ANALISIS KLASTERING LIRIK LAGU INDONESIA

Afdilah Marjuki[1], Herny Februariyanti[2]
[1,2] Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Stikubank
e-mail: [1]bodongben@gmail.com, [2]hernyfeb@edu.unisbank.ac.id

ABSTRAK

Sebuah lirik lagu dapat dengan mudah dikategorikan secara manual oleh manusia, tetapi jika dilakukan secara terkomputerisasi akan membawa permasalahan tersendiri. Untuk itu dibuat sebuah sistem yang mampu membuat klastering lirik lagu berbahasa Indonesia. Sistem dibuat menggunakan bahasa pemrograman R.

Data yang digunakan adalah 254 lirik lagu berbahasa indonesia yang didapatkan dari internet. Sebagai sample digunakan 17 data lirik lagu agar proses analisis dapat dilakukan dengan baik. Sistem akan melakukan data cleaning sebelum dilakukan proses pengklasteran agar data yang diolah berisi term yang dapat mewakili isi lirik lagu. Lirik lagu yang memiliki tingkat similaritas yang sama akan menjadi satu anggota klaster yang sama. Sistem mampu menampilkan hasil klastering dengan menggunakan metode hierarchical clustering dan K-means clustering.

***Kata Kunci**: Clustering, Hierarchical Clustering, K-Means Clustering*

1. PENDAHULUAN

Sebuah lirik lagu dapat dengan mudah dikategorikan secara manual oleh manusia, tetapi jika dilakukan secara terkomputerisasi akan membawa permasalahan tersendiri. Begitu pula dengan mencari tingkat kemiripan atau similaritas suatu lirik lagu dengan lirik lainnya, manusia dapat dengan mudah menentukan apakah lirik memilki tingkat kemiripan atau similaritas dengan lirik lainnya atau tidak.

Dalam penelitian ini teknik yang digunakan untuk memecahkan masalah diatas adalah dengan menggunakan teknik *text mining* untuk pengkategorian lirik lagu indonesia. Sedangkan untuk mencari nilai similaritas suatu lirik lagu dengan lirik lainnya menggunakan kata kunci yang didapat dari hasil query. Penulis akan menganalisis tingkat similaritas atau kemiripan suatu lirik lagu indonesia dengan menggunakan pembobotan *Term Frequency*.

Term Frequency merupakan salah satu metode untuk menghitung bobot setiap term dalam teks. Klastering (*clustering*) didefinisikan sebagai upaya mengelompokkan data ke dalam klaster sedemikian sehingga data-data di dalam klaster yang sama, lebih memiliki kesamaan dibandingkan dengan data-data pada klaster yang berbeda.

Dengan penelitian ini diharapkan proses pengkategorian lirik lagu secara terkomputerisasi hasilnya dapat sesuai dengan pengkategorian secara manual. Dan pengukuran tingkat similaritas lirik dapat menunjukan seberapa besar nilai similaritas. Berdasarkan uraian di atas, penulis tertarik untuk melakukan penelitian dengan judul "Analisis Klastering Lirik Lagu Indonesia".

2. TINJAUAN PUSTAKA

Penelitian sebelumnya yang pernah dilakukan berkaitan dengan masalah yang dihadapi penulis yaitu dengan judul "Algoritma Graph untuk Klasifikasi Perundang-undangan". Penelitian dilakukan oleh Februariyanti H dan Zuliarso E pada tahun 2015.[4]. Pembahasan meliputi pengkategorian dokumen perundang-undangan dengan algoritma graph, sistem yang dibuat terdiri dari 2 (dua) bagian. Bagian pertama adalah bagian untuk mengekstrak file teks dan memasukkan bagian dasar hukum ke sistem. Sedang bagian kedua adalah bagian untuk melakukan klasifikasi dan memvisualisasi perundangan yang telah terklasifikasi.

