Implementasi dan AnalisisAlgoritma Clustering COBWEB (EM) Pada Data Tugas Akhir Universitas Telkom

Jonathan Togatorop[1], Eko Darwianto ST., MT.[2], Dawam Dwi Jatmiko Suwawi, ST., MT.[3]
[1,2,3]Fakultas Informatika Universitas Telkom, Bandung
joeyakuza90@gmail.com

Abstrak

Perkembangan teknologi saat ini menghasilkan perkembangan data yang sangat pesat, mulai dari bidang ekonomi, industri, pendidikan serta berbagai bidang lainnya. Institusi pendidikan saat ini juga memiliki kumpulan data diantaranya data mahasiswa, data pegawai ,data tugas akhir dan data lainnya. Universitas Telkom memiliki kumpulan data tugas akhir terdiri atas dokumen tugas akhir dari seluruh jurusan yang ada di Universitas Telkom. Dokumen tersebut memiliki jumlah yang sangat besar sehingga sulit bagi mahasiswa atau dosen untuk mencari secara tepat dokumen yang mana yang inginkan.

Oleh karena ini dibutuhkan suatu cara pengorganisasian dari dokumen-dokumen tersebut agar lebih terstruktur, yaitu dengan pengelompokan dokumen. *Clustering*, Salah satu algoritma yang dapat digunakan untuk mengelompokkan dokumen adalah *COBWEB*. *COBWEB* merupakan salah satu contoh algoritma *hierarchical* karena mengorganisasikan data menjadi *clasification tree*. Berdasarkan hasil pengujian sistem pengelompokan dokumen menggunakan algoritma *COBWEB* didapat nilai *internal similarity* total akan berkurang jika jumlah dokumen yang dikelompokkan bertambah. Sedangkan nilai *internal similarity* pada suatu *cluster* akan bertambah jika jumlah dokumen yang dikelompokkan bertambah.

.

Kata Kunci: *Clustering, COBWEB, Hierarchical, Internal similarity*

Abstract

Current technological developments resulted in the development of data very rapidly, ranging from economics, industry, education and various other fields. Educational institutions today have a data set including student data, employee data, the final task data and other data. Telkom University have final project data set consists of the final project documents from all departments in the University of Telkom. The Document has a very large number making it difficult for students or lecturer to locate precisely which documents are required.

Therefore, it needed a way to organize these documents to be more structured, with grouping documents. Clustering, is one algorithm that can be used to grouping the document was a COBWEB. COBWEB b is one example of hierarchical algorithm for organizing data into a tree clasification. Based on document grouping test results using cobweb algorithms, total internal similarity value obtained will be reduced if the number of grouped documents increases. While the internal similarity value to a cluster will increase if the number of grouped documents increases.

***Keywords** : Clustering, COBWEB, Hierarchical, Internal similarity*

1 Pendahuluan

Perkembangan teknologi saat ini menghasilkan perkembangan data yang sangat pesat, mulai dari bidang ekonomi, industri, pendidikan serta berbagai bidang lainnya. Institusi pendidikan saat ini juga memiliki kumpulan data diantaranya data mahasiswa, data pegawai ,data tugas akhir dan data lainnya. Universitas Telkom memiliki kumpulan data tugas akhir terdiri atas dokumen tugas akhir dari seluruh jurusan yang ada di Universitas Telkom. Dokumen tersebut memiliki jumlah yang sangat besar sehingga sulit bagi mahasiswa atau dosen untuk mencari secara tepat dokumen yang mana yang inginkan.

Oleh karena ini dibutuhkan suatu cara pengoganisasian dari dokumen-dokumen tersebut agar lebih terstruktur, yaitu dengan pengelompokan dokumen. *Clustering*, dapat menjadi alternatif cara dalam mengelompokkan dokumen tersebut. *Document Clustering* atau pengelompokan dokumen dilakukan untuk mengelompokkan obyek-obyek data berdasarkan kesamaan karakteristik di antara obyek-obyek data tersebut. Obyek data tersebut akan dikelompokkan ke dalam satu atau lebih cluster sehingga obyek-obyek data yang berada di dalam satu cluster akan mempunyai kemiripan karakteristik satu dengan yang lainnya. Algoritma *clustering* yang sering digunakan saat ini yaitu *partitional* (*Expectation-maximization, K-Means*) dan *hierarchical* (*COBWEB, Centroid Linkage, Single Linkage*), *overlapping* (*Fuzzy C-Means*) dan *hybrid* .

