

## INDEXING PADA SISTEM PENALARAN BERBASIS KASUS MENGGUNAKAN METODE COMPLETE-LINKAGE CLUSTERING

Elvery B. Johannes

e-mail: [ebjohannes@fatek.unpatti.ac.id](mailto:ebjohannes@fatek.unpatti.ac.id), [elbenjohannes@gmail.com](mailto:elbenjohannes@gmail.com)

Fakultas Teknik Universitas Pattimura - Ambon

### ABSTRAK

Sistem penalaran berbasis kasus atau *Case-Based Reasoning* (CBR) merupakan bagian dari *artificial intelligent* yang telah banyak diimplementasikan sebagai sistem untuk mendiagnosa penyakit, mendeteksi kerusakan bangunan, mesin, komputer, dan lain sebagainya. Cara kerjanya yaitu dengan membandingkan kasus baru terhadap kasus lama yang disimpan sebagai pengetahuan (*knowledge*) pada basis kasus. Kasus lama dicek tingkat similaritasnya satu per satu terhadap kasus baru. Kasus lama yang memiliki kemiripan tertinggi, diberikan kepada *user* sebagai kandidat solusi untuk menyelesaikan kasus baru. Permasalahannya adalah jika kasus lama pada basis kasus sangat banyak, maka waktu *retrieve* akan menjadi relatif semakin lama. Tujuan dari penelitian ini adalah untuk menghasilkan model sistem CBR yang memiliki kinerja optimal dengan memanfaatkan *clustering* untuk *indexing*. Metode *clustering* yang digunakan yaitu *complete-linkage*. *Clustering* dilakukan terhadap kasus lama yang dibuat dengan variasi 3, 4, 5, dan 6 jumlah *cluster*. Pengujian sistem selanjutnya diimplementasikan terhadap kasus penyakit jantung dengan 25 data uji. Hasil pengujian menunjukkan, dengan menggunakan CBR *cluster-indexing* terjadi peningkatan kecepatan waktu *retrieve* sebesar 34.18%, yaitu pada variasi 4 jumlah *cluster*. Akurasi sistem CBR *cluster-indexing* sama dengan sistem *non-indexing* yaitu sebesar 96% pada variasi 4, 5, dan 6 jumlah *cluster*. Pada variasi 3 jumlah *cluster*, akurasi CBR *non-indexing* sebesar 96% sedangkan CBR *cluster-indexing* sebesar 92%.

**Kata kunci** – CBR, *cluster-indexing*, *complete-linkage*.

### PENDAHULUAN

Richter and Weber (2013) menyatakan, *Case-Based Reasoning* (CBR) merupakan sebuah metodologi penyelesaian masalah dengan memanfaatkan pengalamannya sebelumnya. Penyajian pengetahuan dibuat dalam bentuk kasus-kasus (*cases*). Setiap kasus berisi masalah dan jawaban, sehingga kasus-kasus lebih mirip dengan suatu pola tertentu. Cara kerja CBR adalah dengan membandingkan kasus baru dengan kasus lama yang disimpan sebagai pengetahuan (*knowledge*) pada basis kasus. Kelebihan utama dari CBR adalah dalam hal akuisisi pengetahuan, dimana sistem CBR dapat menghilangkan kebutuhan untuk ekstrak model atau kumpulan dari aturan-aturan seperti yang diperlukan dalam sistem *Rule-Based Reasoning* (RBR). Walaupun demikian, kelemahan CBR yaitu jika kasus lama yang ada pada basis kasus memiliki jumlah yang banyak, maka proses untuk menemukan kasus yang mirip akan memerlukan waktu yang relatif cukup lama (Rismawan dan Hartati, 2012). Agar proses *retrieve* lebih efisien, perlu dilakukan *indexing* sehingga proses pencarian kasus yang mirip tidak perlu melibatkan seluruh data tetapi cukup pada beberapa kasus terdekat saja.

*Indexing* kasus pada CBR yang dibangun dalam penelitian ini, dilakukan dengan pendekatan *clustering*. Pendekatan umum dari algoritma

*clustering* adalah untuk menemukan nilai *cluster centroid* yang akan mewakili setiap kelompok. Nilai *cluster centroid* merepresentasikan letak titik pusat dari setiap *cluster*. *Cluster centroid* memungkinkan pengukuran kesamaan antara data baru dengan semua *cluster centroid* sehingga dapat menentukan kelompok data yang paling mirip (Santoso, 2017). Algoritma *clustering* yang digunakan untuk *indexing* kasus adalah metode *complete-linkage* yang merupakan salah satu pendekatan *linkage* dari metode *Agglomerative Hierarchical Clustering* (AHC). AHC memiliki karakteristik yaitu jika suatu data telah dikelompokkan ke dalam suatu *cluster* pada suatu tahapan proses, maka pada tahapan proses selanjutnya data tersebut tidak dapat berpindah ke *cluster* yang lain. Hal ini memungkinkan AHC cenderung tidak terpengaruh dengan adanya *outlier* (Mooi and Sarstedt, 2011).

