

Sistem Tiket Pengaduan Berbasis *Case-Based Reasoning* Untuk Meningkatkan Pelayanan Pengaduan

Arpan Dadi

Abstract—With the technological use growth of computer as a means of assist in finishing work in company which progressively mount fast, what do not in following behaviorally is good in using computer technology, and its minimize is human resource which with quality vertalis in a company, will be happened by the delay in course of accurate and is not solution in finishing solution. System of *Case-Based Reasoning (CBR)* that is technique of is solving of problem of pursuant to *Case-Based* of passing process Retrieve, Reuse, Revise, Retain to be formed by new knowledge in course of solving of solution. Applying CBR have a lot of weared in solution seeking by exploiting or previous proses case data. CBR own five duty in forming knowledge that is case representation, case Index, case retrieval, case adaptation, and case maintenance. At case adaptation represent the process of case seeking which own of speed and accuracy in seeking of new case-based and knowledge at system of solution ticket. This research to explain about applying CBR in system of solution ticket to process the solving of solution quickly and accurate. Where process of inclusion knowledge represent the important step in course of solving of solution ticket

Keyword: *Case-Based Reasoning, Case Adaptation, Solving of solution.*

1. PENDAHULUAN

Pada saat ini penggunaan komputer membutuhkan akan sumber daya manusia (SDM) teknologi informasi (TI) yang bisa menjembatani pengguna komputer (*user*) dan teknologi komputer itu sendiri apabila ada masalah dalam penggunaannya.

Survey yang di lakukan oleh IPKIN (Ikatan Profesi Komputer dan Informatika Indonesia) dengan SEARCC (*South East Asia Regional Computer Conferedations, 1990*), 100 perusahaan yang ada di Jakarta bahwa perusahaan rata-rata belum sepenuhnya memberdayakan SDM TI secara maksimal, ini di sebabkan tidak padunya unsur pendidikan, pelatihan dan program pemerintah dalam memenuhi kebutuhan perusahaan akan SDM TI. Persoalan keterlambatan dalam proses pengaduan masalah TI yang bisa mengkombinasikan kompetensi dan keahlian teknis disertai pengalaman bisnis dan kemampuan memberikan solusi komprehensif atau versatilis (Wahono, R.S, 2006) sedangkan untuk menjadi seorang versatilis proses tahapannya memerlukan waktu yang lama dalam pembentukannya, menjadi kurangnya SDM TI yang

bersifat versatilis dan akan mengakibatkan proses16 pengaduan dalam permasalahan komputer akan terjadi proses degrading dan keterlambatan dalam hal pengaduan permasalahan TI untuk mendapatkan solusi secara cepat dan menyeluruh dalam konteks ke ilmunan TI dan alur bisnis perusahaan itu sendiri, karena kurang di dukung oleh SDM TI yang mampu memberi solusi secara teknis dan proses bisnis secara komprehensif, proses itu akan berlangsung lama sebanding proses pembentukan SDM TI itu sendiri. Dan untuk mempermudah proses pengaduan masalah TI di butuhkan suatu sistem tiket pengaduan yang sudah terkomputerisasi dengan kategori, keilmuan, fakta, teknik dalam mengambil keputusan yang bersifat *artificial intelligence*, dan solusinya, (Marimia, 1992), yang di ambil dari data kemampuan, pengetahuan dan pengalaman para ahli pada teknis masalah dan sistemnya berjalan secara otomatis, sehingga tidak terjadi ketergantungan kepada seorang ahli itu sendiri, dalam membangun sistem tiket pengaduan di perlukan suatu langkah yang strategi dan sistematis yang sesuai dengan bisnis proses di mana sistem tiket pengaduan itu di terapkan, karena sistem tiket pengaduan harus di lengkapi dengan syarat, tampilan sistem yang mudah di pahami (*user interface*), mencari kesimpulan masalah dari *knowledge-based* serta mempunyai *service quality* yang baik (Davidoff, 1994) Dan dari sistem tiket pengaduan yang bersifat *artificial intelligence* maka di perlukan data masukan dari user dan ahli yang berdasarkan permasalahan, pengetahuan, pengalaman, dan solusi yang sudah di lakukan oleh seorang ahli di tampung menjadi *knowledge-based* yang nanti akan di manfaatkan menjadi solusi permasalahan yang di hadapi oleh pengguna sistem tiket pengaduan, di dalam sistem tiket pengaduan terbentuk *Case-Based Reasoning (CBR)* yang menjadi modal dasar untuk memproses permasalahan yang ada, hingga menemukan pemecahan masalah dari pengaduan, *case-based reasoning* sangat di perlukan dalam sistem tiket pengaduan karena sistem tersebut sangat di pengaruhi oleh *case-based* yang masuk dan akan berkembang terus dalam penyelesaian masalah baru dengan mengadopsi penyelesaian masalah - masalah lama. (Riesbeck & Schank, 1989) dan membuat *knowledge* baru bagi sistem itu sendiri atau seorang ahli di bidangnya, agar terdapat proses untuk belajar dan mengingat (Aamodt & 17 Plaza, 1994: p39) di dalam

CBR terdapat ada 4 tahapan proses yang terbentuk dalam mengambil suatu kasus hingga membentuk satu pemecahan yang berdasarkan pengalaman yang sudah ada, adapun empat tahapan tersebut adalah: mengambil (*Retrieve*) menggunakan kembali (*Reuse*), perbaikan (*Revisi*), mempertahankan (*Retain*). Sehingga dalam penelitian ini sudah selayaknya di buat suatu sistem tiket pengaduan berbasis *Case-Based Reasoning* yang bersifat *artificial intelligence*, yang perlu dilandasi dengan *knowledge* para ahli untuk berbagi pengalaman kasus dan solusi sehingga menghasilkan suatu metode, atau cara – cara baru dalam menyelesaikan masalah yang kompleks serta seiring pesatnya perkembangan teknologi informasi.

2. METODE PENELITIAN

2.1 Analisa

Metode analisa data penelitian ini, di bagi dalam beberapa variable, yaitu variable dependen (Y) dan variable independen (X)

- Variable Dependen (Y) adalah: Variable yang nilainya tergantung pada variable yang lain, dalam hal ini variable dependen bersifat kinerja serta penentuan hasil analisa.
- Variable Independen (X) adalah: Variable yang nilainya mempunyai kebebasan yang nanti hasilnya mempengaruhi variable *dependen* diantaranya,

variable motivasi (X_1), variable persepsi (X_2),

variable pembelajaran (X_3), variable kepribadian

(X_4), variable sikap (X_5) Sedangkan untuk

mengkomparasi variable yang ada dengan metode lain harus memperhatikan jenis permasalahan yang sama dalam penentuan nilai, agar terdapat titik temu dalam hasil akhir yang di inginkan seperti:

- Motivasi (X_1): adalah keinginan untuk mencapai capaian tertentu dengan memperhitungkan potensi sumber daya yang ada.

- Persepsi (X_2): adalah asumsi-asumsi pernyataan

sikap, atau pandangan dengan mengambil data dari fakta-fakta di lapangan untuk menghasilkan sebuah nilai yang di inginkan.

