

# CASE-BASED REASONING UNTUK MENJAGA MUTU PELAYANAN PURNA JUAL OTOMOTIVE

Mochammad Choiri<sup>1</sup>, Purnomo Budi Santoso<sup>2</sup>

<sup>1,2</sup>Universitas Brawijaya, Fakultas Teknik, Jurusan Teknik Industri, Malang 65145, Indonesia

**ABSTRACT** Customer satisfaction was the main business factor. After sales service was one of the most important to ensure customer satisfaction. The main criteria for customer satisfaction we know as speed, fit for use, and fair. Human always use analogical method for make some decisions, such as use historical experience for fit the similar problem. If the old case in after sales service could be saved dan organized efectively and sistematix, so we can use it to solve the new problem that have similar type. Method that use old case as a reference to solve new proble in Artificial Intelligent (AI) we always call Case Based Reasoning (CBR). This paper use CBR System for car sales after service problem. We also use Esteem software for improvement tools. After use this tools we get some resume that process in sevice become effective and efficient.

**Keywords:** Artificial Intelligent, Case Based Reasoning, Customer Relationship Management, Maintenance

## 1. PENDAHULUAN

Persaingan bisnis saat ini tak hanya bertumpu pada kualitas produk, melainkan juga pada kualitas layanan. Hal ini tidak berarti produk berupa barang atau jasa yang ditawarkan boleh yang tidak bermutu sekalipun layanannya baik, tetapi terletak pada keseimbangan antara kualitas produk dengan layanan yang diberikan pada pelanggan. Oleh karena itu, kualitas layanan, terutama layanan purna jual, tidak hanya terletak pada hasil akhir layanan itu saja, tetapi terletak pada proses, yaitu ketepatan dan kecepatan layanan yang sangat diharapkan oleh pelanggan.

Manusia biasanya sering menyelesaikan masalah secara analogi, artinya dalam menyelesaikan masalah sering mengacu pada penyelesaian masalah yang sudah pernah terjadi dan mempunyai kemiripan (*similarity*). Metode penggunaan kasus sebagai referensi langsung penyelesaian masalah dalam bidang *artificial intelligence* (AI) disebut *Case Based Reasoning* (CBR) atau dalam bahasa Indonesia disebut Penalaran Berdasarkan Kasus. CBR mengadaptasi solusi atau jawaban masalah sebelumnya untuk digunakan dalam pemecahan kasus baru. Menurut Kolodner (1993), CBR telah terbukti menjadi sebuah pendekatan yang sangat efektif terhadap permasalahan yang aturan-aturannya tidak memadai [1].

Banyak data dan informasi tertidur dalam filing cabinet atau dalam sistem database dari suatu perusahaan maupun industri. Bila data dan informasi ini secara cerdas dapat diorganisasikan dan dimanfaatkan sebagai basis pengetahuan, maka jadilah suatu aplikasi CBR yang hebat.

## 2. CASE BASED REASONING

CBR adalah metode penyelesaian masalah baru dengan menggunakan solusi masalah lama yang mempunyai kemiripan, jadi CBR berbasis metode berpikir secara analogi. CBR banyak dipakai untuk membangun sistem komputer yang cerdas [1], [2], [3].

Proses aplikasi CBR digambarkan melalui empat putaran proses yang disebut dengan *Four Re's* (4 R), yaitu [3] :

- Retrieve the most similar case* (mencari kembali kasus lama yang paling serupa dengan kasus baru).
- Reuse the case to attempt to solve the problem* (menggunakan kembali solusi kasus lama untuk menyelesaikan kasus baru).
- Revise the proposed solution if necessary* (jika diperlukan, lakukan adaptasi dan revisi atas solusi lama yang diusulkan agar sesuai dengan situasi sekarang).
- Retain the new solution as a part of a new case* (apabila solusi pada langkah nomor 3 dianggap solusi baru, maka solusi ini disimpan/ditambahkan kedalam database kasus untuk digunakan menyelesaikan masalah baru dimasa yang akan datang). Dengan demikian,

