PROPOSAL TUGAS AKHIR

Clustering Tugas Akhir Program Studi Informatika Universitas Sebelas Maret (UNS) Menggunakan Metode Hierarchical K-Means

Disusun oleh :

Desti Yulianingtyas

M0513016

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

SURAKARTA

2021

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

SURAKARTA

JANUARI 2021

UNIVERSITAS SEBELAS MARET
PROGRAM STUDI INFORMATIKA

PROPOSAL SKRIPSI

Nama : Desti Yulianingtyas
NIM : M0513016

PERSETUJUAN PEMBIMBING

Proposal Skripsi ini telah disetujui oleh :

Pembimbing I	Pembimbing II
Drs. Bambang Harjito, M.App.Sc.,Ph.D.	Dr. Umi Salamah, S.Si., M.Kom.
NIP. 196211301991031002	NIP. 197002171997022001

BAB I
PENDAHULUAN

1.1 Latar Belakang

Setiap tahunnya Program Studi Informatika meluluskan mahasiswa dengan penelitian tugas akhir yang beragam. Begitu juga setiap tahunnya jumlah data tugas akhir selalu bertambah. Semakin bertambahnya penelitian tugas akhir dengan mata kuliah terbatas menyebabkan semakin banyak pula mahasiswa yang mengambil penelitian yang mirip tema, objek, atau metode penelitian dengan penelitian sebelumnya. Penelitian tugas akhir dapat dikelompokkan berdasarkan kemiripan tema, objek maupun metode penelitian. Hasil pengelompokan penelitian tugas akhir dapat memperlihatkan bagaimana pola kemiripan penelitian dari waktu ke waktu. Hasil pengelompokan juga dapat memperlihatkan materi yang banyak diambil mahasiswa dan jarang diambil. Dengan dilakukannya penelitian ini, diharapkan dapat membantu dosen dalam mengevaluasi metode pembelajaran yang telah dilakukan.

Pengelompokan data penelitian yang umumnya berbentuk teks dapat dilakukan dengan *text mining*. *Text mining* juga disebut sebagai *Text Data Mining* (TDM) atau *Knowledge Discovery in Text* (KDT), secara umum mengacu pada proses ekstraksi informasi dari dokumen-dokumen teks tak terstruktur (*unstructured*). *Text Mining* merupakan variasi dari data mining yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar. *Text Mining* digunakan pada klasifikasi, *clustering, information extraction retrival* (Tan, 1999). Terdapat beberapa metode *text mining* salah satunya adalah *clustering.*

Clustering dapat mengolah data dalam jumlah banyak dan mengelompokkan dokumen yang belum terstruktur atau belum terkelompok dengan baik (Srivastava & Sahami, 2009). Metode *Clustering* yang kita kenal terbagi menjadi dua, yaitu *Hierarchical Clustering* dan *Partitioned Clustering. Hierarchical clustering* mengelompokkan data secara bertahap, sedangkan *Partitioned clustering* langsung mengelompokkan data dengan menentukan cluster di awal proses *clustering.* Salah satu metode *Partitioned clustering* adalah *K-means clustering.*

Metode *Hierarchical clustering* dapat digunakan untuk mengatasi masalah penentuan pusat *cluster* pada *K-means.* Penelitian ini mengombinasikan *K-means clustering* dengan *Hierarchical clustering.* Hasil dari *Hierarchical clustering* akan digunakan dalam penentuan pusat

cluster K-means clustering. Kombinasi antara kedua metode ini telah diuji dan terbukti bahwa kombinasi ini lebih baik dari pada menggunakan *K-means clustering* saja.

Penelitian ini mengelompokkan laporan tugas akhir dari Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret. Program studi Informatika dipilih karena jumlah data penelitian yang cukup banyak.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan dalam penelitian ini dapat dirumuskan, yaitu bagaimana pola penelitian di Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret setelah dilakukan *text mining* dengan kombinasi metode *K-means Clustering* dan *Hierarchical Clustering?*

1.3 Batasan Masalah

Dalam penulisan tugas akhir ini, batasan masalah yang digunakan, antara lain :

Menggunakan bagian laporan tugas akhir hanya pada bagian abstrak yang berbahasa Indonesia.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengelompokkan laporan penelitian tugas akhir ke dalam *cluster-cluster* dengan kombinasi metode *K-means Clustering* dan *Hierarchical Clustering,* mengamati pola yang terbentuk dari hasil *clustering.*

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah dapat mengelompokkan laporan penelitian tugas akhir dan mengetahui pola penelitian yang terbentuk berdasarkan hasil *clustering.*

1.6 Sistematika Penulisan

Penelitian ini akan disusun dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab I merupakan bab pendahuluan yang menguraikan latar belakang masalah, rumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab II berisi tentang teori-teori yang dijadikan sebagai landasan dalam penelitian.

BAB III METODE PENELITIAN

Bab III menguraikan tentang gambaran objek penelitian, proses pengumpulan data, serta gambaran langkah-langkah yang dilakukan oleh penulis untuk melaksanakan dan menyelesaikan penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

Bab IV berisi tentang bagaimana menyelesaikan masalah yang telah dirumuskan berdasarkan metode yang dipilih dan berusaha untuk mewujudkan tujuan, serta manfaat yang ingin diraih.

BAB V PENUTUP

Bab V menguraikan kesimpulan penelitian tugas akhir dan saran-saran sebagai bahan pertimbangan untuk pengembangan penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Dasar Teori

Pada tugas akhir ini digunakan dasar-dasar teori yang menjadi landasan utama penelitian, antara lain sebagai berikut :

2.1.1 *Text Mining*

Text Mining yang juga disebut sebagai *Text Data Mining* (TDM) atau *Knowledge Discovery in Text* (KDT), secara umum mengacu pada proses ekstraksi informasi dari dokumen-dokumen teks tak terstruktur (*unstructured*). *Text Mining* dapat didefinisikan sebagai penemuan informasi baru dan tidak diketahui sebelumnya oleh computer, dengan secara otomatis mengekstrak informasi dari sumber – sumber teks tak terstruktur yang berbeda (Tan, 1999). Definisi singkatnya adalah suatu proses menganilisa teks untuk mengekstrak informasi yang berguna untuk tujuan tertentu. Perbedaan mendasar dari text mining dan data mining terletak pada sumber data yang digunakan. Pada data mining data yang diekstrak berasal dari pola-pola tertentu dan terstruktur, sedangkam text mining sumber data yang digunakan berasal dari teks yang relatif tidak testruktur karena menggunakan tata bahasa manusia atau biasa disebut (natural language). Secara umum basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks ditulis untuk dibaca langsung oleh manusia (Herast, 2003). Dalam penelitian ini memanfaatkan *text mining* dalam hal *clustering.* Proses *text mining* pada penelitian ini terdiri dari *text preprocessing, term weighting, feature selection,* dan *clustering.*

2.1.2 *Text Preprocessing*

Text preprocessing merupakan salah satu komponen dari *text mining. Text preprocessing* adalah tahapan di mana kita melakukan seleksi data agar data yang akan kita olah menjadi lebih terstruktur. Terdapat beberapa tahapan untuk memproses data teks tersebut, yaitu :

a. *Tokenization* dan *Case Folding*

   Tokenization adalah memotong sebuah kalimat berdasarkan tiap kata yang menyusunnya, sedangkan c*ase folding* yaitu tahap mengubah semua karakter huruf pada sebuah kalimat

menjadi huruf kecil (*lowercase*) dan hanya dapat menerima hurif a sampai z, menghilangkan karakter tidak valid seperti angka, tanda baca serta *Uniform Resources Locator* (URL).

b. *Filtering*

Filtering adalah tahap mengambil kata-kata penting dari hasil *tokenization* dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting).

c. Normalisasi

Normalisasi adalah proses di mana singkatan dalam dokumen dinormalkan, normalisasi yang dimaksud adalah mengubah singkatan menjadi kepanjangannya. Singkatan yang digunakan dalam normalisasi adalah singkatan dalam Bahasa Indonesia.

d. *Stemming*

Stemming adalah proses menghilangkan infleksi kata ke bentuk dasarnya, namun bentuk dasar tersebut tidak berarti sama dengan akar kata (root word). Misalnya kata "mendengarkan", "didengarkan" akan ditransformasi menjadi kata "dengar".

e. *Stopword*

Stopword adalah proses di mana penghapusan kata-kata yang sering muncul seperti kata sambung, kata depan, dan kata ganti orang. Pada sebuah rangkaian kalimat atau paragraf dalam Bahasa Indonesia terdapat sebuah kata yang muncul berulang kali. Kemunculan kata tesebut sangat kecil berpengaruh dalam sebuah dokumen, dalam hal pencocokan deskripsi dikarenakan kata tersebut hanya sebagai penghubung antar kata seperti halnya : dan, ke, atau, yang, atau, sebagai, dari, adalah, ke mana, di mana, tetapi, tersebut, walaupun, dan lain sebagainya. Manfaat penggunaan *stopword* mampu mengurangi jumlah kata dalam suatu kalimat atau paragraf dalam sebuah deskripsi untuk membantu pemilihan kecocokan deskripsi yang akan diproses dan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya.

