ISSN 2085-4579

K-Means Clustering Video *Trending* di Youtube Amerika Serikat

Mencari Pola dan Pengelompokkan Video-video *Trending*

Kevin Widjaja[1], Raymond Sunardi Oetama[2]
[1,2] Fakultas Teknologi dan Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1] kevin.widjaja@student.umn.ac.id
[2] raymond@umn.ac.id

Diterima 05 Maret 2020
Disetujui 10 November 2020

***Abstract*—Youtube is the most popular video platform in the world today. Successful YouTubers can create videos that are widely viewed by many Youtube users around the world. A lot of viral videos on Youtube came from the United States. But, making viral videos on Youtube is a tough challenge, both for seasoned YouTubers and especially for new YouTubers. This research focuses on discovering the properties of these viral videos by clustering them into distinct clusters. K-Means algorithm is used for the clustering process. The purpose of this clustering process is to look for patterns in the data that were previously unseen. The result shows that the videos are divided into three clusters which are built from 3 variables; views, likes and dislikes. The patterns and insights found in this study can be useful for aspiring video makers who want to achieve success as a Youtuber.**

***Index Terms*—clusters, K-Means clustering algorithm, trending video, unsupervised learning, USA, Youtube**

I. PENDAHULUAN

Per 2019, Youtube adalah sebuah *platform* berbagi video yang paling populer di dunia. Sebagai portal video terbesar di Internet, YouTube saat ini menguasai 20% dari lalu lintas di jaringan seluler yaitu sekitar 75, 000 video Youtube ditonton per detik di seluruh dunia [1]. Youtube bukan hanya sekedar sebuah *platform* berbagi video yang ditujukan untuk hiburan, tetapi *platform* ini juga berpotensi untuk dijadikan sarana untuk meningkatkan penjualan sebuah perusahaan. Sekitar 54% pengguna Youtube di Amerika Serikat mengatakan bahwa Youtube penting untuk membantu mereka mengambil keputusan untuk membeli sebuah produk tertentu. Lebih dari 500 jam video diunggah ke Youtube setiap menitnya [2]. Dengan banyaknya video yang diunggah dan ditonton setiap harinya, sulit untuk sebuah video untuk menonjol dan menjadi *viral* atau *trending*. Yang dimaksud dengan istilah viral di sini adalah video tersebut sudah dilihat oleh jutaan manusia [3].

Pada penelitian ini akan dibahas tentang pengelompokan video *trending* di Amerika Serikat dan mengeksplorasi apa saja hal-hal yang mungkin mempengaruhi masing-masing kluster untuk menjadi viral. Bagi beberapa orang, Youtube merupakan tempat untuk mencari uang. Faktor inilah yang menjadi salah satu dorongan untuk membuat video YouTube yang viral. Setiap 1000 *views* di Youtube dapat menghasilkan sampai dengan 0.5 USD. Pada data set video *trending* di Amerika Serikat diketahui bahwa rata-rata *views* adalah 2360785. Ini berarti satu video viral secara rata-rata dapat menghasilkan sekitar 1200 USD (16.7 juta Rupiah). Selain faktor ekonomi, terdapat juga faktor *self-expression* yaitu keinginan seseorang untuk mengekspresikan dirinya, yang mendorong seseorang untuk mengunggah video ke *platform* Youtube. Data diambil dari Youtube Amerika Serikat karena Amerika Serikat memiliki pengguna Youtube terbanyak di dunia dengan 167.4 juta pengguna per November 2018 [4].

Model Clustering dipilih karena lebih pas untuk menggambarkan pengelompokkan karena pengelompokan sifat-sifat video Youtube bukan brrbentuk regresi [5] dan trend yang dibahas bukanlah trend secara analisis numerik stokastik [6]. Video Youtube yang *trending* ini dikelompokkan dengan metode *k-means* karena metode *k-means* mampu *scales to large data sets.*

Untuk membuat sebuah video yang viral, perlu dipelajari terlebih dahulu karakteristik video-video yang viral tersebut. Penelitian ini berfokus pada mencari tahu apakah terdapat pola tertentu pada video-video viral dan apakah terdapat *clustering* atau pola tertentu dari video-video *Trending* Youtube di Amerika Serikat. Bila ada, apa saja yang membedakan kluster-kluster ini? Dengan cara apa perbedaan-perbedaan ini dapat berguna bagi *Youtuber* pemula bila mereka mengunggah video ke *platform* Youtube? Dengan adanya penelitian ini, diharapkan pembaca akan semakin memahami apa saja hal-hal yang mempengaruhi sebuah video agar dapat meraih *views*, *likes*, dan comments yang banyak, sehingga dapat membuat video yang *trending*. Penelitian ini juga akan membantu pembaca yang sekedar ingin

mencari tahu kemampuan dan contoh penggunaan dari algoritma *K-Means Clustering*.

