Jurnal Linguistik Komputasional
Indonesian Association for Computational Linguistics
JLK
Vol 5, No 1
Maret 2022

Pengembangan Perangkat *Microservices* untuk Analisis Media Sosial sebagai Pendukung Pelacakan Penyebaran *Tuberculosis*

Ronaldo Cristover Octavianus, Dzikri Robbi, Laras Ervintyana, Hapnes Toba

Program Studi Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Kristen Maranatha
Jl. Suria Sumantri No. 65 Bandung 40164
{2079006, 2079008, 2079007}@maranatha.ac.id, hapnestoba@it.maranatha.edu

***Abstrak*— Tuberculosis (TBC) adalah suatu penyakit menular yang disebabkan oleh kuman *Mycobacterium Tuberculosis*. Penyakit ini sangat berbahaya dan perlu ditangani segera jika ada yang terindikasi tertular, namun sangatlah sulit untuk melakukan pelacakan bagi mereka yang memiliki gejala karena pada awalnya sangat mirip dengan gejala batuk biasa. Riset yang dipaparkan dalam makalah ini bertujuan untuk mengumpulkan data dari media sosial yang terkait dengan penyakit TBC tersebut. Data media sosial, khususnya Twitter, akan diproses nilai sentimennya. Di samping itu, dikembangkan pula perangkat berbasis *microservices* guna melakukan ekstraksi data Twitter dan pengolahan *dashboard* pemberi informasi untuk wilayah yang disinyalir terdapat penderita TBC. Hasil penelitian menunjukkan bahwa tingkat akurasi sentimen perlu untuk ditingkatkan dengan model yang dilengkapi dengan kata-kata non-formal (*slang*) dan nuansa bahasa daerah. Namun demikian, infrastruktur pengolah data dengan berbasis pada *microservices* telah berhasil dikembangkan dengan baik, dan dapat digunakan pada aplikasi lain yang sejenis.**

***Kata kunci*— Analisis Sentimen, *Dashboard*, Media Sosial, *Microservices*, Pemrosesan Teks, Twitter**

I. PENDAHULUAN

Tuberculosis atau dikenal juga dengan TBC, adalah adalah suatu penyakit menular yang disebabkan oleh kuman *Mycobacterium Tuberculosis* [1]. Penyakit ini termasuk salah satu penyakit yang mematikan dan perlu diobati dalam jangka panjang. Berdasarkan data WHO pada tahun 2015, Indonesia memiliki jumlah penderita TBC terbanyak di dunia dengan jumlah populasi sebanyak 10,4 juta jiwa.

Salah satu cara untuk mengurangi dan mencegah penyebaran TBC adalah dengan melakukan *tracing*. Dengan melakukan *tracing*, penyebaran TBC diharapkan dapat diketahui lebih dini dan lebih cepat untuk ditanggulangi. Saat ini, sudah terdapat sistem *tracing* penyakit TBC bernama SITB (Sistem Informasi *Tuberculosis*) yang merupakan aplikasi yang digunakan oleh semua pemangku kepentingan mulai dari Fasilitas Pelayanan Kesehatan, Dinas Kesehatan, Kementrian Kesehatan untuk melakukan pencatatan dan pelaporan kasus TBC [2]. Tetapi sistem ini baru dapat melakukan *tracing* jika pasien sudah positif TBC, dan melakukan pengobatan di fasilitas kesehatan. Oleh karena itu, untuk dapat melakukan mendeteksi potensi penyebaran TBC yang lebih cepat maka diperlukan adanya mekanisme yang dapat memberikan signal deteksi dini penyebaran TBC di suatu wilayah di Indonesia.

Media sosial saat ini telah menjadi bagian hidup masyarakat Indonesia, dengan jumlah pengguna media sosial di Indonesia telah melebihi 50% dari jumlah keseluruhan total penduduk. Penggunaan media sosial sendiri memberikan banyak manfaat, salah satunya adalah penyebaran informasi yang cepat dan mudah diakses. Salah satu media sosial yang paling berpengaruh dalam penyebaran informasi adalah Twitter [3].