2.1.3 *TF-IDF* (*Term Frequency-Inverse Document Frequency)*

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval* (pencarian informasi). Metode *Term Frequency Inverse Document Frequency (TFIDF)* adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. *TFIDF* ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen atau dalam sekelompok kata. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen.

Sedangkan IDF (*Inverse Document Frequency*) merupakan frekuensi kemunculan term di pada keseluruhan dokumen. Nilai idf berkaitan dengan distribusi *term* di berbagai dokumen. *Term* yang jarang muncul pada keseluruhan dokumen memiliki nilai idf lebih besar dibandingkan dengan *term* yang bersangkutan, maka nilai idf dari *term* tersebut adalah nol. Hal tersebut menunjukkan bahwa setiap *term* yang muncul pada dokumen dalam koleksi tidak berguna untuk membedakan dokumen berdasarkan topik tertentu (Manning, et al., 2008).

Metode ini akan menghitung bobot setiap *term* (kata) t di dokumen d dengan rumus :

$$W_{dt} = tf_{dt} \times IDF_t \quad (1)$$

Di mana :

d : dokumen ke-d

t : kata ke-t dari kata kunci

W : bobot dokumen ke-d terhadap kata ke-t

tf : banyaknya *term* yang dicari pada sebuah dokumen

IDF : *Inverse Document Frequency*, didapatkan dengan rumus

$$log\left(\frac{D}{df}\right) \quad (2)$$

Di mana :

D : total dokumen

Df : banyak dokumen yang mengandung *term* yang dicari

Perhitungan bobot dari *term* tertentu dalam sebuah dokumen dengan menggunakan tf x idf menunjukkan bahwa deskripsi terbaik dari dokumen adalah *term* yang banyak muncul dalam dokumen tersebut dan sangat sedikit muncul pada dokumen lain.

2.1.4 Pengelompokkan Data (*Clustering)*

Clustering adalah proses mengelompokkan atau penggolongan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas *cluster*. (Andayani,2007). Clustering membagi data ke dalam gaup-grup yang mempunyai obyek yang karakteristiknya sama.

Metode clustering dibedakan menjadi dua, yaitu Hierarchical clustering dan Partitioned clustering. Hierarchical clustering mengelompokkan data secara bertahap, sedangkan partitioned clustering langsung mengeloropokkan data dengan menentukan jumlah Aster di awal proses clustering. Salah satu metode *partitioned clustering* adalah *k-means clustering.* Penelitian ini menggunakan kombinasi antara *hierarchical clustering* dan *k-means clustering.*

2.1.5 *Hierarchical Clustering*

Metode pengelompokkan hirarki biasanya digunakan apabila belum ada informasi jumlah kelompok yang akan dipilih. Arah pengelompokkan bisa bersifat *divisive* (top to down) artinya dari satu *cluster* sampai menjadi k buah cluster atau bersifat *agglomerative* (*bottom up*) artinya dari n *cluster* (dari n-buah data yang ada) menjadi k buah *cluster*. Teknik hirarki adalah teknik *clustering* membentuk konstruksi berdasarkan tingkatan tertentu seperti struktur pohon. Dengan demikian proses pengelompokkannya dilakukan secara bertingkat atau bertahap.

- *Cluster-cluster* yang mempunyai poin-poin individu. *Cluster-cluster* ini berada di level yang paling bawah.
- Sebuah *cluster* yang di dalamnya terdapat poin-poin yang dipunyai semua *cluster* di dalamnya. *Single cluster* ini berada di level yang paling atas.

Hierarchical clustering adalah salah satu algoritma *clustering* yang dapat digunakan untuk meng-*cluster* dokumen (*document clustering*). Hasil keseluruhan dari algoritma *Hierarchical clustering* secara grafik dapat digambarkan sebagai *tree* yang disebut degan *dendogram. Tree* ini secara grafik menggambarkan proses penggabungan dari *cluster-cluster* yang ada, sehingga menghasilkan *cluster* dengan level yang lebih tinggi. Cabang-cabang dalam pohon menyajikan *cluster.* Kemudian cabang-cabang bergabung pada node yang posisinya sepanjang sumbu jarak (similaritas) menyatakan tingkat di mana penggabungan terjadi (Salton, 1988).

Kemiripan antar dokumen ditentukan dengan mengukur jarak antar dokumen. Dua doktuuen yang mempunyai jarak paling kecil dikatakan mempunyai kemiripan paling tinggi dan dikelompokkan kedalam satu *cluster* yang sama. Sebaliknya dua dokumen yang mempunyai jarak paling besar dikatakan mempunyai kemiripan paling rendah dan dimasukkan ke dalam cluster yang berbeda. Langkah-langkah dalam algoritma *Hierarchical Agglomerative Clustering* (Johnson, 1967):

1. Meletakkan setiap data sebagai sebuah *cluster*.
2. Hitting *distance matrix*.
3. Gabungkan dua *cluster* yang paling dekat sesuai dengan parameter yang dipilih seperti *single, complete, average.*
4. Memperbarui *distance matrix* dari *cluster* baru dengan *cluster* yang tersisa.
5. Lakukan kembali langkah 3 dan 4 sampai yang tersisa hanya satu *cluster*.

Ada 3 metode *Hierarchical Clustering* yaitu metode *single linkage, complete linkage, average linkage,* dan *centroid linkage.* Single linkage memberikan hasil bila kelompok-kelompok digabungkan menurut jarak antara anggota-anggota yang paling dekat, *Complete linkage* memberikan hasil bila kelompok-kelompok digabungkan menurut jarak antara anggota-anggota yang paling jauh. Untuk *Average linkage* digabungkan menurut jarak rata-rata antara pasangan-

pasangan anggota masing-masing pada himpunannya. Sedangkan *Centroid linkage* digabungkan menurut jarak *centroid* antara pasangan-pasangan anggota masing-masing pada himpumannya (Salton, 1988). Metode dalam *Hierarchical Clustering* yang digunakan untuk menentukan *centroid* awal yaitu dengan *single linkage*. Dipilih metode tersebut karena *single linkage* mudah dalam penerapan.

2.1.6 K-means *Clustering*

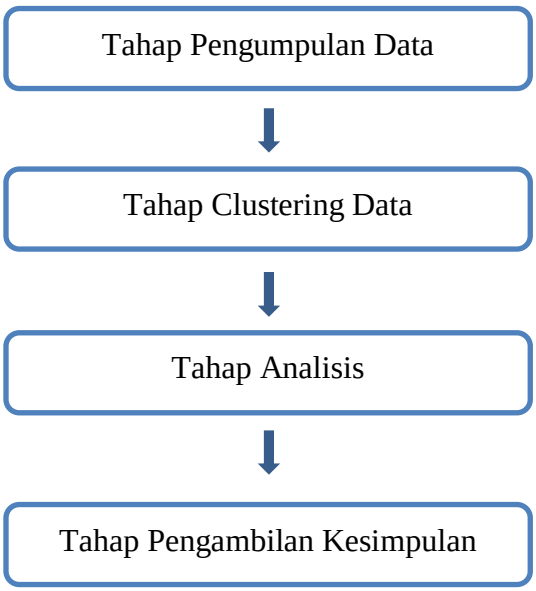
K-means adalah suatu metode penganalisaan data atau metode data mining yang melakukan proses pemodelan tanpa supervisi (*unsupervised*) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi. Metode K-means berusaha mengelompokkan data yang ada ke dalam beberapa kelompok, di mana data dalam satu kelompok mempunyai karakteristik yang sama satu sama lainnya dan mempunyai karakteristik yang berbeda dengan data yang ada di dalam kelompok yang lain. Dengan kata lain, metode ini berusaha untuk meminimalkan variasi antar data yang ada di dalam suatu *cluster* dan memaksimalkan variasi dengan data yang ada di *cluster* lainnya. (Agusta, 2011).

