

## TWITTER TEXT MINING MENGENAI ISU VAKSINASI COVID-19 MENGUNAKAN METODE TERM FREQUENCY, INVERSE DOCUMENT FREQUENCY (TF-IDF)

Edo Harieby, Hoiriyah, Miftahul Walid

Universitas Islam Madura

walangitan07@gmail.com

### ABSTRAK

Penyebaran informasi mengenai vaksin covid-19 menarik perhatian masyarakat. Berbagai macam isu bermunculan terkait halal dan tidaknya vaksinasi covid-19 dilakukan. Media sosial *Twitter* salah satunya yang memberikan ruang pada masyarakat untuk menanyakan dan berkomentar terkait vaksin covid-19 melalui cuitan (*tweet*) ataupun *retweet*. Dengan metode TF-IDF, penelitian ini dilakukan untuk menganalisis *text* (analisis sentimen) dari kumpulan *tweet* sehingga hasilnya diketahui banyaknya kata yang muncul dapat menjadi suatu kata kunci dalam perbincangan di *Twitter*, bahwa banyak masyarakat yang menyetujui adanya wajib vaksin covid-19. Hasil penelitian ini menampilkan 5 kata teratas yang paling banyak muncul, antara lain: vaksin (831.431911 kata), vaksinasi (748.304896 kata), covid (709.626652 kata), sehat (435.356173 kata), dukung (417.387094 kata) dan indonesia (404.432113 kata). Sedangkan hasil pembobotan TF-IDF adalah: mui (0.6436902527847653), vaksin (0.132185733888140), covid (0.1566272932497384), sinovac (0.4762729721904365), suci (0.8634345960912986), halal (0.5720637913580648), dan ncovid (0.543713657254659). Hasil penelitian ini masih memerlukan pembobotan n-gram dengan L1 atau L2 Normalization agar dapat digunakan sebagai *data train* dan *data test* pada proses analisa selanjutnya.

**Kata kunci:** covid-19, halal, isu, mining, text, tf-idf, twitter, vaksin.

### 1. PENDAHULUAN

Isu mengenai vaksin covid-19 mulai muncul dan ramai dibicarakan semenjak pemerintah mengeluarkan kebijakan wajib vaksin bagi warga yang sehat. Berbagai aturan dan persyaratan diterapkan agar masyarakat sadar akan pentingnya vaksin ini, menyebabkan berbagai polemik dan komentar bermunculan di media sosial. Begitupun di media sosial *Twitter*, banyaknya komentar seputar isu vaksin dari pengguna *Twitter* melalui *tweet*-nya, merupakan alasan utama penelitian ini dilakukan. Isu mengenai vaksin yang paling banyak beredar adalah mengenai informasi halal atau tidak vaksin digunakan.

Berbagai macam penelitian mengenai analisis sentimen publik telah banyak dilakukan, namun objek penelitiannya berbeda-beda. Seiring dengan maraknya pengguna *Twitter* yang mencakup semua kalangan usia, membuat media sosial ini menjadi *trend* objek penelitian.

Pengambilan dataset untuk penelitian ini mengunduh data *tweet* yang tersedia di situs [www.kaggle.com](http://www.kaggle.com). Yakni suatu situs yang membantu para peneliti untuk menemukan dataset sebagai sampel penelitian mereka, terutama penelitian yang memerlukan tambang data yang besar seperti *Data Mining* dan *Machine Learning*.

Dataset yang didapat berupa file csv, sehingga memudahkan pada saat pengolahan data. Data *tweet* yang didapat berupa data mentah yang masih acak sehingga membutuhkan *preprocessing* teks terlebih dahulu sebelum dilakukan pembobotan menggunakan metode TF-IDF.