Penelitian ini berfokus pada kecepatan *retrieve* dengan tidak mengabaikan akurasi sistem sebagai aspek utama dari sebuah sistem CBR. Pada CBR *cluster-indexing* waktu *retrieve* terdiri dari waktu untuk proses pencarian *cluster centroid*, dan waktu untuk proses pencocokkan kasus baru dengan kasus lama. Bertambahnya jumlah *cluster* akan mengakibatkan ukuran *cluster* dari setiap *cluster* relatif berkurang tetapi jumlah *cluster centroid* akan bertambah.

Teknik *retrieve* menggunakan metode *cosine coefficient* untuk pencarian *similarity* antara kasus baru dengan *cluster centroid* dan metode *nearest neighbor* digunakan untuk pencarian *similarity* antara kasus baru dengan kasus lama. Sistem CBR yang dibangun selanjutnya akan diujicobakan untuk diagnosis penyakit jantung, yang terdiri dari 6 jenis antara lain; jantung akut, jantung kronik, jantung koroner, jantung hipertensi, jantung katup, dan jantung perikarditif.

## KAJIAN TEORI DAN METODE

### 1. Penelitian Terdahulu

Penelitian terdahulu tentang CBR seperti; *indexing* pada *case-based reasoning* menggunakan *hashing* (Mohsin, *et al.*, 2011). Data yang digunakan merupakan data kasus operasional bendungan timah Tasoh tahun 1997-2005. Hasil penelitian menunjukkan metode *hashing* memberikan hasil yang lebih baik dari pada metode CBR konvensional, dimana proses *retrieve* rata-rata memiliki kecepatan proses lebih cepat 1 ms.

*Indexing* menggunakan metode *backpropagation*, sedangkan teknik *retrieve* menggunakan metode *cosine coefficient* untuk diagnosa penyakit THT (telinga, hidung, dan tenggorok), diteliti oleh Rismawan dan Hartati (2012). Data uji yang digunakan sebanyak 111. Hasil pengujian menunjukkan terdapat 9 kasus yang similaritasnya di bawah 0.8, sedangkan akurasi sistem yang dihasilkan sebesar 91,89%.

Salem *et al.* (2005), membahas tentang teknik *retrieve* untuk diagnosis penyakit jantung yaitu *induction* dan *nearest neighbor*, dengan metode *non-indexing*. Ada sebanyak 110 kasus untuk 4 jenis penyakit jantung. Setiap kasus memiliki 207 atribut yang berhubungan dengan *demografis* dan data klinis. Akurasi sistem yang dihasilkan sebesar 53.8% untuk *induction* dan 100% untuk *nearest neighbor*.

Adawiyah (2016), meneliti *case-based reasoning* untuk diagnosis penyakit akibat virus *dengue*. Metode *indexing* yang digunakan yaitu *non-indexing* dan *bayesian model* dengan teknik *retrieve* yaitu *nearest neighbor*. Data uji yang diambil secara acak dari data 139 kasus dan 85 kasus sebagai *knowledge*. *Non-indexing* menghasilkan nilai sensitivitas 98,14% tingkat akurasi sebesar 99,25%. Nilai sentivitas *Bayesian model* 88,89% dan akurasi 95,56%.

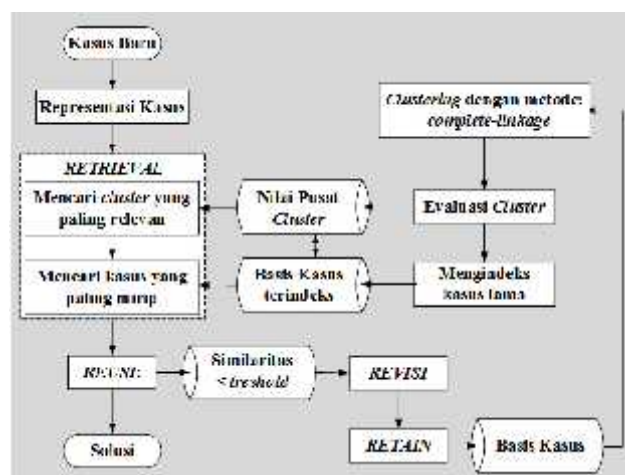
### 2. Deskripsi sistem

Sistem CBR *cluster-indexing* yang dibangun seperti ditunjukkan pada Gambar 1.