2.1.7 Kombinasi Metode *Hierarchical Clustering* dan K-means *Clustering*

Mengombinasikan metode *Hierarchical clustering* dengan *K-means clustering* dimaksudkan agar hasil *clustering* lebih baik. Hasil dari metode *Hierarchical clustering* digunakan untuk menentukan pusat *cluster*. Pusat *cluster* yang dihasilkan *Hierarchical clustering* selanjutnya digunakan sebagai pusat *cluster* awal pada perhitungan *K-means clustering*.

BAB III
METODE PENELITIAN

Metode penelitian mengenai *clustering* tugas akhir dengan kombinasi metode *K-means Clustering* dan *Hierarchical Clustering* ini terdiri dari dua tahapan. Tahapan penelitian terdiri dari tahapan pengumpulan data, tahap implementasi, dan tahap penulisan laporan. Langkah-langkah yang dilakukan digambarkan pada Gambar 3.1.



Gambar 3.1 Metode Penelitian

3.1 Tahap Pengumpulan Data

Pada tahap ini dilakukan pengambilan data yang akan diolah pada penelitian ini, yaitu laporan tugas akhir Program Studi Informatika, Fakultas Matematikan dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret. Laporan tugas akhir kemudian dipilih yang sesuai dengan kriteria yaitu lengkap (tidak kosong di bagian judul, tahun, dan abstrak), abstrak berbahasa Indonesia. Laporan tugas akhir yang digunakan dalam penelitian ini adalah terbit pada tahun 2015 sampai dengan 2020.

3.2 Tahap *Clustering* Data

Pada tahap *clustering* data terdiri dari 3 tahapan, yaitu *preprocessing,* TF-IDF, dan *clustering.* Tahapan *clustering* ditunjukkan pada Gambar 3.2.

3.2.1 *Preprocessing*

Data tugas akhir mahasiswa diolah dalam proses *preprocessing* dengan tujuan untuk memastikam data yang akan diolah pada proses selanjutnya adalah data yang baik. *Preprocessing* terdiri dari lima proses, yaitu *tokenization* dan *case folding, filtering,* normalisasi, *stemming,* dan *stopword.* Berikut penjelasan dari masing-masing proses:

f. *Tokenization* dan *Case Folding*

Tokenization adalah memotong sebuah kalimat berdasarkan tiap kata yang menyusunnya, sedangkan c*ase folding* yaitu tahap mengubah semua karakter huruf pada sebuah kalimat menjadi huruf kecil dan menghilangkan karakter tidak valid seperti angka, tanda baca serta *Uniform Resources Locator* (URL).

g. *Filtering*

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting).

h. Normalisasi

Normalisasi adalah proses di mana singkatan dalam dokumen dinormalkan, normalisasi yang dimaksud adalah mengubah singkatan menjadi kepanjangannya. Singkatan yang digunakan dalam normalisasi adalah singkatan dalam Bahasa Indonesia.

i. *Stemming*

Stemming adalah proses menghilangkan infleksi kata ke bentuk dasarnya, namun bentuk dasar tersebut tidak berarti sama dengan akar kata (root word). Misalnya kata “mendengarkan”, “didengarkan” akan ditransformasi menjadi kata “dengar”.

j. *Stopword*

Stopword adalah proses di mana penghapusan kata-kata yang sering muncul seperti kata sambung, kata deoan, dan kata ganti orang. Contoh *stopword* dalam Bahasa Indonesia adalah “yang”, “dan”, “di”, “dari”, dll. Makna di balik penggunaan *stopword* yaitu dengan

menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat focus pada kata-kata penting sebagai gantinya.

3.2.2 TF-IDF (*Term Frequency-Inverse Document Frequency)*

Data laporan tugas akhir yang sudah melalui proses *preprocessing*, kemudian dilakukan proses pembobotan *term* (kata). Pembobotan *term* menggunakan hasil dari proses *preprocessing* terakhir yaitu *stopword.* Metode TF-IDF ini akan menghitung nilai TF (*Term Frequency*) dan nilai IDF (*Inverse Document Frequency*) pada setiap *term*. Hasil pembobotan *term* digunakan untuk menghitung bobot setiap dokumen dengan cara menjumlahkan semua bobot *term* pada suatu dokumen.

3.2.2 *Clustering*

Pada tahap *clustering* ini menggunakan kombinasi antara dua metode, yaitu *hierarchical clustering* dan *k-means clustering.* Proses dalam metode *hierarchical clustering* ini menggunakan bobot dari setiap dokumen. Hasil dari *hierarchical clustering* ini berupa dendogram, di mana akan dipotong seuai dengan batas *threshold* yang telah dipilih. Batas *threshold* tersebut diperoleh dengan mempertimbangkan keterkaitan antar dokumen. Hasil dari pemotongan dendogram yang berupa *cluster,* akan digunakan dalam algoritma *k-means* untuk proses selanjutnya, yaitu mengelompokkan dokumen ke dalam beberapa *cluster.*

3.3 Tahap Analisis

Pada tahap analisis *clustering* ini dilakukan dengan mengamati hasil dari *cluster-cluster* yang terbentuk. bagaimana keterkaitan antara dokumen satu dengan dokumen yang lainnya dalam satu *cluster* dan menentukan tema pada setiap *cluster.*

3.4 Tahap Pengambilan Kesimpulan

Tahap ini merupakan tahap penarikan kesimpulan dari hasil *clustering* data dan analisis hasil *clustering.* Selain itu juga akan dilakukan penulisan saran untuk penelitian lebih lanjut mengenai *clustering* dokumen laporan tugas akhir.

BAB IV
PEMBAHASAN

4.1 Proses Pengumpulan Data

Proses pengambilan data pada penelitian ini dengan cara *scraping*. Proses *scraping* adalah proses mengambil data dari website dan kemudian diproses lebih lanjut. Data yang digunakan dalam proses *clustering* adalah laporan penelitian skripsi Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret yang diambil dari https://digilib.uns.ac.id/. Di mana data laporan skripsi yang diambil difilter dengan NIM awal M05 yaitu program studi S1 Informatika.

4.2 Proses Clustering Data

4.2.1 *Text Preprocessing*

Terdapat beberapa tahapan untuk memproses data teks tersebut, yaitu :

a. *Case Folding*

Tahap *case folding* adalah tahap mengubah semua karakter huruf pada sebuah kalimat menjadi huruf kecil (*lowercase*) dan hanya dapat menerima hurif a sampai z. Tabel 4.1 adalah contoh penerapan *case folding*.

Tabel 4.1 Hasil Proses *Case Folding*

Sebelum	Sesudah
Usaha Mikro, Kecil, dan Menengah (UMKM) memiliki peranan dalam mendukung ketahanan ekonomi nasional, UMKM mampu menjadi penyedia lapangan kerja di masa-masa yang sulit. Salah satu tren Information Technology (IT) yang mampu mendukung pemasaran UMKM adalah electronic marketing (e-Marketing), namun dibutuhkan biaya besar untuk merambah pemasaran online. Dengan Model e-Marketing bersama, biaya pemasaran online untuk banyak UMKM dapat	usaha mikro kecil dan menengah umkm memiliki peranan dalam mendukung ketahanan ekonomi nasional umkm mampu menjadi penyedia lapangan kerja di masa masa yang sulit salah satu tren information technology it yang mampu mendukung pemasaran umkm adalah electronic marketing e marketing namun dibutuhkan biaya besar untuk merambah pemasaran online dengan model e marketing bersama biaya pemasaran online untuk banyak umkm dapat ditekan

ditekan.Penelitian ini membangun model Web service e-Marketing Kota/Kabupaten dan e-Marketing Provinsi. Pembangunan Application Programming Interface (API) Web Service e-Marketing menerapkan REST Style dengan JavaScript Object Notation (JSON) sebagai format pertukaran data. Pembangunan e-Marketing Provinsi menerapkan Service Oriented Architecture (SOA) dengan memanfaatkan API Web Service e-Marketing di tiap Kota/Kabupaten. Dengan menggunakan API Web Service e-Marketing, e-Marketing Provinsi dapat memanfaatkan dan merekap data di tiap Kota/Kabupaten. Hasil penelitian menunjukan Model e-Marketing Provinsi dapat dibangun dengan memanfaatkan API WebService Kota/Kabupaten. Pengujian API Web Service e-Marketing memiliki rata-rata waktu 454.2 ms untuk method POST dan 288.3 ms untuk method GET.Kata Kunci: JSON, UMKM, REST, SOA, Web Service	penelitian ini membangun model web service e marketing kota kabupaten dane marketing provinsi pembangunan application programming interface api webservice e marketing menerapkan rest style dengan javascript object notation json sebagai format pertukaran data pembangunan e marketing provinsi menerapkan service oriented architecture soa dengan memanfaatkan api web service e marketing di tiapkota kabupaten dengan menggunakan api web service e marketing e marketing provinsi dapat memanfaatkan dan merekap data di tiap kota kabupaten hasil penelitian menunjukan model e marketing provinsi dapat dibangun dengan memanfaatkan api web service kota kabupaten pengujian api web service e marketing memiliki rata ratawaktu ms untuk method post dan ms untuk method get kata kunci json umkm rest soa web service

b. *Stemming*

Stemming adalah proses menghilangkan infleksi kata ke bentuk dasarnya. Beberapa contoh kata yang mengalami proses *stemming* dicetak tebal, dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Proses *Stemming*