*Twitter text mining* merupakan salah satu analisa sentimen dengan sumber data adalah *tweet* (cuitan) dari pengguna *Twitter*. *Tweet* ini berupa sekumpulan teks yang berisi komentar publik mengenai suatu kondisi tertentu, misalnya kebijakan pemerintah mengenai wajib vaksin covid-19 yang akan dianalisa pada penelitian ini. Kumpulan teks yang berisi komentar para pengguna *Twitter* akan diproses melalui *text preprocessing* agar dapat dilakukan pembobotan pada tiap kata untuk mengetahui seberapa sering suatu kata muncul.

### 2. TINJAUAN PUSTAKA

Menurut Kamus Besar Bahasa Indonesia (KBBI), sentimen memiliki arti pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu (bertentangan dengan pertimbangan pikiran).<sup>1</sup>

#### 2.1. Analisis Sentimen

Merupakan proses penggunaan *text analytics* untuk mendapatkan berbagai sumber data dari internet dan media sosial. Tujuannya untuk memperoleh opini dari pengguna yang terdapat pada *platform* tersebut.<sup>2</sup>

Sedangkan menurut Aliandu, analisis sentimen adalah riset komputasional opini, sentimen, dan emosi yang diekspresikan secara tekstual.<sup>3</sup>

#### 2.2. Twitter

Berdasarkan pada sebuah data yang dirilis suatu situs *Semiocast Dot Com* pada 1 Juli 2012 jumlah *tweet* di Indonesia mencapai 29,5 juta orang.<sup>4</sup> Jumlah tersebut menempati posisi kelima dunia. Sedangkan

data yang dirilis situs *A World of Tweets Dot Com* menempatkan Indonesia sebagai negara ketiga terbanyak di dunia yang menulis *tweet* (kicauan), yakni sebanyak 11,39% diperoleh berdasarkan rekaman total jumlah *tweet* seluruh dunia sejak November 2010 dari 383 juta profil pengguna *Twitter* yang dibuat sebelum 1 Januari 2012.<sup>5</sup>

Jumlah pengguna *Twitter* di Indonesia merupakan pangsa pasar yang menjanjikan. Maka tidak heran berbagai produsen mulai kelas kecil hingga besar berlomba - lomba mengelola potensi ekonomi besar ini agar produk-produk mereka laku di pasaran sekurang-kurangnya menjadi referensi.<sup>6</sup>

### 2.3. Text Mining

Menurut Harlian, *Text Mining* memiliki definisi menambang data yang berupa teks, dimana sumber data biasanya didapatkan dari sebuah dokumen dan tujuannya untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.<sup>7</sup>

*Text Mining* memiliki sifat yang sama seperti *Data Mining*, tetapi lebih fokus pada pengolahan teks daripada data yang terstruktur. Analisa *text mining* ini merupakan langkah pertama yang dilakukan dalam penambangan teks, yaitu mengatur dan menyusun data dengan cara tertentu sehingga dapat menjadi sasaran analisis kualitatif dan kuantitatif.<sup>8</sup>

*Text Preprocessing* adalah langkah awal dalam analisa *text mining*. Dalam proses ini, tahapan yang dilakukan adalah untuk mempersiapkan agar teks menjadi lebih terstruktur, sehingga dapat digunakan sebagai sumber data baru untuk proses analisa berikutnya. Tahapan *text processing* pada penelitian ini ada 3 proses, yaitu : (1). *Case Folding*, (2) *Tokenizing dan Filtering*, dan (3) *Stemming*.

Menurut Susilowati<sup>9</sup>, *Case folding* merupakan tahapan untuk melakukan penyamaan *case* dalam suatu dokumen. Hal ini dilakukan untuk mempermudah proses pencarian. *Tokenizing* bertugas untuk memisahkan kata pada suatu kalimat menjadi potongan kata tunggal serta untuk menghilangkan karakter tanda baca. *Filtering* atau lebih dikenal sebagai *cleansing* berfungsi menghapus beberapa simbol atau kata yang tidak dibutuhkan, misalnya *hashtag*, emoji, url, email dan kata kunci.<sup>10</sup> Sedangkan *stemming*, berfungsi menghapus awalan dan akhiran pada kata.