Pada tahap awal, kasus-kasus lama disimpan dalam basis kasus, selanjutnya sistem melakukan *clustering* menggunakan metode *complete-linkage*. Hasil *clustering* dievaluasi/ menggunakan *silhouette index*

(SI). Nilai SI semakin mendekati 1.0, menunjukkan kualitas *cluster* yang terbentuk semakin baik dan sebaliknya. Setelah kasus-kasus lama di-*cluster*-kan, data kasus lama diperbaharui dengan menambahkan pengetahuan baru berupa nilai *cluster centroid*, kemudian disimpan sebagai basis kasus yang sudah terindeks.

Misalkan terdapat kasus baru maka sistem menginisialisasi umur, jenis kelamin dan gejala penyakit yang dialami/diderita pasien dan merepresentasikannya sebagai kasus baru. Selanjutnya sistem melakukan pencarian *cluster* yang paling relevan dengan cara menghitung kemiripan gejala kasus baru dengan nilai *cluster centroid*. Setelah memperoleh *cluster* yang relevan dengan kasus baru, dilakukan perhitungan untuk mencari similaritas antara kasus baru (*target case*) dengan *source case* atau kasus-kasus yang tersimpan dalam basis kasus tetapi hanya yang berada pada *cluster* yang relevan (*retrieve*). Jika similaritas 0.9 menunjukkan bahwa kasus baru tersebut memiliki kesamaan dengan kasus lama sehingga solusi dari *source case* akan diberikan kepada *user (reuse)*, dan sebaliknya jika nilai similaritas < 0.9, menunjukkan bahwa kasus baru tersebut semakin tidak mirip dengan kasus lama yang ada di basis kasus terindeks. Kasus tersebut selanjutnya akan disimpan dalam *database* adaptasi kasus yang nantinya akan dilakukan penyesuaian dari solusi kasus-kasus sebelumnya oleh pakar (*revise*). Kasus baru yang telah dilakukan penyesuaian oleh pakar, kemudian disimpan ke dalam basis kasus untuk menjadi pengetahuan baru (*retain*).

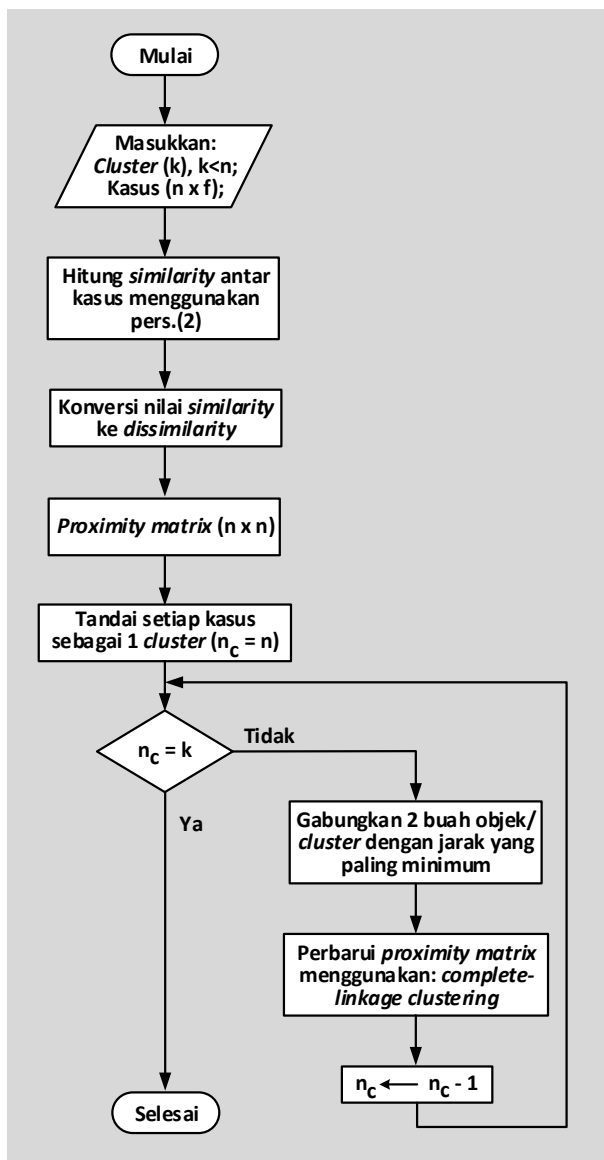


Gambar 1. Gambaran umum sistem

### 3. Clustering

*Clustering* dapat mengelompokkan sejumlah data yang tidak berlabel berdasarkan kemiripan (*similarity*) dan ketidakmiripan (*dissimilarity*) ke dalam kelompok yang disebut *cluster*, sehingga dalam setiap *cluster* akan berisi data yang semirip mungkin (Witten, *et al.*, 2011). Gambar 2 memperlihatkan diagram alur algoritma AHC yang

terdiri dari tahap *preprocessing* dan proses *clustering* menggunakan pendekatan *complete-linkage*.