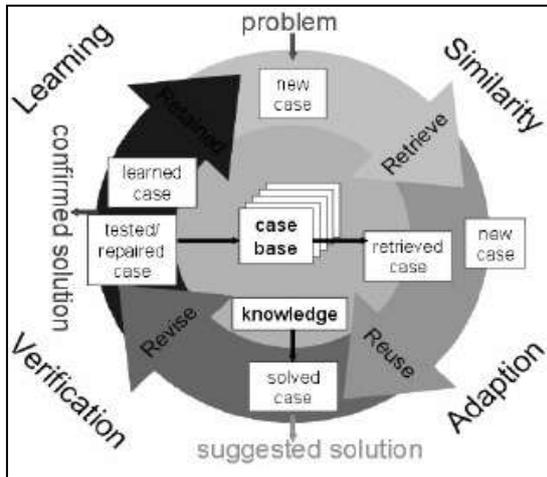
\* Corresponding author: Mochammad Choiri, Purnomo Budi Santoso

[psti.choiri@yahoo.com](mailto:psti.choiri@yahoo.com)

Published online at <http://JEMIS.ub.ac.id>

Copyright ©2015 JTI UB Publishing. All Rights Reserved

CBR ini makin lama makin cerdas mengingat sistem ini dapat menyimpan hal yang baru dan dapat mengikuti perkembangan kasus baru terus menerus. Keempat proses 4Re ini saling mempengaruhi satu sama lain dan dapat digambarkan seperti pada Gambar 1.



Gambar 1. Siklus Case Based Reasoning [3]

### 3. STRUKTUR KASUS PADA CBR

Kasus (*case*) merupakan elemen pengetahuan primer pada aplikasi CBR. Kasus menggambarkan suatu pengalaman. Kasus berisi kumpulan ciri-ciri (*features*) beserta nilainya. Struktur kasus mempunyai ciri-ciri yang dapat dikelompokkan menjadi tiga kelompok berikut [4]:

- Descriptive Feature*, ciri-ciri yang paling mudah untuk mengelompokkan kasus. Pada umumnya : nama, nomor identitas, diskripsi penjelas lainnya, misal nama orang / barang, jenis kelamin, jenis penyakit, jenis gangguan, merek mobil, warna dan lain-lain.
- Adjustment Feature*, ciri-ciri yang bersifat dinamis yang sangat menentukan solusi kasus. Misalnya untuk penyembuhan penyakit, berat badan, umur manusia atau untuk perbaikan mesin adalah kondisi suku cadang mesin. *Ajustment feature* ini akan digunakan untuk mencari kembali kasus lama yang paling mirip dengan kasus baru dan untuk menyesuaikan solusi lama agar tetap *up to date*.
- Solution Feature*, yaitu ciri-ciri yang menunjukkan jawaban. Misalnya kasus orang sakit , obat apa yang diberikan dan berapa takarannya, kasus perbaikan mesin, mengganti suku cadang apa yang diperlukan.

Berikut ini contoh kasus seorang dokter yang telah lama praktek, atas dasar kasus yang pernah dihadapi puluhan tahun, maka dapat disusun database kasus yang terdiri dari ribuan kasus pasien yang telah ditanganinya selama ini (misal selama 20

tahun praktek). Contoh kecil database kasus diagnosis penyakit yang sangat sederhana dan fiktif dapat dilihat pada Tabel 1.