II. LANDASAN TEORI

A. *Tentang Algoritma K-Means Clustering*

Clustering adalah sebuah proses untuk membagi data menjadi kelompok-kelompok (kluster) berdasarkan sebuah pola tertentu. *Clustering* bersifat *unsupervised learning* artinya algoritma ini tidak menerima variabel output untuk dijadikan contoh. Algoritma ini berfungsi untuk mencari pola dari sebuah input tertentu. Berbeda dengan *supervised learning* yang menerima pasangan input dan output, mempelajari polanya, lalu menggunakannya untuk membuat prediksi.

K-*Means Clustering* adalah salah satu metode *clustering* yang paling banyak digunakan, di mana dataset sebanyak "n" dikelompokkan ke dalam grup/kluster sebanyak "k". Tujuan utama dari algoritma K-*Means Clustering* adalah untuk memperkecil jumlah jarak antara titik-titik dengan masing-masing centroid klusternya. Algoritma yang digunakan dalam K-*Means clustering* adalah sebagai berikut [7]:

(1) Tentukan titik *centroid* sebanyak K secara acak,

(2) Atur data sehingga terbentuk K kluster dengan titik-titik *centroid* yang telah ditentukan sebelumnya. Pengelompokan data dalam K-*means* dilakukan dengan menghitung jarak data dengan titik *centroid* terdekat. Perhitungan jarak dilakukan dengan jarak *Euclidean*. Rumus jarak *Euclidean* secara umum:

$$d(A, C) = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2} \quad (1)$$

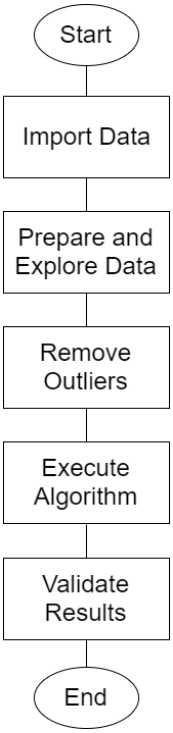
Di mana A adalah data anggota suatu kluster dan C adalah titik *centroid*.

(3) Hitung nilai titik *centroid* dengan menghitung posisi titik tengah dari masing-masing anggota kluster.

(4) Ulangi langkah 2 dan langkah 3 sampai nilai dari titik *centroid* sudah tidak perubahan.

B. *Kelebihan dan Kelemahan Algoritma K-Means Clustering*

Algoritma K-*Means Clustering* berguna untuk mencari kelompok atau kluster dalam sebuah data yang tidak ditentukan sebelumnya. Algoritma ini dapat bermanfaat untuk mencari tahu pola yang sebelumnya tidak terlihat dari sebuah data. Kelebihan dari K-*Means Clustering* adalah implementasi yang sederhana, dapat mengolah data yang besar, dan dapat menyesuaikan dengan data yang berbeda-beda. Sedangkan kelemahan dari K-*Means Clustering* adalah kita harus secara manual menentukan berapa banyak kluster (k) yang mau kita cari sebelum menjalankan algoritmanya. Kualitas hasil algoritma K-*Means Clustering* juga dapat berkurang bila terdapat *outliers* sehingga lebih baik untuk mengurangi atau menghapus *outliers* dari data. Selain itu, k-*means* juga membutuhkan sumber daya yang besar karena algoritma k-*means* berulang-ulang kali menyisir data sampai memusat ke satu hasil [5].



Gambar 1. *Flow Chart* proses pembuatan model algoritma

III. METODOLOGI

Objek yang diteliti pada penelitian ini adalah video yang viral di Youtube Amerika Serikat. Video-video yang diteliti memiliki atribut antara lain berapa kali video tersebut ditonton (*views*), berapa banyak orang yang menyukai video tersebut (*likes*), dan berapa banyak orang yang tidak menyukai video tersebut (*dislike*). Sebagaimana terlihat pada Gambar 1, kerangka kerja dari penelitian ini terdiri dari *import data*, *prepare and explore data*, *remove outliers*, *execute algorithm*, dan *validate results.*