Gambar 2. Diagram alur clustering dengan algoritma AHC

3.1 Tahap preprocessing

1) Normalisasi data

Sebelum melakukan clustering menggunakan algoritma AHC, data input perlu dinormalisasikan. Tujuannya adalah untuk mendapatkan data dengan ukuran yang lebih kecil yang mewakili data asli tanpa kehilangan karakteristik datanya. Data yang dinormalisasi adalah data numerik yaitu fitur usia karena memiliki rentang nilai yang cukup besar. Normalisasi data dihitung menggunakan (Prasetyo, 2014) seperti pada persamaan (1). Nilai *min* dan *max* merupakan nilai minimum dan nilai maksimum fitur usia. Fitur usia pada penyakit jantung diukur dalam satuan tahun. Dalam perancangan sistem CBR ini, diasumsikan bahwa domain usia = [*m*, *m*] dimana, nilai *m* merupakan batas bawah (*lower bound*) dan nilai *m* merupakan batas atas (*upper bound*).

$$v' = \frac{v - m}{m - m} \dots\dots\dots(1)$$

2) Proximity matrix

Setelah fitur usia dinormalisasikan, selanjutnya dihitung similaritas antar objek (data kasus) dalam bentuk matriks kedekatan (*proximity matrix*) dengan ukuran *n x n* (*n* = banyaknya objek/kasus). Untuk mendapatkan *proximity matrix* dilakukan perhitungan *similaritas* antar objek menggunakan metode *cosine coefficient similarity* (Prasetyo, 2014) seperti pada persamaan (2).

$$C(X, Y) = \frac{\langle X|Y \rangle}{\|X\| \|Y\|} \dots\dots\dots(2)$$

dimana, “*<*” perkalian dari vektor *X* dan *Y*, sedangkan “*||*” menunjukkan norm pada masing-masing vektor.

3.2 Tahap Clustering

Langkah-langkah proses clustering dengan algoritma AHC melalui pendekatan *complete-linkage* :

1. Masukkan *n* objek/cluster (buat dalam bentuk matriks jarak dengan ukuran *n x n*).
2. Cari pasangan objek/cluster dengan jarak terdekat (paling mirip), gabungkan menjadi satu cluster.
3. Hitung jarak antara himpunan objek/cluster dengan cluster yang telah terbentuk pada langkah 2, pilih jarak terjauh.
4. Update entries pada matriks dengan cara :
  - a. Hapus baris dan kolom yang bersesuaian dengan objek/cluster yang sudah digabungkan dalam satu cluster (langkah 2).
  - b. Tambahkan baris dan kolom yang memberikan jarak terdekat (yang telah dipilih pada langkah 3) antara cluster yang telah terbentuk (pada langkah 2) dengan himpunan objek/cluster yang tersisa.
5. Ulangi langkah 2 sebanyak jumlah cluster yang diinginkan.

3.3. Evaluasi Cluster

Cluster yang telah terbentuk selanjutnya dievaluasi menggunakan *Silhouette Index* (SI) (Prasetyo, 2014) seperti pada persamaan (3), (4), dan (5) :

- SI data ke-*i* :

$$S_i^j = \frac{b_i^j - a_i^j}{\max\{a_i^j, b_i^j\}} \dots\dots\dots(3)$$

$$a_i^j = \frac{1}{m_j - 1} \sum_{r=1, r \neq i}^{m_j} d(x_i^j, x_r^j)$$

$$i = 1, 2, \dots, m_j$$

*d(x<sub>i</sub><sup>j</sup>, x<sub>r</sub><sup>j</sup>)* = jarak data ke-*i* dengan data ke-*r* dalam cluster *j*, sedangkan *m<sub>j</sub>* adalah jumlah data dalam cluster ke-*j*.

$$b_i^j = m_{n-1, \dots, k; n \neq j} \left\{ \frac{1}{m_n} \sum_{r=1, r \neq i}^{m_n} d(x_i^j, x_i^r) \right\}$$

$$i = 1, 2, \dots, m_n$$

- SI lokal :

$$S_j = \frac{1}{m_j} \sum_{i=1}^{m_j} S_i^j \dots\dots\dots (4)$$

- SI global :

$$S = \frac{1}{k} \sum_{j=1}^k S_j \dots\dots\dots (5)$$

3.4. Proses retrieve

Proses *retrieve* dilakukan dengan 2 tahap, (i) melakukan pencarian similaritas antara kasus baru dengan nilai *cluster centroid* menggunakan persamaan (2), dengan *X* = vektor *cluster centroid* dan *Y* = vektor kasus baru dimana fitur numerik telah dinormalisasi, dan (ii) mencari similaritas antara kasus baru dengan kasus lama yang menjadi *cluster membership* dari *cluster* yang paling relevan yang telah diperoleh pada tahap pertama. Proses similaritas pada tahap kedua adalah mengitung nilai similaritas lokal dan nilai similaritas global.