Salah satu algoritma yang dapat digunakan untuk mengelompokkan dokumen adalah *COBWEB*. *COBWEB* merupakan salah satu contoh algoritma *hierarchical* karena mengorganisasikan data menjadi *clasification tree*. Algoritma ini menerapkan *incremental concept* yang artinya proses *clustering* dapat dilakukan secara terus-menerus tanpa harus menganalisis keseluruhan dataset[9]. *COBWEB* melakukan clustering data dengan membangun pohon klasifikasi di mana tiap

node dari pohon tersebut menggambarkan cluster yang berisi objek-objek data. Dalam membangun pohon, *COBWEB* menggunakan *category utility* (CU) untuk mengevaluasi tree dan mendapatkan pengelompokan data yang paling tepat[2]. Sebelum dilakukan pengelompokan dokumen dilakukan terlebih dahulu proses *processing*, yaitu *cleansing, tokenizing, parsing, stopword elimination,* dan *stemming.* Proses ini diperlukan untuk mengurang jumlah kata yang diproses pada saat *clustering.*

Pada tugas akhir ini akan diteliti bagaimana proses bagaimana proses *clustering* dokumen, khususnya dokumen berbentuk teks menggunakan algoritma *COBWEB* sebagai algoritma clustering incremental dan hierarkhi . Dengan menggunakan *clustering COBWEB* diharapkan dapat menghasilkan cluster yang berkualitas baik dan menyajikan dokumen sesuai dengan kebutuhan user.

2 Landasan Teori

2.1 Text Mining

Jumlah data yang sangat besar dapat membuat kita sulit untuk menemukan informasi yang terdapat dari data itu sendiri. Salah satu teknik untuk menemukan informasi yang terdapat dalam data adalah *data mining.*

Data mining merupakan proses pengambilan informasi atau korelasi antar data dari sebuah *dataset.* Dimana *dataset* ini memiliki 2 atau lebih parameter yang menggambarkan karakteristik dari sebuah populasi atau kumpulan data. Salah satu bagian dari *data mining* yang cukup menarik adalah *text mining.* Metode ini digunakan untuk menggali informasi dari data-data dalam bentuk teks. Yang membedakan *data mining* dan *text mining* adalah proses analisis terhadap suatu data. *Data mining* adalah proses untuk menemukan informasi dari dari sejumlah besar data yang disimpan baik di dalam *database*, *data warehouse* atau penyimpanan data lainnya[4]. Sedangkan *text mining* adalah sebuah analisis yang mengumpulkan *keywords* dan *terms* yang sering muncul secara bersamaan dan kemudian menemukan hubungan asosiasi dan korelasi diantara *keywords* dan *terms* tersebut[7].
Teks mining merupakan pencarian pola yang menarik atau pola yang berguna pada sebuah informasi textual yang tidak terstruktur, atau bisa didefenisikan sebagai proses menganalisis teks untuk mengekstraksi informasi dengan tujuan tertentu. Teks mining dapat menganalisis keyword dari suatu dokumen atau hanya daftar kata dalam dokumen yang bersangkutan dengan aturan-aturan tertentu[3].

2.2 Clustering

Data mining adalah kegiatan ekstraksi atau menambang informasi dan pola yang berguna dari data yang berukuran besar[4].

Secara umum, data mining terbagi dalam 2 sifat [4]:

a. Predictive: menghasilkan model berdasarkan sekumpulan data yang dapat digunakan untuk memperkirakan nilai data yang lain. Metode-metode yang termasuk Predictive Data Mining adalah: Klasifikasi, Regresi, dan Time series Analysis.
b. Descriptive: mengidentifikasi pola atau hubungan dalam data untuk menghasilkan informasi baru. Metode yang termasuk dalam Descriptive Data Mining adalah:

- *Clustering*: identifikasi kategori untuk mendeskripsikan data.
- *Association Rules*: identifikasi hubungan antara data yang satu dengan lainnya.
- *Summarization*: pemetaan data ke dalam subset dengan deskripsi sederhana.
- *Sequence Discovery*: identifikasi pola sekuensial dalam data.