Sebelum	Sesudah
usaha mikro kecil dan **menengah** umkm **memiliki** peranan dalam mendukung **ketahanan** ekonomi nasional umkm mampu **menjadi penyedia** lapangan kerja di masa masa yang sulit salah satu tren information technology it yang mampu mendukung pemasaran umkm adalah electronic marketing e marketing namun dibutuhkan biaya besar untuk **merambah pemasaran** online dengan	usaha mikro kecil dan **tengah** umkm milik peran dalam mendukung **tahan** ekonomi nasional umkm mampu **jadi sedia** lapangan kerja di masa masa yang sulit salah satu tren information technology it yang mampu mendukung pasar umkm adalah electronic marketing e marketing namun dibutuhkan biaya besar untuk **rambah pasar** online dengan model e marketing bersama biaya pasar

model e marketing bersama biaya pemasaran online untuk banyak umkm dapat ditekan penelitian ini membangun model web service e marketing kota kabupaten dane marketing provinsi pembangunan application programming interface api webservice e marketing menerapkan rest style dengan javascript object notation json sebagai format pertukaran data pembangunan e marketing provinsi menerapkan service oriented architecture soa dengan memanfaatkan api web service e marketing di tiapkota kabupaten dengan menggunakan api web service e marketing e marketing provinsi dapat memanfaatkan dan merekap data di tiap kota kabupaten hasil penelitian menunjukan model e marketing provinsi dapat dibangun dengan memanfaatkan api web service kota kabupaten pengujian api web service e marketing memiliki rata ratawaktu ms untuk method post dan ms untuk method get kata kunci json umkm rest soa web service	online untuk banyak umkm dapat tekan teliti ini bangun model web service e marketing kota kabupaten dane marketing provinsi bangun application programming interface api webservice e marketing terap rest style dengan javascript object notation json bagai format tukar data bangun e marketing provinsi terap service oriented architecture soa dengan manfaat api web service e marketing di tiapkota kabupaten dengan guna api web service e marketing e marketing provinsi dapat manfaat dan rekap data di tiap kota kabupaten hasil penelitian menunjukan model e marketing provinsi dapat bangun dengan manfaat api web service kota kabupaten uji api web service e marketing milik rata ratawaktu ms untuk method post dan ms untuk method get kata kunci json umkm rest soa web service

c. *Filtering*

Filtering adalah tahap menghilangkan *stopword* pada dokumen. Proses ini dilakukan dengan membuat *library stopwords* yang berisi kata hubung yang ingin dihapus, kemudian mencocokkan seluruh kata pada dokumen ke *library stopword* tersebut, apabila ada kata yang masuk ke *library stopwords*, maka kata tersebut akan dihilangkan dari dokumen.

```
 words = article.split()
for word in words:
                    if word not in stopwords:
                                filtered += word+” “
article= filtered
```

Pada program terdapat *library stopword* yang bernama *stopwords,* lalu *string* dokumen diubah menjadi list per kata menggunakan fungsi *split().* List ini akan dicek satu per satu ke *library stopwords*, apabila kata itu tidak ada dalam *library stopwords*, maka kata

tersebut akan ditambahkan ke *string filtered*. Setelah seluruh kata pada dokumen dicek, maka *string filtered* akan disimpan ke list dokumen. Contoh penerapan filter*ing* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Proses *Filtering*

Sebelum	Sesudah
usaha mikro kecil dan tengah umkm milik peran dalam mendukung tahan ekonomi nasional umkm mampu jadi sedia lapangan kerja di masa masa yang sulit salah satu tren information technology it yang mampu mendukung pasar umkm adalah electronic marketing e marketing namun dibutuhkan biaya besar untuk rambah pasar online dengan model e marketing bersama biaya pasar online untuk banyak umkm dapat tekan teliti ini bangun model web service e marketing kota kabupaten dane marketing provinsi bangun application programming interface api webservice e marketing terap rest style dengan javascript object notation json bagai format tukar data bangun e marketing provinsi terap service oriented architecture soa dengan manfaat api web service e marketing di tiapkota kabupaten dengan guna api web service e marketing e marketing provinsi dapat manfaat dan rekap data di tiap kota kabupaten hasil penelitian menunjukan model e marketing provinsi dapat bangun dengan manfaat api web service kota kabupaten uji api web service e marketing milik rata ratawaktu ms untuk method post dan ms untuk method get kata kunci json umkm rest soa web service	usaha mikro kecil tengah umkm milik peran dalam mendukung tahan ekonomi nasional umkm mampu jadi sedia lapangan kerja masa masa sulit salah satu tren information technology it mampu mendukung pasar umkm electronic marketing e marketing namun dibutuhkan biaya besar rambah pasar online model e marketing bersama biaya pasar online banyak umkm tekan teliti bangun model web service e marketing kota kabupaten dane marketing provinsi bangun application programming interface api webservice e marketing terap rest style javascript object notation json bagai format tukar data bangun e marketing provinsi terap service oriented architecture soa manfaat api web service e marketing tiapkota kabupaten api web service e marketing e marketing provinsi manfaat rekap data tiap kota kabupaten hasil penelitian menunjukan model e marketing provinsi bangun manfaat api web service kota kabupaten uji api web service e marketing milik rata rata waktu ms method post ms method get kata kunci json umkm rest soa web service

d. Tokenization

Pada tahap ini, *string* dokumen akan dipecah per kata, kemudian kata tersebut akan dicari frekuensinya dan disimpan dalam bentuk *dict* dengan *key* kata dan *value* frekuensi kata tersebut. Contoh penerapan *tokenization* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Proses *Tokenization*

usaha mikro kecil tengah umkm milik peran dalam mendukung tahan ekonomi nasional umkm mampu jadi sedia lapangan kerja masa masa sulit salah satu tren information technology it mampu mendukung pasar umkm electronic marketing e marketing namun dibutuhkan biaya besar rambah pasar online model e marketing bersama biaya pasar online banyak umkm tekan teliti bangun model web service e marketing kota kabupaten dane marketing provinsi bangun application programming interface api webservice e marketing terap rest style javascript object notation json bagai format tukar data bangun e marketing provinsi terap service oriented architecture soa manfaat api web service e marketing tiapkota kabupaten api web service e marketing e marketing provinsi manfaat rekap data tiap kota kabupaten hasil penelitian menunjukan model e marketing provinsi bangun manfaat api web service kota kabupaten uji api web service e marketing milik rata rata waktu ms method post ms method get kata kunci json umkm rest soa web service	"api": 5, "application": 1, "architecture": 1, "bagai": 1, "bangun": 4, "banyak": 1, "besar": 1, "biaya": 2, "dalammendukung": 1, "dane": 1, "data": 2, "e": 10, "ekonomi": 1, "electronic": 1, "format": 1, "get": 1, "hasil": 1, "information": 1, "interface": 1, "it": 1, "jadi": 1, "javascript": 1, "json": 2, "kabupaten": 4, "kata": 1, "kecil": 1, "kota": 3, "kunci": 1, "lapangankerja": 1, "mampu": 1,

	"mampumendukung": 1, "manfaat": 3, "marketing": 10, "marketingbersama": 1, "marketingprovinsi": 1, "masa": 2, "method": 2, "mikro": 1, "milik": 2, "model": 3, "ms": 2, "namundibutuhkan": 1, "nasional": 1, "notation": 1, "object": 1, "online": 2, "pasar": 3, "penelitianmenunjukan": 1, "peran": 1, "post": 1, "programming": 1, "provinsi": 3, "rambah": 1, "rata": 1, "ratawaktu": 1, "rekap": 1, "rest": 2, "salah": 1, "sedia": 1, "service": 5, "serviceoriented": 1, "soa": 2, "style": 1, "sulit": 1, "tahan": 1, "technology": 1, "tekan": 1, "teliti": 1,

	"tengah": 1, "terap": 2, "tiap": 1, "tiapkota": 1, "tren": 1, "tukar": 1, "uji": 1, "umkm": 5, "usaha": 1, "web": 5, "webservice": 2

4.2.2 Term Weighting

Tahap *term weighting* berfungsi untuk menentukan bobot kata pada setiap dokumen. Kata yang disimpan harus bersifat unik, sehingga kata hanya dihitung satu kali pada pembobotan TF-IDF. Perhitungan bobot dokumen dilakukan secara bertahap dimulai dengan menghitung TF, DF, IDF, dan TF-IDF. Proses perhitungan bobot ini dilakukan pada semua dokumen sebanyak 275. Proses perhitungan bobot setiap dokumen dengan menggunakan metode TF-IDF adalah sebagai berikut :

1. Menghitung nilai TF

   TF (*Term Frequency*) adalah frekuensi dari kemunculan sebuah *term* dalam dokumen. Semakin besar jumlah kemunculan suatu *term* (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. Sehingga kata yang paling sering muncul memiliki nilai TF 1, sedangkan kata yang tidak pernah muncul memiliki nilai 0. Contoh hasil perhitungan TF terdapat pada Tabel 4.5.