1) Similarity lokal

*Similarity* lokal yaitu *similarity* yang terdapat pada level fitur yang dibedakan menjadi 2 jenis, yaitu simbolik dan numerik. Fitur yang termasuk dalam jenis simbolik adalah fitur jenis kelamin, fitur gejala dan faktor risiko, sedangkan fitur untuk jenis numerik adalah fitur usia. Jika sebuah fitur memiliki sub fitur maka dihitung nilai data *range*-nya.

a. Data bernilai numerik dihitung menggunakan persamaan (6), (Shi, et al., 2011).

$$f(S_i, T_i) = 1 - \frac{|S_i - T_i|}{f_m - f_m} \dots\dots\dots (6)$$

$f(S_i, T_i)$  = similaritas lokal atribut ke-*i* dari kasus *S* (*source*) dengan kasus *T* (*target*)

$S_i$  = nilai fitur ke-*i* dari kasus lama (*source*)

$T_i$  = nilai fitur ke-*i* dari kasus baru (*target*)

$f_m$  = nilai maksimum fitur ke-*i* pada basis kasus

$f_m$  = nilai minimum fitur ke-*i* pada basis kasus.

b. Data bernilai simbolik dihitung menggunakan persamaan (7), (Shi, et al., 2011).

$$f(s, t) = \begin{cases} 1 & \text{jika } s = t \\ 0 & \text{lain} \end{cases} \dots\dots\dots (7)$$

$s, t$  { benar, salah }

2) Similarity global

*Similarity* global merupakan similaritas yang terdapat pada kasus atau level objek. *Similarity* global digunakan untuk menghitung keserupaan antar permasalahan baru dengan kasus yang tersimpan dalam *base-case*. Metode yang digunakan untuk menghitung *similarity* global adalah *minkowski distance similarity* seperti pada persamaan (8), (Nunez, et al., 2004).

$$Si(S, T) = \left( \frac{\sum_{i=1}^n w_i^r \times |f(S_i, T_i)|^r}{\sum_{i=1}^n w_i^r} \right)^{1/r} \dots\dots\dots (8)$$

$Si(S, T)$  = nilai similaritas antara kasus lama (*S*) dan kasus baru (*T*)

$f(S_i, T_i)$  = kesamaan fitur ke-*i* dari *case* dan *target*

*T* = kasus baru (*target*)

*S* = kasus lama (*cases*) yang ada dalam penyimpanan basis kasus

*n* = jumlah fitur pada masing-masing kasus

*i* = fitur individu antara 1 sampai *n*

*w* = bobot yang diberikan pada fitur ke-*i*

*r* = faktor minkowski (integer positif).

4. Pengujian sistem

Pengujian sistem dilakukan untuk mengetahui akurasi dan kecepatan *retrieve* masing-masing terhadap 25 data uji. Akurasi sistem dihitung dengan menggunakan persamaan (9), (Elhadad and Badran, 2017)

$$A = \frac{k}{n} \times 100\% \dots\dots\dots (9)$$

$k_i$  = keputusan ke-*i*, bernilai 1 jika benar dan 0 jika salah

*n* = banyaknya data uji.

Kecepatan *retrieve* dihitung dengan membuat *script* program (Gambar 3) untuk mencatat waktu awal dan waktu akhir pada proses *retrieve*.

Kecepatan *retrieve* merupakan selisih antara waktu akhir proses dengan waktu awal proses, yang diukur dalam satuan detik.

5. Implementasi sistem

Aplikasi dibuat menggunakan bahasa pemrograman PHP dengan *framework codeigneter*, dan dbMaria untuk *data base*. Perangkat keras terdiri dari Prosesor Intel(R) Core(TM) i3-5005U CPU @ 2.00 GHz, Hard Disk Drive 500 GB, RAM 4 GB, dan perangkat lunak adalah OS Windows 7 Ultimate 64 bit, *text editor* menggunakan Notepad++, XAMPP v3.2.2 sebagai *local web server*.