- Pembelajaran (X_3): adalah satu proses yang

terjadi di lapangan yang tahapannya memerlukan waktu yang lama guna menghasilkan suatu nilai yang di inginkan.

- Kepribadian (X_4): adalah satu sikap atau ciri-ciri

tertentu yang di punyai oleh setiap orang atau unit yang di gunakan sebagai objek penelitian dalam penentuan nilai yang di inginkan.

- Sikap (X_5): adalah keputusan yang di ambil

dalam penentuan satu masalah yang akan menghasilkan nilai tertentu.

- Keputusan (Y): adalah nilai akhir yang di ambil melalui proses yang panjang dengan mempertimbangkan data dan fakta yang ada. Cara pengukuran validitas angket kompetensi menggunakan teknik korelasi dengan *r pearson* atau koefisien korelasi produk momen pearson dengan taraf signifikan 5% yaitu dengan rumus

$$R = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{(n(\sum x^2) - (\sum x)^2)(n(\sum y^2) - (\sum y)^2)}}$$

n adalah banyaknya pasangan pengamatan

$\sum x$ adalah jumlah pengamatan variable x

$\sum y$ adalah jumlah pengamatan variabel y

$(\sum x^2)$ adalah jumlah kuadrat pengamatan variabel x

$(\sum x)^2$ adalah kuadrat jumlah pengamatan variabel x

$(\sum y^2)$ adalah jumlah kuadrat pengamatan variabel y

$(\sum y)^2$ adalah kuadrat jumlah pengamatan variabel y

$\sum xy$ adalah jumlah hasil kali variabel x dan y

Dengan menggunakan rumusan diatas, kita bisa menentukan nilai akan kebutuhan sistem tiket pengaduan berbasis CBR yang di kelompokkan menjadi beberapa bagian.

TABEL 1. PERUMUSAN PERNYATAAN

KETERANGAN	FAVORABLE	UNFAVORABLE
Sangat Setuju	4	0
Setuju	3	1
Abitan	2	2
Tidak Setuju	1	3
Sangat Tidak Setuju	0	4

Pernyataan *favorable* adalah pernyataan yang menyatakan persetujuan akan adanya sistem tiket pengaduan, sedangkan pernyataan *unfavorable* adalah pernyataan ketidaksukaannya akan sistem tiket pengaduan, dah hasil yang di dapat dengan jumlah responden sebanyak 20 orang. Berdasarkan analisis validitas terhadap 15 butir item kuisioner kompetensi, maka semuanya dinyatakan valid. Koefisien korelasi yang diperoleh dari hasil perhitungan terletak pada kisaran- 0,3971 - 0,187 dan berarti pernyataan *favorable* lebih besar nilainya yang berarti kebutuhan akan sistem tiket pengaduan berbasis CBR diperlukan oleh pengguna komputer.

Pada bagian ini akan di lakukan analisa perbandingan, untuk mengetahui antara kasus yang ada, dengan pemasukan kasus yang baru, sehingga akan diketemukan kecocokan jumlah fitur kasus yang ada, masalah yang timbul dalam perbandingan ini adalah untuk menentukan bobot fitur kasus, jadi keterbatasan perbandingan ini adalah apabila jumlah perbandingannya terlalu besar sehingga waktu dan pengambil keputusan yang tepat dalam solusi kasus agak sulit untuk mencari linier perbandingan jumlah kasus. Sebaiknya pendekatan ini lebih efektif bila digunakan pada kasus dasar yang relatif lebih kecil.

Beberapa implementasi CBR sudah menggunakan rumus di bawah ini, untuk mencocokkan penentuan pengambilan kasus semisal, Broadway (Skalk, 1992) untuk mengambil model mobil, *The Compaq SMART system* (Acorn & Walden, 1992) untuk pendukung bantuan costumer produk, serta ANON (Owens, 1993) untuk situasi assisment plan dalam ke gagalan.

Perumusan logika untuk menentukan pencocokan tetangga yang paling terdekat dengan kasus yang ada oleh perangkat lunak adalah sebagai berikut:

1. W bobot kwalitas suatu fitur atau slot,
2. sim adalah fungsi kemiripan
3. FI dan FR adalah nilai nilai untuk fitur

4. i dalam penginputan kasus yang di ambil masing-masing

$$\frac{\sum_{i=1}^n W_i \cdot sim(f_i^T, f_i^R)}{\sum_{i=1}^n W_i}$$

2.1.1 Pengujian Sistem (Testing)

Pengujian sistem (testing) sangat penting untuk suatu jaminan kualitas perangkat lunak yang merupakan representasi dari kajian pokok dari perancangan, analisa, desain, dan pengkodean dan untuk menemukan kesalahan (Berard, 1994) Pentingnya pengujian perangkat lunak serta implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat bisa dipaksakan karena melibatkan rangkaian aktivitas proses pembuatan, yang pasti ada peluang terjadinya kesalahan manusia sangat besar, dan karena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas.

Dan untuk menentukan jaminan kualitas perangkat lunak, ada beberapa cara pengujian yang di lakukan diantaranya : "Whitebox testing" dan "Blackbox Testing".

Dari hasil pengujian di harapkan meningkatnya visibilitas (kemampuan) suatu perangkat lunak yang mempunyai kemampuan sistem yang terstruktur dengan baik, adapun aturan dalam pengujian perangkat lunak meliputi:

- a. Pengujian proses eksekusi suatu program dengan maksud menemukan kesalahan.
- b. *Test case* program dengan cara probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
- c. Pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Berikut beberapa teknik pengujian perangkat lunak dan hasil pengujiannya dengan menggunakan tehnik "White Box" dan "Black Box"

