

Deteksi Trending Topik Terkait Covid-19 Pada Tweet Bahasa Indonesia Menggunakan Metode Maximum Capturing

Prasetyo Ariwibowo¹, Indra²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi
Universitas Budi Luhur

Jl. Ciledug Raya, Petungkang Utara, Jakarta Selatan, DKI Jakarta, Indonesia 12260

¹po.prasetyo99@gmail.com, ²indra@budiluhur.ac.id

Abstrak. Twitter merupakan layanan jejaring sosial yang sering digunakan sebagai sumber informasi, seperti topik mengenai Covid-19. Banyaknya informasi yang terdapat pada *tweet* terkadang terdapat informasi yang sudah lampau dan tidak penting. Oleh karena itu diperlukan sebuah *trending topics* agar memudahkan mengetahui informasi terbaru dan yang sedang banyak dibicarakan oleh masyarakat. Pada penelitian ini membangun sistem untuk mendeteksi *trending topics* pada sosial media Twitter. *Dataset* yang digunakan sebanyak 500 data *tweet*. Sistem yang dibuat menggunakan metode *Maximum Capturing* untuk membentuk suatu kluster dengan perhitungan *Jaccard Similarity* sebagai penentuan nilai kesamaan antar *tweet* yang digunakan dalam pembentukan kluster. Dari kluster yang terbentuk diproses menggunakan Algoritme Apriori sebagai penentuan *frequent itemset* yang hasilnya sebagai kumpulan kata *trending topics*. Dalam pengujian menggunakan 12 data *ground truth* dengan membandingkan keterkaitan topik pada sebuah media berita online, yaitu Kompas.com yang menghasilkan nilai *topic recall* = 58.3%, *keyword precision* = 66.6%, dan *keyword recall* = 83.3%.

Kata kunci: *Twitter, Trending Topics, Klusterisasi*

1 Pendahuluan

Teknologi informasi yang dengan cepat berkembang membuat media sosial menjadi konsumsi yang banyak digunakan pengguna internet seperti Facebook, Twitter, Instagram, Path, dan lainnya. Twitter adalah layanan *microblogging* yang banyak digunakan untuk memberi informasi dan mendapatkan informasi dan layanan ini menyediakan fitur mengirimkan pesan ke dalam aplikasi yang disebut *tweet* [1]. Twitter saat ini menjadi media sosial dengan pengguna terbanyak yaitu sebanyak 140 juta pengguna dan memproduksi 400 juta *tweet* setiap harinya [2]. Berdasarkan dari data yang dirilis oleh *World of Tweets* pada tahun 2010, Indonesia menjadi peringkat 3 dunia dengan *tweet* sebanyak 13.39% dari total pengguna twitter dunia dan menjadi yang terbanyak di asia dengan aktifitas Twitter 59.97% [3]. Analisis terhadap media sosial berguna untuk mengetahui suatu informasi dan berita mengenai sebuah informasi, preferensi dan opini masyarakat.

Masa pandemi covid-19, pengguna Twitter meningkat dikarenakan berita-berita dan informasi terbaru tentang covid-19 tersedia dan lengkap. Arus informasi dan berita pada aplikasi Twitter sangat cepat dalam penyebarannya. Perangkat pemerintahan yang mempublikasikan sebuah informasi tentang covid-19 antara lain Twitter menyediakan sumber informasi yang besar dan mudah didapat. Sifat informasinya mengalir secara *stream* dan *up-to-date*. Maka dari itu twitter bisa digunakan sebagai sumber *real-time* untuk identifikasi peristiwa di dunia nyata. Topik pembicaraan di twitter sangat beragam, sehingga dibutuhkan pemroses *tweet* untuk memperoleh informasi yang berharga. Salah satu fitur yang menarik pada Twitter adalah *trending topics*. *Trending Topic* adalah menggambarkan suatu kejadian yang sedang umum dibicarakan atau dibahas pada Twitter [4]. *Trending topics* berfungsi untuk mengetahui percakapan yang sedang dibahas saat ini dan membantu pengguna untuk selalu *update* tentang kejadian terbaru dan menemukan masalah utama dari masyarakat. *Trending topics* memiliki peran penting dalam menemukan berita atau kejadian terhangat dan aktual dengan cepat.

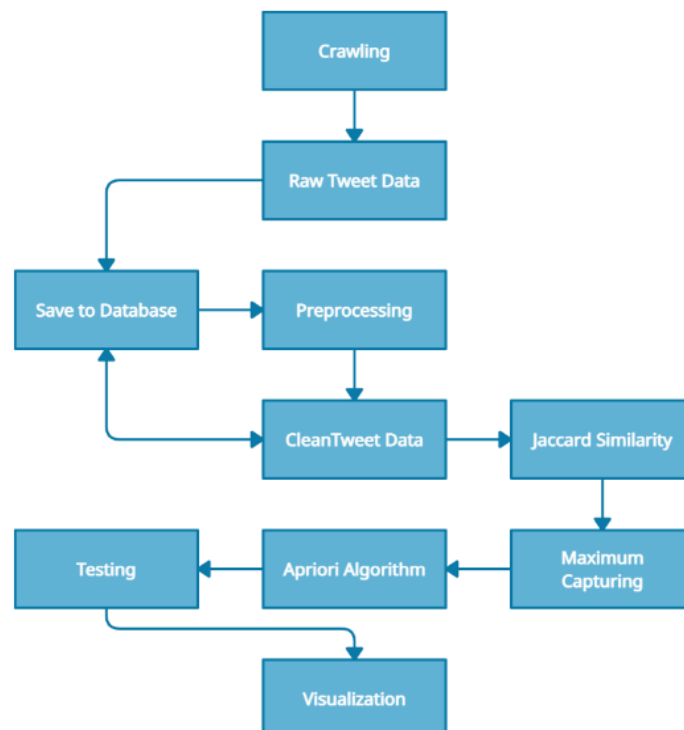
Beberapa penelitian telah melakukan untuk menggali dan mencari *trending topics* dengan berbagai macam teknik yang ditawarkan. Seperti pada penelitian yang dilakukan oleh [5] mengenai proses deteksi dan *tracking topics* mengenai covid-19 dari *tweets* Italia menggunakan NLP dan *graph analysis techniques*. Selain itu penelitian serupa dilakukan oleh [6] mengenai deteksi *trending topics* dan deskripsi secara *real time* dari *tweet* dengan NLP dan proses *clustering* menggunakan *Hierarchical Dirichlet Processes* (HDP). Penelitian terhadap frekuensi topik