Tabel 1. Contoh Database Struktur Kasus Diagnosis Penyakit

Descriptive Feature			Adjustment Feature		Solution Feature		
Nama	Jenis kelamin	Penyakit	Berat	Umur	Obat	Dosis	Takaran
			Badan	Tahun			
Ana	P	Demam	50 kg	18	Panadol	3x sehari	1 tablet
Antok	L	Demam	30 kg	8	Panadol	3x sehari	½ tablet
Alex	L	Demam	70 kg	70	Panadol	3x sehari	1 tablet
Dani	L	Diare	65 kg	23	Diapet	2x sehari	1 tablet
Elisa	P	Maag	90 kg	25	Promag	1 x sehari	3 tablet

### 4. KESERUPAAN

CBR berdasarkan metode berpikir analogi/kemiripan (*similarity*), yaitu seberapa mirip suatu kasus lama dengan kasus baru. Pencarian kembali (*Retrieval*) kasus-kasus lama yang mirip dengan kasus baru sangat tergantung kepada cara mengukur keserupaan antar fitur. Secara umum dikenal dua cara proses *retrieval* secara komersial, yaitu: *nearest neighbor retrieval* dan *inductive retrieval*. Berikut ini akan dijelaskan konsep *nearest neighbor* secara sederhana, misalnya terjadi pada kasus peminjaman (loan) pada suatu bank. Lihat gambar x, Kasus (Case) A dan B adalah kasus lama, kasus T (target) adalah kasus baru. Kasus mana (A atau B) yang lebih mirip dengan kasus T? Untuk itu perlu dihitung jarak target terhadap sumber dapat menggunakan grafik yang sama, tetapi dengan skala yang berbeda, sehingga formula *nearest neighbor* berubah menjadi :

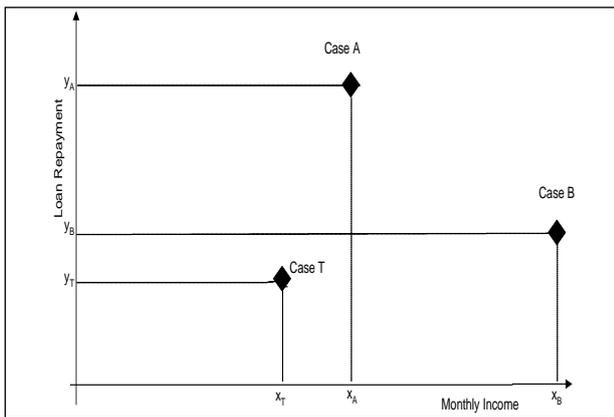
$$\text{Jarak T dari A} = dA = \sqrt{|w_x(x_T - x_A)|^2 + |w_y(y_T + y_A)|^2}$$

$$\text{Jarak T dari B} = dB = \sqrt{|w_x(x_T - x_B)|^2 + |w_y(y_T + y_B)|^2}$$

Dengan notasi:  $w_x$  = bobot dari atribut x,  $w_y$  = bobot dari atribut y, d = distance

Sumbu X adalah *monthly income* (pendapatan bulanan), sumbu Y adalah *loan repayment* (derajat pembayaran hutang kembali). Jadi seorang nasabah yang akan pinjam (hutang) dinilai dari sisi penghasilan dan histori bagaimana ia mengangsur hutangnya di masa lalu (kasus lama).

Dengan pembobotan atribut ini akan terjadi kemungkinan perubahan jarak kedekatan antara T dari A dan T dari B. Sebagai contoh untuk menggambarkan keadaan ini dapat dilakukan dengan menggambar ulang grafik yang telah ada dengan sumbu x n kali skala sumbu y seperti terlihat pada Gambar 2.



Gambar 2. Kekeragaman dengan pembobotan

### 5. IMPLEMENTASI CBR UNTUK MENJAGA MUTU PELAYANAN PURNA JUAL OTOMOTIVE

#### Identifikasi Kasus

Kasus-kasus yang diangkat sebagai data adalah pengalaman para teknisi perawatan otomotif, khususnya mesin bensin. Identifikasi gangguan mesin otomotif yang sering terjadi biasanya sebagai berikut :

- Mesin sulit dihidupkan
- Motor starter mati
- Mesin sulit dihidupkan jika masih dalam keadaan panas
- Mesin dapat dihidupkan tapi mati lagi
- Mesin dapat hidup tapi tersendat-sendat
- Akselerasi kurang
- Mesin mendadak mati
- Mesin tersendat pada putaran tinggi
- Motor starter mati hanya berbunyi klik, dan gangguan yang lainnya.