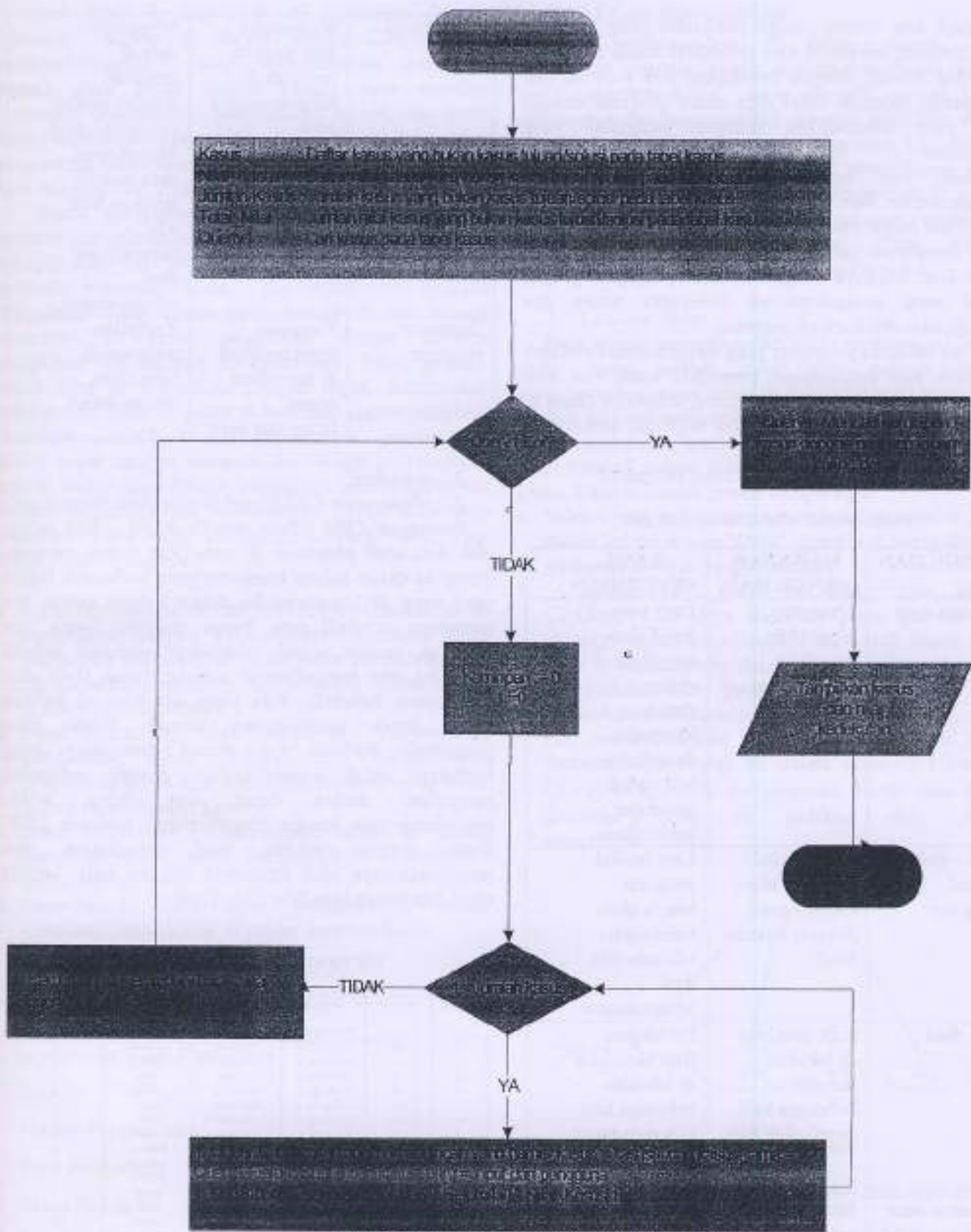
1. White Box Testing

White box testing adalah suatu pengujian yang di lakukan pada saat kontruksi program secara bertahap, *whitebox testing* berfokus pada struktur program control. Atau *Test case* dilakukan untuk memastikan bahwa semua *statemen* pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian *basic path*, tehnik pengujian *white box*, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang

independent secara linear yang akan memastikan cakupan.

- memberikan jaminan bahwa semua jalur *independent* suatu modul digunakan minimal satu kali,
- menggunakan semua keputusan logis untuk semua kondisi *true* atau *false*,
- mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi,
- menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

Sehingga di harapkan sistem program yang terbentuk bisa di minimal kesalahan modul atau pengulangan modul yang tidak di perlukan, berikut hasil pengujian *white box testing*, di gambarkan dalam bentuk *flowchart* di bawah ini



Gambar 1. Flowchart White Box Testing Nilai Kemiripan Kasus

2.1.2 Black Box Testing

Black box testing adalah pengujian yang berfokus pada domain informasi dari perangkat lunak atau *user interface* modul, dengan melakukan *test case* dengan mempartisi domain *input* dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Black box testing merupakan pendekatan komplementer dari teknik *white box testing*, karena pengujian *black box* diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik *white box*. Caranya dengan metode pengujian *graph-based* serta mengeksplorasi hubungan antara dan tingkah laku objek-objek program.

Dan cakupan pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. *GUI (grafik user interface)*, arsitektur *client & server*, dokumentasi dan fasilitas *help* dan sistem *real time* masing-masing.

Berikut tabel pengujian *black box* program:

TABEL 2. HASIL PENGUJIAN *BLACK BOX*

PENGUJIAN	HARAPAN PENGUJIAN	HASIL PENGUJIAN
Inputan data	Data yang di input akan masuk ke dalam <i>database</i> kasus dan <i>knowledge</i>	Data yang di input akan masuk ke dalam <i>database</i> kasus dan <i>knowledge</i> bila sudah mendapat persetujuan
Link-link modul program	Link modul program akan terintegrasi dengan laporan hasil	Link modul program hanya akan terintegrasi bila ada izin dari administrator
Edit data	Edit data bisa di lakukan dengan beberapa kali perubahan data	Perubahan data akan bisa di lakukan beberapa kali bila mendapat izin dari administrator
Laporan data	Data masukan otomatis akan berbentuk laporan aktifitas	Data masukan bisa di buat laporan dengan keinginan user

Mengambil data	Mengambil data yang di cari bisa di lakukan secara acak	Mengambil data di tentukan dengan modul pencarian data
Menghapus data	Menghapus data bisa di lakukan dengan keinginan user	Menghapus data bisa di lakukan bila ada persetujuan dari administrator
Tampilan program	Tampilan program akan di tampilkan sesuai keinginan user	Tampilan program di tampilkan secara detail

2.2 Arsitektur

2.2.1 Penerapan CBR Untuk Sistem Tiket Pengaduan

Penerapan CBR dalam penelitian ini, yaitu pengolahan data hasil penelitian di masukkan dalam rumus sistem ke dalam bahasa komputer yang berbentuk logika yang nanti di komparasi ke dalam bahasa mesin dan outputnya menjadi satu frame program yang diinginkan, berikut contoh *computing approach* sebagai tabel list data pengaduan di sebuah divisi *Help Desk* PT. Parastar Echrindo, data yang ada bisa di jadikan acuan untuk membangun sebuah sistem tiket pengaduan berbasis *Case-Based Reasoning* yang berfungsi untuk mengefektifkan fungsi pelayanan pengaduan secara cepat dan tepat walaupun penyelesaiannya, karena dengan sistem berbasis CBR dapat dengan mudah tiket pengaduan penyelesaiannya bisa terkontrol dengan baik berdasarkan tabel data pengaduan *Div, Help Desk*.