   Tabel 4.5 Hasil Perhitungan TF

{ "api": 0.005030181086519115, "application": 0.001006036217303823,

"architecture": 0.001006036217303823,
"bagai": 0.001006036217303823,
"bangun": 0.004024144869215292,
"banyak": 0.001006036217303823,
"besar": 0.001006036217303823,
"biaya": 0.002012072434607646,
"dalammendukung": 0.001006036217303823,
"dane": 0.001006036217303823,
"data": 0.002012072434607646,
"e": 0.01006036217303823,
"ekonomi": 0.001006036217303823,
"electronic": 0.001006036217303823,
"format": 0.001006036217303823,
"get": 0.001006036217303823,
"hasil": 0.001006036217303823,
"information": 0.001006036217303823,
"interface": 0.001006036217303823,
"it": 0.001006036217303823,
"jadi": 0.001006036217303823,
"javascript": 0.001006036217303823,
"json": 0.002012072434607646,
"kabupaten": 0.004024144869215292,
"kata": 0.001006036217303823,
"kecil": 0.001006036217303823,
"kota": 0.0030181086519114686,
"kunci": 0.001006036217303823,
"lapangankerja": 0.001006036217303823,
"mampu": 0.001006036217303823,
"mampumendukung": 0.001006036217303823,
"manfaat": 0.0030181086519114686,
"marketing": 0.01006036217303823,
"marketingbersama": 0.001006036217303823,
"marketingprovinsi": 0.001006036217303823,
"masa": 0.002012072434607646,
"method": 0.002012072434607646,
"mikro": 0.001006036217303823,
"milik": 0.002012072434607646,
"model": 0.0030181086519114686,

"ms": 0.002012072434607646,
"namundibutuhkan": 0.001006036217303823,
"nasional": 0.001006036217303823,
"notation": 0.001006036217303823,
"object": 0.001006036217303823,
"online": 0.002012072434607646,
"pasar": 0.0030181086519114686,
"penelitianmenunjukan": 0.001006036217303823,
"peran": 0.001006036217303823,
"post": 0.001006036217303823,
"programming": 0.001006036217303823,
"provinsi": 0.0030181086519114686,
"rambah": 0.001006036217303823,
"rata": 0.001006036217303823,
"ratawaktu": 0.001006036217303823,
"rekap": 0.001006036217303823,
"rest": 0.002012072434607646,
"salah": 0.001006036217303823,
"sedia": 0.001006036217303823,
"service": 0.005030181086519115,
"serviceoriented": 0.001006036217303823,
"soa": 0.002012072434607646,
"style": 0.001006036217303823,
"sulit": 0.001006036217303823,
"tahan": 0.001006036217303823,
"technology": 0.001006036217303823,
"tekan": 0.001006036217303823,
"teliti": 0.001006036217303823,
"tengah": 0.001006036217303823,
"terap": 0.002012072434607646,
"tiap": 0.001006036217303823,
"tiapkota": 0.001006036217303823,
"tren": 0.001006036217303823,
"tukar": 0.001006036217303823,
"uji": 0.001006036217303823,
"umkm": 0.005030181086519115,
"usaha": 0.001006036217303823,
"web": 0.005030181086519115,

```
        "webservice": 0.002012072434607646
}
```


2. Menghitung nilai IDF

Nilai IDF didapatkan dengan melakukan inverse pada nilai DF di mana nilai DF adalah jumlah dokumen yang memiliki kata tertentu dalam corpus. Semakin kecil nilai DF, semakin penting kata tersebut. Pada DF berkebalikan dengan TF, yang semakin besar berarti kata semakin penting, maka dari itu perlu dilakukan inverse pada nilai DF.

Inverse pada nilai DF dilakukan dengan mencari nilai log dari pembagian jumlah dokumen di corpus dengan nilai DF. Log pada hitungan diperlukan agar nilai IDF satu derajat dengan nilai TF. Pada corpus berisi 275 dokumen, nilai IDF tertinggi adalah log (275/1)=2,439332693830263 di mana nilai masih satu derajat dengan nilai TF. Contoh hasil perhitungan IDF terdapat pada Tabel 4.6.

Rumus : $IDF = log\left(\frac{N}{df}\right)$

Tabel 4.6 Hasil Perhitungan IDF


```
{
        "api": 0.00747027255729848,
        "application": 0.0013683616174875633,
        "bagai": 0.0003809203757310372,
        "bangun": 0.003014634260771626,
        "banyak": 0.0003847362600541148,
        "besar": 0.0003012611744758815,
        "biaya": 0.0026667793592020646,
        "data": 0.0004624098174801443,
        "e": 0.013333896796010321,
        "ekonomi": 0.0016038578006197241,
        "format": 0.0014063782783420903,
        "get": 0.0016712086956203413,
        "hasil": 1.4538286920719948e-05,
        "information": 0.0011675846004803158,
        "interface": 0.0013333896796010323,
        "it": 0.0009981636443541686,
        "jadi": 0.00032068285978311155,
```

```
        "json": 0.0032077156012394483,
        "kabupaten": 0.006182063206592834,
        "kata": 0.0001803433660414365,
        "kecil": 0.000968019556449296,
        "kota": 0.004219134835026271,
        "kunci": 0.0001803433660414365,
        "mampu": 0.0007448316795716739,
        "manfaat": 0.0024321858297304106,
        "masa": 0.0028127565566841805,
        "method": 0.0021682190298041647,
        "milik": 0.0007467743100283724,
        "model": 0.0020859496986641714,
        "nasional": 0.0014063782783420903,
        "online": 0.0025417332691641476,
        "pasar": 0.004482163534379088,
        "peran": 0.0014063782783420903,
        "programming": 0.001448020818742719,
        "provinsi": 0.005013626086861023,
        "rata": 0.0005735422274640709,
        "salah": 0.0003012611744758815,
        "sedia": 0.0009981636443541686,
        "service": 0.005956037166634591,
        "sulit": 0.0010305426014682543,
        "tahan": 0.001494054511459696,
        "tekan": 0.0016712086956203413,
        "teliti": 6.867689291602252e-05,
        "tengah": 0.001494054511459696,
        "terap": 0.0013016921616338206,
        "tiap": 0.0005286909873707706,
        "tukar": 0.0014063782783420903,
        "uji": 0.0002561783330107283,
        "usaha": 0.0010305426014682543,
        "web": 0.0051527130073412706
}
```

3. Menghitung TF-IDF

Perhitungan bobot setiap *term* dengan TF-IDF diperoleh dari perkalian TF dan nilai IDF. Hasil bobot setiap *term* pada satu dokumen dijumlahkan untuk mendapatkan nilai bobot setiap dokumen. Nilai TF-IDF ini yang akan digunakan sebagai bobot dari kata panghitungan *cluster*. Hasil perhitungan nilai TF-IDF dari data sampel ditunjukkan pada Tabel 4.7.