Beranjak dari situasi di atas, penelitian ini bertujuan menghasilkan sebuah sistem yang fleksibel untuk memperlihatkan penyebaran penyakit TBC melalui informasi lokasi dan prediksi sentimen pada media sosial Twitter. Dengan memanfaatkan pendekatan ini diharapkan dapat membantu *tracing* penyebaran penyakit TBC di Indonesia.

II. METODE

A. Perangkat Penelitian

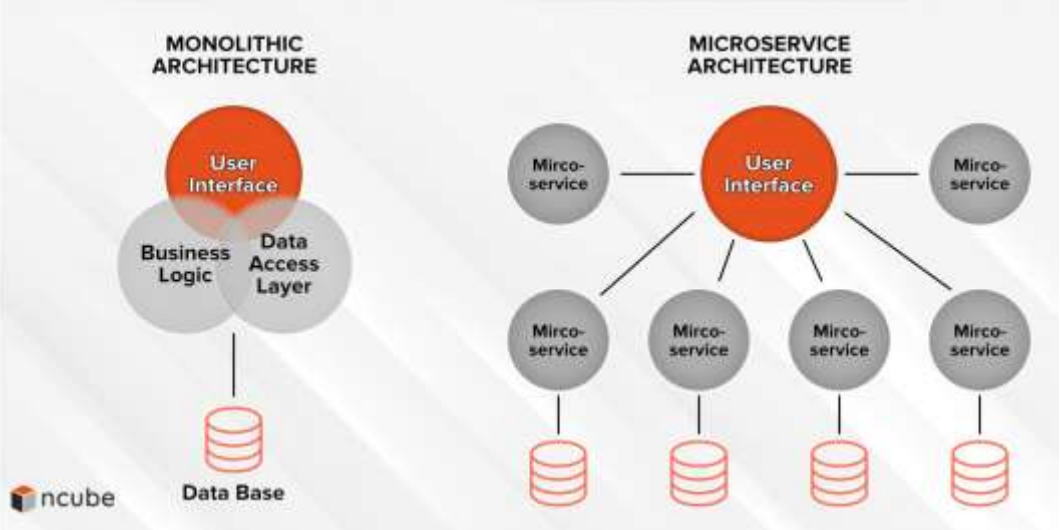
1) *Text Mining: Text mining* adalah bidang pengetahuan yang mencakup area *information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning*, dan *data mining* [4]. Dalam beberapa tahun terakhir, obyek dari *text mining* yang banyak diteliti adalah konten halaman web yang memiliki banyak konten dokumen teks yang dapat diolah lebih lanjut. Tahap pertama dari *text mining* adalah *text processing*. *Text processing* mengolah data tidak terstruktur menjadi data terstruktur. Keluaran dari *text preprocessing* digunakan sebagai masukan ke dalam berbagai algoritma *text mining*. Luaran dari algoritma *text mining*, misalnya hasil analisis

sentimen, dapat menjadi sumber pengetahuan dan digunakan sebagai pendukung pengambilan keputusan.

2) *Pemrosesan Teks:* Tahapan dalam *text mining* dimulai dengan *text preprocessing*. *Text preprocessing* menyiapkan data teks menjadi kata/token yang siap diolah lebih lanjut. *Text preprocessing* berpengaruh terhadap keberhasilan algoritma *text mining* yang digunakan [4]. Beberapa proses yang dilakukan dalam *text preprocessing* adalah tokenisasi, menghilangkan *stop word*, *stemming* dan lematisasi [4]. Pada tahapan *preprocessing* media Twitter, semua kata yang ada dalam sebuah cuitan diubah menjadi *lowercase* dan menghilangkan beberapa konten yang dianggap tidak penting, seperti [5]: *space pattern, URL, Twitter mentions, retweet Symbols*, dan *stop words.*

3) *Term Frequency - Inverse Document Frequency (TF-IDF):* Metode pembobotan TF-IDF biasanya digunakan dalam *text mining*. TF-IDF merupakan metode pembobotan secara statistik yang menunjukkan seberapa pentingnya sebuah kata pada suatu koleksi dokumen. *Term frequency* adalah jumlah sebuah kata pada dokumen, sedangkan *inverse document frequency* adalah nilai yang digunakan untuk mengukur seberapa penting sebuah kata pada sebuah koleksi dokumen [4]. Dalam suatu bahasa, terdapat kata-kata yang tidak memiliki makna penting (*stop words*), contohnya dalam bahasa Inggris adalah *the* dan *of*, kata-kata tersebut harus dihilangkan dengan cara menentukan *threshold* yang dipakai dalam TF-IDF sehingga terdapat filter awal terlebih dahulu sebelum kata-kata tersebut mulai untuk diproses [6].