### 2.4. Term Frequency-Inverse Document Frequency (TF-IDF)

Merupakan salah satu proses dari teknik ekstraksi fitur dengan proses memberikan nilai pada masing - masing kata yang ada pada *data training*. Untuk mengetahui seberapa penting sebuah kata dalam mewakili suatu kalimat, akan diberi perhitungan. Pemberian nilai pada TF-IDF tergantung besarnya frekuensi kemunculan kata pada dokumen.<sup>11</sup>

Pada ekstraksi dengan fitur TF-IDF rumus yang digunakan untuk menghitung nilai (w) dari masing -

masing dokumen terhadap kata kunci dengan formula pada persamaan (1) dibawah ini.<sup>12</sup>

$$W_{dt} = TF_{dt} * IDF_{ft} \quad (1)$$

Keterangan :

$W_{dt}$  = nilai dokumen ke d pada kata ke t

$TF_{dt}$  = Jumlah kata yang dicari dalam suatu dokumen

$IDF_{ft}$  = Inverse Document Frequency  $(\log(\frac{N}{df}))$

N = Jumlah dokumen

Df = Jumlah dokumen yang mengandung kata yang dicari.

## 3. METODE PENELITIAN

Data penelitian diambil dari situs *www.kaggle.com*. Data memuat tentang *tweet* opini publik mengenai berbagai macam isu vaksinasi *covid-19*. Kebijakan pemerintah yang menganjurkan wajib vaksin menuai pro dan kontra. Di tahun 2021, pemerintah mulai melaksanakan vaksin tahap pertama. Awal pelaksanaan vaksin inilah, banyak bermunculan isu - isu mengenai vaksin, terutama komentar masyarakat di media sosial *Twitter*. Kemudian data *tweet* tersebut diunduh sebagai dataset dalam bentuk data teks. Data *tweet* kemudian diolah dan dirapikan melalui *preprocessing* teks.

*Preprocessing* teks bertujuan untuk merapikan data *tweet* yang masih mengandung kata-kata tidak penting, simbol, huruf, spasi ganda, dan lain-lain. Dan juga membagi kalimat menjadi potongan kata sehingga memudahkan pada saat pembobotan.

Penelitian ini menggunakan metode *Term Frequency, Inverse Document Frequency* (TF-IDF) metode TF-IDF merupakan suatu metode yang menggunakan cara pembobotan pada kata (*term*) yakni menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut. Semakin sering kata muncul dalam suatu dokumen menunjukkan seberapa pentingnya kata tersebut.

Analisa dilakukan dengan menggunakan bahasa pemrograman *Python* yang terdapat di dalam *Jupyter Notebook*. Dataset berupa teks mengenai kumpulan *tweet* isu vaksinasi dari pengguna *Twitter* dalam bentuk *file csv* akan diolah menjadi bentuk angka untuk kemudian dilakukan pembobotan. Sebelumnya data teks melalui proses *preprocessing* terlebih dahulu agar mudah mencari kata yang sering muncul.

## 4. HASIL DAN PEMBAHASAN

### 4.1. Pengambilan data

Data *tweet* diperoleh dari situs *www.kaggle.com*. Data yang berupa teks dari kumpulan *tweet* para pengguna *Twitter* berisi komentar - komentar mereka tentang vaksin *covid-19* disimpan dalam database *vaksin\_indo.csv* untuk memudahkan saat pengolahan data. Data *tweet* sebanyak 36.117 akan diolah dengan menggunakan *Jupyter Notebook*. Data *tweet* dirapikan

kolom-kolomnya sehingga didapat tampilan seperti gambar berikut ini :

```
In [13]: vaksin Indo[['date', 'id', 'text', 'user_location']]
Out[13]:
```