juga dilakukan oleh [7] yaitu dilakukan perbandingan terhadap lima metode *clustering* pada sembilan *dataset* berbasis *frequent itemset*. Salah satu teknik yang digunakan dalam menemukan *trending topics* adalah metode *Maximum Capturing Clustering*. Menurut [8] *clustering* atau pengklasteran merupakan teknik pada data mining yang digunakan untuk menganalisis data dan memecahkan permasalahan dengan cara mengelompokkan data atau membagi *dataset* kedalam partisi. *Clustering* dapat digunakan untuk menganalisis topik pada *tweet* dengan mengelompokkan secara otomatis *tweet* yang memiliki kesamaan. *Clustering* digunakan pada teks atau dokumen berbeda dengan *clustering* pada data terstruktur.

Pada penelitian *trending topics* ini menggunakan metode *clustering* Maximum Capturing dengan *Jaccard Similarity* untuk mencari nilai kesamaan antar dokumen atau tweet dan Algoritme Apriori untuk menentukan *frequent itemset*. Menurut jurnal [9] Teknik *Maximum Capturing* merupakan salah satu algoritme klasterisasi dokumen berbasis *frequent itemsets* yang mampu menghasilkan kualitas klasterisasi yang lebih baik dibandingkan dengan yang dihasilkan oleh algoritma sejenis lainnya. Dalam jurnal [10] menjelaskan bahwa *Jaccard Similarity* atau *Jaccard Coefficient* merupakan algoritme yang fungsinya untuk membandingkan dua *sample* yaitu dokumen yang satu dengan yang lainnya berdasarkan kata yang dimilikinya. Algoritme apriori adalah satu algoritme dasar yang diusulkan oleh Agrawal dan Srikan pada tahun 1994 untuk menemukan *frequent itemsets* pada aturan asosiasi Boolean [11].

Penelitian mengenai *trending topics* merupakan topik yang menarik untuk diteliti. Oleh karena itu, penulis ingin melakukan penelitian untuk menemukan *trending topics* menggunakan data *tweet* yang berkaitan dengan Covid-19 pada tahun 2021 di Indonesia. Pengambilan data Twitter dengan kata kunci tertentu seperti *satgascovid19id*, *KemendesRI*, *BNPB_Indonesia*, dan *KawalCOVID19*. Aplikasi *trending topics* akan dibuat dengan menggunakan bahasa pemrograman *Python*.

2 Metode Penelitian



Gambar 1. Tahapan Metode

Alur penelitian dalam penerapan klusterisasi untuk mendeteksi *trending topics* terdapat beberapa tahapan awal hingga akhir yang dinyatakan seperti pada gambar 1 berikut.

2.1 Data Penelitian

Data yang digunakan dalam penelitian ini merupakan dataset yang dikumpulkan dari Twitter dalam bentuk format *list tweet* yang di ambil pada bulan Juni – Juli 2021 sebanyak 500 data yang terintegrasi dengan *database* sehingga otomatis tersimpan ke dalam *database*. Proses pengambilan data dengan melakukan proses *crawling* menggunakan API Twitter, *library tweepy* dari bahasa pemrograman *Python*. Dalam pengambilan *dataset* menggunakan *keyword*: covid, vaksin, satgascovid19id, KemenkesRI, BNPB_Indonesia, KawalCOVID19.

Pada tabel 1 merupakan Tabel Sampel Data *Tweet* yang telah melalui proses *preprocessing* sebagai contoh yang akan digunakan dalam pembahasan sistem deteksi *trending topics*.

Tabel 1. Tabel Sampel Data

id_text	tweet_text
1	lonjakan kasus corona indonesia
2	vaksinasi indonesia astrazeneca
3	swab antigen vaksin sinovac
4	vaksin corona serentak nasional
5	isolasi mandiri masyarakat terpapar covid

2.2 Preprocessing

Berdasarkan data yang diperoleh, dilakukan *preprocessing data*. Preprocessing bertujuan memproses data agar lebih terstruktur dan menyiapkan data yang akan digunakan pada tahap selanjutnya [12]. Berikut tahapan yang dilakukan dalam *preprocessing* :

1. *Filtering* : proses untuk menghapus karakter selain a sampai z atau karakter selain huruf, seperti simbol, angka, tanda, dan url.
2. *Spaces Cleaning* : proses menghapus spasi yang lebih dari satu spasi, menghapus *leading spaces* atau spasi berada pada depan kalimat, serta kalimat yang mengandung kata yang menyatu atau tidak menggunakan spasi (*unused spaces*).
3. *Case Folding* : proses penyetaraan kata yang mengandung huruf besar untuk diubah menjadi huruf kecil.
4. *Slangword* : kata-kata tidak baku diubah menjadi kata yang utuh dan baku. Perubahan kata ini berdasarkan kamus *slangword*.
5. *Stopword* : proses menghapus kata-kata yang terkandung dalam *stoplist* berdasarkan kamus Tala.
6. *Stemming* : proses menghilangkan imbuhan dan mengubah menjadi kata dasar.