5.1. Implementasi halaman login.

Terdapat tiga hak akses bagi user yaitu administrator, paramedis, pakar. Administrator berfungsi sebagai operator sistem yang memiliki hak akses untuk menambahkan pengguna baru pada sistem baik itu pakar maupun paramedis, mengubah serta menghapus data pengguna. Selain itu, administrator juga bertugas untuk melakukan *clustering* basis kasus.



Gambar 3. Implementasi login

Paramedis mempunyai hak akses untuk memasukkan data pasien baru dan diagnosis penyakit berdasarkan kasus yang telah tersimpan sebelumnya oleh pakar. Pakar mempunyai hak akses menginput data, yang terdiri dari data penyakit, data gejala, dan data kasus. Selain itu pakar juga dapat melakukan *clustering* serta adaptasi pada proses *revise* dan selanjutnya menyimpan kasus ke dalam basis kasus *retain*. Implementasi halaman *login* seperti ditunjukkan pada Gambar 3.

5.2. Implementasi menu utama.

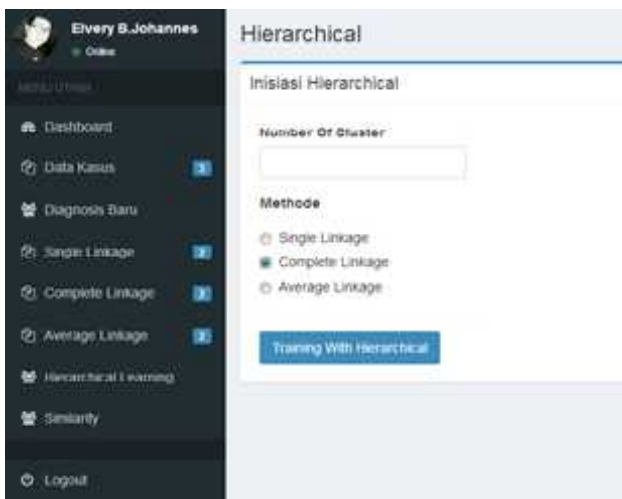
Dashboard menginformasikan (i) Jumlah kasus yang telah disimpan dalam basis kasus, (ii) Diagnosis baru yang merupakan kasus-kasus baru yang *similarity*-nya  $< threshold$  yang disimpan dalam basis kasus sementara, sebelum di-*revisi* dan yang selanjutnya akan di-*retain* oleh pakar, serta (iii) Jenis penyakit dari sistem CBR *cluster-indexing*. Implementasi menu utama seperti ditunjukkan pada Gambar 4.



Gambar 4. Implementasi menu utama

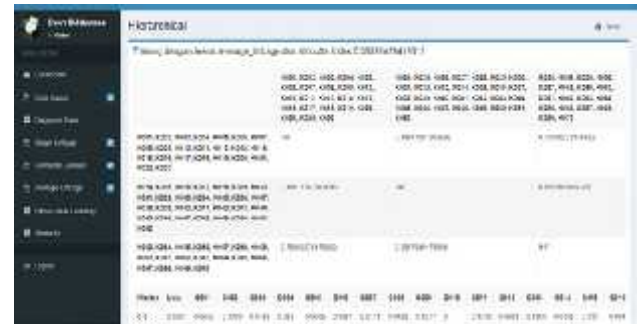
5.3. Implementasi halaman *clustering*.

Halaman ini menampilkan menu *indexing* dengan pendekatan *complete-linkage*, dan jumlah *cluster* sebagai nilai *input*-an seperti ditunjukkan pada Gambar 5.



Gambar 5. Implementasi halaman *clustering*

*Output* dari proses ini berupa nilai *cluster centroid*, dan *cluster membership* dari masing-masing *cluster* sesuai teknik *complete-linkage*. Gambar 6 menampilkan *output* dari proses *clustering*.



Gambar 6. *Output* proses *clustering*

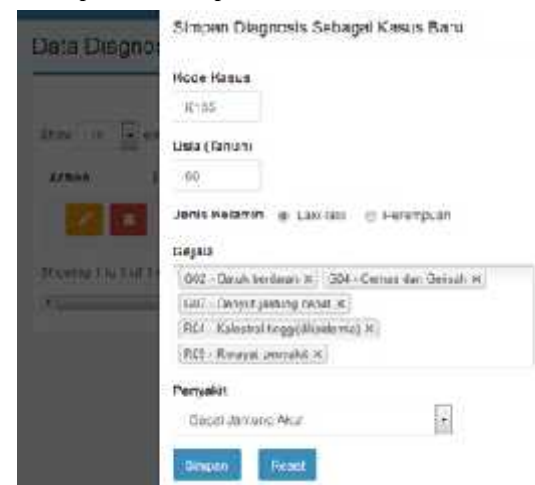
5.4. Implementasi proses *retrieve* serta kecepatan prosesnya dan *reuse*.