	date	id	text	user_location
0	2021-01-10 15:13:42	134028990352776995	#vaksin untuk #Indonesia https://t.co/URgUjRckUn	Bekasi, Jawa Barat
1	2021-01-10 16:14:21	1340302160240340401	Tak Kenal maka Tak Kebal. Vaksin menjadi bukti...	Kota Bogor - Jawa Barat
2	2021-01-10 16:33:46	1340307055040340160	Sistem satu data mendukung Vaksinasi di indone...	Nah!
3	2021-01-10 17:35:47	134032266025969377	https://t.co/TeId94jhhm\nsiap menerima vaksin ...	Nah!
4	2021-01-11 00:24:53	1340425614584268753	Apa Perbedaan Vaksin China Sinovac dan Sinopha...	Indonesia
...	...	...	...	...
13486	2021-04-20 15:42:48	138453301356974856	Wah @Bani_inamas dan @at_inamas hebat juga.	Tagar Timur, Indonesia
13487	2021-04-20 19:00:00	1384562641664544709	Kunci Pemulihan Ekonomi adalah Keterlaksanaan Va...	Jakarta, Indonesia
13488	2021-04-20 20:37:37	1384607207594595201	#bandarjalongline #kocorantogel #rangkaiju #o...	Nah!
13489	2021-04-20 20:49:53	1384609517680231363	Kepada seluruh wanita Indonesia #triSELAMAT HA...	Kota Surabaya, Jawa Timur
13490	2021-04-20 23:36:34	1384632330701086209	Mungkin ini informasi penting untuk diketahui.	Kota Tangerang, Banten

13491 rows x 4 columns

Gambar 1. Tampilan data tweet

#### 4.2. Text Preprocessing

Selanjutnya data tweet akan diproses melalui *text preprocessing* untuk kemudian dilakukan proses pembobotan. Data tweet yang berupa data teks, diproses untuk menemukan kata (*term*) yang dianggap penting karena seringnya muncul di setiap tweet yang ada. Berikut ini tahapan proses *text preprocessing*:

##### 1. Case Folding

Data teks yang mengandung huruf kapital (besar) akan dijadikan huruf kecil semua melalui proses *case folding*. Sebab, dalam bahasa pemrograman Python mengenal *case sensitive*. Sehingga mempengaruhi hasil jika dilakukan analisa nantinya. Hasil dari proses *case folding* dapat dilihat pada gambar dibawah ini :

Case Folding Result :

```
0 #vaksin untuk #Indonesia https://t.co/urgujrckun
1 tak kenal maka tak kebal. vaksin menjadi bukti...
2 sistem satu data mendukung vaksinasi di indone...
3 https://t.co/TeId94jhhm\nsiap menerima vaksin ...
4 apa perbedaan vaksin china sinovac dan sinopha...
5 dua lagi obat yang dapat menyelamatkan nyawa pasi...
6 vaksin buatan tiionggkok kembali dipertanyakan, ...
7 kabar duka, dokter pribadi paus fransiskus men...
8 masker bekas rawan tertular virus. pakai maske...
9 27 wakil supervisor kab/kota sudah terlatih, 6...
10 mengenal vaksin covid-19 !!\n#vaksin\n#vaksinc...
11 top news koran rakyat merdeka\n\njelang imlek ...
12 lawan #covid19 dengan #solidaritas bukan stigm...
13 vaksinasi covid-19 segera dimulai, tetap jaga ...
14 sampurasun wargi jabar 🙏\n.\nyuk wargi kita li...
15 aceh salah satu provinsi yang menerapkan syari...
16 mari siapkan diri untuk vaksinasi saat vaksin ...
17 vaksinya halal yaa\n\n#vaksin\n#pandemi \n#ind...
18 menjadi orang pertama yang di suntik\n\n#vaksi...
19 lebih baik mencegah toh\n\n#vaksin\n#pandemi \...
20 sudah mendapat status halal yaa\n\n#vaksin\n#p...
21 simak yang boleh dan tidak\n\n#vaksin\n#pandem...
22 video tenaga kesehatan tolak vaksin covid-19 v...
23 jadi, #sobatbp sudah terima sms dari kemenkes ...
24 top news koran rakyat merdeka\n\nndpr selesai r...
Name: Isi Tweet, dtype: object
```

Gambar 2. Tampilan hasil proses Case Folding

##### 2. Tokenizing dan Filtering

Data tweet yang berupa teks tidak beraturan dipotong untuk memisahkan kalimat menjadi kata, sebelumnya dilakukan juga proses penghapusan karakter tanda baca, hastag, simbol, dan lain-lain.