2.3 Jaccard Similarity

Setelah melalui *preprocessing* dilakukan perhitungan menggunakan *Jaccard Similarity* yang berfungsi sebagai metode untuk membandingkan antar data *tweet* berdasarkan kata yang dimiliki [10]. Berikut adalah rumus perhitungan dari *Jaccard Similarity*:

$$(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \times 100\% \quad (1)$$

Dimana:

x: Dokumen i

y: Dokumen j

Berdasarkan persamaan (1) yang merupakan rumus *Jaccard Similarity* yang digunakan untuk mencari nilai persamaan pada dua sampel data sebagai contoh diketahui X *tweet* ke-11 “swab virus corona vaksinasi astrazeneca”, dan Y *tweet* ke-14 “swab antigen vaksinasi astrazeneca sinovac”.

$$\text{Similarity (11, 14)} = \frac{|\text{swab vaksinasi astrazeneca}|}{|\text{swab virus corona antigen vaksinasi astrazeneca sinovac}|}$$

$$\text{Similarity (11, 14)} = \frac{|3|}{|7|} = 0.43$$

Berdasarkan contoh perhitungan di atas, pada tabel 2 merupakan contoh hasil dari nilai *similarity* perbandingan antar *tweet*.

Tabel 2. Contoh Hasil Jaccard Similarity

No.	ID Pasangan Perbandingan	Tweet <i>i</i>	Tweet <i>j</i>	Nilai Similarity
1	16	3	14	0.5
2	78	14	3	0.5
3	59	11	14	0.43
4	82	14	11	0.43
5	27	5	15	0.38

2.4 Maximum Capturing

Langkah selanjutnya pembentukan kluster menggunakan *Maximum Capturing* dengan mengelompokkan data *tweet* berdasarkan nilai *similarity* terbesar. Berikut tahapan pembentukan kluster pada metode *Maximum Capturing* [7]:

1. Membangun matriks kesamaan dokumen.
2. Menemukan nilai *similarity* minimum selain 0 (nol).
3. Menemukan nilai *similarity* maksimum dokumen, lalu mencari semua dokumen yang belum terkluster.
4. Nilai *similarity* terbesar yang telah ditemukan maka dokumen dibentuk sebagai suatu *cluster*. Jika, nilai *similarity* terbesar sama dengan nilai *similarity* minimum, semua dokumen dalam pasangan perbandingan tersebut yang belum terkluster digunakan untuk membentuk *cluster* baru. Jika, nilai *similarity* terbesar tidak sama dengan nilai *similarity* minimum semua dokumen dalam pasangan perbandingan tersebut, disatukan dalam membentuk suatu *cluster*, untuk setiap pasangan perbandingan dokumen jika salah satu dari dokumen dalam pasangan perbandingan milik *cluster* yang sudah terbentuk, maka dokumen lain yang ada pada pasangan perbandingan yang tidak terlampir dalam suatu *cluster* akan dilampirkan ke dalam *cluster* ini.
5. Kemudian jika terdapat dokumen yang belum termasuk ke dalam *cluster* manapun, maka dokumen tersebut akan digunakan untuk membentuk *cluster* baru.

Dalam proses implementasi metode *Maximum Capturing* pada penelitian ini dilakukan dengan cara menemukan nilai *similarity* terbesar dari seluruh pasangan perbandingan dokumen atau *tweet* yang belum terbentuk sebagai *cluster* yang sebelumnya menentukan nilai *similarity* minimum sebesar 0.1. Sebagai contoh terdapat 10 (sepuluh) dokumen atau data *tweet* yang telah memiliki nilai *similarity* masing-masing pasangan dokumen yang di representasikan dengan matriks sebagai berikut.

	1	2	3	4	5
1	0	0.45	0.55	0.1	0.3
2		0	0.25	0.3	0.5
3			0	0.15	0.4

4	0	0.2
5		0

Berdasarkan matriks tersebut maka mencari nilai *similarity* terbesar, yaitu terdapat pada pasangan dokumen [1,3]. Selanjutnya pasangan dokumen tersebut membentuk menjadi kluster [1,3].

	1	2	3	4	5
1	0	0.45	0.55	0.1	0.3
2		0	0.25	0.3	0.5
3			0	0.4	0.4
4				0	0.2
5					0

Kemudian mencari nilai terbesar kembali selain dokumen yang telah terbentuk kluster dan nilai 0 (nol). Nilai *similarity* terbesar terdapat pada dokumen [2,5] maka terbentuk menjadi kluster [1,3], [2,5].

	1	2	3	4	5
1	0	0.45	0.55	0.1	0.3
2		0	0.25	0.3	0.5
3			0	0.4	0.4
4				0	0.2
5					0

Selanjutnya nilai *similarity* terbesar terdapat pada pasangan dokumen [1,2]. Dokumen [1,2] tidak akan membentuk kluster baru karena telah terbentuk kluster sebelumnya, yaitu kluster [1,3], [2,5].

	1	2	3	4	5
1	0	0.45	0.55	0.1	0.3
2		0	0.25	0.3	0.5
3			0	0.4	0.4
4				0	0.2
5					0

Setelah itu, pada pasangan dokumen [3,4] dan [3,5] memiliki nilai *similarity* terbesar. Sebelumnya telah terbentuk kluster [1,3] dan [2,5] karena dokumen 3 dan 5 telah terbentuk kluster sebelumnya maka hanya akan memasukkan dokumen 4 ke dalam kluster menjadi kluster [1,3,4] dan [2,5] karena terdapat salah satu dokumen yang telah membentuk kluster sehingga dokumen yang belum terbentuk kluster dalam pasangan dokumen tersebut akan di masukkan ke dalam kluster salah satu pasangan dokumen tersebut.

	1	2	3	4	5
1	0	0.45	0.55	0.1	0.3
2		0	0.25	0.3	0.5
3			0	0.4	0.4
4				0	0.2
5					0

Karena seluruh dokumen telah terbentuk kluster maka proses pembentukan kluster selesai dengan menghasilkan 2 buah kluster, yaitu kluster [1,3,4] dan [2,5].

Pada tabel 3 merupakan contoh dari hasil kluster yang telah terbentuk dan kandungan *tweet* secara *unique* dari setiap kluster.