Gangguan mesin otomotif berbahan bakar bensin biasanya terjadi akibat terganggunya sistem pengapian, sistem pengaturan bahan bakar dan keausan sistem torak silinder. Kasus-kasus seperti diatas beserta solusinya, yang menumpuk dibagian *customer service*, apabila dikumpulkan dan diorganisasikan secara cerdas dalam database kasus, misalnya menggunakan alat Esteem, akan menjadi basis solusi (*knowledge base*) untuk mengatasi kasus gangguan mesin otomotif yang terjadi kemudian. Dengan demikian mutu pelayanan akan terjaga dan tidak berbeda antara customer yang satu ke customer yang lain.

#### Desain Struktur Kasus untuk Perawatan Otomotive

- Descriptive Feature*: berguna sebagai pengenalan kasus, dalam hal ini berupa jenis gangguan dan penyebabnya.

b. *Adjustment Feature*: fitur yang sangat mempengaruhi solusi, meliputi keadaan bagian-bagian system mesin yang mengalami gangguan atau kerusakan fungsinya.

c. *Solution Feature*: fitur-fitur yang merupakan solusi kasus, meliputi penyelesaian masalah yang terjadi atau saransaran perbaikan yang diperlukan.

Desain struktur kasus untuk perawatan otomotive dapat dilihat pada Tabel 2.

Tabel 2. Struktur Database Kasus untuk Perawatan dan Perbaikan Mesin Bensin

DESCRIPTIVE FEATURE		ADJUSTMENT FEATURE							SOLUTION FEATURE		
Jenis	Brand	Keadaan							Diagnosis	Tindakan	
Gangguan	Name	Umur Mesin	Tega-ngan Accu	Koil	Distributor	Busi	Dinamo ampere	Pompa Bahan Bakar	Karburator	Gangguan	Perbaikan
Mesin distart sulit hidup	Toyota	8	13	Normal	Baik	Normal	Normal	Normal	Normal	Kondisi & Posisi delko	Atur posisi/ ganti delko
Motor starter lemah	Da-hatsu	6	9	Normal	Baik	Nyala api kecil	Normal	Normal	Normal	Tegangan accu drop	Charge/ ganti baru
Mesin hidup lalu mati	Da-hatsu	10	11	Normal	Baik	Nyala api kecil	Tidak normal	Normal	Normal	Dinamo koersling	Ganti baru
Mesin distart Tdk hidup	Toyota	7	13	Tidak normal	Baik	Nyala api kecil	Normal	Normal	Normal	Kabel danKondisi koil	Tes koil rusak ganti
Akselerasi Mesin kurang	Da-hatsu	5	13	Normal	Buruk	Buruk	Normal	Normal	Tidak normal	Plna, busi & karburator kotor	Bersihkan karbur.ganti yang lain
Jalan mesin tersendat	Toyota	6	13	Normal	Baik	Normal	Normal	Tersen-dat	Tidak normal	Saluran, file Karburator kotor	Ganti fil-r, Ter bersih kan y g lain
Mesin mendadak mati	Da-hatsu	5	13	Normal	Baik	Normal	Normal	Normal	Tidak normal	Sel karburator & Saensi b.b saluran nya	Perbaiki atau ganti
Motor starter mati	Da-hatsu	7	13	Normal	Baik	Normal	Normal	Normal	Normal	Sakelar & saluran listrik putus	Perbaiki atau ganti baru
Mesin distart Sulit hidup	Toyota	9	10	Normal	Baik	Nyala api kecil	Normal	Normal	Normal	Accu drop, Busi aus	Ganti yang baru