A. *Proses 1: Import Data*

Data set yang digunakan pada penelitian ini adalah data set "*Trending YouTube Video Statistics*" yang diunduh dalam bentuk file .csv dari Kaggle.com. Karena keterbatasan perangkat keras, dari data yang disediakan, hanya 500 observasi dari data Amerika Serikat yang digunakan untuk perhitungan k-*means*

clustering. Bentuk data yang digunakan berupa tabel dengan 16 kolom (variabel). Dari ke 16 kolom ini, yang digunakan untuk algoritma *clustering* hanya kolom 8, 9, dan 10 (*views*, *likes*, dan *dislikes*) yang semuanya merupakan variabel numerik. Kolom-kolom lain tidak digunakan karena tidak relevan dan tipe datanya tidak sesuai untuk proses k-*means clustering*. Data yang digunakan pada penelitian ini valid karena data dikumpulkan menggunakan YouTube Data API oleh seorang pengguna Kaggle bernama Mitchell J. Untuk setiap video yang terdapat di dataset dilakukan pencarian di Google, dan terbukti video tersebut sungguh ada dan bukan video palsu [8].

B. *Proses 2: Prepare and Explore Data*

Proses ini menerima input berupa data yang sudah diimpor ke dalam RStudio. Dari kolom yang terdapat pada data, hanya 3 yang akan digunakan untuk *clustering* yaitu *views*, *likes*, dan *dislikes*. Pada proses ini, sebuah dataset baru dibuat dari data video YouTube yaitu myData yang menampung hanya ketiga kolom yang dibutuhkan untuk menjalankan algoritma. Mean, median, min, max, dan kuartil dihitung dengan fungsi *summary* pada RStudio. Lalu, untuk memvisualisasikan data, digunakan *boxplot* untuk melihat persebaran data. Uji Normalitas Anderson-Darling digunakan untuk menguji normalitas dari variabel-variabel *views*, *likes*, dan *dislikes*.

C. *Proses 3: Remove Outliers*

Outliers adalah pengamatan yang terletak pada jarak yang tidak normal dari nilai-nilai lain dalam sampel acak dari suatu populasi. Pada penelitian ini, kami menggunakan box plot untuk mendeteksi *outliers*. Box plot dibuat dengan menggambar kotak di antara kuartil atas dan bawah dengan garis solid yang ditarik melintasi kotak untuk menandakan median. Untuk mengidentifikasi *outliers*, kita harus menghitung *fence* seperti berikut:

$$lower\ inner\ fence = Q1 - 1.5 * IQ \quad (2)$$

$$upper\ inner\ fence = Q3 + 1.5 * IQ \quad (3)$$

di mana: Q1 adalah kuartil 1, Q3 adalah kuartil 3, dan IQ adalah jarak antar kuartil (*Interquartile range*) yaitu Q3-Q1. Nilai-nilai yang berada "di luar" *fence* tersebut dikategorikan sebagai sebuah pencilan (*outlier*). "Di luar" di sini maksudnya adalah lebih kecil daripada *lower inner fence* atau lebih besar daripada *upper inner fence* [9].

Karena sifat mean yang sangat mudah terpengaruh dengan nilai yang ekstrem seperti pencilan, algoritma K-*means Clustering* juga sangat sensitif terhadap *outliers*, sehingga pada penelitian ini *outliers* dari myData dihapus. Proses menerima input berupa myData dari proses sebelumnya, lalu menggunakan fungsi *for* untuk berulang-ulang menghapus *outliers* sampai tidak ada *outliers* lagi. *Outlier* adalah data-data yang berada di luar *range* dari batas bawah dan batas atas. Proses *remove* data *outliers* dilakukan dengan menggunakan program R [10].

D. *Proses 4: Execute Algorithm*

Pada proses ini, input diterima berupa myData yang sudah bersih dari *outliers*. Dikarenakan keterbatasan kekuatan komputasi, data yang diambil hanya sebanyak 500 observasi yang dipilih secara *random sampling*, sehingga ukuran myData menjadi sebanyak 500 observasi. Setelah mengurangi jumlah observasi, myData digunakan untuk membuat model k-*means clustering* menggunakan fungsi *kmeans*() pada RStudio. K dimulai dengan nilai 1 sampai dengan 10 kemudian nilai K dipilih dengan menggunakan *elbow method*. *Elbow Method* merupakan metode yang bersifat visual. Ide dari *elbow method* adalah meningkatkan k sebanyak 1 setiap kali, lalu hitung *within sum of squares* (wss) masing-masing k. Pada satu titik, nilai wss turun secara dramatis, lalu menjadi mendatar. K pada titik inilah yang paling optimal untuk dijadikan jumlah kluster [11].