Clustering juga dikenal sebagai *unsupervised learning* yang membagi data menjadi kelompok-kelompok atau clusters berdasarkan suatu kemiripan atribut-atribut di antara data tersebut. Secara umum pembagian algoritma clustering terdiri atas *partitional clustering* dan *hierarchical clustering*. *Partitional* mengelompokkan data dengan memilah-milah data yang dianalisa ke dalam cluster-cluster yang ada. *Clustering* dengan pendekatan hirarki atau sering disebut dengan *hierarchical clustering* mengelompokkan data dengan membuat suatu hirarki berupa dendogram dimana data yang mirip akan ditempatkan pada hirarki yang berdekatan dan yang tidak pada hirarki yang berjauhan. Contoh beberapa algoritma *hierarchical clustering* adalah *ROCK, CHAMELEON, COBWEB,* dan *SNN*. Sedangkan contoh dari algoritma *partitional clustering* diantaranya adalah *K-Means, Clara, EM, dan Bond Energy*.

2.3 Algoritma COBWEB

COBWEB adalah suatu sistem *incremental* untuk *hierarchical conceptual clustering* Metode ini melakukan melakukan *hill-climbing searching* melalui skema ruang hierarchical classification menggunakan operator yang memungkinkan perjalanan dua arah melalui ruang ini[2].

Dalam algoritma *COBWEB* ada beberapa hal yang diperlukan untuk mengimplementasi algoritma tersebut[2]. Antara lain:

- Fungsi evaluasi heuristik untuk mengetahui di cluster mana, suatu obyek disimpan.
- Representasi state yang berguna untuk menentukan struktur hierarchy dan representasi concept.
- Operator yang digunakan dalam

membangun skema klasifikasi.

- Strategi kontrol yang digunakan untuk mendeskripsikan algoritma COBWEB, termasuk deskripsi tingkat tinggi sistem.

✓ Fungsi evaluasi heuristic

Fungsi yang digunakan dalam *COBWEB* yaitu *category utility* (CU) . Fungsi ini digunakan untuk mengevaluasi kegunaan suatu kategori atau partisi dengan menggunakan strategi perhitungan probabilitas.

Category utility merupakan fungsi yang digunakan untuk memberikan nilai dan mengetahui kualitas suatu partisi . Suatu partisi cluster mempunyai nilai category utility. Berikut ini fungsi category utility

$$CU_p[C_1,\dots,C_K] = \frac{\sum_{k=1}^{K} P(C_k)\sum_i \sum_j [P(A_i = V_{ij}|C_k)^2 - P(A_i = V_{ij}|C_p)^2]}{K}$$

Dimana P(C_k)= probabilitas bahwa sebuah dokumen acak termasuk dalam cluster anak C_k

A_i = atribut ke –i dari suatu dokumen

V_{ij} = nilai ke –j dari atribut ke -i

✓ Representasi state (representasi concept)

Dalam *COBWEB*, suatu concept yang umum memiliki semua obyek dari concept-concept khusus yang ada dibawahnya. Concept juga harus memiliki kepekaan terhadap perubahan dari attribute-value yang dimilikinya. Selain itu, sebuah concept harus dapat dievaluasi dengan fungsi CU . Oleh karena itu, dalam *COBWEB* sebuah concept direpresentasikan dengan node.

✓ Operator

COBWEB secara bertahap menggabungkan objek ke dalam classification tree, dimana setiap node adalah probabilistic concept yang merepresentasikan kelas. Penggabungan objek merupakan proses pengelompokan objek dengan menurunkan pohon di jalur yang tepat, memperbarui jumlahnya sepanjang jalan. Operator dalam algoritma *COBWEB* meliputi:

- Mengklasifikasikan objek ke dalam node
- Membuat node baru
- Mengkombinasikan 2 node menjadi 1 node (merge node)
- Membagi sebuah node menjadi 2 node (split node)

✓ Strategi kontrol

Algoritma *COBWEB* memiliki strategi kontrol yang digunakan dalam tahap- tahap pembentuk hierarchy concept. Berikut ini akan ditampilkan strategi kontrol algoritma *COBWEB*.