4) *Microservices*: *Microservices* atau layanan mikro adalah suatu pendekatan arsitektur berbasis awan, dimana satu aplikasi terdiri dari banyak komponen atau layanan kecil yang digabungkan secara independen dan dapat digunakan secara mandiri. Layanan ini biasanya memiliki *stack* teknologi yang berbeda-beda termasuk dengan model database dan juga manajemen data. *Microservices* ini juga berkomunikasi satu dengan yang lainnya melalui *Application Programming Interface* (API) atau *Event Stream* atau *Message Broker* dan lainnya. *Microservices* itu sendiri diatur sesuai dengan kebutuhan aplikasi dengan layanan-layanan yang ada dibagi dan dipetakan sesuai dengan kebutuhan. Dengan menggunakan *microservices*, pengelolaan aplikasi menjadi lebih mudah. Setiap layanan tidak memiliki hubungan satu dengan lain dan terpisah (*loose coupled*) sehingga sangat memungkinkan untuk melakukan *scaling* secara mandiri tanpa mengganggu layanan lain. Secara umum arsitektur *microservices* digambarkan pada Gambar 1 [7].

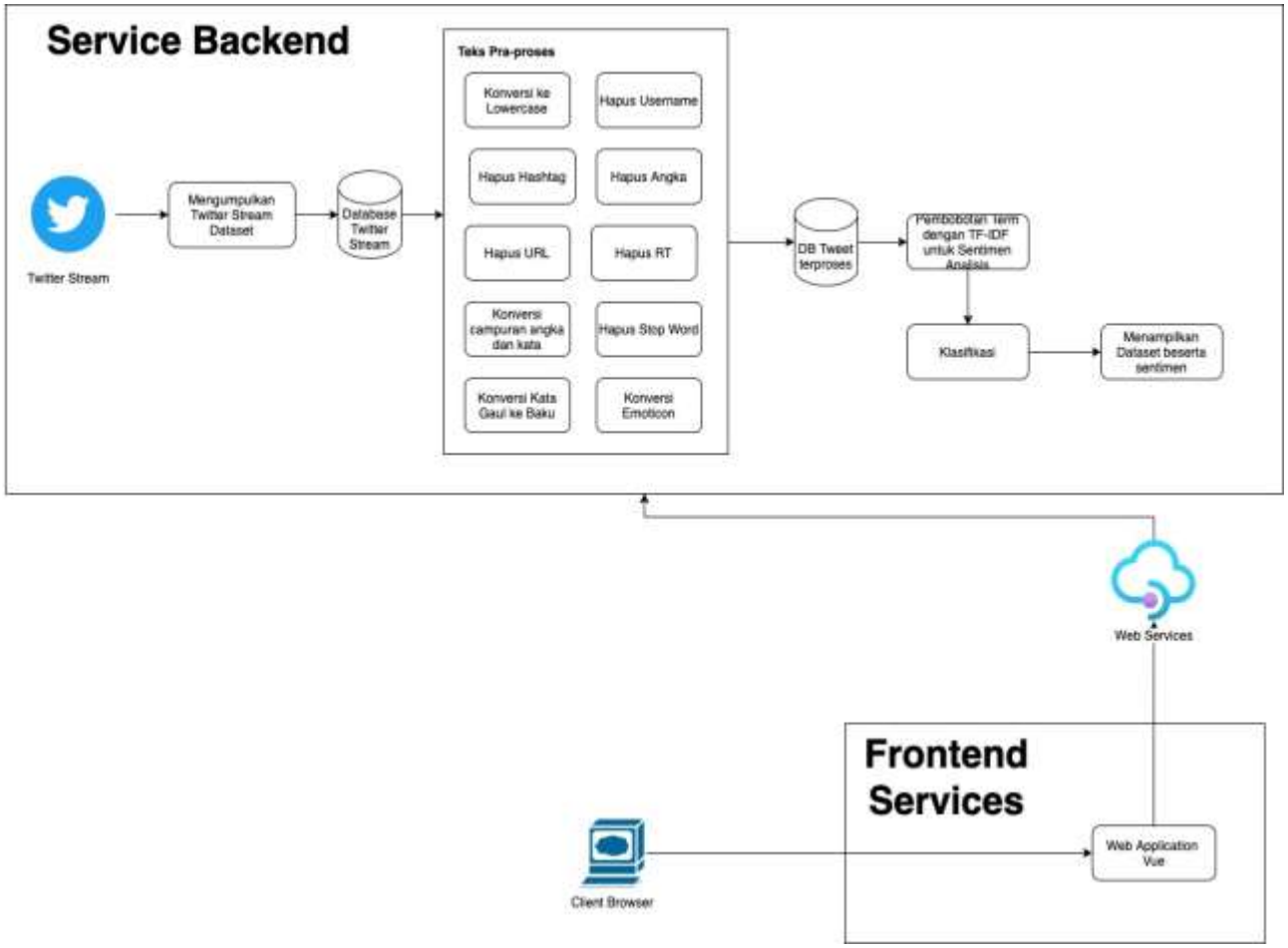


Gambar 1. Perbedaan konsep arsitektur monolitik dan layanan mikro (*microservices*), dengan interaksi berbagai komponen secara independen dalam sebuah aplikasi. [7].

B. Pendekatan dan Metode Kerja

1) Konsep aplikasi: Aplikasi yang dikembangkan dalam riset ini merupakan aplikasi yang memiliki beberapa proses seperti proses melakukan *stream* data secara *real time*, proses untuk melakukan perhitungan dan kalkulasi terhadap data tekstual yang ada, dan proses untuk menampilkan data agar pengguna bisa melihat ringkasan informasi secara visual. Dalam tiga bagian besar proses ini diperlukan adanya suatu penggambaran model yang dapat membantu pengguna dan pengembangan aplikasi ke depannya.