Pada penelitian ini sistem mampu melakukan klasifikasi dan memvisualisasikan perundangan-undangan dengan dasar hukum yang sama. Pada penelitian ini teori graph digunakan untuk menampilkan visualisasi dokumen perundang-undangan. Teori graph sebagai alat bantu untuk merumuskan masalah-masalah yang ada serta mendefinisikan struktur hubungan antara perundang-undangan yang satu dengan yang lain menggunakan dasar hukum yang sama.

Penelitian lainnya adalah "Implementasi *Cosine Similarity* dan Algoritma *Smith-Waterman* untuk Mendeteksi Kemiripan Teks". Penelitian dilakukan oleh Imbar, Radiant Victor, dkk pada tahun 2014.[6]. Pembahasan meliputi pengembangan sebuah aplikasi yang mengimplementasikan *cosine similarity* dan algoritma Smith-Waterman untuk mendeteksi kemiripan teks.

Aplikasi yang dibangun dapat mendeteksi tingkat kemiripan teks dari sangat mirip hingga sangat tidak mirip berdasarkan kemunculan kata di dalamnya dengan menggunakan *cosine similarity*. Aplikasi juga dapat mendeteksi tingkat kemiripan teks dari sangat mirip hingga sangat tidak mirip berdasarkan urutan kata pembentuknya dengan menggunakan algoritma Smith-Waterman.

Penelitian lainnya adalah " Klastering Dokumen Menggunakan Hierarchical Agglomerative Clustering ". Penelitian dilakukan oleh Februariyanti, H dan Winarko, E pada tahun 2010.[5]. Pada penelitian ini membahas bagaimana sistem temu kembali informasi dapat memberikan informasi yang relevan karena semakin banyaknya informasi yang tersedia. Untuk mengetahui kemiripan antar dokumen menggunakan pembobotan term frequency

dan cosine similaritas. Sistem yang dibangun dapat menampilkan dokumen apa saja yang mempunyai kedekatan similaritas dari query yang diinputkan user.

Dalam penelitian ini, sistem dapat menampilkan dokumen yang membahas tentang topik yang sama, cenderung akan mengelompok menjadi satu kluster. Dengan adanya klaster, akan membantu menemukan dokumen yang berada dalam satu klaster dengan query yang diinputkan user. Pada penelitian ini dilakukan pengujian *recall* dan *precision*. Dari hasil pengujian mampu menghasilkan nilai rata-rata *recall*= 0.6 dan nilai rata-rata *precision*= 0.5 serta nilai *F-measure*= 0.5.

Dalam penelitian ini penulis mempunyai persamaan dengan penelitian yang sudah dijelaskan diatas, diantaranya adalah *text mining*. Perbedaan yang dilakukan yaitu penulis menggunakan pembobotan *term frequency* dan analisis data menggunakan aplikasi R Studio.

3. METODE PENELITIAN

3.1 K-Means Clustering

K-Means merupakan salah satu metode data clustering non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih cluster/kelompok. Metode ini mempartisi data ke dalam cluster/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain. Adapun tujuan dari data clustering ini adalah untuk meminimalisasikan *objective function* yang diset dalam proses clustering, yang pada umumnya berusaha meminimalisasikan variasi di dalam suatu cluster dan memaksimalisasikan variasi antar cluster [1].

Data clustering menggunakan metode *K-Means* ini secara umum dapat dilakukan dengan algoritma dasar sebagai berikut [1]:

1. Tentukan jumlah cluster
2. Alokasikan data ke dalam cluster secara random
3. Hitung *centroid*/rata-rata dari data yang ada di masing-masing cluster
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat
5. Kembali ke Step 3, apabila masih ada data yang berpindah cluster atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan.