Pseudo code COBWEB:

1 FUNCTION COBWEB (Object, Root (of a classification tree))
2 Update count of the root
3 IF Root is a leaf
THEN Return the expanded leaf to accommodate the new object
ELSE Find that child of Root that beat host Object and perform one of the following

a) Consider creating a new class and do so if appropriate
b) Consider node merging and so do if appropriate and call COBWEB (Object, Merge Node)
c) Consider node splitting and do so if appropriate and call COBWEB (Object, Root)
d) IF none of the above (a, b, or c) were performed THEN call COBWEB (Object, Best child of the Root)

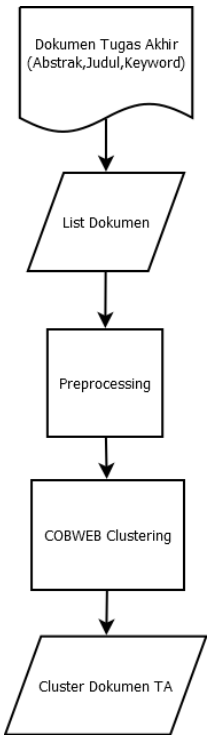
Dari penjelasan per tahap diatas dapat dilihat bahwa dalam tiap tahapnya dilakukan pengambilan keputusan yang terbaik sehingga mendapatkan output yang baik (hill climbing search).

3 Metode Penelitian Dan Perancangan

3.1 Gambaran Umum Sistem

Pada tugas akhir ini sistem yang dibangun adalah perangkat lunak untuk clustering dokumen teks berbahasa Indonesia. Sistem yang akan dibangun adalah sistem yang mampu mengelompokkan document teks dengan menggunakan algoritma *COBWEB* .

Pada awalnya pengguna meng-*input*-kan dokumen ke database sehingga menjadi list dokumen. Lalu list dokumen tersebut akan melalui proses *preprocessing* . Setelah melalui proses *preprocessing* maka langkah selanjutnya adalah *document clustering* dengan menggunakan algoritma *COBWEB* yang akan menghasilkan *Cluster-Cluster* dari dokumen TA. Gambar 3.1 di bawah ini menunjukkan gambaran umum sistem secara flowchart.



Gambar 3-1 Gambaran Umum Sistem

3.2 Preprocessing

Tahap Preprocessing adalah tahap pengubahan dokumen ke dalam bentuk term-term. Tahap ini mempunyai beberapa proses yaitu cleansing, tokenizing, parsing, stopword removal dan stemming.

3.3 Clustering

Tahap *clustering* tahapan untuk melakukan pengelompokan dokumen menggunakan algoritma *COBWEB* berdasarkan abstraksi.

3.3 Pelabelan

Setelah proses pengelompokan selesai, maka didapatkan *Cluster* yang telah memiliki satu atau lebih anggota. Setiap *Cluster* ini selanjutnya akan diberi nama yang biasa disebut label. Label dari cluster parentnya diambil dari label aktual, sedangkan untuk label dari cluster childnya diambil dari kata yang paling banyak muncul dalam suatu cluster

4 Implementasi, Pengujian dan Analisis

4.1 Implementasi

Sistem pengelompokan dokumen mempunyai tiga proses utama yaitu *preprocessing, , clustering* COBWEB, dan pelabelan. Data yang diinputkan berbentuk teks yang merupakan abstraksi dari data Tugas Akhir D-3 Universitas Telkom. Abstraksi tersebut akan melalui tahap *preprocessing* yaitu *cleansing, tokenizing, parsing,, stopword elimination (Removal)* dan *stemming.* Lalu dilakukan pengelompokan menggunakan algoritma COBWEB sehingga dokumen akan dikelompokkan menurut cluster-clusternya . Kemudian *Cluster-Cluster* tersebut akan dilabeli dengan menggunakan label yang paling banyak muncul untuk parent clusternya dan kata yang paling banyak muncul pada cluster anaknya.Tabel berikut adalah tabel fungsi-fungsi utama yang digunakan pada sistem.