2) Diagram Perancangan: Aplikasi ini dibuat dengan menggunakan arsitektur *microservices*. Setiap *service* atau layanan dipecah sesuai dengan tujuannya masing-masing. Alasan dibuat arsitektur tersebut adalah:



Gambar 2. Arsitektur aplikasi berbasis *micorservices*

- Mempermudah pengembangan aplikasi karena perbedaan bahasa pemrograman yang digunakan untuk setiap proses;
- Mempermudah pengembangan aplikasi karena setiap *service* terpaut namun dalam prosesnya bisa berjalan masing-masing secara mandiri;
- Dapat menerima *request* dengan jumlah yang lebih banyak karena fungsi yang dipanggil bersifat mikro dan tidak memakan banyak *resource* karena *package*, *library* dan *server* dapat menampung *request* yang cukup tinggi.
- Dalam proses kedepannya, jumlah data akan semakin tinggi oleh karena itu proses pengolahan data akan semakin tinggi juga. Dengan menggunakan arsitektur ini, proses *scaling* service dapat dilakukan pada *service* yang memiliki *load* tinggi saja dan menjadi lebih hemat *resource*.

Arsitektur aplikasi yang dikembangkan dalam riset ini dapat dilihat dalam Gambar 2. Pada Gambar 2 tersebut dapat dilihat bahwa arsitektur dibuat menjadi 2 bagian yaitu: *service backend* dan *service frontend*.

- *Service Backend*

Service backend adalah layanan yang bertugas untuk melakukan pengolahan data di belakang layar. Setiap data yang masuk akan diolah dan nantinya akan dikembalikan apabila *service frontend* meminta suatu data. Pada *service* ini, dibuat menjadi beberapa *microservices* berikut ini:

a. Twitter Stream Service

Service ini berfungsi untuk melakukan *stream* API Twitter dan menyimpannya ke dalam basis data. API DataStream yang dikumpulkan memiliki format JSON. Dari *response stream* tersebut dipilih objek mana saja yang akan disimpan dan dipisah serta sisanya akan disimpan sebagai data mentah atau data acuan dalam suatu kolom.