E. *Proses 5: Validate Results*

Proses ini menerima input berupa model yang dihasilkan oleh algoritma. Pada proses ini, hasil kluster yang sudah dibuat oleh algoritma akan diuji kualitasnya. Proses akan menghasilkan nilai tertentu yang menentukan kualitas dari model. Terdapat 2 jenis *clustering validation statistics*, yaitu Internal kluster *validation* dan *External* kluster *validation*. *Internal cluster validation* menggunakan informasi internal untuk mengevaluasi kluster, sedangkan *external cluster validation*, membandingkan hasil analisis kluster dengan hasil yang diketahui secara eksternal, seperti label kelas yang disediakan dari luar data yang digunakan. *External cluster validation* mengukur sejauh mana label dari klaster cocok dengan label kelas yang disediakan secara eksternal. Karena kita sudah tahu kluster yang "benar" sebelumnya, *external cluster validation* lebih sering digunakan untuk menilai dan memilih algoritma pengelompokan yang tepat untuk sebuah set data tertentu, daripada untuk menilai kinerja algoritma pada data yang belum terkelompokkan sebelumnya. Sehingga pada penelitian ini, kluster *validation* yang digunakan hanya *Internal kluster validation* karena tidak terdapat label eksternal yang diketahui untuk dijadikan acuan pada *external cluster validation*. Fungsi yang digunakan adalah *cluster.stats* dari *library fpc* pada RStudio [12].

Metode pertama yang akan digunakan untuk validasi hasil adalah menguji Dunn Index. Dunn Index adalah perbandingan jarak terkecil antara pengamatan yang tidak berada dalam kluster yang sama dengan jarak terbesar antar 2 titik di dalam kluster yang sama. Indeks Dunn memiliki nilai antara nol dan *infinity*.

Semakin tinggi nilai Dunn Index, maka semakin baik klusternya. Nilai Dunn index didapatkan dengan menjalankan perintah pada RStudio yaitu *stat$dunn*. Formula Dunn Index adalah:

$$DI = \frac{\min \delta(C_i, C_j)}{\max \Delta k} \quad (4)$$

di mana: $\delta(C_i, C_j)$ adalah jarak antar kluster Ci dengan kluster Cj, dan $\max \Delta k$ adalah jarak di dalam kluster k, atau bisa dibilang metrik untuk mengukur besar kecilnya kluster. Nilai DI yang semakin tinggi mengindikasikan jarak antar kluster yang semakin besar (kluster terpisah jauh satu sama lain) dan jarak di dalam kluster yang semakin kecil (ukuran kluster yang lebih kecil) [13].

Metode kedua yang akan digunakan untuk validasi hasil adalah memeriksa *Average-Between* dan *Average-Within* dari kluster. *Average-Between* adalah rata-rata jarak antar kluster. Semakin besar nilai *Average-Between* maka antara kluster semakin terpisah jauh dan model kluster semakin baik. *Average-Within* adalah rata-rata jarak antar titik di 1 kluster yang sama. Semakin kecil *Average-Within* maka anggota kluster akan semakin mendekati pusat kluster, dan model kluster semakin baik. *Average Between* didapatkan dengan menjalankan perintah *stat$average.between* pada RStudio. *Average within* didapatkan dengan menjalankan perintah *stat$average.within* pada RStudio [12].

Metode ketiga yang akan digunakan untuk memvalidasi hasil adalah membuat *silhouette* plot dengan pertama-tama menghitung *silhouette coefficient* (Si) terlebih dahulu.

Silhouette coefficient dihitung dengan menggunakan rumus sebagai berikut:

$$Si = \frac{(b - a)}{\max(a, b)} \quad (5)$$

di mana a adalah rata-rata jarak intra-kluster (*mean intra-cluster distance*), dan b adalah rata-rata jarak terdekat antar kluster (*mean nearest-cluster distance*). Dengan kata lain, b adalah jarak antara sebuah sampel dan kluster terdekat yang bukan kluster sampel tersebut. Nilai Si berkisar antara -1 sampai 1. Semakin nilai Si mendekati 1 maka titik-titik di dalam kluster semakin mirip, berarti data terkelompokkan dengan baik. Sebaliknya, semakin nilai Si mendekati -1 maka titik-titik di kluster semakin berbeda, dan data tidak terkelompokkan dengan baik. *Silhouette* didapatkan dengan menjalankan perintah *silhouette()* pada RStudio [14].