TABEL 3
PENYELESAIAN TIKET PENGADUAN

Kasus	Dept	Alamat Pengaduan	Type Pengaduan	Nama Pengaduan	Waktu Penyelesaian (hari)
1	FA	Gedung 1	Penting	Indah	5
2	FA	Gedung 1	Tunda	Indah	2
3	FA	Gedung 2	Tunda	Indah	3
4	FA	Gedung 1	Penting	Indah	1
5	FA	Gedung 1	Penting	Indah	2
6	GA	Gedung 2	Segera	Putri	2
7	GA	Gedung 3	Segera	Putri	2
8	GA	Gedung 3	Segera	Molus	2
9	GA	Gedung 2	Tunda	Molus	3
10	GA	Gedung 2	Tunda	Molus	4
11	IT	Gedung 4	Tunda	Budi	4
12	IT	Gedung 4	Tunda	Budi	3
13	IT	Gedung 4	Batal	Agus	0
14	IT	Gedung 4	Batal	Agus	0
15	ACC	Gedung 2	Batal	Heri	0
16	ACC	Gedung 2	Batal	Heri	0
17	ACC	Gedung 2	Penting	Leo	4
18	HRD	Gedung 3	Penting	Leo	1
19	HRD	Gedung 3	Segera	Roma	2
20	HRD	Gedung 3	Segera	Roma	2

Dari tabel di atas akan di selesaikan dengan beberapa metode di dalam sistem CBR dengan memperhitungkan *case representation* (penyajian kasus), *case index* (Indek kasus), *case similirty assisment* (Persamaan penyelesaian kasus), dan *case-based maintenance* (memelihara data kasus) dalam tiket pengaduan untuk mencari informasi pengaduan dengan baik dalam divisi *Help Desk*, contoh kasus:

"Suatu pengaduan pada nomor 1 di dalam tabel.3 terdapat pengaduan dari Dept.FA (Finance) di gedung 1 mengadukan hal yang sangat penting dari sumber pengadu bernama "indah", dan selanjutnya terdapat pengaduan yang sama pada nomor 2 dan hingga mencapai 20 kasus pengaduan dengan berbeda pengaduan dan sumber informasi dan lokasi gedung. untuk bisa di selesaikan dengan cepat, dikarenakan ketebatasan SDM IT maka di butuhkan sistem yang bisa mencatat pengaduan dan memberikan penyelesaian secara cepat dengan mencatat *knowledge* penyelesaian, agar di waktu akan datang, pengaduan bisa terselesaikan dengan cepat dengan mengadopsi kasus lama atau membuat *knowledge* penyelesaian yang baru. Dan dengan teknik metode:

- Case representation* dan *index*. penyelesaian kasus pengaduan dengan cara merepresentasikan suatu kasus dan *index* kasus dalam tabel pengaduan agar mendapatkan similirty (persamaan) suatu kasus untuk penyelesaiannya.
- Case matching* dan *retrieval*. Penyelesaian pengaduan kasus dengan cara menstabilkan persamaan kasus agar mendapat formula yang sama dalam penyelesaian kasus
- Case adaptation*. Menkontruksi kasus ke dalam alqoritma tabel guna mendapatkan penyelesaian *query* kasus
- Case-based maintenance*. Memperbaiki secara konsisten setiap kasus di dalam database kasus

TABEL 4 CASE REPRESENTATION

Kasus 1	
Penyelesaian Tiket Pengaduan	
Dept.	FA
Alamat Pengaduan	Gedung 1
Type Pengaduan	Penting
Nama Pengadu	Indah
Penyelesaian	
Waktu penyelesaian	1Hari

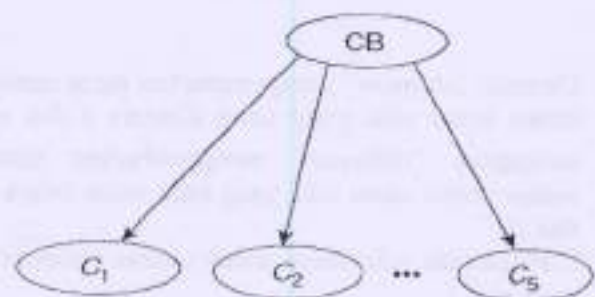
2.2.1.1 Case Representation

Case representation di dalam sistem CBR merupakan penentu dalam memutuskan suatu kasus, suatu representasi kasus harus menjadi *expresi* kasus sebenarnya dan harus menjadi *describe* bagi user untuk menentukan ke akuratan kasus yang ada. Di sisi lain CBR harus berbagi kasus dengan suatu kasus yang terkomputasi secara *flexible*, suatu kasus bisa di wakili dengan suatu tampilan form sebagai gambaran representatif kasus sebenarnya. Dan bentuk representatif kasus bisa terlihat dalam tabel.3.5

2.2.1.2 Case Index

Pada tabel.3 merupakan deskripsi representasi kasus. *Index* kasus adalah salah satu metode dalam mengumpulkan suatu kasus di dalam memori komputer, berikut ini salah satu contoh bagaimana mengorganisasi suatu kasus dalam bentuk hirarki sehingga suatu kasus akan menjadi subset dalam pencarian selama *retrieval* kasus. Berikut contoh proses tahapannya:

- Seleksi yang paling dominan dalam data tabel di atas dalam hal ini, kolom "Dept" apabila di *index* akan mempunyai corak yang berbeda.
- Clasifikasikan kasus ke dalam lima kasus, $C_1, C_2, C_3, \dots, C_5$ bisa di lihat pada gambar 3.2 strukture *index case-based*. Dan setiap kasus baru akan di masukkan ke dalam hirarki *index*, lalu akan di lanjutkan ke prosedur *index* untuk bisa digunakan. clasifikasinya (ne, CL) (i.e....mengklasifikasi ne setiap satu kasus yang terdaftar akan masuk ke dalam *class* CL) dimana $CI = \{C_1, C_2, \dots, C_5\}$ pengimputan kasus baru yang *describel* akan di jadikan satu *fiture* $ne = \{F_1, F_2, \dots, F_5\}$



Gambar 2
Strukture Index Case-Based

- Step 1.* Start dari $i = 1$, lalu di seleksi satu *case* secara random melalui *class* C_i .
- Step 2.* Di komparasi masukan *case* yang lebih awal (ie, "Dept") sampai ne dan e_j
- Step 3.* Jika dua masukan diketemukan kesamaan, maka $ne \in C_i$, else $i = i + 1$; lanjut ke langkah 1

Step 4. Ulangi step 1,2 dan 3 sampai ne dengan bertemu class C_i

2.2.1.3 Case Retrieval

Case retrieval adalah proses pencarian di dalam sistem *case-based*, setiap kasus yang berakhir akan di catat dalam *case history*, berikut contohnya: diberikan suatu case $\hat{e} = (FA, Gedung 1, Sangat penting, Indah, 0)$ dan di ketahui bahwa informasi tersebut ternyata \emptyset , *case retrieval* akan memproses untuk mencari kasus yang ada dan akan di akhiri untuk \hat{e} .

Dan *retrieval* algoritma untuk penyelesaian kasus dengan cara membagi dalam beberapa class kasus untuk diberikan seluruhnya ke kasus \hat{e} , pembagiannya sebagai berikut (\hat{e}, C_i) , Dimana $C_L = \{C_1, C_2, \dots, C_5\}$ dan \hat{e} adalah *query* kasus.

Step 1. Dimulai dari $i = 1$, lalu di seleksi satu kasus e_j dan akan di acak secara asal untuk class C_i .

Step 2. Di komparasi ciri-ciri masukan "Alamat" untuk \hat{e} dan e_j

Step 3. Jika di ketemukaan dua masukan yang teridentifikasi, $\hat{e} \in C_i$, else $i = i + 1$; lanjut ke langkah 1.