Data yang diambil melalui *streaming* tentunya perlu untuk dilakukan *filtering* sehingga tidak semua data akan masuk. Hanya data-data yang berhubungan dengan *tuberculosis* saja. Dalam proses filtering *data stream*, diberikan kata-kata kunci sebagaimana ditampilkan dalam Tabel 1.

TABEL I
PRE-DEFINED KATA KUNCI TERKAIT TBC

Kata Kunci
Batuk
Tuberkulosis
Tuberculosis
TB
TB Paru
Batuk Berdarah

b. Preprocessing Service

Tahap *preprocessing* merupakan suatu proses untuk mempersiapkan data mentah sebelum dilakukan proses lainnya. Pada umumnya, praproses data dilakukan dengan cara menghilangkan data yang tidak sesuai dan mengubahnya menjadi bentuk yang mudah diproses oleh sistem. Praproses ini perlu dilakukan untuk melakukan analisis sentimen, khususnya pada media sosial yang sebagian besar berisi kata-kata atau kalimat yang tidak formal dan tidak terstruktur dan memiliki *noise* yang besar.

Terdapat tiga model praproses untuk kalimat atau teks dengan *noise* besar. Ketiga model tersebut adalah:

- *Orthographic Model*
  Model ini dipergunakan untuk memperbaiki kata atau kalimat yang memiliki kesalahan dari segi bentuk kata atau kalimat.
- *Error Model*
  Model ini dipergunakan untuk memperbaiki kesalahan dari segi kesalahan eja atau kesalahan penulisan. Ada dua jenis kesalahan yang dikoreksi dengan model ini yaitu kesalahan penulisan dan kesalahan eja. Kesalahan penulisan dapat mengacu pada kesalahan pengetikan sedangkan kesalahan eja muncul ketika penulis tidak mengetahui ejaan yang benar.
- *White Space Model*
  Model ini mengacu pada pengoreksian tanda baca. Contoh kesalahan dalam model ini adalah tidak menggunakan titik di akhir kalimat namun model ini tidak terlalu efektif untuk diterapkan pada media sosial yang tidak mengindahkan tanda baca.

Tahap praproses ini bisa disebut juga dengan proses ekstraksi data dimana data yang sebelumnya telah dikumpulkan dalam database akan diproses melalui proses ekstraksi. Dalam proses ini, akan dilakukan beberapa proses seperti:

- *Case Folding*, yaitu membuat semua teks menjadi huruf kecil
- *Remove Punctuation*, yaitu menghapus semua karakter non alfabet misalnya simbol, spasi dan lainnya.
- *Remove Username*, yaitu menghapus nama user yang diawali dengan simbol '@' karena dianggap tidak penting
- *Remove Hashtag*, yaitu menghapus karakter "#" yang biasanya dijadikan judul topik.
- *Clean Number*, yaitu menghapus angka yang ada didepan atau dibelakang suatu kata, seperti: jalan2, makan2, 22nya dan lainnya.
- *Clean One Character*, yaitu menghapus jika terdapat hanya satu huruf saja seperti g, a, f, dan lain-lain.
- *Remove URL*, yaitu menghapus *URL* yang terdapat pada twit atau kata.
- *Remove RT*, yaitu menghapus *prefix* RT yang menandakan bahwa teks tersebut merujuk pada suatu *tweet* dan *username*.
- *Convert Number*, yaitu melakukan proses perubahan number menjadi suatu kata khususnya pada kata-kata yang memiliki campuran seperti s4y4ng, b4tuk dimana bila diubah menjadi sayang dan batuk.
- *Remove Stop Word. Stop word* diproses pada sebuah kalimat jika mengandung kata-kata yang sering keluar dan dianggap tidak penting seperti waktu, penghubung, dan lain-lain.
- *Convert Word*, yaitu melakukan konversi dari kata-kata dengan bahasa *alay*/gaul menjadi bahasa baku, contoh sebagian kecilnya dapat dilihat dalam Tabel 2.