Tabel 3. Tabel *Cluster Candidate Topics*

Cluster ID	Cluster Document	Candidate Topics
1	[3, 14, 11, 2, 8, 7, 1]	['swab', 'antigen', 'vaksin', 'sinovac', 'vaksinasi', 'astrazeneca', 'virus', 'corona', 'indonesia', 'masyarakat', 'positif', 'isoman', 'selama', 'satu', 'minggu', 'dirumah', 'lonjakan', 'kasus']
2	[5, 15]	['isolasi', 'mandiri', 'masyarakat', 'terpapar', 'covid', 'isoman', 'positif', 'corona']
3	[10, 12, 9]	['isoman', 'obat', 'antivirus', 'covid', 'vaksin', 'sinovac', 'konsumsi', 'vitamin']
4	[6, 13, 4]	['dosis', 'vaksin', 'sinovac', 'kurang', 'astrazeneca', 'langka', 'corona', 'serentak', 'nasional']

2.5 Algoritme Apriori

Hasil dari kluster yang telah terbentuk kemudian dilakukan pencarian *frequent itemset* berdasarkan kumpulan *tweet* yang terkandung dalam kluster menggunakan Algoritme Apriori. Pada tahap ini digunakan perhitungan *support* yang merupakan suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan kluster [13]. Berikut persamaan untuk mencari nilai *support*:

$$Support(A) = \frac{\sum \text{kluster mengandung } A}{\sum \text{kluster}} \quad (2)$$

Dalam implementasi ini menggunakan nilai minimum *support* sebesar 0.33. Kemudian menghitung dan *scan* data untuk mendapatkan pola *frequent* dari nilai *support*. Sebagai contoh akan mencari nilai *support* dari kata “vaksin” dengan cara *scan* dan menghitung jumlah kata yang muncul dari setiap kluster berdasarkan data yang terdapat pada tabel 4. Kemudian menghitung nilai *support* dari kata “vaksin” berdasarkan persamaan (2) dengan cara jumlah kemunculan kata “vaksin” dari setiap kluster dibagi dengan jumlah kluster seperti pada persamaan dibawah ini.

$$Support(A) = \frac{3}{4} = 0.7$$

2.6 Pengujian

Pengujian dilakukan untuk mengetahui tingkat presisi dan akurasi dari *trend topics* yang dihasilkan. Pengujian metode pada penelitian ini menggunakan tiga pengukuran yaitu, *Topic Recall* (TR), *Keyword Precision* (KP) dan *Keyword Recall* (KR). Data *Ground Truth* yang digunakan dalam pengujian ini bersumber dari Kompas.com. Dalam pengambilan Data *Ground Truth* dilakukan secara manual. Rumus perhitungan yang digunakan untuk melakukan pengujian seperti pada persamaan (3.2), (3.3), dan (3.4) sebagai berikut:

$$TR = \frac{|GT \cap BT|}{|GT|} \quad (3)$$

$$KP = \frac{|KGT \cap KBT|}{|KBT|} \quad (4)$$

$$KR = \frac{|KGT \cap KBT|}{|KGT|} \quad (5)$$

Dimana:

GT (*Ground Truth Topic*) : sekumpulan topik pada suatu *ground truth*.

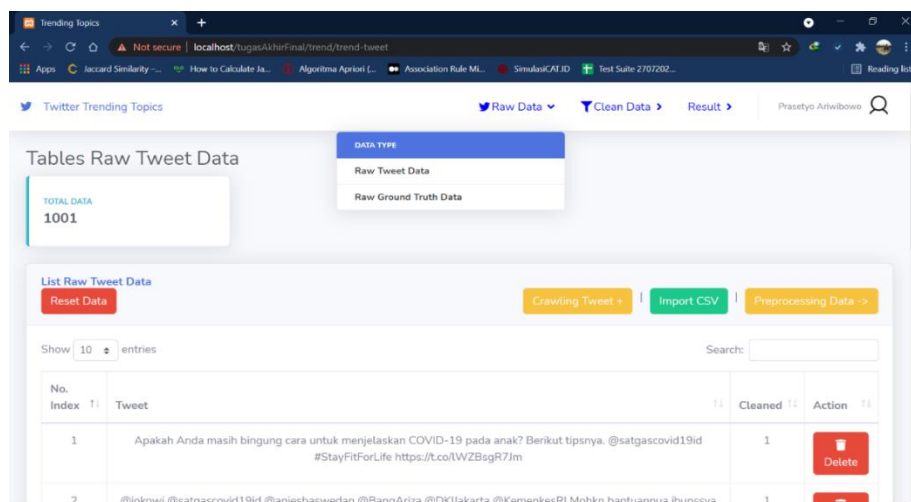
BT (*Trending Topic*) : sekumpulan *trending topic*.

KGT (*Keyword Ground Truth Topic*) : sekumpulan *keyword* pada *ground truth*.

KBT (*Keyword Trending Topic*) : sekumpulan *keyword* pada *trending topic*.

2.7 Visualisasi

Visualisasi dalam sistem deteksi *trending topics* ini menggunakan tampilan melalui *web* agar memudahkan untuk mengetahui dan memahami hasil dari alur sistem ini. Di dalam tampilan *web* tersebut terdiri dari fitur *crawling*, CRUD(Create, Read, Update, Delete) *dataset*, *preprocessing*, *clustering*, pengujian serta hasil *trend topics*. Pada gambar 2 merupakan salah satu visualisasi dari halaman *web trend topics*.