Adapun tahapan pada proses *tokenizing* antara lain:

- a) Proses penghapusan karakter spesial
- b) Menghilangkan angka
- c) Menghilangkan tanda baca (*punctuation*)
- d) Menghilangkan spasi di awal dan diakhir
- e) Menghilangkan beberapa spasi menjadi spasi tunggal
- f) Menghilangkan karakter tunggal
- g) *Stopword Removal*

Pada proses *stopword removal* ini dilakukan proses *stopword* menggunakan bahasa Indonesia yang didapatkan dari library NLTK untuk *filtering dataframe*.

Dan berikut ini hasil dari proses *tokenizing* dapat dilihat pada gambar dibawah ini:

Tokenizing Result:

```
0 [untuk]
1 [Tak, Kenal, maka, Tak, Kebal, Vaksin, menjadi...]
2 [Sistem, satu, data, mendukung, Vaksinasi, di,...]
3 [Siap, menerima, vaksin, covid]
4 [Apa, Perbedaan, Vaksin, China, Sinovac, dan, ...]
5 [Dua, Lagi, Obat, yang, Dapat, Selamatkan, Nya...]
6 [Vaksin, buatan, Tiongkok, kembali, dipertanya...]
7 [Kabar, Duka, Dokter, Pribadi, Paus, Fransisku...]
8 [Masker, bekas, rawan, tertular, virus, Pakai,...]
9 [wakil, supervisor, kabkota, sudah, terlatih, ...]
10 [Menegal, Vaksin, Covid]
11 [Top, News, Koran, Rakyat, Merdeka, Jelang, Im...]
12 [Lawan, dengan, bukan, stigma]
13 [Vaksinasi, Covid, segera, dimulai, tetap, jag...]
14 [Sampurasun, wargi, jabar, Yuk, wargi, kita, l...]
15 [Aceh, Salah, Satu, Provinsi, Yang, Menerapkan...]
16 [Mari, siapkan, diri, untuk, vaksinasi, saat, ...]
17 [Vaksinya, halal, yaa]
18 [Menjadi, orang, pertama, yang, di, suntik]
19 [Lebih, baik, mencegah, toh]
20 [Sudah, mendapat, status, halal, yaa]
21 [Simak, yang, boleh, dan, tidak]
22 [Video, Tenaga, Kesehatan, Tolak, Vaksin, Covi...]
23 [Jadi, sudah, terima, SMS, dari, Kemenkes, seb...]
24 [Top, News, Koran, Rakyat, Merdeka, DPR, Seles...
Name: tweet_tokens, dtype: object
```

Gambar 3. Data tweet hasil tokenizing

##### 3. Stemming

Selanjutnya akan dilakukan proses *stemming* untuk menghapus awalan dan akhiran pada kata sehingga menjadi kata dasar. Pada proses *stemming* ini akan menggunakan *library* sastrawi untuk mengembalikan kata ke bentuk dasarnya. Hasil dari *stemming* dapat dilihat pada gambar berikut ini :

```
menjadi : menjadi
berjuang : juang
menjaga : jaga
eksistensinya : eksistensi
dunia : dunia
men : men
Sistem : sistem
data : data
mendukung : dukung
Vaksinasi : vaksinasi
Indonesia : indonesia
Siap : siap
menerima : terima
vaksin : vaksin
covid : covid
Apa : apa
Perbedaan : beda
China : china
Sinovac : sinovac
Sinopharm : sinopharm
```

Gambar 4. Tampilan hasil proses stemming

Setelah di dapat kata dasar maka akan dilakukan pembobotan kata menggunakan TF-IDF untuk menghitung frekuensi kemunculan kata pada proses berikutnya.