TABEL II
CUPLIKAN KAMUS KONVERSI KATA

Sebelum	Sesudah
Akyu	Aku
Akuwh	Aku
Akku	Aku
Aq	Aku
Aquwh	Aku
Awak	Aku
Amaca	Ah masa
Alluw	Hallo
Alo	Hallo
Atw	Atau

- *Convert Emotion*, yaitu merubah *emoticon* dalam Twitter menjadi suatu kata sentimen. Contoh konversi yang dilakukan dapat dilihat dalam Tabel 3.

TABEL III
CUPLIKAN KAMUS KONVERSI *EMOTICON*

Emoticon	**Konversi**	**Masuk ke dalam Kelas**
:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)	Senang	Positif
>:D :-D :D 8- D 8D x-D xD XD XD =-D =D =-3 =3	Tertawa	Positif
> :[:-(:(:-c :c :-< :< :-[:[:{ > .><> .< :'(	Sedih	Negatif
D :< D : D 8 D ; D = D X v.v D-':	Horror	Netral
> :P :-P :P X-P x-p xp XP :- p :p =p :-Þ :Þ	*Tongue*	Netral
>:o>:O :-	*Shock*	Positif

Emoticon	Konversi	Masuk ke dalam Kelas
O :O °o° °O° :O o_O o.O 8-0		
> :\ >:/ :-/ :-. :/ :\ =/ =\ :S	*Kesal*	Negatif
:| :-|	Ekspresi Datar	Negatif

c. Term and Classification Service

Service ini memiliki fungsi untuk melakukan perhitungan:

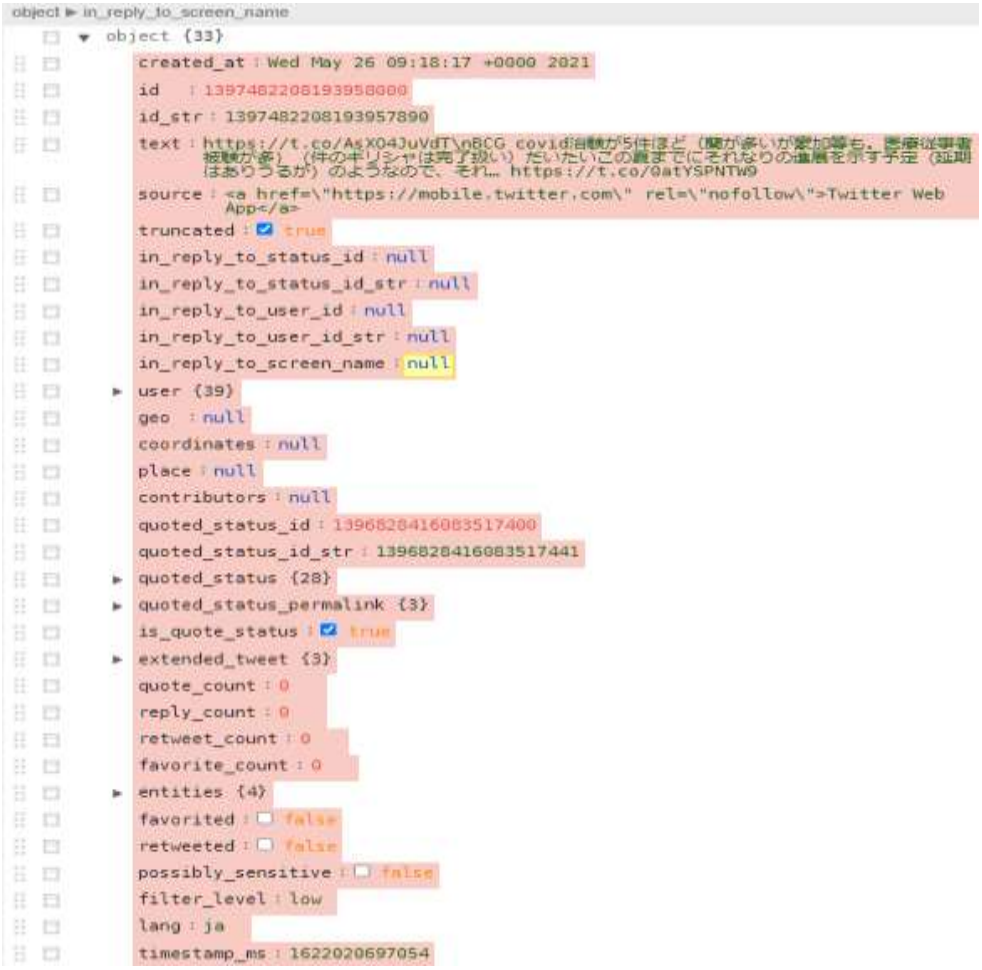
- Tokenisasi
  Pada proses ini, pada kalimat-kalimat yang sudah dibersihkan akan dilakukan tokenisasi dan indeksasi yang selanjutnya akan dilakukan proses pembobotan, TF-IDF.
- Proses TF-IDF
  Pembobotan kata dilakukan dengan mendapatkan *context of discussion* dari dataset Twitter. *Context of discussion* dapat digunakan untuk analisis tren percakapan yang terjadi di Twitter terkait dengan penyakit TBC.
- Proses Klasifikasi
  Dari hasil pengolahan seperti tokenisasi, pembobotan TF IDF maka dilakukan proses klasifikasi dimana setiap *tweet* akan dinilai dan diklasifikasi apakah *tweet* tersebut masuk kategori positif atau negatif.