IV. HASIL DAN PEMBAHASAN

A. *Proses 1: Import Data*

Sebagaimana terlihat pada Gambar 2, output dari proses 1 adalah sebuah data *frame* yang dapat digunakan di RStudio. Data *frame* ini sangat besar dan berisi 16 variabel yang berbeda. *Library* yang digunakan untuk mengimpor data adalah *library reader*. Data *frame* disimpan dalam variabel data.

```
> head(data)
# A tibble: 6 x 16
  video_id trending_date title channel_title category_id publish_time        tags   views
  <chr>    <chr>         <chr> <chr>                <dbl> <dttm>              <chr>  <dbl>
1 2kyS6Sv~ 17.14.11      WE W~ CaseyNeistat            22 2017-11-13 17:13:01 SHAN~ 7.48e5
2 1zAPwfr~ 17.14.11      The ~ LastWeekToni~           24 2017-11-13 07:30:00 "las~ 2.42e6
3 5qpjK5D~ 17.14.11      Raci~ Rudy Mancuso            23 2017-11-12 19:05:24 "rac~ 3.19e6
4 puqaWrE~ 17.14.11      Nick~ Good Mythica~           24 2017-11-13 11:00:04 "rhe~ 3.43e5
5 d38OmeD~ 17.14.11      I Da~ nigahiga                24 2017-11-12 18:01:41 "rya~ 2.10e6
6 gHZ1QZ0~ 17.14.11      2 We~ ijustine                28 2017-11-13 19:07:23 "iju~ 1.19e5
# ... with 8 more variables: likes <dbl>, dislikes <dbl>, comment_count <dbl>,
#   thumbnail_link <chr>, comments_disabled <lgl>, ratings_disabled <lgl>,
#   video_error_or_removed <lgl>, description <chr>
```

Gambar 2. *Head* dari data

B. *Proses 2: Prepare and Explore Data*

Sebagaimana terlihat pada Gambar 3, output proses 2 adalah data yang lebih ringkas yang hanya terdiri dari 3 variabel, ciri-ciri data, dan *boxplot* dari data.

```
> summary(myData)
     views               likes             dislikes       
 Min.   :      549   Min.   :     0   Min.   :     0  
 1st Qu.:   242329   1st Qu.:  5424   1st Qu.:   202  
 Median :   681861   Median : 18091   Median :   631  
 Mean   :  2360785   Mean   : 74267   Mean   :  3711  
 3rd Qu.:  1823157   3rd Qu.: 55417   3rd Qu.:  1938  
 Max.   :225211923   Max.   :5613827   Max.   :1674420
```

Gambar 3. *Summary* dari ketiga variabel yang diteliti

Karena data yang diterima dari Kaggle.com ini sudah bersih, maka tidak perlu dilakukan *cleansing*, namun perlu dilakukan penghapusan *outliers* yang akan dijelaskan pada bagian berikut.

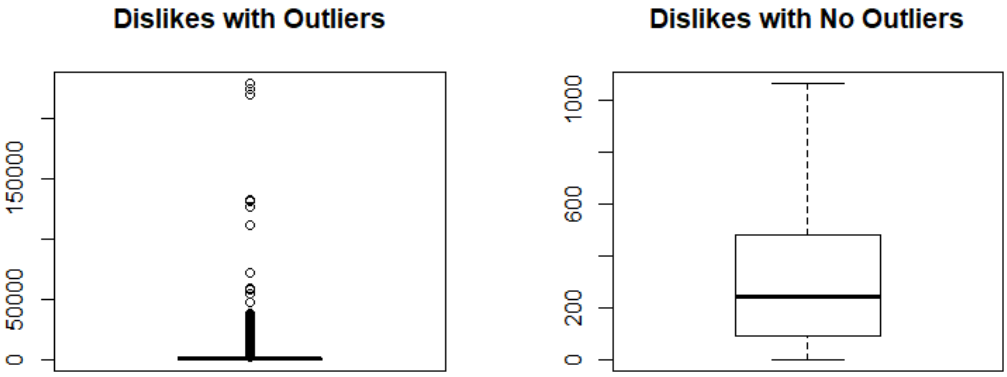
C. *Proses 3: Remove Outliers*

Output proses 3 adalah data yang sudah bersih dari *outliers*, jumlah data berkurang dari 40949 menjadi 22930.

Cara menghapus *outliers* adalah dengan menggunakan *for loop* pada Rstudio. Berikut adalah penjabaran langkah-langkahnya:

1. Buat *boxplot* menggunakan data, dan cari *outliers*-nya,
2. Bila ada *outliers*, maka hapus *outliers* yang ada di *boxplot* tersebut lalu ulangi langkah 1 dengan data yang sudah dikurangi *outliers*-nya.
3. Bila tidak ada *outliers* maka *loop* selesai.