Tabel 4-1 Tabel Fungsi pada sistem

No.	Proses	Fungsi	Keterangan
1.	Preprocessing	function cleansing()	Fungsi untuk menghilangkan karakter-karakter selain huruf seperti tanda baca, dan simbol
		function tokenizing()	Fungsi untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Huruf yang diterima hanya huruf ‘a’ sampai dengan ‘z’
		function parsing()	Fungsi untuk mengubah dokumen menjadi kumpulan kata atau daftar kata (term)
		function stopword()	Fungsi untuk membuang kata-kata yang sering muncul dan tidak memiliki arti deskriptif terhadap isi dokumen. Kata-kata yang termasuk dalam stopwords, misalnya kata ‘yang’, ‘di’, ‘dari’ dan sebagainya
		function stemming()	Fungsi untuk mencari *root*

			atau kata dasar dari setiap kata hasil stopwords removal
2.	Clustering COBWEB	function COBWEB ()	Fungsi untuk melakukan pengelompokan dokumen kedalam beberapa klaster
3.	Pelabelan	function labelfromDB ()	Fungsi untuk memberikan label terhadap setiap klaster yang terbentuk

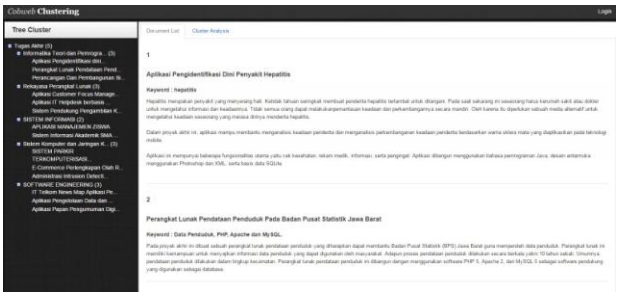
4.2 Pengujian

Pengujian dilakukan dengan menghitung nilai peformansi algoritma *COBWEB* dengan menghitung nilai *internal similarity* berdasarkan hasil pengelompokan yang telah ditentukan jumlah dokumen per clusternya

Tabel 4-2 Tabel Data Pengujian

No	*Jumlah Dokumen*	Jumlah Label Aktual	Jumlah *Cluster* (K)
1	*3*	5	6
2	*5*	5	7
3	*7*	5	9

- **Percobaan 1**
  Pada percobaan pertama jumlah dokumen yang digunakan perclusternya adalah 3 dokumen



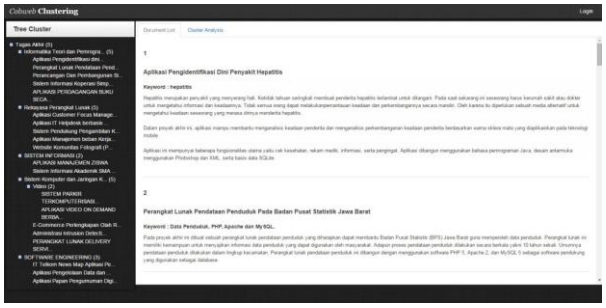
Gambar 4-1 Pengelompokan Percobaan 1

Tabel 4-3 Pengelompokan Percobaan 1

No.	Cluster	Label *Cluster*	Anggota	*Internal similarity*	*Internal similarity* Total
1	1	Tugas Akhir	Semua dokumen	0.1479	0.16131472442223
2	2	Informatika Teori dan Pemrograman	1,2,3	0.0933	
3	3	Rekayasa Perangkat Lunak	77,78,79	0.1065	
4	4	Sistem Informasi	404,405	0.2929	
5	5	Sistem Komputer dan Jaringan Komputer	406,407, 408	0.1106	
6	6	Software Engineering	427,428, 429	0.2164	

- **Percobaan 2**
  Pada percobaan kedua jumlah dokumen yang digunakan perclusternya adalah 5 dokumen

Gambar 4-2 Pengelompokan Percobaan 2



Tabel 4-4 Pengelompokan Percobaan 2

No.	Cluster	Label *Cluster*	Anggota	*Internal similarity*	*Internal similarity* Total
1	1	Tugas Akhir	Semua dokumen	0.14869	0.16032071240948
2	2	Informatika Teori dan Pemrograman	1,2,3,4,5	0.1329	
3	3	Rekayasa Perangkat Lunak	77,78,79 80,81	0.1261	
4	4	Sistem Informasi	404,405	0.2929	
5	5	Sistem Komputer dan Jaringan Komputer	406,407, 408,409, 410	0.11942	
6	6	Video	406,410	0.0856	
7	7	Software Engineering	427,428,429	0.2164	