Gambar 2. Visualisasi *Web Trend Topics*

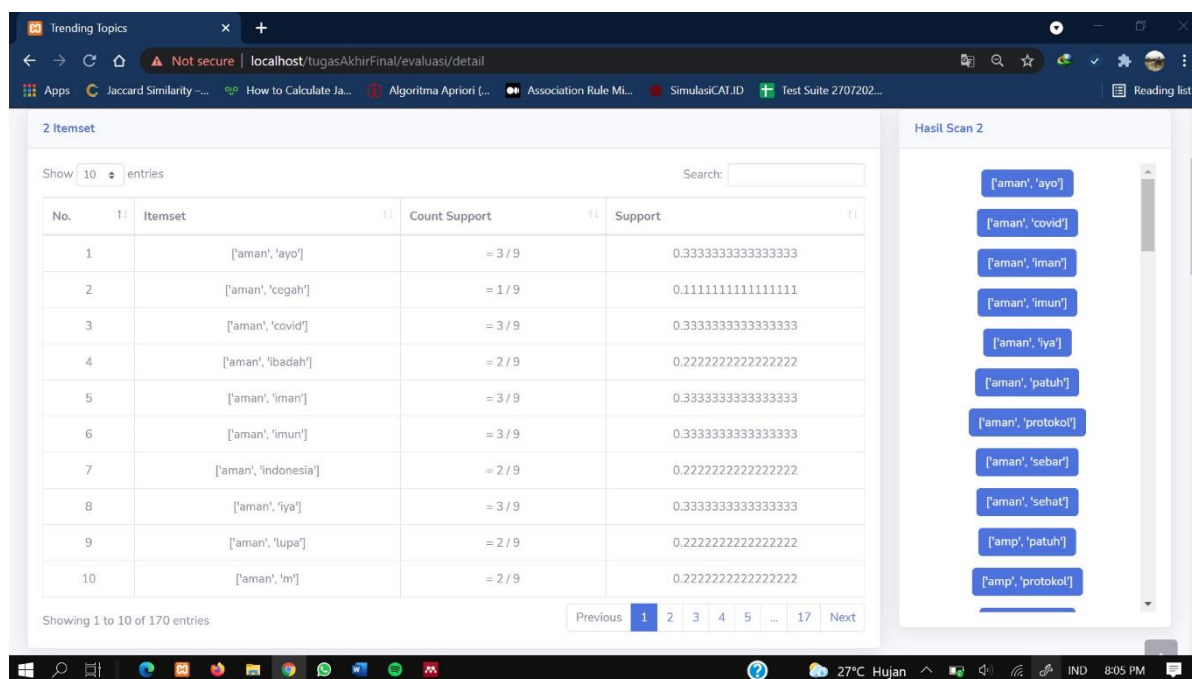
3 Hasil dan Pembahasan

3.1 Hasil Pembentukan Kluster

Pada proses metode *Maximum Capturing* dalam pembentukan kluster dengan menggunakan nilai dari hasil *Jaccard Similarity* berdasarkan *dataset* yang digunakan menghasilkan sebanyak 9 kluster, yaitu [365, 358], [179, 183, 249, 186, 197, 140, 137, 251, 194, 200, 236], [247, 14, 246, 150, 195, 106], [80, 168, 88, 169, 164, 107, 102], [174, 134], [430, 445], [232, 199], [60, 36], [115, 114]. Masing-masing kluster tersebut berisi indeks dari setiap *tweet*. Kemudian kandungan kata-kata *tweet* yang ada dalam setiap kalster dijadikan *unique* dan digabungkan.

3.2 Pembentukan *Frequent Itemset*

Berdasarkan hasil kluster yang diperoleh kemudian dilakukan pencarian *frequent itemsets* menggunakan algoritme apriori. Pada gambar 3 merupakan proses pembentukan dari dua *itemset*, hasil dari *scan* iterasi kedua dan hasil perhitungan nilai *support* dari masing-masing *itemset*.



The screenshot shows a web browser window with a table titled '2 Itemset' and a list titled 'Hasil Scan 2'. The table lists 10 itemsets with their counts and support values. The list shows the results of a scan iteration, listing various itemsets in blue buttons.

No.	Itemset	Count Support	Support
1	['aman', 'ayo']	= 3 / 9	0.3333333333333333
2	['aman', 'cegah']	= 1 / 9	0.1111111111111111
3	['aman', 'covid']	= 3 / 9	0.3333333333333333
4	['aman', 'libadah']	= 2 / 9	0.2222222222222222
5	['aman', 'iman']	= 3 / 9	0.3333333333333333
6	['aman', 'imun']	= 3 / 9	0.3333333333333333
7	['aman', 'indonesia']	= 2 / 9	0.2222222222222222
8	['aman', 'iya']	= 3 / 9	0.3333333333333333
9	['aman', 'lupa']	= 2 / 9	0.2222222222222222
10	['aman', 'm']	= 2 / 9	0.2222222222222222