3.2 Hierarchical Clustering

Pada algoritma *clustering*, *data* akan dikelompokkan menjadi *cluster-cluster* berdasarkan kemiripan satu *data* dengan yang lain. Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu *cluster* dan meminimumkan kesamaan antar anggota *cluster* yang berbeda. Kategori algoritma *clustering* yang banyak dikenal adalah *Hierarchical Clustering*. *Hierarchical Clustering* adalah salah satu algoritma *clustering* yang dapat digunakan untuk meng-*cluster* dokumen *(document clustering*). [2].

Dari teknik *hierarchical clustering*, dapat dihasilkan suatu kumpulan partisi yang berurutan, dimana dalam kumpulan tersebut terdapat:

a. *Cluster – cluster* yang mempunyai poin – poin individu. *Cluster – cluster* ini berada di level yang paling bawah.

b. Sebuah *cluster* yang didalamnya terdapat poin – poin yang dipunyai semua *cluster* didalamnya. *Single cluster* ini berada di *level* yang paling atas.

Hasil keseluruhan dari algoritma *hierarchical clustering* secara grafik dapat digambarkan sebagai *tree*, yang disebut dengan *dendogram*. *Tree* ini secara grafik menggambarkan proses penggabungan dari *cluster – cluster* yang ada, sehingga menghasilkan *cluster* dengan level yang lebih tinggi. [2].

3.3 Agglomerative Hierarchical Clustering

Metode ini menggunakan strategi disain *Bottom-Up* yang dimulai dengan meletakkan setiap obyek sebagai sebuah *cluster* tersendiri (*atomic cluster*) dan selanjutnya menggabungkan *atomic cluster – atomic cluster* tersebut menjadi *cluster* yang lebih besar dan lebih besar lagi sampai akhirnya semua obyek menyatu dalam sebuah *cluster* atau proses dapat pula berhenti jika telah mencapai batasan kondisi tertentu. Adapun ukuran jarak yang digunakan untuk menggabungkan dua buah obyek cluster adalah *Minimum Distance,* yang dapat dilihat pada persamaan (1). [2].

$$d_{\min}(C_i, C_j) = \min_{p \in Ci,\, p' \in Cj} |p - p'| \quad (1)$$

Dimana |p-p'| jarak dua buah obyek p dan p'

3.4 Pemrograman R

Bahasa R merupakan versi sumber terbuka (*open-source*) dari bahasa pemrograman S. Versi komersial yang berbasis bahasa S adalah S plus. Bahasa R memiliki kemampuan yang tidak kalah dangan paket-paket program pengolahan data komersial bahkan dalam beberapa hal kemampuannya lebih baik. Bahasa R mendapat sambutan yang baik dari kalangan statistikawan di seluruh dunia, sayangnya di Indonesia belum banyak dikenal. [3]. R tersedia untuk berbagai sistem operasi selain Windows, seperti Macintosh, Linux dan UNIX, memiliki

kemampuan membuat grafik yang canggih. Sintaksnya mudah dipelajari dengan banyak fungsi-fungsi statistik yang terpasang. R dapat dengan mudah diperluas dengan menciptakan fungsi-fungsi buatan pengguna sendiri disamping juga tersedia banyak program-program kecil sebagai tambahan (*add in*), yang disebut *package* yang juga dapat diunduh gratis [3].

R merupakan bahasa pemrograman komputer, sehingga bagi pemrogram menjadi lebih akrab, sedangkan bagi pemakai awal akan merupakan langkah yang mudah untuk memulai sebagai pemrogram komputer. Bagi sebagian pengguna yang terbiasa mengguna GUI dengan sistem menu, R juga menyediakan banyak GUI yang berbasiskan sistem menu, antara lain R Studio, Tinn-R, R Commander dan bayak lagi yang lainnya, dan dapat diunduh gratis juga. GUI standar R diperlihatkan oleh gambar di atas yang menyertai tulisan ini. Walaupun R adalah gratis kemampuannya tidak kalah dengan program-program paket statistik yang komersial, bahkan dalam hal tertentu adalah lebih baik. Penggunaan R tidak dibatasi, bahkan dapat digunakan untuk tujuan-tujuan komersial [3].