### 6. METODE PENGEMBANGAN APLIKASI CBR DENGAN ESTEEM

*Esteem* adalah salah satu alat pengembang aplikasi *Case Based Reasoning* dengan berbagai macam tujuan seperti, diagnosis penyakit, *trouble shooting* permesinan, peramalan cuaca, pengaturan lalu lintas pesawat terbang, pencampuran komposisi logam dan lain-lain. Secara spesifik, pengembangan sistem CBR memakai *software Esteem* akan melalui tahapan-tahapan sebagai berikut :

- Dimulai dengan mendefinisikan kasus yang dipilih.
- Memasukkan nama kasus.
- Memasukkan nama-nama *feature* dan tipe-tipe *feature*
- Mendefinisikan keserupaan (*Similarity Definition*) dengan memasukkan nama-nama *feature*.
- Memasukkan tipe keserupaan, *threshold*, memilih tipe *Feature Matching & Weight / Rule Base Name*
- Memasukkan kasus-kasus lama sebagai data (*casebase*)
- Memakai aplikasi *Interface*.
- Memilih *Target Case Entry Feature, Retrieved Case Feature, and Case Feature*

- i. Run User Interface
- j. Memasukkan kasus baru sebagai target (*Target Case*)
- k. Retrieve Application
- l. Memilih kasus yang serupa dari daftar kasus yang tersedia.
- m. Melakukan adaptasi bila diperlukan.
- n. Bila tidak dilakukan adaptasi, proses dianggap selesai dari hasil solusi yang didapat dari Retrieve Application.

**Contoh Perancangan Prototipe Aplikasi CBR**

Prosedur yang dilakukan membuat prototipe program aplikasi untuk *trouble shooting* mesin otomotif dengan alat *Esteem* menggunakan metode *Case Based Reasoning* akan menampilkan aplikasi-aplikasi dari *Esteem* sebagai berikut :

- a. Case Based Definition Editor
- b. Similarity Definition Editor
- c. Cases Editor
- d. User Interface Editor
- e. Browser Editor dan
- f. Run User Interface



Gambar 3. Top Menu dari Esteem

**Case Based Definition Editor**

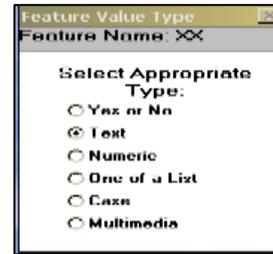
*Editor* pertama yang dipakai untuk membangun aplikasi *Case Based Reasoning* adalah *Case Based Definition Editor*. *Editor* ini dipakai untuk mendeklarasikan komponen-komponen dasar dari sistem, yaitu *feature* apa saja yang digunakan dalam suatu kasus. *Features* yang dideklarasikan dalam *Case Based Definition Editor* ini harus mempunyai nama dan tipe yang sesuai. Tipe fitur dapat dipilih dari tipe menu yang disediakan oleh *Esteem*, dan tampilan *Case Based Definition Editor* ini seperti pada Gambar 4.



Gambar 4. Case Based Definition Editor

Memilih tipe fitur dapat dilakukan dengan cara mengklik kanan daerah *Feature Value*

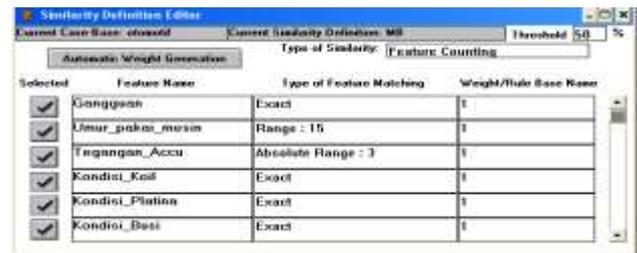
*Type* pada *Feature Name* yang sesuai, maka akan keluar jenis pilihan *Feature Value Type*, kemudian dipilih yang sesuai satu dari beberapa pilihan yang ada. Bila setiap nama fitur telah diberi tipe yang sesuai, maka pengisian *Case Based Definition* adalah lengkap. Gambar 5 menunjukkan tampilan dari *Feature Value Type*.