Step 4. Diulang ke step 1, 2 dan 3 sampai \hat{e} lalu di lanjutkan ke class C_i , lalu di cari kesamaan kasus yang ada. Dan apabila

Step 1. Jika $\hat{e} \in C_i$, untuk setiap kasus e_i di kalkulasikan derajat kesamaan \hat{e} dan e_i seluruhnya.

$$SM(\hat{e}, e_j) = \frac{\text{common}}{\text{common} + \text{different}}$$

Dimana "common" menggambarkan suatu nomor utama suatu nilai yang sama diantara \hat{e} dan e_i , sedangkan "different" menggambarkan suatu nomor utama suatu nilai yang tidak sama antara \hat{e} dan e_i .

Step 2. Persamaan nilai kasus untuk ukuran computasi pada step 1

Step 3. Memilih kasus seperti yang ada di e_1, e_2, \dots, e_k dari C_i menentukan kesamaan dengan \hat{e} adalah sebagai berikut:

$$SM(\hat{e}, e_j) = \max \{SM(\hat{e}, e_j), e_j \in C_i\} \quad i = 1, 2, \dots, k$$

Dan untuk mendapatkan *query* kasus $\hat{e} = (FA, Gedung, Penting, 0)$ setelah diproses dengan algoritma diatas, untuk mencari kesamaan kasus *retrieved* dari C_1 : e_1, e_2, e_4 dan e_5

$e_1 = (FA, Gedung 1, Penting, Indah, 5)$

$e_2 = (FA, Gedung 1, Tunda, Indah, 2)$

$e_4 = (FA, Gedung 1, Penting, Indah, 3)$

$e_5 = (FA, Gedung 1, Penting, Indah, 2)$

dan hasilnya sebagai berikut :

$$SM(\hat{e}, e_1) = SM(\hat{e}, e_2) = SM(\hat{e}, e_4) = SM(\hat{e}, e_5) = 0.8$$

2.2.1.4 Case Adaptation

Case adaptation adalah suatu proses di dalam sistem CBR yang berfungsi untuk mencari solusi permasalahan kasus yang ada dengan menganalisis sumber dari proses *retrieved* kasus sebelumnya, untuk digunakan dalam perbandingan kasus yang sejenis, bertahapannya:

Step 1. Semua *case retrieved* di kelompokkan yang mempunyai kesamaan kasus pada *case* \hat{e} untuk di jadikan sebagai masukan data base pemecahan kasus (i, e, \dots Penyelesaian Tiket Pengaduan) yang di bagi dalam beberapa kategori seperti, C_1, C_2, \dots, C_m , dengan catatan setiap masukan list data pengaduan sesuai dengan, a_1, a_2, \dots, a_m .

Step 2. Semua nomor kasus di jumlahkan pada *case retrieved* di masukkan dalam katagori ($i = 1, 2, \dots, m$), dan semuanya di masukkan dalam record sebagai n_1, n_2, \dots, n_m .

Step 3. Memilih kategori C_i untuk $n_i = \max \{n_1, n_2, \dots, n_m\}$.

Step 4. Jika kasus yang ada hanya mengeluarkan satu kategori di dalam C_i maka $n_i > n_j, j \neq i$ maka solusinya (Penyelesaian Tiket Pengaduan) langsung di masukkan ke dalam database kasus CBR sebagai solusi kasus yang pasti; else jika senyatanya tidak termasuk dalam kategori, seperti $C_{i1}, C_{i2}, \dots, C_{ik}$ disamakan dengan nilai $n_i = \max \{n_1, n_2, \dots, n_m\}$, dan bisa diambil rata-rata dengan nilai b dan di definisikan sebagai berikut.

$$b = \frac{a_{i1} + a_{i2} + \dots + a_{ik}}{k}$$

dan hasil solusi yang pasti untuk kasus, dimana $a_{i1}, a_{i2}, \dots, a_{ik}$ merupakan nilai dari "penyelesaian tiket pengaduan" sebagai catatan untuk $C_{i1}, C_{i2}, \dots, C_{ik}$.

2.2.1.5 Case-Based Maintenance

Case-based maintenance merupakan hal yang sangat penting dalam sistem CBR, karena teknik ini merupakan representasi sistem aplikasi dalam memecahkan persoalan pengaduan untuk ke akuratan sistem CBR, untuk mengetahui cara kerja case-base maintenance berikut contoh kasus redudance kasus pada tabel 3.1 pada kasus nomor 14 dan 15, apabila di urut ke bawah sistem petunjuknya, kita pasti akan mengidentifikasi dan menghapus salah satu kasus yang ada, adapun langkah prosesnya sebagai berikut :

Step 1. Untuk setiap kasus C_j dalam tahapan 1.9.1, kita akan mengklasifikasikan di dalam case-based dalam 5 kasus, semua klasifikasi di mulai dari keseluruhan pengaduan dan di general menjadi satu kasus $C_{i1}, C_{i2}, \dots, C_{in}$ ($i=1,2,\dots,5$)

Step 2. Penjumlahan nomor kasus dalam tiap kasus C_{ij} , ($i = 1,2,\dots,5, j = 1,2,\dots,n$).

Step 3. Jika salah satu nomor kasus dalam C_{ij} apabila equal untuk nomor 2, lalu di komparasi dengan nilai kasus yang ada.

Step 4. Jika semua kasus yang teridentifikasi mempunyai nilai keistimewaan, maka harus ada salah satu kasus yang di abaikan.

Demikian contoh kasus penerapan computing approach untuk bisa di terapkan dalam software aplikasi terapan.

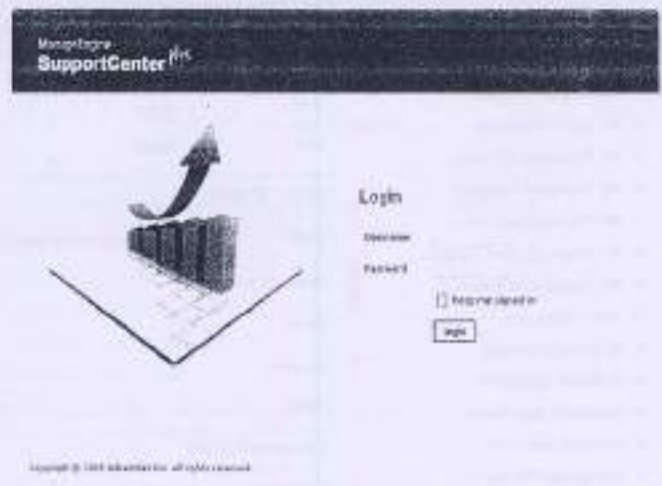
2.2.2 Perancangan Analisa

Perancangan analisa ini di buat untuk membentuk suatu konsep sistem tiket pengaduan yang sederhana dan mudah di pahami, serta membuat suatu knowledge based untuk mendukung sistem tiket pengaduan yang di sesuaikan dengan kebutuhan sistem, berikut desain flow lifecycle sistem tiket pengaduan.