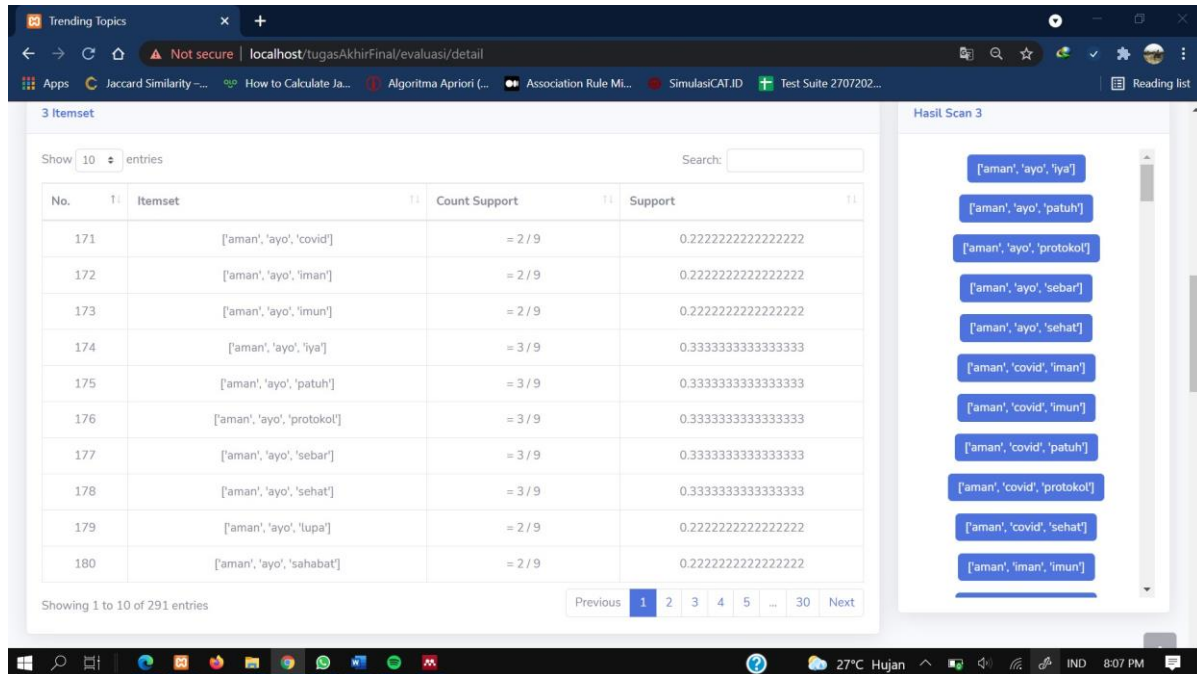
Showing 1 to 10 of 170 entries

Hasil Scan 2

- ['aman', 'ayo']
- ['aman', 'covid']
- ['aman', 'iman']
- ['aman', 'imun']
- ['aman', 'iya']
- ['aman', 'patuh']
- ['aman', 'protokol']
- ['aman', 'sebar']
- ['aman', 'sehat']
- ['amp', 'patuh']
- ['amp', 'protokol']

Gambar 3. Pembentukan 2 *itemset*

Pada gambar 4 merupakan pembentukan dari tiga *itemset*, hasil dari *scan* iterasi ketiga dan hasil perhitungan nilai *support* dari masing-masing *itemset*.



The screenshot shows a web browser window with a table of 3-itemsets and a list of scan results. The table has columns for No., Itemset, Count Support, and Support. The scan results are displayed as blue buttons with itemset strings.

No.	Itemset	Count Support	Support
171	['aman', 'ayo', 'covid']	= 2 / 9	0.2222222222222222
172	['aman', 'ayo', 'imam']	= 2 / 9	0.2222222222222222
173	['aman', 'ayo', 'imun']	= 2 / 9	0.2222222222222222
174	['aman', 'ayo', 'iya']	= 3 / 9	0.3333333333333333
175	['aman', 'ayo', 'patuh']	= 3 / 9	0.3333333333333333
176	['aman', 'ayo', 'protokol']	= 3 / 9	0.3333333333333333
177	['aman', 'ayo', 'sebar']	= 3 / 9	0.3333333333333333
178	['aman', 'ayo', 'sehat']	= 3 / 9	0.3333333333333333
179	['aman', 'ayo', 'lupa']	= 2 / 9	0.2222222222222222
180	['aman', 'ayo', 'sahabat']	= 2 / 9	0.2222222222222222

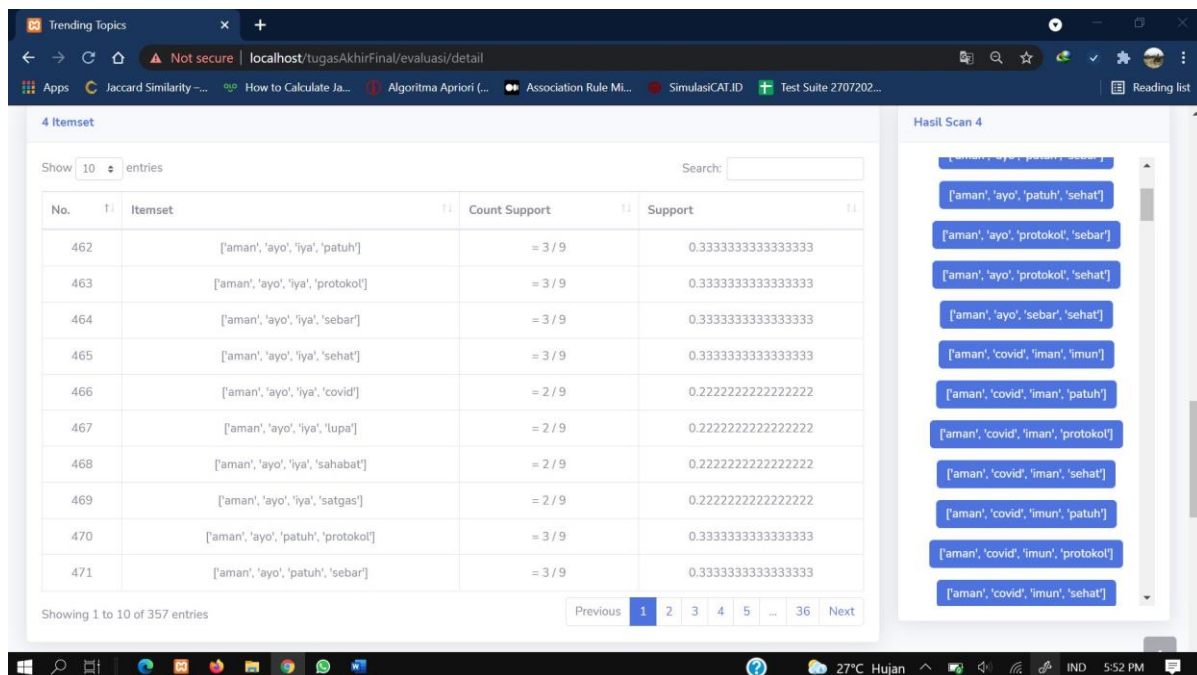
Showing 1 to 10 of 291 entries

Hasil Scan 3

- ['aman', 'ayo', 'iya']
- ['aman', 'ayo', 'patuh']
- ['aman', 'ayo', 'protokol']
- ['aman', 'ayo', 'sebar']
- ['aman', 'ayo', 'sehat']
- ['aman', 'covid', 'imam']
- ['aman', 'covid', 'imun']
- ['aman', 'covid', 'patuh']
- ['aman', 'covid', 'protokol']
- ['aman', 'covid', 'sehat']
- ['aman', 'imam', 'imun']

Gambar 4. Pembentukan 3 *itemset*

Pada tahapan ini dilakukan hingga empat iterasi hasil akhir dari apriori ini seperti pada gambar 5 yang merupakan pembentukan empat *itemset*, hasil *scan* dari iterasi keempat dan hasil perhitungan nilai *support* dari masing-masing *itemset*.