Aktivitas ini diulang-ulang sampai *outliers* sepenuhnya hilang dari data. Gambar 6 memperlihatkan hasil sebelum dan setelah *outliers* dihilangkan.



Gambar 6. Boxplot *dislikes* sebelum dan setelah *outliers* dibuang

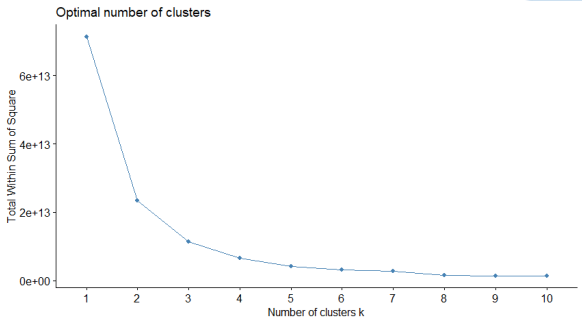
D. Proses 4: Execute Algorithm

Output proses 4 adalah hasil k-*means clustering* dari data yang sudah dibuang *outliersnya*. Jumlah sampel yang digunakan untuk penelitian dikurangi dengan metode *random sampling* sebanyak 500 observasi. Hasilnya dapat dilihat pada Gambar 4.

```
> result
K-means clustering with 3 clusters of sizes 153, 65, 282

Cluster means:
      views     likes dislikes
1  509122.2 15572.248 447.1961
2 1114810.9 23397.677 614.8615
3  143187.1  5302.511 184.4362
```

Gambar 4. Hasil *clustering*



Gambar 5. Grafik untuk mengidentifikasi *elbow point*

Dari Gambar 5, terlihat penurunan dari k = 2 ke k = 3, lalu pelandaian mulai dari k = 3 sampai k = 4 dan seterusnya. Sehingga dari grafik tersebut kita dapat k yang paling optimal yaitu k = 3. Sehingga nilai k=3 dipilih untuk penelitian ini.

E. Proses 5: Validate Results

Output proses 5 adalah hasil validasi algoritma dan informasi mengenai kualitas *clustering* hasil algoritma k-*means clustering*. Hasilnya terlihat pada Gambar 7.

Validasi hasil metode 1 menggunakan Dunn Index. *Clustering* yang dibuat memiliki Dunn Index sebesar 0.004403429. Hasil ini menandakan kluster hasil algoritma kurang baik karena hasil ini menunjukkan bahwa jarak terbesar 2 titik intra-kluster jauh lebih besar daripada jarak terkecil 2 titik beda kluster. Hal ini dapat dipahami sebagai kluster-kluster yang berdempetan, tidak berjarak jauh satu sama lain, dan kluster-kluster yang cenderung "lebar".

Validasi hasil metode 2 menggunakan *Average-Within* dan *Average-Between*. *Clustering* yang dibuat memiliki *Average-Between* sebesar 555105.1. Nilai ini cukup besar, sehingga diketahui bahwa jarak antar kluster cukup jauh. Sedangkan nilai *Average-Within* dari *clustering* ini adalah 145126.5. Nilai ini cukup kecil, sehingga diketahui bahwa titik-titik cenderung berkumpul ke arah pusat klusternya.

```
> stat$dunn
[1] 0.004403429
> #Average between and within
> stat$average.between
[1] 555105.1
> stat$average.within
[1] 145126.5
>
```

Gambar 7. Hasil kalkulasi Dunn Index, *Average-Between* dan *Average-Within*

Validasi hasil metode 3 menghitung *Silhouette Coefficient* dan membuat *Silhouette* Diagram. Setelah dikalkulasi oleh RStudio, didapat hasil *Silhouette coefficient* per kluster. Terlihat bahwa kluster 1 memiliki *silhouette coefficient* 0.51, kluster 2 memiliki *silhouette coefficient* 0.54, dan kluster 3 memiliki *silhouette coefficient* 0.64. Hal ini berarti kluster 1,2, dan 3 sudah terkelompokkan dengan cukup baik. Dari *silhouette* diagram terlihat bahwa rata-rata *silhouette coefficient* juga cukup baik, yaitu 0.59. Hal ini berarti *clustering* yang terbentuk cukup baik. Hasilnya terlihat pada Gambar 8.

```
> sil <- silhouette(result$cluster, dist(myData2))
> fviz_silhouette(sil)
  cluster size ave.sil.width
1       1  153          0.51
2       2   65          0.54
3       3  282          0.64
>
```