### 4.3. Pembobotan TF-IDF

Pada proses ini akan memproses teks ke dalam sebuah matriks berisi numerik. Pengubahan kata ke dalam bentuk vektor juga harus memperhatikan apakah kata tersebut informatif atau tidak. Pada TF-IDF, kata yang sering muncul memiliki nilai yang cenderung kecil sedangkan kata yang jarang muncul memiliki nilai yang cenderung besar. Tahapan prosesnya sebagai berikut:

- 1) Mempersiapkan data *tweet* yang akan diolah.

Data *tweet* yang akan diolah, diambil hanya kolom *tweet* saja. Setelah proses *text preprocessing* sebelumnya, membaca file tersebut dalam Pandas lalu *rename* kolom *tweet\_token\_stemmed* menjadi *tweet*. Sehingga didapat hasil tampilan sebagai berikut:

	tweet
0	[]
1	['kenal', 'kebal', 'vaksin', 'bukti', 'manusia...']
2	['sistem', 'data', 'dukung', 'vaksinasi', 'ind...']
3	['terima', 'vaksin', 'covid']
4	['beda', 'vaksin', 'china', 'sinovac', 'sinoph...']
5	['obat', 'selamat', 'nyawa', 'pasien', 'covid'...]
6	['vaksin', 'buat', 'tiionggkok', 'orang', 'tua',...]
7	['kabar', 'duka', 'dokter', 'pribadi', 'paus',...]
8	['masker', 'bekas', 'rawan', 'tular', 'virus',...]
9	['wakil', 'supervisor', 'kabkota', 'latih', 'r...']
10	['kenal', 'vaksin', 'covid']
11	['top', 'news', 'koran', 'rakyat', 'merdeka', ...]
12	['lawan', 'stigma']
13	['vaksinasi', 'covid', 'jaga', 'protokol', 'se...']
14	['sampurasun', 'wargi', 'jabar', 'yuk', 'wargi...']
15	['aceh', 'salah', 'provinsi', 'terap', 'syaria...']
16	['mari', 'siap', 'vaksinasi', 'vaksin', 'laksa...']

Gambar 5. Data teks yang akan dibobot

Data tweet yang masih berupa teks akan ditransform menjadi bentuk numerik dalam sebuah matriks melalui pembobotan TF-IDF. Berikut ini cara menghitung TF dan IDF suatu *corpus*.

- 2) Menghitung TF

*Term Frequency* (TF) menghitung frekuensi kemunculan kata pada sebuah dokumen. *Term frequency* merupakan frekuensi kemunculan *term* *i* pada dokumen *j* dibagi dengan total *term* pada dokumen *j* maksudnya, karena panjang dari setiap dokumen berbeda-beda, maka umumnya nilai TF ini dibagi dengan panjang dokumen (jumlah seluruh kata pada dokumen).

Berikut ini baris perintah dan hasil perhitungan TF (*Term Frequency*):

```
def calc_TF(document):
    # Counts the number of times the word appears in review
    TF_dict = {}
    for term in document:
        if term in TF_dict:
            TF_dict[term] += 1
        else:
            TF_dict[term] = 1
    # Computes tf for each word
    for term in TF_dict:
        TF_dict[term] = TF_dict[term] / len(document)
    return TF_dict
```

term	TF
mui	0.14285714285714285
vaksin	0.14285714285714285
covid	0.14285714285714285
sinovac	0.14285714285714285
suci	0.14285714285714285
halal	0.14285714285714285
ncovid	0.14285714285714285

Gambar 6. Tampilan proses dan hasil hitung TF

- 3) Menghitung IDF

IDF (*Inverse Document Frequency*) memiliki fungsi untuk mengurangi bobot suatu *term* jika kemunculannya banyak tersebar di seluruh dokumen.