- **Percobaan 3**

Pada percobaan ketiga jumlah dokumen yang digunakan perclusternya adalah 7 dokumen

Gambar 4-3 Pengelompokan Percobaan 3

Tabel 4-5 Pengelompokan Percobaan 3

No.	Cluster	Label *Cluster*	Anggota	*Internal similarity*	*Internal similarity* Total
1	1	Tugas Akhir	Semua dokumen	0.1437	0.13532365026682
2	2	Informatika Teori dan Pemrograman	1.2.3.4.5.6.7	0.1343	
3	3	Bahasa	2.6	0.06598	
4	4	Rekayasa Perangkat Lunak	77,78,79,80, 81, 82.83	0.12563	
5	5	Sistem Informasi	404,405	0.2929	
6	6	Sistem Komputer dan Jaringan Komputer	406,407,408, 409,410,411,412	0.1132	
7	7	Video	406, 410	0.0856	
8	8	IDS	408, 412	0.0272	
9	9	Software Engineering	427,428,429	0.21643	

4.3 Analisis Hasil Pengujian

Pada percobaan di atas dapat dilihat bahwa nilai *internal similarity* total akan berkurang jika jumlah dokumen yang dikelompokkan bertambah. Hal ini disebabkan oleh semakin banyaknya jumlah kata yang diproses sehinga jumlah total kesamaan antar kata pada dokumen pada seluruh *cluster* semakin berkurang sehingga nilai *internal similarity* totalnya semakin kecil. Sedangkan untuk nilai *internal similarity* per *cluster*nya, jika dokumen yang dikelompokkan bertambah nilai *internal similarity*nya semakin besar. Hal ini disebabkan oleh jumlah kesamaan antar kata pada dokumen yang terdapat pada suatu *cluster*, dimana *cluster* itu dikelompokkan berdasarkan kesamaan dari kata-kata yang terdapat dalam dokumen dalam *cluster* tersebut, sehingga nilai *internal similarity* semakin besar

5 Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian sistem pengelompokan dokumen menggunakan algoritm COBWEB maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma COBWEB dapat mengelompokkan dokumen tugas akhir Universitas Telkom dengan mencari nilai CU tertinggi dalam membentuk tree.
2. Nilai *internal similarity total* akan berkurang jika jumlah dokumen yang dikelompokkan bertambah .
3. Nilai *internal similarity* pada suatu *cluster* akan bertambah jika jumlah dokumen yang dikelompokkan bertambah.

5.2 Saran

Perlu ditambahkan proses sinonim kata pada bagian *preprocessing* untuk mengurangi jumlah term yang akan diproses, sehingga nilai *internal similarity* antar dokumen semakin tinggi.

Daftar Pustaka

[1] Arifin,A.Z. & Setiono, A.N. Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering. Surabaya : Institut Teknologi Sepuluh November

[2] Fisher, D.H. 1987. *Knowledge Acquisition Via Incremental Conceptual Clustering* . California; University of California .

[3] Francis, L., Flynn, M. 2010. *Text Mining Handbook*., Casualy Actuarial Society E-forum

[4] Han, J. & Amber, M. 2006. *Data Mining Concept and Techniques Second Edition.* San Fransisco; Morgan Kauffman.

[5] Sahoo, N. Callan, J & Krishnan, R A ,*Incremental Hierarchical Clustering of Text Documents*

[6] Steiibach, M. Karypis, G & Kumar, V. *2000 A Comparison of Document Clustering Techniques* ; University of Minnesota

[7] Tan, A.H. *Text Mining*: *The state of the art and the challenges* Singapore; Kent Ridge Digital Labs

[8] Vijayarani,S. Ilamathi,J. & Nithya. *Preprocessing Techniques for Text Mining* .Coimbatore, Tamilnadu, India ; Bharathiar University,.

[9] Yova,R. & Adolf, P. 2008. Implementasi Dan Annalisis Algoritma COBWEB Dan ITERATE Dalam Conceptual . Bali ; Universitas Indonesia .