Gambar 8. Hasil kalkulasi *silhouette coefficient*

F. *Diskusi*

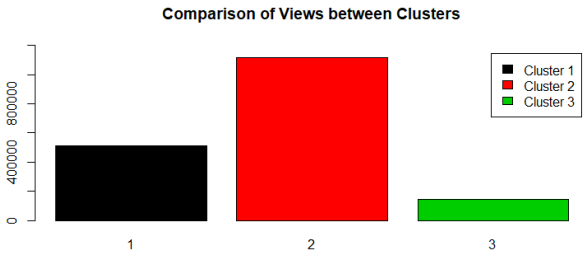
Sebagaimana terlihat pada Gambar 9, hasil *clustering* dari algoritma tersebut membagi data menjadi 3 kluster. Kluster pertama memiliki anggota sebanyak 153 titik dari total 500 titik. Kluster ini memiliki ciri-ciri *views*, *likes*, dan *dislikes* berada di tengah-tengah 2 kluster lainnya. Terdapat 34.82 *likes* setiap 1 *dislikes* artinya secara keseluruhan video ini cukup disukai oleh penonton, walaupun jumlah *views* jauh di bawah kluster kedua, jumlah *views* hanya sekitar setengah dari kluster kedua.

Kluster kedua memiliki anggota paling sedikit yaitu hanya 65 dari 500 video. Sebagaimana terlihat pada gambar 10, gambar 11, dan gambar 12, kluster ini memiliki jumlah *views*, *likes*, dan *dislikes* terbanyak dari antara kluster lainnya, dan memiliki jumlah *likes* terbanyak setiap 1 *dislikes*, walaupun berbeda tipis dengan kluster pertama. Ini menunjukkan bahwa video pada kluster ini secara umum yang paling disukai, dan paling populer dari antara ketiga kluster lainnya. Tetapi, kluster ini adalah yang paling sulit dicapai, tercermin dari anggotanya yang paling sedikit dari antara kedua kluster lainnya.

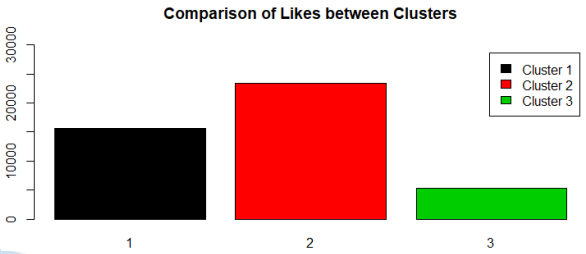
Kluster ketiga memiliki jumlah *views*, *likes*, dan *dislikes* yang paling sedikit dari antara kluster lainnya. Rata-rata *views* pada kluster ini hanya 10% dari rata-rata *views* pada kluster kedua. Jumlah *likes* setiap 1 *dislikes* juga paling kecil, artinya dibanding dengan kedua kluster lainnya, video pada kluster ini secara umum paling tidak disukai. Namun sebagaimana terlihat pada gambar 13, anggota pada kluster ini paling banyak yaitu 282 video, yang berarti sebagian besar video masuk pada kluster ini.

```
> cbind(result$centers,(result$centers[,2]/result$centers[,3]))
      views     likes dislikes
1  509122.2 15572.248 447.1961 34.82197
2 1114810.9 23397.677 614.8615 38.05357
3  143187.1  5302.511 184.4362 28.74984
```

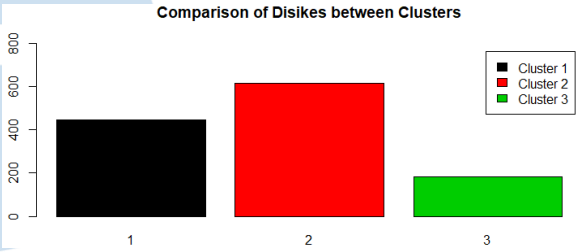
Gambar 9. Menghitung jumlah *likes* setiap 1 *dislikes*



Gambar 10. Perbandingan rata-rata *views* antar kluster



Gambar 11. Perbandingan rata-rata *likes* antar kluster



Gambar 12. Perbandingan rata-rata *Dislikes* antar kluster

Gambar 13. Perbandingan ukuran ketiga kluster

V. SIMPULAN

A. *Kesimpulan*

Penelitian ini memberi informasi tentang pengelompokan video *trending* di YouTube Amerika Serikat. Diketahui bahwa terdapat 3 kluster dengan ciri-cirinya masing-masing. Kluster dengan *views*, *likes*, dan *dislikes* paling sedikit adalah kluster dengan anggota terbanyak, sedangkan kluster dengan *views*, *likes*, dan *dislikes* terbanyak memiliki anggota paling

sedikit. Semakin populer sebuah video, maka cenderung semakin disukai oleh banyak orang.