4. HASIL DAN PEMBAHASAN

4.1 Data Cleaning

Sebelum melakukan analisa untuk pembentukan klaster pada lirik lagu, dilakukan *data cleaning* agar konten yang terdapat pada lirik lagu tidak mengandung kata-kata yang tidak perlu. Pertama yang dilakukan adalah dengan menghapus tanda baca, tanda baca seharusnya tidak ada dalam lirik lagu karena dalam proses analisis lirik lagu tidak bisa untuk mewakili isi dari lirik lagu tersebut. Untuk itu tanda baca yang terkandung dalam lirik lagu dihapus. Kedua, menghapus nomor yang terkandung dalam lirik lagu. Seperti halnya tanda baca, nomor juga harus dilakukan proses penghapusan karena tidak digunakan dalam proses analisis.

Tahap ketiga adalah dengan mengubah konten lirik lagu menjadi huruf kecil agar proses analisa lebih maksimal karena konten lirik lagu sama rata menggunakan huruf kecil. Berikut ini adalah script untuk melakukan *data cleaning*.

```
#Menghapus tanda baca
sotu_corpus <- tm_map(sotu_corpus, removePunctuation)
inspect(sotu_corpus[1])

#Mengahapus Nomor
sotu_corpus <- tm_map(sotu_corpus, removeNumbers)
inspect(sotu_corpus[1])

#Mengubah ke huruf kecil
sotu_corpus <- tm_map(sotu_corpus, tolower)
inspect(sotu_corpus[1])
```

Gambar 1. Proses data cleaning

4.2 Menghapus Stopword

Langkah pertama yang dilakukan dalam menghapus kata tidak penting atau stopword adalah mengimport data stopword yang sudah ada dalam bentuk file .CSV kedalam R studio menjadi sebuah data frame. Setelah data frame stopword berhasil dibuat, kemudian melakukan proses penghapusan stopword yang terkandung dalam lirik lagu.

Setelah dilakukan penghapusan stopword, kemudian dilakukan proses penghapusan spasi agar konten lirik lagu setelah dilakukan penghapusan stopword bisa tersusun rapi dan tidak mengandung spasi dobel. Dengan dilakukannya serangkaian proses data cleaning, diharapkan dalam proses analisis menjadi lebih maksimal karena konten lirik lagu berisi hanya kata-kata penting yang mewakili isi dari lagu tersebut. Berikut ini adalah script untuk melakukan proses penghapusan *stopword* dan spasi.

```
#Menyiapkan stoplist
stopwords <- read.csv("C:/stopwordID.csv", header = FALSE)
stopwords <- as.character(stopwords$V1)
stopwords <- c(stopwords, stopwords())

#Menghapus stopword
sotu_corpus <- tm_map(sotu_corpus, removeWords, c(stopwords("english"),stopwords))
inspect(sotu_corpus[1])

#Menghapus spasi
sotu_corpus <- tm_map(sotu_corpus, stripWhitespace)
inspect(sotu_corpus[1])
```

Gambar 2. Menghapus stopword dan spasi

Berikut ini akan disajikan lirik lagu sebelum melewati proses *data cleaning* dan lirik lagu setelah dilakukan proses *data cleaning*. Sebagai contoh adalah lirik lagu dengan no 1 pada data frame. Yang pertama adalah lirik lagu sebelum dilakukan proses *data cleaning*.

```
> inspect(sotu_corpus[1])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 1

[1] Dari hatiku terdalam  Sungguh aku cinta padamu  Cintaku buka
nlah cinta biasa  Jika kamu yang memiliki  Dan kamu yang temanik
u seumur hidupku  Terimalah pengakuanku  Percayalah kepadaku  Se
mua ini ku lakukan  Karena kamu memang untukku  Cintaku bukanlah
 cinta biasa  Jika kamu yang memiliki  Dan kamu yang temaniku  S
eumur hidupku  Cintaku bukanlah cinta biasa  Jika kamu yang memi
liki  Dan kamu yang temaniku seumur hidupku  Cintaku bukan cinta
 biasa  Jika kamu yang menemani  Dan kamu yang temaniku seumur h
idupku  Terimalah pengakuanku
```