Gambar 3 Lifecycle Sistem Tiket Pengaduan

2.2.3 Perancangan Antar Muka



Gambar 4 Menu login

Home Requests Solutions Accounts Contacts Admin Reports Support Add New

Views **Requests > New Request**

New Request

Request Details

Status: Mode:

Level: Priority:

Owner Details

Group: Support Rep:

Contact Details

Name:

Account: Email:

Phone: Mobile:

Category Details

Category:

Sub Category:

Item:

Subject:

Description:

Recent Items

- windows error
- hardware
- hardware
- Troubel Jaringan

Gambar 5. Menu Pengaduan

Home Requests Solutions Accounts Contacts Admin Reports Support Add New

Views **Solutions > New Solution**

New Solution

Subject:

Contents:

Attachments: Attached files:

Topic:

Keywords:

Keywords should be relevant support
 Keywords should be less than 100 characters and should be lower case letters, digits, periods, underscores, hyphens, and spaces.

Add Add And Approve Cancel

Recent Items

- windows error
- hardware
- hardware
- Troubel Jaringan
- Admin

Gambar 6. Menu Jawaban

Address: <http://localhost:8080/CustomerDef.do?showView=allForm> Go

All Accounts
 Recently Added Account...
 Accounts With SubAccounts...

Recent Items

- windows error
- hardware
- hardware
- Troubel jaringan
- Admin

New Account

Account details

Account Name * Annual Revenue (\$)

Email Website

Phone Fax

Industry Select Industry Type... TimeZone Select TimeZone...

Account Manager Select Account Manager...

Address

Door No Street

LandMark City

State/Province Country

Zip/Postal Code

Description

Description

Attached Files:

Gambar 7. Form User

Address: <http://localhost:8080/CustomerDef.do?mode=addsubacct&301> Go

All Accounts
 Recently Added Account...
 Accounts With SubAccounts...

Recent Items

- Admin
- Troubel jaringan
- windows error
- hardware
- hardware

New Sub Account

Sub Account details

Account Name: Admin

Sub Account Name * Annual Revenue (\$)

Email Website

Phone Fax

Industry Select Industry Type... TimeZone Select TimeZone...

Address

Door No Street

LandMark City

State/Province Country

Zip/Postal Code

Description

Description

Attached Files:

Gambar 8. Form Pakar

5. HASIL DAN PEMBAHASAN

5.1 Implementasi

Apabila hasil pengujian dan *testing* telah dilakukan dengan baik maka sistem tiket pengaduan berbasis CBR telah siap untuk digunakan dan siap diimplementasikan pada user yang berkepentingan dengan modul sistem, sehingga kebutuhan user akan modul sistem akan terpenuhi dengan baik, adapun sasaran dari sistem tiket pengaduan berbasis CBR adalah *user* tiap divisi dan divisi IT yang berkepentingan akan sistem tersebut dalam suatu organisasi perusahaan.

1. Target User

Adapun target *user* dalam penerapan modul system tersebut adalah:

- Teknisi dan System Administrator* adalah seseorang yang harus memelihara (*maintain*) sistem komputer yang berbeda pengoperasian dan penanganannya beserta sarana pendukungnya.
- Director, General Manager, Manager, Supervisor* adalah jabatan yang sangat strategis dalam suatu organisasi. Ia memiliki peran ganda. Di satu sisi mereka adalah pemimpin yang membimbing, memotivasi dan mengendalikan karyawan. Di sisi lain, ia adalah wakil manajemen yang harus mempertanggungjawabkan semua tugas yang diberikan padanya, termasuk penerapan sistem secara terintegrasi guna mengaktifkan dan mengefesifensikan struktur birokrasi dengan sistem yang ada.
- User implement (*trainer*)

User implement adalah pengguna langsung yang menjalankan sistem yang di buat, user tersebut biasanya terdapat pada tiap-tiap departemen dalam organisasi perusahaan, mereka yang akan merasakan langsung hasil dari rekayasa perangkat lunak dan hasil testing akan mempengaruhi kualitas sistem modul pada saat digunakan serta penentu layak atau tidak sistem tersebut digunakan.

2. Strategi Implementasi

Dalam penerapan sistem di perlukan strategi implementasi yang baik agar sistem yang telah dibuat bisa di terapkan dengan baik, dan memerlukan strategi khusus dalam implementasinya adapun strategi yang digunakan adalah dengan cara:

1. Pelatihan

Metode pelatihan adalah proses untuk melatih pengguna dalam mengenalkan proses bisnis perusahaan dan fitur serta fungsi sistem yang akan diimplementasikan dengan tujuan pengembangan kompetensi untuk menjamin keberhasilan implementasi sistem yang telah dibuat. Adapun modul pelatihan implementasi sistem terdiri dari:

- Penggunaan Sistem : ialah tata cara menjalankan sistem yang akan di implementasikan
- Konsep Umum Komputer : ialah pelatihan pengetahuan komputer secara menyeluruh ser fungsi dan teknik dalam mengoperasikan komputer
- Konsep Sistem Informasi : ialah garis besar atau alur cerita sistem yang telah di buat dalam pelaksanaan tugas dan fungsi organisasi dalam sistem informasi.
- Konsep Pengorganisasian : ialah suatu pengenalan yang di berikan dalam struktur organisasi, serta tugas dan fungsi peranan dalam organisasi
- Manajemen Sistem : ialah pencatatan sistem secara berkala yang akan menghasilkan suatu data dan nantinya akan di jadikan bahan informasi untuk perbaikan sistem informasi.
- Instalasi Sistem : ialah proses pemasangan sistem kedalam sebuah perangkat komputer, yang hasilnya akan di jalankan dalam implementasi sistem.

2. Penyebaran Kuisisioner

Penyebaran kuisisioner di lakukan untuk mengetahui pengukuran sistem tiket pengaduan berbasis CBR bisa di terapkan di lingkungan perusahaan dengan nilai standar pengukuran kecepatan dan keakuratan sistem dalam penerapan sistem tiket pengaduan berbasis CBR.

Kuisisioner di lakukan pada tiap divisi perusahaan minimal 3 user pada tiap divisi dan kalau jumlahnya menjadi 20 user dan di ambil secara acak dengan rincian 5 user pemakai komputer yang sering mengadu permasalahan komputer, 10 user pemakai komputer yang tidak terlalu sering mengadu permasalahan komputer, dan 5 user yang tidak pernah mengadu permasalahan komputer.

Dari 20 user sampel tersebut akan diuji coba dengan sistem tiket pengaduan berbasis CBR dalam penerapan pelayanan pengaduan untuk tingkat pengukuran kecepatan dan keakuratan dalam pelayanan pengaduan dan solusi pemecahannya.

Metode yang digunakan untuk pengukuran hasil implementasi ini berupa kuisioner. ada 2 jenis kuisioner yang di berikan yaitu kuisioner *pretest* dan *posttest*, kuisioner *pretest* di berikan sebelum sistem tiket pengaduan berbasis CBR di terapkan untuk memperoleh nilai awal dan kuisioner *posttest* di berikan setelah sistem tiket pengaduan berbasis CBR di gunakan guna mendapatkan informasi data setelah di terapkannya sistem tiket pengaduan berbasis CBR.