Berikut ini implementasinya dalam bahasa Python:

```
def calc_DF(tfDict):
    count_DF = {}
    # Run through each document's tf dictionary and increment countDict's (term, doc) pair
    for document in tfDict:
        for term in document:
            if term in count_DF:
                count_DF[term] += 1
            else:
                count_DF[term] = 1
    return count_DF

DF = calc_DF(TWEET_DATA["TF_dict"])
n_document = len(TWEET_DATA)

def calc_IDF(__n_document, __DF):
    IDF_Dict = {}
    for term in __DF:
        IDF_Dict[term] = np.log(__n_document / (__DF[term] + 1))
    return IDF_Dict
```

Gambar 7. Tampilan perintah hitung IDF

Kemudian dilakukan perkalian TF dan IDF Serta tampilan hasil proses perhitungan IDF pada gambar dibawah ini:

term	TF	TF-IDF
mui	0.14285714285714285	0.6436902527847653
vaksin	0.14285714285714285	0.13218573388814045
covid	0.14285714285714285	0.1566272932497384
sinovac	0.14285714285714285	0.4762279721904365
suci	0.14285714285714285	0.8634345960912986
halal	0.14285714285714285	0.5720637913580648
ncovid	0.14285714285714285	0.5437136572546594

Gambar 8. Tampilan hasil hitung TF-IDF

- 4) Membuat *sparse matrix*

Data series TF-IDF yang didapat akan dirubah menjadi vektor data berbentuk *sparse matrix* dengan dimensi (*n\_sample*, *n\_feature*), *sparse matrix* yang ditampilkan untuk sample ke 0 dengan *n\_feature* = 50. Hasilnya setelah dijalankan tampak seperti gambar berikut ini :



- [3] Aliandu, Paulina. "Analisis sentimen tweet berbahasa indonesia di twitter". Tesis. Fakultas MIPA. Pasca Sarjana Ilmu Komputer, Universitas Gajah Mada. Yogyakarta.2012.
- [4] Campagne, J.C., Dux, J., Guyot, P. dan Julien, D." *Twitter reaches half a billion account more than 140 million in the U.S*". diakses tanggal 16 Oktober 2012, dari semiocast: [http://semiocast.com/publications/2012\\_07\\_30\\_Twitter\\_reaches\\_half\\_a\\_billion\\_accounts\\_140m\\_in\\_the\\_US](http://semiocast.com/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US).
- [5] Markdalen, A dan Zapponi, C. "*Top 20 Countries Chart*". diakses tanggal 17 September 2012, dari *awordlof*: [http://awordlof\\_tweets\\_frogdesign.com](http://awordlof_tweets_frogdesign.com).
- [6] Putranti, D.V., "Analisis sentimen twitter untuk teks berbahasa Indonesia dengan *maximum entropy* dan *support vector machine*". Indonesia Journal of Computing and Cybernetics System (IJCCS), vol.8.no.1, Januari, pp.91-100. 2014. ISSN 1978-1520.
- [7] Binus, "Apa Itu Text Mining", diakses tanggal 23 Mei 2022, dari Binus: <https://sis.binus.ac.id/2021/04/23/apa-itu-text-mining/>.
- [8] Dqlab, "Salah Satu Contoh Teknik Analisis Data dengan Mengenal Text Mining", diakses pada tanggal 26 Mei 2022, dari dqlab: <https://dqlab.id/salah-satu-contoh-teknik-analisis-data-dengan-mengenal-text-mining>.
- [9] Susilowati, E., Sabariah, M.K & Gozali, A.A., "Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas pada Twitter". E-Proceeding of Engineering, 2(1),1-7. 2015.
- [10] Aditya, B.R. "Penggunaan Web Crawler untuk Menghimpun Tweets dengan Metode Pre-processing Text Mining". Jurnal Infotel, 7(2), 92-100. 2015.
- [11] Pravina, A.M., Cholissodin, L., Adikara, P.P. "Analisa Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.e-ISSN, 2548,964X.2019.
- [12] Melita, R., Amrizal, V., Suseno, H.B., & Dirjam, T. "Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Study Kasus: Syarah Umdatil Ahkam)". Jurnal Teknik Informatika (JTI) UIN Syarif Hidayatullah,11(2), Hal.149-164.2018.