Gambar 3. Lirik lagu sebelum data cleaning

Berikutnya adalah lirik lagu setelah dilakukan proses *data cleaning*. Perbedaan yang terjadi sangat terlihat setelah dilakukan proses data cleaning agar dalam pembuatan klaster lebih fokus pada konten lirik lagu.

```
> inspect(sotu_corpus[1])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 1

[1]  hatiku terdalam sungguh cinta padamu cintaku cinta memiliki
 temaniku seumur hidupku terimalah pengakuanku percayalah kepada
ku ku lakukan untukku cintaku cinta memiliki temaniku seumur hid
upku cintaku cinta memiliki temaniku seumur hidupku cintaku cint
a menemani temaniku seumur hidupku terimalah pengakuanku
```

Gambar 4. Lirik lagu setelah data cleaning

4.3 Hierarchical Clustering

Langkah pertama dalam pemodelan adalah membuat Document Term Matrix, dengan dokumen sebagai baris, kata-kata individual di sepanjang kolom, dan hitungan frekuensi sebagai konten. Setelah dilakukan proses *Document Term Matrix*, langkah berikutnya adalah membuat Term Frequency - Inverse Document Frequency untuk mendapatkan bobot antar lirik lagu.

Setelah frekuensi dihitung, semua dokumen diproses pencarian jarak untuk pengelompokan dokumen yang dihitung sebagai kemiripan kosinus. Tujuan pengelompokan adalah untuk mempelajari tentang kelompok dokumen yang terbentuk menjadi sebuah klaster.

```
doc_term <- DocumentTermMatrix(sotu_corpus)
doc_term$dimnames$Docs <- sotu$file_name

tf_idf <- weightTfIdf(m = doc_term, normalize = TRUE)
tf_idf_mat <- as.matrix(tf_idf)

tf_idf_dist <- dist(tf_idf_mat, method = 'euclidian')

clust_h <- hclust(d = tf_idf_dist, method = 'complete')
plot(clust_h,
     main = 'Cluster Dendrogram: complete euclidian Distance',
     xlab = '', ylab = '', sub = '')
```

Gambar 5. Script hierarchical clustering

4.4 K-Means Clustering

Untuk mengkonfirmasi analisis cluster hirarkis, analisis cluster K-Means dihitung yang akan memberikan representasi visual yang lebih baik dari ruang cluster. Karena K-Means bergantung pada Euclidean Distance daripada Cosine Dissimilarity, pertama yang perlu dilakukan adalah dengan menormalkan matriks TF-IDF. Proses K-Means sendiri akan berkelompok selama 5 centroid, dan meningkatkan iterasi maksimal dari standar 10 sampai 25.