The screenshot shows a web browser window with a table of 4-itemsets and a list of scan results. The table has columns for No., Itemset, Count Support, and Support. The scan results are displayed as blue buttons with itemset strings.

No.	Itemset	Count Support	Support
462	['aman', 'ayo', 'iya', 'patuh']	= 3 / 9	0.3333333333333333
463	['aman', 'ayo', 'iya', 'protokol']	= 3 / 9	0.3333333333333333
464	['aman', 'ayo', 'iya', 'sebar']	= 3 / 9	0.3333333333333333
465	['aman', 'ayo', 'iya', 'sehat']	= 3 / 9	0.3333333333333333
466	['aman', 'ayo', 'iya', 'covid']	= 2 / 9	0.2222222222222222
467	['aman', 'ayo', 'iya', 'lupa']	= 2 / 9	0.2222222222222222
468	['aman', 'ayo', 'iya', 'sahabat']	= 2 / 9	0.2222222222222222
469	['aman', 'ayo', 'iya', 'satgas']	= 2 / 9	0.2222222222222222
470	['aman', 'ayo', 'patuh', 'protokol']	= 3 / 9	0.3333333333333333
471	['aman', 'ayo', 'patuh', 'sebar']	= 3 / 9	0.3333333333333333

Showing 1 to 10 of 357 entries

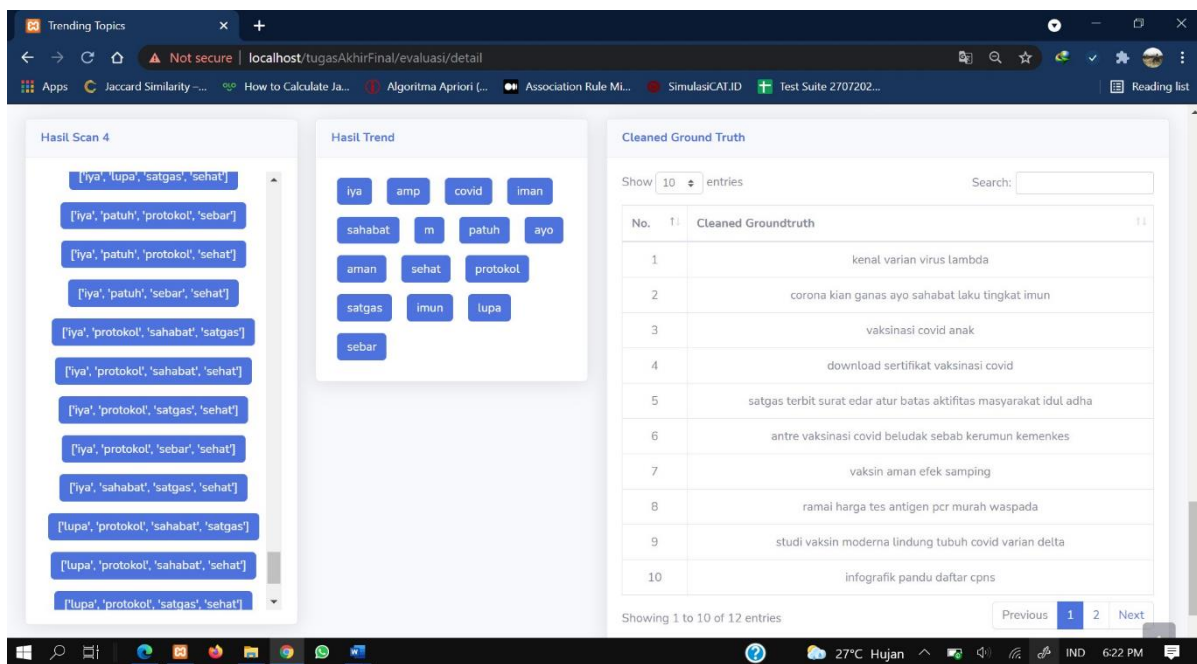
Hasil Scan 4

- ['aman', 'ayo', 'patuh', 'sebar']
- ['aman', 'ayo', 'patuh', 'sehat']
- ['aman', 'ayo', 'protokol', 'sebar']
- ['aman', 'ayo', 'protokol', 'sehat']
- ['aman', 'ayo', 'sebar', 'sehat']
- ['aman', 'covid', 'imam', 'imun']
- ['aman', 'covid', 'imam', 'patuh']
- ['aman', 'covid', 'imam', 'protokol']
- ['aman', 'covid', 'imam', 'sehat']
- ['aman', 'covid', 'imun', 'patuh']
- ['aman', 'covid', 'imun', 'protokol']
- ['aman', 'covid', 'imun', 'sehat']

Gambar 5. Pembentukan 4 *itemset*

3.3 Hasil Trending Topics

Hasil yang didapatkan terdiri dari kumpulan kata, seperti patuh, aman, imun, iman, sebar, lupa, ayo, sahabat, satgas, protokol, sehat, covid seperti pada tampilan gambar 6 berikut. Kumpulan *trend topics* tersebut diperoleh berdasarkan *scan* iterasi keempat dari *frequent itemset* dengan nilai *support* yang memenuhi nilai minimum yang sebelumnya telah ditentukan. Maka kumpulan kata tersebut memiliki nilai *support* dan frekuensi kemunculan yang paling tinggi diantara iterasi pertama hingga iterasi keempat dapat dikatakan bahwa kumpulan kata tersebut merupakan hasil dari *trend topics*. Hasil yang didapatkan dapat terpengaruh berdasarkan *dataset* yang digunakan, nilai minimum *support* yang ditentukan dan jumlah iterasi yang dilakukan.