Proses ini dimulai dengan meng-*input* usia, jenis kelamin, dan gejala pasien oleh *user*. Setelah dilakukan proses pencarian *similarity*, maka *output* ditampilkan berupa nilai *similaritas*, waktu *retrieve*, rekomendasi kasus, jenis penyakit, dan *cluster membership*-nya merupakan rekomendasi kasus yang dipilih, seperti ditunjukkan pada Gbr. 8.



Gambar 7. Halaman proses *retrieve* dan kecepatan prosesnya

5.5. Implementasi proses *revise* dan *retain*

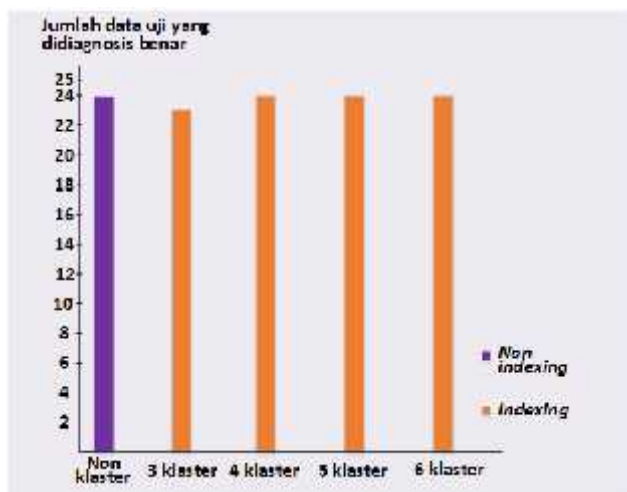


Gambar 8. Halaman proses *revise* dan *retain*

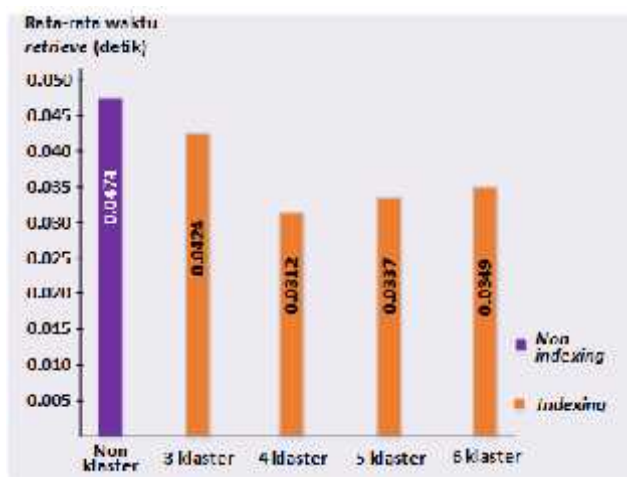
Proses *revise* dan *retain* dilakukan oleh pengguna dengan hak akses sebagai pakar. Proses *revise* dilakukan ketika nilai similaritas tertinggi kasus yang di-*retrieve* tidak mencapai nilai *threshold*. Kasus yang berada di bawah *threshold* tersebut akan disimpan terlebih dahulu oleh sistem CBR kemudian direvisi oleh pakar sehingga dapat digunakan sebagai solusi untuk diagnosis kasus baru. Implementasi dari proses *revise* dan *retain* seperti ditunjukkan dalam Gambar 8.

## HASIL DAN PEMBAHASAN

Sebagaimana telah dijelaskan di awal bahwa penyakit jantung yang akan di diagnosis terdiri dari 6 jenis penyakit. Pengujian kasus penyakit jantung dilakukan terhadap 25 kasus data uji dengan data latih sebanyak 115 kasus. Data uji yang digunakan diambil secara acak namun tetap mempertimbangkan aspek keterwakilan terhadap data latih dari tiap-tiap jenis penyakit yang disimpan dalam basis kasus. Rekapitulasi hasil pengujian akurasi sistem dan kecepatan *retrieve* disajikan dalam bentuk grafik seperti pada Gambar 9, dan 10.



Gambar 9. Hasil pengujian akurasi



Gambar 10. Hasil pengujian kecepatan *retrieve*

Gambar 9 menunjukkan CBR *non-indexing* dan CBR *cluster-indexing* dengan variasi 4, 5, dan 6 jumlah *cluster* menghasilkan akurasi sebesar 96% dimana 1 data uji (kasus) atau = 4% didiagnosis salah. Menurut pakar, kasus tersebut merupakan jenis penyakit jantung katup/endocarditis, sedangkan yang didiagnosis sistem adalah jenis penyakit jantung perikarditif. Pada CBR *cluster-indexing* dengan variasi 3 jumlah *cluster*, akurasi sistem sebesar 92%, atau terdapat 2 data uji (= 8%) yang didiagnosis salah, masing-masing jenis penyakit jantung katup/endocarditis menjadi jenis penyakit jantung perikarditif dan jenis penyakit jantung koroner menjadi jenis penyakit gagal jantung akut.