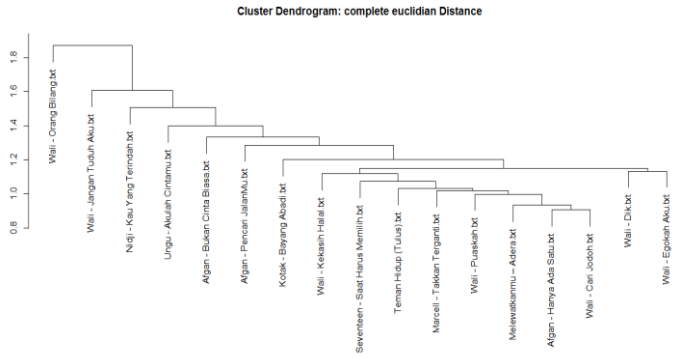
Data berisi ribuan dimensi, dimensi untuk setiap istilah. Meskipun klaster dibangun menggunakan ruang fitur dimensi penuh, tidak akan praktis untuk memvisualisasikan banyak dimensi ini. Untuk membuat visualisasi lebih sesuai, Analisis Komponen Utama dilakukan dan 2 komponen terpenting dipetakan ke plot beserta meta-data untuk tujuan markup.

```
tf_idf_norm <- tf_idf_mat / apply(tf_idf_mat, MARGIN = 1, FUN = function(x) sum(x^2)^0.5)
km_clust <- kmeans(x = tf_idf_norm, centers = 5, iter.max = 25)

pca_comp <- prcomp(tf_idf_norm)
pca_rep <- data_frame(sotu_name = sotu$file_name,
                      pc1 = pca_comp$x[,1],
                      pc2 = pca_comp$x[,2],
                      clust_id = as.factor(km_clust$cluster))

ggplot(data = pca_rep, mapping = aes(x = pc1, y = pc2, color = clust_id)) +
  scale_color_brewer(palette = 'Set1') +
  geom_text(mapping = aes(label = sotu_name), size = 2.5, fontface = 'bold') +
  labs(title = 'K-Means Cluster: 5 clusters on PCA Features',
       x = 'Principal Component Analysis: Factor 1',
       y = 'Principal Component Analysis: Factor 2') +
  theme_grey() +
  theme(legend.position = 'right',
        legend.title = element_blank())
```

Gambar 6. Script K-Means Clustering

4.5 Pembentukan Klaster Hierarchical Clustering

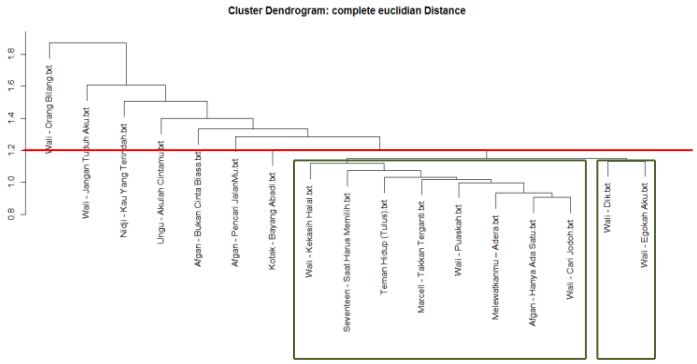
Untuk mempermudah dalam pembacaan diagram yang terbentuk, dalam melakukan pembentukan klaster dilakukan menggunakan sampel berupa 17 lirik lagu. Berikut ini adalah gambar klaster dendrogram yang terbentuk dengan menggunakan method complete.



Gambar 7. Klaster Dendrogram Method Complete

Berdasarkan gambar 7 dilakukan pembuatan klaster dengan *hierarchical clustering* menggunakan metode *complete*, dapat dilihat bahwa klaster dendrogram yang terbentuk menggambarkan lirik lagu yang berada dalam satu klaster memiliki tingkat similaritas yang sama dengan lirik lagu yang berada dalam satu klaster. Sebagai contoh adalah lirik lagu melewatkanmu-adera berada dalam satu klaster dengan lirik lagu afgan-hanya ada satu dan wali-cari jodoh.

Klaster dendrogram yang terbentuk pada gambar 7 adalah klaster yang menggambarkan keseluruhan lirik lagu, sehingga akan sulit dalam menganalisa lirik lagu mana saja yang berada dalam satu klaster maupun berapa jumlah klaster yang terbentuk. Untuk itu dilakukan pemotongan pada titik tertentu agar lebih mudah dalam menganalisa, sebagai contoh pada gambar dibawah ini dilakukan pemotongan pada titik 1.2.