Gambar 6. Hasil Trending Topics

3.4 Pengujian

Berdasarkan hasil *trend topics* tersebut maka dilakukan pengujian dengan menggunakan 12 data *ground truth* yang di ambil dari media online, yaitu Kompas.com seperti pada tabel 4 berikut.

Tabel 4. Tabel Data *Ground Truth*

<i>Ground Truth Topics</i>	<i>Ground Truth Keyword</i>
kenal Varian Baru Virus Lambda	Kenal varian virus lambda
corona Kian Ganas, Ayo Sahabat Lakukan 6 Cara Ini untuk Tingkatkan Imun	corona kian ganas ayo sahabat laku tingkat imun
apan Vaksinasi Covid-19 untuk Anak 1-17 Tahun Dimulai?	vaksinasi covid anak
Caranya Download Sertifikat Vaksinasi Covid-19	download sertifikat vaksinasi covid
satgas Terbitkan Surat Edaran Aturan Pembatasan Aktifitas Masyarakat Selama Idul Adha	satgas terbit surat edar atur batas aktifitas masyarakat idul adha

ntrean Vaksinasi Covid Membeludak ngga Sebabkan Kerumunan, Ini Kata emenkes	antre vaksinasi covid beludak sebab kerumun kemenkes
pa yang dimaksud Vaksin Aman, adalah Tetap Ada Efek Samping	vaksin aman efek samping
amai soal Harga Tes Antigen dan PCR lurah, Ini yang Perlu Diwaspadai	ramai harga tes antigen pcr murah waspada
udi: Vaksin Moderna Lindungi tubuh dari Covid-19 Varian Delta	udi vaksin moderna lindung tubuh covid varian delta
INFOGRAFIK: Panduan Pendaftaran PNS 2021	infografik pandu daftar cpns
Masjid Ditutup Selama PPKM Darurat, Ini Kata Dewan Masjid	masjid tutup pppm darurat dewan masjid
swa ini Perilaku Hidup Bersih dan sehat Saat Belajar Tatap Muka	siswa perilaku hidup bersih sehat ajar tatap muka

Berdasarkan hasil perhitungan pengujian diatas, maka besaran nilai dari *topic recall* = 58.3%, *keyword precision* = 66.6%, dan *keyword recall* = 83.3%.

4 Kesimpulan

Setelah melakukan studi literatur, perancangan, analisis, implementasi dan pengujian sistem dapat disimpulkan bahwa Sistem Deteksi *Trending Topics* Terkait Covid-19 Pada *Tweet* Bahasa Indonesia Menggunakan Metode *Maximum Capturing* telah dapat mendeteksi *trend topics* pada sebuah data *tweet*. Berdasarkan hasil pengujian data *trend topics* yang ditemukan dengan melakukan perbandingan menggunakan 12 data *ground truth* yang bersumber dari Kompas.com menghasilkan nilai dari *topic recall* = 58.3%, *keyword precision* = 66.6%, dan *keyword recall* = 83.3%.

Referensi

- [1] BA. N. Assidyk et al., "Analisis Perbandingan Pembobotan TF-IDF dan TF-RF pada Trending Topic di Twitter dengan Menggunakan Klasifikasi K-Nearest Neighbor," vol. 7, no. 2, pp. 7773–7781, 2020.
- [2] F. Atefeh and W. Khreich, "A survey of techniques for event detection in Twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 133–164, 2015, doi: 10.1111/coin.12017.
- [3] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion using word embedding for tweet topic classification," *Proceeding 2016 10th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2016 Spec. Issue Radar Technol.*, no. 2011, 2017, doi: 10.1109/TSSA.2016.7871085.
- [4] R. Rafif, E. B. Setiawan, and I. Kurniawan, "Analisis dan implemenasi algoritma C4 . 5 dan pembobotan TF-IDF untuk menentukan trending topik pada media sosial twitter," vol. 7, no. 2, pp. 7661–7672, 2020.
- [5] E. De Santis, A. Martino, and A. Rizzi, "An Infoveillance System for Detecting and Tracking Relevant Topics from Italian Tweets during the COVID-19 Event," *IEEE Access*, vol. 8, pp. 132527–132538, 2020, doi: 10.1109/ACCESS.2020.3010033.
- [6] A. Madani, O. Boussaid, and D. E. Zegour, "Real-time trending topics detection and description from Twitter content," *Soc. Netw. Anal. Min.*, vol. 5, no. 1, pp. 1–13, 2015, doi: 10.1007/s13278-015-0298-5.
- [7] W. Zhang, T. Yoshida, X. Tang, and Q. Wang, "Text clustering using frequent itemsets," *ELSEVIER Knowledge-Based Syst.*, vol. 23, no. 5, pp. 379–388, 2010, doi: 10.1016/j.knosys.2010.01.011.
- [8] M. A. Bianto, S. Rahayu, M. Huda, and Kursini, "Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering Dan Model Bayesian," *Semin. Nas. Teknol. Inf. dan Multimed.*, pp. 19–24, 2018.
- [9] G. Pradnyana and A. Djunaidy, "Metode Weighted Maximum Capturing Untuk Klasterisasi Dokumen Berbasis Frequent Itemsets," *J. Ilmu Komput.*, vol. 6, no. 2, pp. 1–10, 2013.
- [10] Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma WInnowing," *Transmisi*, vol. 20, no. 3, p. 105, 2018, doi: 10.14710/transmisi.20.3.105-110.

- [11] R. Agrawal and R. Srikant, "A fast algorithm for mining association rules," in Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 1994, pp. 513–516, doi: 10.1109/ICSESS.2014.6933618.
- [12] S. Syarif, Anwar, and Dewiani, "Trending topic prediction by optimizing K-nearest neighbor algorithm," Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017, vol. 2018-Janua, pp. 1–4, 2018, doi: 10.1109/CAIPT.2017.8320711.
- [13] L. Kurniawati, A. E. Kusuma, and B. Dewansyah, "Implementasi Algoritma Apriori Untuk Menentukan," CESS (Journal Comput. Eng. Syst. Sci., vol. 4, no. 1, pp. 6–10, 2019.