Gambar 5. Feature Value Type

**Similarity Definition Editor**

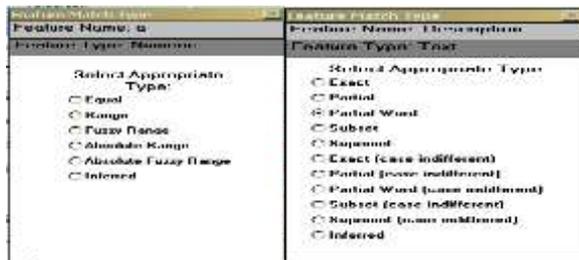
Tahapan yang kedua dalam pengembangan proses adalah bagaimana kasus-kasus akan dicari kembali berdasarkan pada *Case Based Definition*. Dengan *Similarity Definition Editor* untuk menetapkan metode dan ukuran untuk menentukan keserupaan pencarian kembali (*retrieval*).



Gambar 6. Similarity Definition Editor

Dilihat dari gambar diatas, editor ini mempunyai lima tampilan penting, yaitu :

- a. Kolom pertama dalam editor menunjukkan *selected*, ini digunakan untuk memberitahu sistem, *feature name* mana saja yang digunakan untuk *similarity matching* (pencocokan yang mirip).
- b. Tampilan nama (*feature name*) akan dimunculkan dan diikuti oleh bidang disisi kanannya untuk tampilan tingkat kecocokan (*Type of Feature Matching*). Pengembang akan mengklik kanan mouse dalam bidang ini untuk menampilkan menu “jenis-jenis fitur yang cocok” (*Types of Feature Matching*). Jenis fitur yang paling cocok satu sama lain sifatnya tergantung dari jenis fitur tersebut.

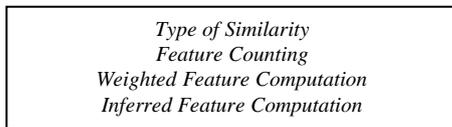


Gambar 7. Feature Match Type



Gambar 8. Case Editor

- c. Kolom sebelah kanan *editor*, yang bersebelahan dengan *scrollbars*, digunakan untuk memberikan kisaran yang menjelaskan hal yang penting yang saling berhubungan dari beberapa fitur dalam proses *retrieval* atau sebuah nama berbasis aturan (*rule base*) yang diberikan. Nama yang berbasis aturan adalah sebuah petunjuk terhadap sebuah koleksi (kumpulan) aturan-aturan untuk menentukan bagaimana pentingnya fitur tersebut dalam proses *retrieval*.
- d. Bidang tipe kemiripan (*Field Type of Similarity*) digunakan untuk menentukan teknik keserupaan (*similarity*) apa yang akan digunakan dalam sebuah dasar kasus perkasus. Pengembang meng-klik kanan untuk menampilkan sebuah menu teknik keserupaan yang dapat digunakan, *Esteem* menyediakan tiga tipe keserupaan (*similarity*) tersebut sebagai berikut :



- e. *Thresold* (ambang batas) dapat diatur sedemikian rupa untuk digunakan selama proses *retrieval*. *Threshold* menunjukkan nilai prosentasi dari keserupaan (*similarity*). Keserupaan ditawarkan kepada pemakai sebagai nilai kemiripan yang berkisar antara 0 sampai dengan 100. *Thresold* dimungkinkan sebagai pembatas untuk kasus yang dicari kemiripannya (*retrieved*).