Adapun tahapan implementasi kuisioner sebagai berikut :

- User diberikan panduan bagaimana cara mengisi kuisioner *pretest*
- Pengguna mengis kuisioner *pretest*
- Kuisioner *pretest* di kumpulkan untuk di analisa
- User diberikan panduan bagaimana cara mengisi kuisioner *posttest*
- Pengguna mengisi kuisioner *posttest*
- Kuisioner *posttest* di kumpulkan untuk di analisa

5.2 Pengukuran Penelitian

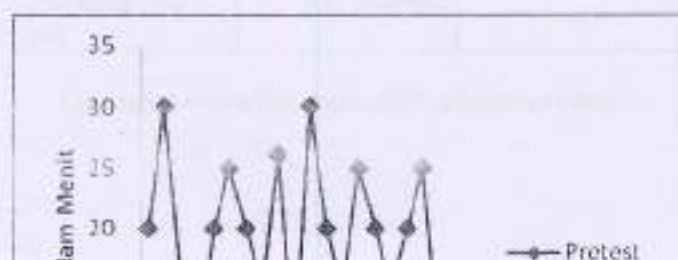
5.2.1 Analisa Hasil Pretest dan Posttest

Pengukuran penelitian ini secara umum untuk mengetahui adanya peningkatan kecepatan dan keakuratan dalam sistem tiket pelayanan pengaduan sebelum di gunakannya sistem tersebut dan sesudah di terapkannya sistem tiket pelayanan pengaduan. Data hasil kuisioner akan di analisa oleh divisi IT

Untuk pengukuran variable data hasil dari pengumpulan kuisioner *pretest* dan *posttest* devisi IT menentukan siapa saja user yang akan di berikan lembaran kuisioner yang nanti hasil datanya akan di analisa, apakah ada peningkatan setelah di terapkannya sistem tiket pengaduan berbasis CBR dalam meningkatkan pelayanan sebelum dan sesudah sistem di terapkan. Adapun hasil data pengumpulan akan di bandingkan factor kecepatan dan keakuratan dalam pelayanan pengaduan dan solusinya. Hasil *pretest* dan *posttest* di sajikan dalam bentuk tabel 4. untuk pengukuran kecepatan sistem, dan data pretest dan *posttest* pada tabel 5. untuk pengukuran keakuratan sistem dalam adaptasi kasus.

No	Data Pengaduan	Waktu dalam Menit	
		<i>Pretest</i>	<i>Posttest</i>
1	1	20	5
2	2	30	3
3	3	15	2
4	4	10	5
5	5	20	4
6	6	25	3
7	7	20	2
8	8	15	1
9	9	26	4
10	10	10	3
11	11	30	1
12	12	20	1
13	13	15	1
14	14	25	2
15	15	20	3
16	16	15	2
17	17	20	2
18	18	25	1
19	19	10	1
20	20	15	2
Nilai Rata - rata		19.3	1.25

Adapun hasil data *pretest* untuk pengukuran kecepatan untuk sistem tiket pengaduan kalau di gambarkan dalam bentuk grafik maka bentuk grafiknya sebagai berikut.



TABEL 5. PRETEST DAN POSTTEST PENGUKURAN KECEPATAN

No	Data Perhitungan	Pengukuran Keakuratan Pretest	Pengukuran Keakuratan Posttest
1	1	AKURAT	AKURAT
2	2	ELUKE AKURAT	AKURAT
3	3	AKURAT	AKURAT
4	4	AKURAT	AKURAT
5	5	AKURAT	AKURAT
6	6	AKURAT	AKURAT
7	7	AKURAT	AKURAT
8	8	ELUKE AKURAT	AKURAT
9	9	ELUKE AKURAT	AKURAT
10	10	ELUKE AKURAT	AKURAT
11	11	AKURAT	ELUKE AKURAT
12	12	AKURAT	AKURAT
13	13	AKURAT	AKURAT
14	14	AKURAT	AKURAT
15	15	AKURAT	AKURAT
16	16	AKURAT	AKURAT
17	17	ELUKE AKURAT	AKURAT
18	18	ELUKE AKURAT	AKURAT
19	19	AKURAT	AKURAT
20	20	AKURAT	AKURAT
21	21	AKURAT	AKURAT
Rata-rata Keakuratan		74,7%	80,5%

Apabila hasil pengukuran keakuratan *Pretest* dan *Posttest* di bentuk dalam grafik maka akan terlihat jelas perbedaan yang sangat mencolok untuk hasil dan perhitungannya, bisa di lihat dalam gambar 9 dan gambar 10 berikut ini:



Gambar 9

Grafik Pengukuran *Pretest* Kecepatan

Dari grafik di atas terlihat perbedaan yang sangat jauh dalam kecepatan penanganan pengaduan dan penyelesaiannya.

TABEL.6

PRETEST DAN POSTTEST PENGUKURAN KEAKURATAN



Gambar. 10

Pengukuran *Pretest* Keakuratan Penyelesaian Pengaduan



Gambar. 11

Pengukuran *Posttest* Keakuratan Penyelesaian Pengaduan

Dari grafik kedua di atas terlihat jelas perbedaan yang sangat jauh dalam keakuratan dalam penyelesaian pengaduan.

5.2.2 Implikasi Penelitian

5.2.2.1 Uji Statistik Pretest dan Posttest

Pengujian dilakukan untuk mengetahui efektifitas sistem tiket pengaduan berbasis CBR terhadap kecepatan dan keakuratan dalam penyelesaian proses pengaduan dan penyelesaiannya yang dilakukan dengan cara membandingkan waktu dan nilai kesesuaian *pretest* dan *posttest*. Pada tahapan ini akan dilakukan proses perbandingan hasil pengukuran penelitian yang telah didapatkan sebelumnya. Adapun metode perbandingan ini adalah dengan menganalisa *t-test*.

Metode ini digunakan karena *t-test* dapat digunakan untuk menguji kecocokan atas perbedaan pada suatu percobaan yang menggunakan satu kelompok contoh. Apabila sebelum melakukan percobaan, peneliti melakukan *pretest*, maka peneliti akan mempunyai dua kelompok nilai yang berasal dari satu kelompok contoh. Apabila percobaan itu mempunyai dampak terhadap hasil atau tujuan percobaan, maka kedua kelompok skor tersebut akan menunjukkan perbedaan yang signifikan. *T-test* membandingkan hasil perhitungan perbedaan hipotesa dengan *t* tabel. Apabila hasil perhitungan tersebut berbeda secara signifikan, maka hipotesa diterima.

Dan diketahui beberapa variabel yang menjadi parameter perhitungan pada *t-test* sebagai berikut:

- Derajat kebebasan (*df*), yaitu suatu angka yang menjelaskan sekumpulan skor sampel yang bebas dari kesalahan. Nilai *df* diperoleh dari jumlah sampel-1. Jadi dalam hal ini, nilai *df*=19 karena jumlah sampel adalah 20.
- Alpha, yaitu tingkat signifikansi pengujian. Besaran nilai yang umumnya digunakan adalah 0,05.
- Simpangan baku (*Sd*) yang dapat dihitung dengan

rumus sebagai berikut: $Sd = (\sum x - X^2) / n - 1$

- Standard error (Sx)* yang dapat dihitung berdasarkan rumus sebagai berikut:

$$Sx = Sd / n$$

Sedangkan untuk nilai *t*, dapat dihitung dengan rumus berikut:

$$t = X - \mu / Sx$$

Dari hasil perhitungan tersebut, maka *t* hitung akan dibandingkan dengan *t* tabel. Jika perbedaannya signifikan, maka disimpulkan bahwa hipotesa diterima. Untuk perhitungan ini, dapat disederhanakan dengan menggunakan fungsi dari *Microsoft Excel* untuk *data analysis*.