Waktu *retrieve* terbaik (Gambar 10) yaitu pada CBR *cluster-indexing* yang memiliki kecepatan melebihi CBR *non-indexing* untuk semua variasi jumlah *cluster*. Kecepatan *retrieve* tertinggi terjadi pada variasi 4 jumlah *cluster* sebesar 0.0312 detik atau terjadi peningkatan dari CBR *non-indexing* sebesar 34.18%

## KESIMPULAN

Dengan menggunakan CBR *cluster-indexing*, kecepatan *retrieve* meningkat untuk semua variasi jumlah *cluster*. Jika dibandingkan dengan CBR *non-indexing*, terjadi peningkatan kecepatan *retrieve* sebesar 34.18% yaitu pada variasi 4 jumlah *cluster* (*retrieve* tercepat).

Secara umum akurasi sistem CBR *cluster-indexing* sama dengan sistem CBR *non-indexing* (=96%), kecuali pada variasi 3 jumlah *cluster*, sistem CBR *non-indexing* lebih baik dari sistem CBR *cluster-indexing* (=92%).

Kelemahan utama dari metode *linkage* pada AHC adalah bahwa, *clustering* harus dilakukan pada beberapa variasi jumlah *cluster* sampai ditemukannya kualitas *cluster* yang paling optimal.

## DAFTAR PUSTAKA

- Adawiyah, R., 2016, Case-Based Reasoning untuk Diagnosis Penyakit akibat Virus Dengue. *Tesis*. Jurusan Ilmu Komputer dan Elektronika FMIPA-UGM. Yogyakarta. 87p
- Elhadad, M. K., and Badran, K. M., 2017, A Novel Approach for Ontology-Based Dimensionality reduction for web text document classification. p373–378
- Mohsin, M. F. N., Manaf, M., Norwawi, N. Md., and Wahab, M. A. H., 2011, Faster Case Retrieval Using Hash Indexing Technique. *International Journal of Artificial Intelligence and Expert Systems (IJAE)*. Vol. 2: Issue (2). [Online]. Tersedia:

- <http://www.cscjournals.org/manuscript/Journals/IJAE/Volume2/Issue2/IJAE-42.pdf> [24 Juni 2017]
- Mooi, E., and Sarstedt, M., 2011, *A Concise Guide to Market Research. The Process, Data, and Methods Using IBM SPSS Statistics*. Springer Heidelberg Dordrecht London New York. 252p
- Nunez, H., Marre, M. S., Cartez, U., Comas, J., Martinez, M., Roda, I. R. and Poch, M., 2004, *A Comparative Study on the Use of Similarity Measures in Case-based Reasoning to Improve the Classification of environmental system situations*. Environmental Modelling & Software XX (2003) XXX–XXX. Catalonia-Girona. p2-11
- Prasetyo, E., 2014, *Data Mining. Mengolah Data Menjadi Informasi Menggunakan Matlab*. Andi Yogyakarta. 19p. 46p
- Richter, M. M., and Weber, R.O., 2013, *Case-Based Reasoning, A Textbook*. ©Springer-Verlag Berlin Heidelberg, p17-22
- Rismawan, T., dan Hartati, S., 2012, Case-Based Reasoning untuk Diagnosis Penyakit THT (Telinga Hidung dan Tenggorokan). *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol.6, no.2, Juli 2012. Yogyakarta. p67-78
- Salem, A. B. M., Roushdy, M. and Hodhod, R. A. 2005, A Case Based Expert System for Supporting Diagnosis of Heart Diseases. *AIML Journal*. March 2005. Volume (5). Issue (1). Cairo. p33-39
- Santoso, H., 2017. Metode Indexing pada Case-Based Reasoning (CBR) Menggunakan Density Based Spatial Clustering Application With Noise (DBSCAN). *Tesis*. Jurusan Ilmu Komputer dan Elektronika FMIPA-UGM. Yogyakarta. 2p
- Shi, H., Xin, M., and Dong, W., 2011, A Kind of Case Similarity Model Based on Case-Based Reasoning. *International Conferences on Internet of Things, and Cyber, Physical and Social Computing IEEE*. p453-457
- Witten, I. H., Frank, E., and Hall, M. A., 2011, *Data Mining: Practical Machine Learning Tools and Techniques*. USA: Morgan Kauffmann. Burlington. 337p