**Case Editor**

*Case Editor* adalah editor yang digunakan untuk memasukkan data suatu kasus dengan solusinya. Apabila editor ini sudah diisi data lengkap, maka tombol "Save Case" digunakan untuk menyimpan data kasus tersebut. Setiap kasus yang dimasukkan dalam editor ini harus diberi nama sendiri-sendiri yang berbeda antara kasus satu dengan kasus lainnya, misalnya *Case Name* : MB1, MB2, MB3 dan seterusnya. Tampilan dari *Case Editor* dapat dilihat pada gambar berikut.

**End User Interface Editor**

*End-User Interface Editor* merupakan editor yang digunakan untuk memilih *feature* apa saja yang akan ditampilkan pada saat aplikasi di eksekusi. Pada saat eksekusi akan ada tiga *windows* yang muncul, yaitu *Target Case Entry Feature*, *Retrieved Case Feature*, dan *Selected Case Feature*. Dalam *End-User Interface Editor* ini juga terdapat tombol 'Additional End User Functionally' yang berfungsi untuk mendeklarasikan aspek penting dari *end-user interface*. Pada Gambar 9 *feature* yang diblok biru merupakan *feature* yang akan ditampilkan pada masing-masing layar.



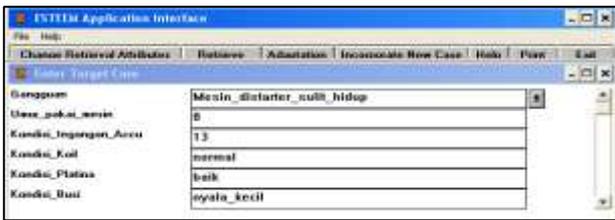
Gambar 9. End-User Interface Editor

**Menjalankan Aplikasi (Run user Interface)**

Untuk menjalankan aplikasi yang telah dibuat dapat menggunakan tombol 'Run' atau dengan mengklik option menu kemudian klik pilihan *execute*, maka layar yang akan muncul ada tiga, yaitu :

**a. Target Case Entry Screen**

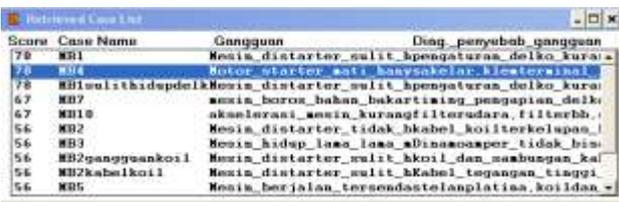
Merupakan layar dimana *user* dapat mengisikan nilai *feature* dari kasus yang baru. Setelah nilai *feature-feature* yang diperlukan diinputkan maka dapat dilakukan pencarian dengan menekan tombol 'Retrieve'. Bentuk *Target Case Entry Screen* dapat dilihat seperti pada Gambar 10.



Gambar 10. Target Case Entry Screen

**b. Retrieved Case List Screen**

Merupakan layar yang muncul setelah menekan tombol 'Retrieve'. Pada layar ini akan ditampilkan nama dari kasus-kasus lama beserta nilai score kasus itu dibandingkan dengan kasus target. Tampilan *Retrieved Case List Screen* seperti pada Gambar 11.



Gambar 11. Retrieve Case List Screen

**c. Selected Case Screen**

*Selected Case Screen* adalah layar tampilan dimana pengguna dapat mengamati dan membandingkan antara kasus barunya dengan kasus hasil *retrieval* dengan mempertimbangkan nilai dari solusi hasil *retrieval* tersebut. Selain itu pengguna juga dapat memasukkan kasus baru yang sudah diadaptasi ini kedalam *case base system*; dengan cara ini sistem dapat belajar atau menambah jumlah data kasus secara terus menerus yang kemudian dapat digunakan untuk menjawab kasus-kasus baru yang akan datang.