Tabel 4.7 Hasil Perhitungan TF-IDF


```
{
        "api": 0.00747027255729848,
        "application": 0.0013683616174875633,
        "bagai": 0.0003809203757310372,
        "bangun": 0.003014634260771626,
        "banyak": 0.0003847362600541148,
        "besar": 0.0003012611744758815,
        "biaya": 0.0026667793592020646,
        "data": 0.0004624098174801443,
        "e": 0.013333896796010321,
        "ekonomi": 0.0016038578006197241,
        "format": 0.0014063782783420903,
        "get": 0.0016712086956203413,
        "hasil": 1.4538286920719948e-05,
        "information": 0.0011675846004803158,
        "interface": 0.0013333896796010323,
        "it": 0.0009981636443541686,
        "jadi": 0.00032068285978311155,
        "json": 0.0032077156012394483,
        "kabupaten": 0.006182063206592834,
        "kata": 0.0001803433660414365,
        "kecil": 0.000968019556449296,
        "kota": 0.004219134835026271,
        "kunci": 0.0001803433660414365,
        "mampu": 0.0007448316795716739,
        "manfaat": 0.0024321858297304106,
        "masa": 0.0028127565566841805,
        "method": 0.0021682190298041647,
        "milik": 0.0007467743100283724,
        "model": 0.0020859496986641714,
```

```
        "nasional": 0.0014063782783420903,
        "online": 0.0025417332691641476,
        "pasar": 0.004482163534379088,
        "peran": 0.0014063782783420903,
        "programming": 0.001448020818742719,
        "provinsi": 0.005013626086861023,
        "rata": 0.0005735422274640709,
        "salah": 0.0003012611744758815,
        "sedia": 0.0009981636443541686,
        "service": 0.005956037166634591,
        "sulit": 0.0010305426014682543,
        "tahan": 0.001494054511459696,
        "tekan": 0.0016712086956203413,
        "teliti": 6.867689291602252e-05,
        "tengah": 0.001494054511459696,
        "terap": 0.0013016921616338206,
        "tiap": 0.0005286909873707706,
        "tukar": 0.0014063782783420903,
        "uji": 0.0002561783330107283,
        "usaha": 0.0010305426014682543,
        "web": 0.0015127130073412706
}
```


4.2.4 Clustering

Hierarchical clustering yang digunakan dalam penelitian ini adalah agglomerative clustering, di mana setiap dokumen dianggap satu *cluster,* kemudian digabungkan satu per satu berdasarkan jarak yang paling dekat hingga hanya tersisa satu *cluster*. Metode perhitungan jarak yang digunakan adalah metode *ward*. Metode *ward* menentukan jarak dengan menghitung *euclidean distance* dari dua dokumen, lalu akan dicari 2 *cluster* dengan jarak terdekat untuk dijadikan satu *cluster.*

Sebagai contoh ditunjukkan proses *clustering* 4 dokumen dengan 10 kata sebagai berikut :

D1 = (3.74; 2.91; 2.42; 9.02; 1.25; 1.45; 2; 1.3; 1.25; 4.06)

D2 = (0; 0; 0; 14.04; 2.03; 1.25; 2.1; 0; 1.25; 1.4)

D3 = (0; 2.12; 0; 0.17; 0.41; 3.17; 0.31; 1.36; 1.28; 0)

D4 = (0.54; 3.52; 0; 0.13; 5.14; 0.03; 3.51; 0.07; 2.3; 3.42)

Proses *hierarchical clustering* pada penelitian ini mengasumsikan setiap dokumen sebagai *cluster,* lalu menghitung jarak antar *cluster* dengan *euclidean distance*. Penghitungan jarak *cluster* dihitung dengan cara berikut :

$$d_{12} = \sqrt{(3.74-0)^2 + (2.91-0)^2 + (2.42-0)^2 + \cdots . + (4.06-1.4)^2}$$

= 7.9332

$$d_{13} = \sqrt{(3.74-0)^2 + (2.91-2.12)^2 + (2.42-0)^2 + \cdots . + (4.06-0)^2}$$

= 11.0362

$$d_{14} = \sqrt{(3.74-0.54)^2 + (2.91-3.52)^2 + (2.42-0)^2 + \cdots . + (4.06-3.42)^2}$$

= 10.8774

Jarak *cluster* dihitung untuk semua *cluster.* Sehingga diperoleh data jarak yang dapat dilihat pada Tabel 4.8 sebagai berikut:

Tabel 4.8 Jarak Antar *Cluster* Data Sampel (1)

C1	C2	Jarak
1	2	7.9332
1	3	11.0362
1	4	10.8774
2	3	14.4981
2	4	14.9835
3	4	7.177

Lalu jarak dua *cluster* dengan jarak terkecil akan digabungkan, pada table di atas jarak *cluster* 3 dan *cluster* 4 merupakan jarak paling kecil. Maka *cluster* 3 dan *cluster* 4 akan digabungkan. Setelah menggabungkan *cluster*, akan dilakukan pembaruan jarak gabungan *cluster* 3 dan *cluster* 4. Memperbarui jarak dapat dilakukan dengan cara sebagai berikut:

$$i_{(3,4)1} = \frac{1+1}{2+1}11.0362 + \frac{1+1}{2+1}10.8774 - \frac{1}{2+1} + 7.177 = 12.2167$$

$$i_{(3,4)2} = \frac{1+1}{2+1}14.4981 + \frac{1+1}{2+1}14.9835 - \frac{1}{2+1} + 7.177 = 17.2621$$

Setelah perbaruan jarak, jarak yang terbaru akan dimasukkan ke matriks jarak untuk kemudian dicari jarak yang terkecil lagi untuk pembentukan cluster berikutnya. Tabel jarak kemudian diperbarui dengan jarak yang baru dapat dilihat pada Tabel 4.9 berikut.

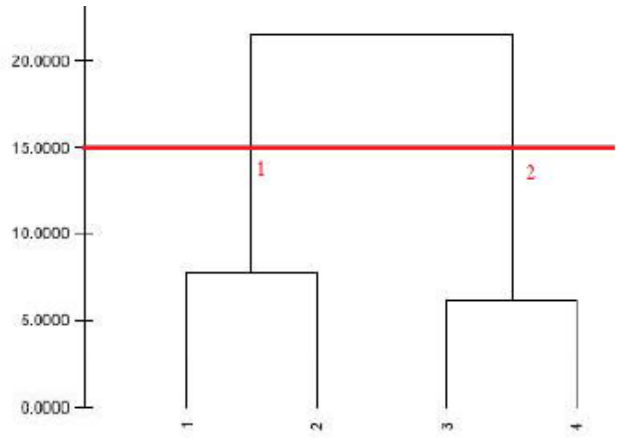
Tabel 4.9 Jarak Antar *Cluster* Data Sampel (2)

C1	C2	Jarak
1	2	7.9332
1	3,4	12.2167
2	3,4	17.2621

Jarak terkecil yang diperoleh adalah jarak antara *cluster* 1 dan 2, maka kedua *cluster* tersebut digabungkan. Setelah digabungkan kembali dilakukan pembaruan jarak

$$i_{(12)34} = \frac{1+1}{2+1}12.2167 + \frac{1+1}{2+1}17.2621 - \frac{2}{2+2}7.9332 = 15.6858$$

Setelah penggabungan *cluster* 1 dan 2, tersisa 2 cluster di matriks jarak, maka 2 cluster tersebut akan digabungkan dan algoritma *agglomerative hierarchical clustering* selesai karena jumlah *cluster* = 1. Hasil dari *hierarchical clustering* ini dapat digambarkan sebagai sebuah dendogram. Dendogram kemudian dipotong dengan menggunakan *threshold* tertentu sehingga terbentuk *cluster-cluster.* Gambar dendogram hasil *hierarchical clustering* dan *threshold* tertentu dapat dilihat pada Gambar 4.1.



Gambar 4.1 Dendogram Hasil Data Sampel

Untuk menggunakan hasil dari *hierarchical clustering* sebagai centroid awal metode *k-means*, perlu ditentukan *threshold* dari dendogram. Hal ini dapat dilakukan dengan memotong dendogram pada jarak tertentu. Jumlah *cluster* dari *k-means* juga ditentukan oleh jarak potongan dendogram.

4.2.5 K-Means Clustering

Setelah memperoleh dendogram dan dapat menentukan *threshold*, proses *clustering* dilanjutkan dengan *k-means clustering*. *K-means clustering* diawali dengan menentukan *cluster* dan pusat *cluster*. Pada penelitian ini menggunakan hasil *hierarchical clustering* pada penentuan jumlah *cluster* dan pusat *cluster*. Pemotongan dendogram data sampel menghasilkan dua *cluster* yaitu *cluster* 1 dan *cluster* 2. Pusat *cluster* 1 dan *cluster* 2 dapat diperoleh dengan menghitung rata-rata bobot kata pada *cluster*.

