosa secara signifikan, karena semakin banyak *case* tersimpan di *case memory* menunjukkan bahwa sistem cerdas telah memiliki banyak pengalaman dalam menangani berbagai kasus, sehingga kemungkinan kebenaran *case* dan solusi yang disarankan oleh sistem cerdas menjadi lebih tinggi.

Sistem cerdas berbasis CBR pada studi kasus ini tidak dapat mempertahankan *case* yang berguna dan menghapus *case* yang tidak berguna secara otomatis. Selain itu juga tidak mampu menghitung kemiripan antar *case* ketika menghadapi kata-kata berupa sinonim. Oleh karena itu, saran yang dapat dipertimbangkan untuk pengembangan selanjutnya :

1. Mengembangkan sistem cerdas berbasis *Adaptive Case Based Reasoning* yang memiliki kemampuan untuk mempertahankan *case* yang berguna dan menghapus *case-case* yang salah secara otomatis untuk meningkatkan efektifitas pemeliharaan *case* sehingga fase *retrieval* menjadi lebih cepat dan efisien.
2. Mengembangkan sistem cerdas baik berbasis CBR atau *Adaptive CBR* yang mampu menghitung kemiripan antar *case* walaupun deskripsi gejala penyakit yang diinputkan merupakan sinonim dari deskripsi gejala penyakit yang tersimpan di *case memory*.



## 5. DAFTAR RUJUKAN

- Ahn, H., & Kim, K.-j., 2009. Bankruptcy Prediction Modeling With Hybrid Case-based Reasoning and Genetic Algorithms Approach. *Applied Soft Computing*, 9 (2),pp.599-607.
- Balasubramaniam, A., Boyle, R. A., & Voulvoulis, N., 2007. Improving Petroleum Contaminated Land Remediation. *Chemosphere*, 66 (5),pp.791-798.
- Direktorat Jenderal Peternakan. ,2010. *Blue Print Program Swasembada Daging Sapi 2014*. Jakarta: Kementrian Pertanian.
- Ellenius, J., & Groth, T., 2008. Dynamic Decision Support Graph—Visualization of ANN-Generated Diagnostic Indications of Pathological Conditions Developing Over Time. *Artificial Intelligence in Medicine* , 42 (3),pp.189-198.
- Iqbal, N., & Ashraf, M. H. 2006. *Evaluation of jCOLIBRI*. Master. Vasteras: Malardalen University.
- Lees, B., & Corchado, J., 1999. Integrated Case-Based Neural Network Approach to Problem Solving. In: *5th Biannual German Conference on Knowledge-Based Systems: Knowledge-Based Systems - Survey and Future Directions*, Würzburg. Germany 3-5 March 1999. Springer: London.
- Pal, S., 2004. *Foundations of Soft Case-Based Reasoning*. Hoboken: Wiley-Interscience publication.
- Purwanto, G. D., 2011. *Penerapan Algoritma Artificial Neural Network Pada Sistem Cerdas Untuk Pendeteksian dan Penanganan Dini Penyakit Sapi*. Sarjana. Surabaya: Institut Teknologi Sepuluh Nopember
- Salamó, M., & Sánchez, M.L., 2010. Adaptive Case-based Reasoning Using Retention And Forgetting Strategies. *Knowledge-Based Systems* , 24 (2),pp.230-247.
- Subroto, I. M., 2007. *Case-based Retrieval On Question Items Generation*. Master. Kuala Lumpur: Universiti Teknologi Malaysia.