Gambar 12. Selected Case Screen

**7. KESIMPULAN**

Beberapa kesimpulan yang dapat diperoleh dalam penelitian ini adalah sebagai berikut:

- Banyak kasus tersimpan nganggur didalam laci filing cabinet atau dalam sistem database suatu perusahaan, padahal kasus-kasus itu merupakan pengetahuan yang luar biasa bermanfaat untuk solusi kasus baru.
- CBR menggunakan solusi kasus lama untuk solusi kasus baru yang memiliki kemiripan.

- CBR mempunyai 4 siklus yaitu: *Retrieve, Reuse, Revise, dan Retain*.
- Pelayanan purna jual, misalnya dalam hal automotive, penuh dengan pemanfaatan kasus-kasus lama untuk solusi kasus baru.
  - Dengan CBR, pelayanan purna jual otomotive (mobil) dapat dijaga ketepatan, kecepatan layanan, mutu dan keajegannya.

**DAFTAR PUSTAKA**

- Kolodner, J.L. 1993. "Case-Based Reasoning", Morgan Kaufmann Publisher Inc., San Francisco, California.
- Riesbeck, C., dan Schank, R. 1989. *Inside case-based reasoning*. Lawrence Erlbaum, New Jersey.
- Aamodt, A. dan E. Plaza. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Communications, IOS Press, Vol 7:1,halaman 39-59
- Stottler, Richard H. 1994. *CBR for Cost and Sales Prediction*. AI Expert August 1994
- Watson, Ian. 1997. *Applying Case-Based Reasoning : Technique for Enterprise System*. San Francisco: Morgan Kaufmann Publisher Inc.

Lampiran 1 Contoh Gangguan Mesin Berbahan Bakar Bensin

<i>DESCRIPTIVE FEATURE</i>		<i>ADJUSTMENT FEATURE</i>								<i>SOLUTION FEATURE</i>	
Jenis Gangguan	Brand Name	Keadaan Komponen Mesin								Diagnosis Gangguan	Tindakan Perbaikan
		Umur Mesin (tahun)	Tegangan Accu	Koil	Distributor arus listrik (delco)	Busi	Dinamo	Pompa Bb,	Karburator		
1. Dinamo starter mati	A	5	Normal 12 V	normal	normal	normal	normal	Normal	Normal	1. Saklar pengapian Rusak. 2. Sambungan accu putus atau longgar	1.Ganti dengan yang baru 3.Perbaiki saluran kabel starter-accu
2. Starter hidup, mesin tidak mau hidup	B	4	Normal 12 V	normal	Penutupnya panas	Api kecil	normal	normal	normal	1.Platina terbakar atau berlubang 2.Terminal-terminal Delco berkarat (kotor) 3.Tutup koil dan kabel-kabel bertegangan tinggi basah 4.Kumparan koil rusak 5.Tahanan primer tidak tepat	1.Ganti dengan yang baru 2. Bersihkan atau ganti yang baru 3.kabel keringkan dan cari penyebab Basahnya kabel 4. Cabut kabel koil, bila tak ada listrik, ganti koil. 5.Periksa penurunan tegangan pada kabel, bila tegangan kecil, ganti dengan yang baru
3. Mesin sulit dinyalakan	B	5	Normal 12 V	normal	normal	Api kecil	normal	Normal	Normal	1.Api busi lemah 2. Rumah busi pecah atau rusak. 3.Sambungan sirkuit terbuka atau korsleting dengan massa	1.Periksa platina dan sirkuit primer, mungkin salah sambung. 2.Ganti baru 3.Tes aliran listrik antara terminal koil dan dengan massa
4.Boros bensin	B	5	Normal 12 V	normal	Kurang normal	kurangnormal	normal	Normal	Normal	1.Terminal Delco kotor atau stelan kurang tepat 2.Timing pengapian kurang tepat. 3.Busi kotoratau stelan gap kurang tepat	1.Bersihkan, stel lagi atau ganti baru. 2. Di stel menurut ketentuan. 3. Bersihkan dan stel gap atau ganti busi baru