$C1 = (1.85; 1.455; 1.21; 11.53; 1.64; 1.35; 2.05; 0.65; 1.25; 2.73)$

$C2 = (0.27;\ 2.82;\ 0;\ 0.15;\ 2.775;\ 1.6;\ 1.91;\ 0.715;\ 1.79; 1.71)$

Setelah menentukan jumlah *cluster* dan pusat *cluster* awal ditentukan, proses *clustering* dengan *k-means* dilanjutkan menghitung *euclidean distance* dokumen ke pusat *cluster* 1 dan 2. Penghitungan jarak antara dokumen 1 dengan pusat *cluster* 1 dan 2 adalah sebagai berikut:

$$d_{11} = \sqrt{(3.74 - 1.85)^2 + (2.91 - 1.455)^2 + (2.42 - 1.21)^2 + \cdots . + (4.06 - 2.73)^2}$$

= 3.4983

$$d_{12} = \sqrt{(3.74 - 0.27)^2 + (2.91 - 2.82)^2 + (2.42 - 0)^2 + \cdots . + (4.06 - 1.71)^2}$$

= 10.2515

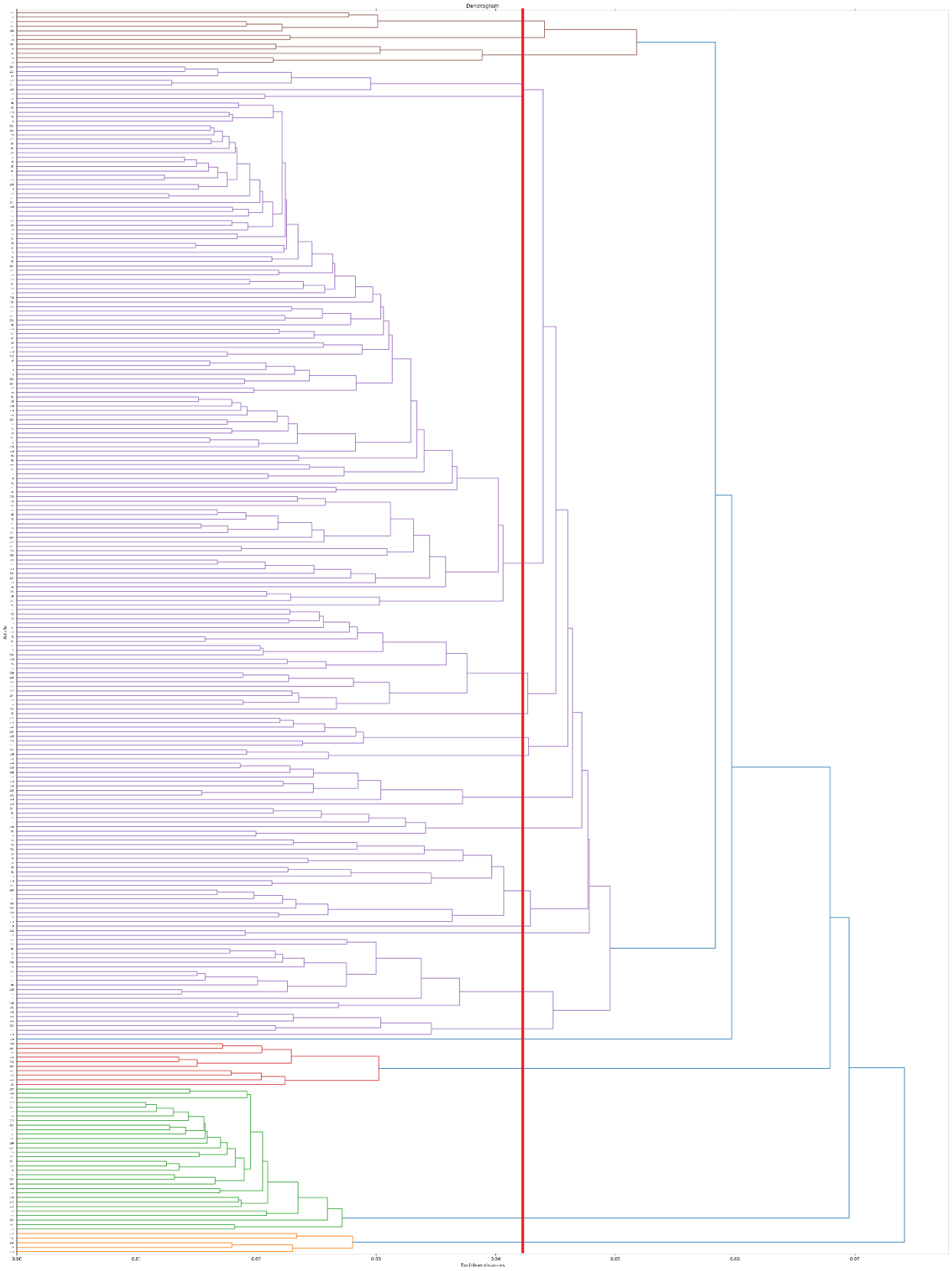
Perhitungan ini juga dilakukan untuk dokumen lain sehingga diperoleh hasil perhitungan *k-means clustering* yang ditunjukkan pada Tabel 4.10. Semua dokumen akan dihitung *euclidean distance* ke semua pusat *cluster,* kemudian dokumen akan dimasukkan ke *cluster* dengan jarak paling kecil dengan dokumen tersebut.

Tabel 4.10 Hasil *K-means Clustering* Data Sampel

Dokumen	C1	C2	Cluster
D1	3.4983	10.2515	1
D2	3.9572	14.2327	1
D3	12.2551	3.8446	2
D4	12.4677	3.8446	2

Pada Tabel 4.10 hasil menunjukkan dokumen 1 dan 2 cenderung masuk ke dalam *cluster* 1 dan dokumen 3 dan 4 cenderung masuk ke dalam *cluster* 2. Setelah semua dokumen dihitung *cluster*-nya, selanjutnya akan dicek ada atau tidak dokumen yang berpindah *cluster*. apabila ada dokumen yang berpindah *cluster*, maka pusat *cluster* harus diperbarui. Pembaruan ini dilakukan dengan menghitung kembali rata-rata bobot kata dari anggota *cluster* tersebut. Setelah jarak diperbarui, maka akan dilakukan kembali penghitungan jarak untuk setiap dokumen.

Pada Tabel 4.10 di atas menunjukkan bahwa dokumen 1 dan 2 cenderung masuk ke *cluster* 1 dan dokumen 3 dan 4 cenderung masuk ke *cluster* 2. Sehingga dapat disimpulkan bahwa tidak ada dokumen yang berpindah *cluster,* sehingga perhitungan *k-means* berhenti. Apabila algoritma *k-means* sudah berhenti, maka *cluster* yang terakhir terbentuk adalah hasil dari algoritma ini. Pada Tabel 4.10 di atas, tidak terjadi perpindahan dokumen, maka penghitungan dihentikan dan hasilnya adalah seperti Tabel 4.10. Hasil dari penelitian ini adalah hasil *clustering* dari *k-means.*



Gambar 4.2 Dendogram Hasil *Hierarchical Clustering*

Dendogram hasil data sampel dapat dilihat pada Gambar 4.2. Penelitian ini menggunakan 275 dokumen yang telah melalui proses *pre-processing, term-weighting, hierarchical clustering* dan *k-means clustering*. Garis merah pada dendogram tersebut adalah *threshold* pemotongan dendogram hasil *hierarchical clustering*. *Threshold* ini dipilih atas pertimbangan keterkaitan antar dokumen pada *cluster* hasil pemotongan dendogram. Ketika *threshold* dinaikkan, ada *cluster* yang mengakibatkan adanya dokumen-dokumen dengan tema berbeda masuk ke dalam satu *cluster.* Sedangkan, ketika *threshold* diturunkan, ada *cluster* dengan dokumen-dokumen yang memiliki tema yang sama terpisah. Oleh karena itu, *threshold* ini dipilih.