Untuk membuat video yang viral, tidak cukup hanya membuat video saja, tetapi pembuat video juga harus paham dengan kondisi video-video di *platform* YouTube. Mereka harus menargetkan videonya agar berada di puncak, dalam konteks penelitian ini berarti masuk ke kluster kedua. Pembuat video yang hendak mengunggah videonya ke YouTube dapat melihat bahwa ada pola di dalam video-video *trending*. Dari hasil *clustering* terlihat bahwa kluster kedua adalah kluster dengan anggota yang paling "sukses" dan di saat yang sama adalah kluster yang paling sedikit anggotanya.

B. Saran

Algoritma K-*Means Clustering* juga membutuhkan peneliti untuk menentukan k atau jumlah kluster sebelum menjalankan algoritmanya. Untuk menentukan jumlah kluster, peneliti harus mengandalkan grafik dan melihat secara subjektif. Algoritma lain seperti *hierarchial clustering* dapat menentukan kluster tanpa peneliti menentukannya terlebih dahulu.

C. Limitasi

Limitasi dari penelitian ini adalah jumlah kluster harus ditentukan terlebih dahulu oleh peneliti, dan tidak terdapat metode yang objektif untuk mencari tahu jumlah kluster yang optimal untuk k-*means clustering*.

Keterbatasan lain pada penelitian ini adalah *outliers* sepenuhnya dihapus. Dengan dihapusnya *outliers*, ada kemungkinan yang besar terdapat informasi penting yang hilang bersama dengan *outliersnya*. Penelitian selanjutnya mungkin dapat menggunakan algoritma yang tidak terganggu oleh *outliers*.

UCAPAN TERIMA KASIH

Penelitian ini didukung oleh pendanaan dari Universitas Multimedia Nusantara, Tangerang, Indonesia.

DAFTAR PUSTAKA

[1] Schwind, Anika, M. Cise, A. Özgü, G. Carste, and W. Florian. "Dissecting the performance of YouTube video streaming in mobile networks." International Journal of Network Management, vol.30, no. 3, 2020.

[2] A. Smith, S. Toor dan P. V. Kessel, "Many Turn to YouTube for Children's Content, News, How-To Lessons," Pew Research Center Internet & Technology, 7 November 2018.

[3] Lexico, "Viral," Lexico Powered by Oxford, 2020. [Online]. Available: https://www.lexico.com/definition/viral. [Diakses 17 May 2020].

[4] J. Clement, "Hours of video uploaded to YouTube every minute as of May 2019," Statista, 9 August 2019.

[5] Merfin dan R. S. Oetama, "Prediksi Harga Saham Perusahaan Perbankan Menggunakan Regresi Linear Studi Kasus Bank BCA Tahun 2015-2017", ULTIMATICS. Vol. 11, No. 1, 2019:

[6] R. S. Oetama, "Analisis Titik Tertinggi dan Terendah dengan Model Stokastik pada Perdagangan Mata Uang Modern: Studi Kasus Perdagangan Harga Emas Bulan April - September 2016", ULTIMA INFOSYS, Vol.7 No. 2, 2016.

[7] I. E. A. Parlina, "Memanfaatkan Algoritma K-Means dalam Menentukan Pegawai yang Layak Mengikuti Assessment Center untuk *Clustering* Program SDP," Computer Engineering, Science and System Journal, vol. 3, no. 1, hal. 87-93, 2018.

[8] Michele. J, "*Trending* YouTube Video Statistics," 3 June 2019. [Online].

[9] NIST/SEMATECH, e-Handbook of Statistical Methods, 2013.

[10] U. C. Academy, "Removing outliers - quick & dirty," RPubs by RStudio, 2018.

[11] Nainggolan, Rena, Resianta Perangin-angin, Emma Simarmata, and Astuti Feriani Tarigan. "Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method." In Journal of Physics: Conference Series, vol. 1361, no. 1, p. 012015. IOP Publishing, 2019.

[12] Nerurkar, Pranav, Aruna Pavate, Mansi Shah, and Samuel Jacob. "Performance of internal cluster validations measures for evolutionary clustering." In Computing, Communication and Signal Processing, hal. 305-312. Springer, Singapore, 2019.

[13] Jin X., Han J. "K-Medoids Clustering". In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA, 2019.

[14] Dinh, D. Tai, T. Fujinami, and V. N. Huynh. "Estimating the Optimal Number of Clusters in Categorical Data Clustering by Silhouette Coefficient." In International Symposium on Knowledge and Systems Sciences, hal. 1-17. Springer, Singapore, 2019.