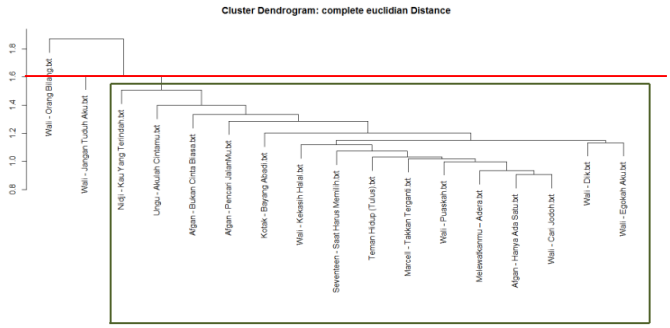


Gambar 8. Dendrogram pada titik 1.2

Pada gambar 8 dapat dilihat klaster dendrogram yang terbentuk dilakukan pemotongan pada titik 1.2, sehingga pada titik tersebut menghasilkan 2 buah klaster.

a. Klaster yang pertama adalah lirik lagu dengan judul wali-kekasih halal, seventeen-saat harus memilih, tulus-teman hidup, marcell-takkan terganti, wali-puaskah, adera-melewatkanmu, afgan-hanya ada satu dan yang terakhir wali-cari jodoh.
b. Klaster yang kedua adalah lirik lagu dengan judul wali-dik dan wali-egokah aku.

Dari hasil klaster yang terbentuk, pengguna dapat mengetahui lirik lagu mana saja yang berada dalam satu klaster, sehingga dapat disimpulkan bahwa lirik lagu yang berada dalam satu klaster adalah yang memiliki nilai similaritas yang sama. Selanjutnya dilakukan percobaan kedua dengan menggunakan titik potong 1.6, yang dapat dilihat pada gambar dibawah ini.



Gambar 9. Dendrogram pada titik 1.6

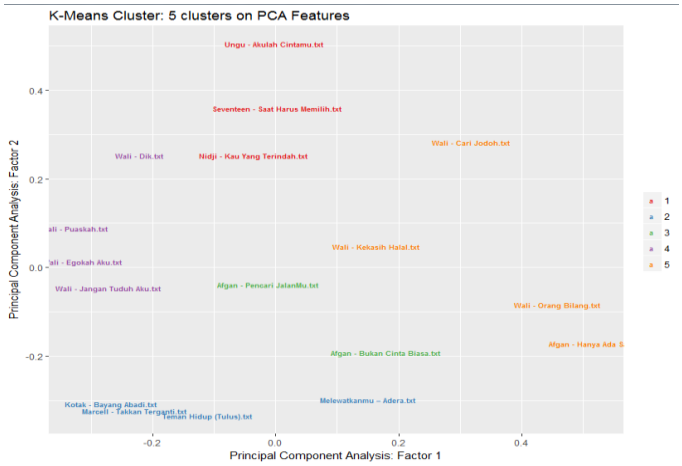
Pada gambar 9 dapat dilihat klaster yang terbentuk dengan titik potong 1.6 menghasilkan satu klaster dengan banyak anggota lirik lagu. Dari total keseluruhan lirik lagu yang diuji, terdapat 15 lirik lagu yang berada dalam satu klaster dan 2 lagu yang bukan anggota klaster yaitu lirik lagu dengan judul wali-orang bilang dan wali-jangan tuduh aku.

4.6 K-Means Clustering

K-Means clustering yang terbentuk menggambarkan lirik lagu mana saja yang termasuk kedalam satu klaster. Dapat dilihat pada gambar 10, klaster yang terbentuk berjumlah 5 klaster. Lirik lagu yang menjadi anggota satu klaster ditandai dengan warna yang sama. Dalam pembacaannya, K-means clustering adalah yang paling mudah untuk mengidentifikasi lirik lagu apa saja yang menjadi satu klaster. Dibandingkan dengan grafik-grafik yang dibuat sebelumnya, K-means clustering dapat menampilkan lirik lagu yang berbeda klaster dengan membedakannya dengan warna yang berbeda.