Hasil dari *hierarchical clustering* berupa dendogram ditunjukkan pada Gambar 4.2 di mana menunjukkan bahwa dengan threshold yang dipilih, akan terbentuk 16 *cluster*. *Cluster* yang dihasilkan oleh *hierarchical clustering* menjadi penentu jumlah *cluster* dan pusat *cluster* pada metode *k-means.* Setelah jumlah *cluster* diperoleh dan pusat *cluster* awal, proses dilanjutkan dengan *k-means clustering.* Hasil yang didapatkan dari metode *k-means* merupakan hasil akhir dari proses *clustering.*

4.3 Analisis Hasil *Clustering*

Pada proses *clustering* data menghasilkan 16 *cluster* yang terdiri dari beberapa dokumen setiap *cluster*-nya. Proses analisis yang pertama dilakukan dengan meneliti pola keanekaragaman tema berdasarkan judul dokumen di setiap *cluster.* Tabel 4.11 memperlihatkan hasil analisis tema pada setiap *cluster.*

Tabel 4.11 Analisis Tema Hasil *Clustering*

NO	TEMA CLUSTER	JUMLAH DOKUMEN SESUAI TEMA	JUMLAH DOKUMEN TIDAK SESUAI TEMA (TEMA SEHARUSNYA)	JUMLAH TOTAL DOKUMEN
1	Clustering	17	3 (1 Fuzzy, 1 RPL, 1 Klasifikasi)	20
2	RPL	8	2 (1 Kriptografi, 1 Pengolahan citra)	10
3	Filtering	7	2 (Similaritas)	9
4	Similaritas	6	1 (Ontologi)	7
5	JST	9	1 (SPK)	10
6	Naïve Bayes	14	12 (2 Similaritas, 1 RPL, 1 Probabilitas, 3 Klasifikasi, 1 Data Mining, 3 Filtering, 1 SPK)	26

7	Similaritas	4	1 (Naïve Bayes)	5
8	RPL	2	Tidak ada	2
9	SPK	1	Tidak ada	1
10	Ontologi	4	1 (RPL)	5
11	Fuzzy	5	30 (1 Pengolahan Citra, 3 KC, 11 RPL, 6 SPK, 1 Machine Learning, 1 Filltering, 1 Naïve Bayes, 2 Data Mining, 1 JST, 3 Clustering)	35
12	Pengolahan Citra	12	3 (1 Fuzzy, 1 SPK, 1 JST)	15
13	Similaritas	5	1 (Fuzzy)	6
14	RPL	5	Tidak ada	5
15	Naïve Bayes	2	Tidak ada	2
16	Tidak spesifik	0		117

Tabel 4.11 memperlihatkan bahwa pada hasil *clustering* terdapat beberapa *cluster* yang memuat dokumen dengan tema yang berbeda atau tidak sesuai dengan tema pada *cluster* tersebut. Penentuan tema *cluster* sudah spesifik atau belum ditentukan dengan persentase rumus berikut :

$$\text{persentase tema} = \frac{jumlah\ dokumen\ terbanyak\ pada\ suatu\ tema}{jumlah\ dokumen\ total\ cluster}\ x\ 100\%$$

Dokumen dianggap sudah spesifik apabila jumlah presentase tema untuk salah satu tema lebih dari atau sama dengan 80%. Sedangkan apabila jumlah persentase tema masih kurang dari 80% akan dianggap belum spesifik. Setiap *cluster* memiliki persentase tema yang berbeda-beda. Pada *cluster* 1 dengan jumlah 20 dokumen, terdapat 3 dokumen yang tidak sesuai dengan tema, sedangkan tema terbanyak sejumlah 17 dokumen. Hasil persentasi dengan tema *clustering* pada *cluster* 1 adalah 85% (lebih dari 80%) dan dokumen dengan tema yang berbeda ada 15%. Nilai perhitungan persentase tema *clustering* dianggap memiliki tema spesifik. *Cluster* yang dianggap memiliki tema spesifik adalah *cluster* 1 dengan hasil presentase tema 85%, *cluster* 2 dengan hasil presentase tema 80%, *cluster* 4 dengan hasil presentase tema 86%, *cluster* 5 dengan hasil presentase tema 90%, *cluster* 7 dengan hasil presentase tema 80%, *cluster* 8 dengan hasil presentase tema 100%, *cluster* 9 dengan hasil presentase tema 100%, *cluster* 10 dengan hasil presentase tema 80%, *cluster* 12 dengan hasil presentase tema 80%, *cluster* 13 dengan hasil

presentase tema 83%, *cluster* 14 dengan hasil presentase tema 100%, *cluster* 15 dengan hasil presentase tema 100%.

Hasil perhitungan persentase tema pada *cluster* 3 diperoleh nilai 78%, *cluster* 6 diperoleh nilai 54%, dan *cluster* 11 diperoleh nilai 14%. Nilai persentase tema tersebut kurang dari batas minimal sehingga *cluster* dianggap belum spesifik. Hasil penelitian memuat 3 *cluster* dengan tema yang belum spesifik yaitu *cluster* 3, 6 dan 11. Hal ini kemungkinan disebabkan karena diprosesnya semua kata pada abstrak. Data pada abstrak hanya sebagian gambaran kecil dari sebuah penelitian dan kata-kata yang berada pada abstrak kurang dapat merepresentasikan atau menjelaskan penelitian.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan di atas dapat disimpulkan bahwa, penelitian *clustering* dokumen dengan menggunakan metode *hierarchical clustering* dan *k-means clustering* ini menghasilkan 16 *cluster.* Pada 16 *cluster* tersebut ada 11 *cluster* yang memiliki persentase tema lebih dari 80% yang dianggap tema spesifik, 3 *cluster* yang memiliki persentase tema kurang dari 80% dianggap tema belum spesifik. Selain itu terdapat 1 *cluster* tidak dapat dimasukkan ke dalam satu tema karena ada dokumen sejumlah 117 dengan tema yang berbeda-beda. Hal ini disebabkan karena Hal ini kemungkinan disebabkan karena diprosesnya semua kata pada abstrak. Data pada abstrak hanya sebagian gambaran kecil dari sebuah penelitian dan kata-kata yang berada pada abstrak kurang dapat merepresentasikan atau menjelaskan penelitian.

5.2 Saran

Dalam mengembangkan lebih lanjut penelitian ini agar hasil *clustering* lebih baik disarankan untuk menggunakan laporan tugas akhir pada bagian Bab 2 atau Bab 3 yang dapat merepresentasikan keseluruhan dari isi laporan. Selain itu, pemilihan kata sebaiknya lebih terbatas pada kata-kata kunci yang signifikan pada penentuan jenis penelitian yang dilakukan.

DAFTAR PUSTAKA

Andrea Tri Rian Dani dkk. 2019. "Penerapan Hierarchical Clustering Metode Agglomerative pada Data Runtun Waktu". Jambura Journal of Mathematics Volume 1 Nomor 2 Juli 2019

Aris Tri Jaka. 2015. "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining". Jurnal Informatika UPGRIS Volume 1 Edisi Juni 2015

Februariyanti, Herny. 2017. Hierarchical Agglomerative Clustering Untuk Pengelompokan Skripsi Mahasiswa. Prosiding SINTAK 2017.

G. Salton. 1989. *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Pennsylvania : Addison Wesley

Igg Adiwijaya. 2006. "Text Mining dan Knowledge Discovery". Kolokium Bersama Komunitas Datamining Indonesia & Soft-Computing Indonesia September 2006

Indraloka, D. Smaradahana. 2017. Penerapan *Text Mining* untuk Melakukan *Clustering* Data *Tweet* Shopee Indonesia. Jurnal Sains dan Seni ITS Vol. 6, No. 2 (2017) 2337-3520 (2301-928X Print).

Musfiroh Nurjannah dkk. 2013. "Penerapan Algoritma Term Frequency-Inverse Document Frequency (TD-IDF) untuk Text Mining". Jurnal Informatika Mulawarman Vol. 8 No. 3 September 2013

Oliveira and Pedrycz. 2007. Advances in Fuzzy Clustering and Its Application. John Wiley and Sons, Ltd

Rahmawati, Lynda. 2015. Analisa Clustering Menggunakan Metode K-Means dan Hierarchical Clustering (Studi Kasus : Dokumen Skripsi Jurusan Kimia, FMIPA, Universitas Sebelas Maret).

Sri Andayani. 2007. "Pembentukan Cluster dalam Knowledge Discovery in Database dengan Algoritma K-Means". Article of Lumbung Pustaka Universitas Negeri Yogyakarta

Wicaksana, A. Danang. 2018. Clustering Dokumen Skripsi Dengan Menggunakan HIerarchical Agglomerative Clustering. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 12 Desember 2018,hlm.6227-6234.

Widodo. 2017. Implementasi Algoritma K-Means Clustering Untuk Mengetahui Bidang Skripsi Mahasiswa Multimedia Pendidikan Teknik Informatika Dan Komputer Universitas Negeri Jakarta. Jurnal Pinter Vol.1 No.2 Desember 2017.

Yudhi Agusta. 2007. "K-Means : Penerapan, Permasalahan dan Metode Terkait". *Jurnal Sistem dan Informatika* Vol. 3 (Pebruari 2007)