- Uji perbedaan kecepatan *Pretest dan Posttest* Untuk menguji apakah ada perbedaan signifikan antara kecepatan sebelum dan sesudah menggunakan sistem, maka dapat dilakukan penyusunan hipotesa dibawah ini:

Ho : Tidak terdapat perbedaan antara hasil kecepatan sebelum dan sesudah implementasi sistem.

Ha : Terdapat perbedaan antara hasil kecepatan sebelum dan sesudah implementasi sistem.

Pengujian kecepatan ini menggunakan data pada tabel pengukuran kecepatan hasil rangkuman dari hasil kuesioner yang sudah dijelaskan. Dari data tersebut, lalu data di-generate menggunakan fungsi *data analysis* dari *Microsoft Excel* dan akan menghasilkan tabel berikut.

TABEL 7
HASIL Uji KECEPATAN *T-TEST: PAIRED TWO SAMPLE FOR MEANS*

<i>t-Test: Paired Two Sample for Means</i>		
Mean	<i>Pretest</i>	<i>Posttest</i>
<i>Variance</i>	18,90	1,10
<i>Observations</i>	51,57631580	0,144210527
<i>Pearson Correlation</i>	20	20
<i>Hypothesized Mean</i>	0,689874708	
<i>df</i>	19	
<i>t Stat</i>	11,94563706	
<i>P(T<=t) one-tail</i>	7,02299E-10	
<i>t Critical one-tail</i>	1,829132793	
<i>P(T<=t) two-tail</i>	1,2046E-09	
<i>t Critical two-tail</i>	3,09302406	

Dari tabel 6 tersebut dapat dilihat bahwa *t* tabel

(*t critical one-tail*) bernilai 1,829132793 sedangkan *t* hitung (*t Stat*) bernilai 11,94563706. Terlihat perbedaan signifikan, berarti terdapat perbedaan kecepatan yang signifikan antara sebelum dan sesudah penerapan sistem. Berarti penerapan sistem membawa efek positif. Dengan melihat nilai probabilitas, *P-value* adalah 7,02299E-11 lebih kecil dari 0,05 berarti H_0 ditolak atau penerapan sistem meningkatkan kecepatan dalam proses sistem tiket pengaduan.

b. Uji perbedaan akurasi *Pretest dan Posttest*

Untuk menguji apakah ada perbedaan signifikan antara keakuratan sebelum dan sesudah menggunakan sistem, maka dapat dilakukan penyusunan hipotesa dibawah ini:

H_0 : Tidak terdapat perbedaan antara hasil keakuratan sebelum dan sesudah implementasi sistem.

H_a : Terdapat terdapat perbedaan antara hasil keakuratan sebelum dan sesudah implementasi sistem.

Pengujian keakuratan ini menggunakan data sampel yang tepat dikerjakan oleh para pakar yang memasukkan *explecte knowledge* dan sistem hasil rangkuman dari hasil kuesioner. Dari data tersebut, setelah di-generate menggunakan fungsi *data analysis* dari *Microsoft Excel* dan akan menghasilkan tabel berikut.

TABEL 8
HASIL UJI AKURAT T-TEST: PAIRED TWO SAMPLE FOR MEANS

<i>t-Test: Paired Two Sample for Means</i>		
	<i>Pretest</i>	<i>Posttest</i>
<i>Mean</i>	0,9	0,97
<i>Variance</i>	0,231052632	0,06
<i>Observations</i>	20	20
<i>Pearson Correlation</i>	0,360438322	
<i>Hypothesized Mean</i>	0	
<i>df</i>	19	
<i>t Stat</i>	-2,616611479	
<i>P(T<=t) one-tail</i>	0,020495753	
<i>t Critical one-tail</i>	1,829132793	
<i>P(T<=t) two-tail</i>	0,030991506	
<i>t Critical two-tail</i>	2,10302406	

Dari tabel 7 tersebut dapat dilihat bahwa *t* tabel (*t critical one-tail*) bernilai 1,829132793 sedangkan *t* hitung (*t Stat*) bernilai -2,616611479. Terlihat

perbedaan, berarti terdapat perbedaan kecepatan an sebelum dan sesudah penerapan sistem. Dengan melihat nilai probabilitas, *P-value* adalah 0,020495753 lebih kecil dari 0,05 berarti H_0 ditolak atau penerapan sistem meningkatkan keakuratan dalam proses penyelesaian pengaduan.

5. KESIMPULAN

Dengan terbentuknya sistem tiket pengaduan berbasis *case-based reasoning* dengan konstruksi *work-based*. Proses pengaduan di proses secara cepat, sehingga proses waktu pengaduan dan penyelesaian tiket tidak memerlukan waktu yang lama dan *cost* yang tinggi. Dan terbentuknya *explicit knowledge-based* di sistem CBR untuk menentukan pemecahan masalah secara akurat melalui proses *retrieve, reuse, revise, reselect* sehingga proses pencarian akan menemukan keakuratan data.

6. SARAN

Hasil dari penelitian untuk sistem tiket pengaduan berbasis CBR ini membawa nilai yang positif bagi perusahaan dalam mengefektifkan kinerja perusahaan, terhadap pelayanan pengaduan pelanggan atau karyawan perusahaan itu sendiri. Namun peneliti menyarankan beberapa hal untuk efektivitas sistem bekerja.

- Inputan *case* dan *knowledge* sangat di perlukan lebih banyak lagi untuk kasus penanganan pengaduan.
- Penggunaan sistem ini akan lebih efektif bila kebijakan khusus di dalam satu organisasi perusahaan untuk menggunakannya secara simultan

7. DAFTAR PUSTAKA

- Compen, D. R., & Higgins, C. A. 1995. *Application of Cognitive Theory To Training For Computer Skills*. *Inform Systems Research*.
- Wahono, R. S. 2006. <http://www.satriawahono.net/2006/02/24/sdm-ti-dari-spesialis-ke-versatilis/>. Retrieved <http://www.satriawahono.net>.
- Sunkar, K. Pal & Simon C.K. Shiu 2003. *Foundations of soft base reasoning*, P.1-27
- Setiawo, Bambang. 2005. Knowledge sharing in Indonesia res centre : models and mechanism. *Proceedings on the 9 th World Conferent on Systemics,Cybernetics and Informatics*, Orlando, July 10-13, pp.14
- Aamodt, A. and Plaza, E. 1994. *Case-Based Reasoning: Four Issues: Methodological Variations, and System Approache Communications*, 39-59.
- Aha, D., and Breslow, L. 1997. *Refining conversational libraries*. *Proceedings of the Second International Conferent Case-Based Reasoning*. Providence, RI: Springer-Verlag.
- Aha, D., Macey, T. and Breslow, L. 1998. *Supporting Die Inferencing in Conversational Case-Based Reasoning. 7 European Workshop on Case-Based Reasoning*. Dublin, In Springer.