Dari grafik K-Means yang terbentuk dapat disimpulkan lirik lagu mana saja yang berada dalam satu klaster, berikut ini adalah klaster-klaster yang terbentuk :

a. Klaster yang pertama adalah ditandai dengan warna ungu, anggota dari klaster pertama adalah lirik lagu dengan judul wali-dik, wali-puaskah, wali-egokah aku, dan wali-jangan tuduh aku.
b. Klaster yang kedua ditandai dengan warna biru, anggota dari klaster kedua ini adalah kotak-bayang abadi, marcell-takkan terganti, tulus-teman hidup dan adera-melewatkanmu.
c. Klaster yang ketiga ditandai dengan warna merah, anggota klaster ketiga memiliki 3 anggota lirik lagu yaitu ungu-akulah cintamu, seventeen-saat harus memilih, dan nidji-kau yang terindah.
d. Klaster yang keempat ditandai dengan warna hijau yaitu afgan-pencari jalanMu dan afgan-bukan cinta biasa.
e. Klaster yang terakhir ditandai dengan warna orange, anggota klaster yang terakhir yaitu wali-cari jodoh, wali-kekasih halal, wali-orang bilang, dan afgan-hanya ada satu.



Gambar 10. K-Means clustering

5. KESIMPULAN

Dari penjelasan pada bab-bab sebelumnya, penulis dapat menarik beberapa kesimpulan yaitu sebagai berikut:

a. Sistem yang dibuat berhasil menerapkan *data cleaning* pada teks lirik lagu berbahasa Indonesia.
b. Pembobotan kata dilakukan dengan algoritma TF-IDF (Term Frequency dan Invers Document Frequency).
c. Klastering digunakan menggunakan metode *hierarchical clustering*.
d. Sistem dapat menampilkan lirik lagu berbahasa Indonesia hasil klastering dengan *hierarchical clustering* dan *K-Means Clustering*.
e. Lirik lagu yang memiliki tingkat similaritas yang sama akan berada dalam satu klaster.

6. SARAN

Dari kesimpulan yang diambil, saran yang dapat penulis berikan untuk perbaikan sistem ini adalah sebagai berikut:

a. Sistem belum bisa menerapkan stemming nazief-adriani dalam mencari kata dasar teks berbahasa Indonesia dengan bahasa pemrograman R.
b. Diharapkan dengan adanya penelitian ini, peneliti-peneliti selanjutnya dapat membuat sistem klastering dengan bahasa pemrograman lain.

DAFTAR PUSTAKA

[1] Agusta, Y., 2007, *K-Means – Penerapan, Permasalahan dan Metode Terkait*, Jurnal Sistem dan Informatika, 3, 47-60.

[2] Budhi, Gregorius S., dkk., 2008 , *HIERARCHICAL CLUSTERING* UNTUK APLIKASI *AUTOMATED TEXT INTEGRATION*, Seminar Nasional Aplikasi Teknologi Informasi, C27-C32

[3] Fatima, S., 2014, Bahasa Pemrograman R, [Online], Tersedia: https://soniafatima.wordpress.com/2014/12/19/bahasa-pemrograman-r/. [7 Desember 2016]

[4] Februariyanti, H., Zuliarso, E. 2015, *Algoritma Graph Untuk Klasifikasi Perundang-Undangan,* Dinamika Informatika, 7, (1), 17-25

[5] Februariyanti, H., Winarko, E., 2010, *KLASTERING DOKUMEN MENGGUNAKAN HIERARCHICAL AGGLOMERATIVE CLUSTERING*, [Online], Tersedia: https://www.unisbank.ac.id/ojs/index.php/eksternalfti/article/download/1687/573. [28 Juli 2017]

[6] Imbar, R.V., dkk., 2014, *Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks*, Jurnal Informatika, 10, (1), 31-42