d. API Gateway Services

Service ini digunakan sebagai penghubung antara *Frontend Services* dan *Backend Services*. Setiap *request* yang dikirimkan dari *frontend* akan diterima oleh API Gateway terlebih dahulu sebelum melakukan pengambilan data yang sudah diolah.

- *Service Frontend*

Service frontend adalah *service* yang bertemu langsung dengan pengguna. *Service* ini berbasis web dengan teknologi CSR (*Client Side Rendering*). *Service* ini akan melakukan *generate script* dan melakukan *render* pada *browser* pengguna, sehingga *load* di sisi *server* akan menjadi lebih rendah dan interaksi pengguna pada aplikasi menjadi lebih baik juga.



Gambar 3. Struktur JSON untuk pengelolaan *stream* data

C. Struktur JSON

Data yang ditarik dan di-*stream* dari API Twitter, perlu disimpan di dalam suatu basis data. Sebelum menyimpan data *stream*, perlu dilakukan analisis terhadap respons balikan dari API *Stream*. Untuk itulah diperlukan suatu struktur JSON pada Gambar 3. Dari struktur objek yang ada pada Gambar 3, dibuatlah suatu struktur tabel sebagaimana ditampilkan dalam Tabel 4.

TABEL IV
STRUKTUR TABEL UNTUK PENYIMPANAN HASIL *STREAM*

Nama Field	**Struktur Data**	***Keterangan***
idx	int(11) primary, auto_increment	*auto generated* id
id	varchar(255)	nomor otomatis
json	text	*response* asli dari twitter stream api
tweet_text	text	*tweet text*
tweet_text_clean	text	*tweet text* setelah dilakukan pra proses.
user_id	varchar(255)	id dari user
user_screen_name	varchar(255)	nama *screen twitter*
user_avatar_url	varchar(255)	gambar profil *user*
geo	varchar(255)	alamat geo lokasi
coordinates	varchar(255)	*coordinate user update*
places	varchar(255)	tempat
created_at	timestamp	tanggal *insert table*
updated_at	timestamp	tanggal *update tabel*
sentiment_analysis	varchar(25)	hasil dari perhitungan sentimen
lang	varchar(10)	bahasa aplikasi yang digunakan user
source	varchar(255)	perangkat yang digunakan *user* untuk melakukan *tweet*

III. HASIL DAN PEMBAHASAN

A. Twitter Stream Dataset

Proses pengumpulan data dilakukan oleh *service* yang dibangun menggunakan *framework* Laravel dengan melakukan *streaming* pada media sosial Twitter. Untuk dapat melakukan *streaming* pada media sosial Twitter tersebut, dibutuhkan API Key yang diperoleh secara resmi melalui laman pengembang aplikasi pada media sosial Twitter. Selain itu, untuk mendukung layanan *streaming* juga digunakan *library* Phirehose, yaitu sebuah *library* berbasis PHP yang dapat melakukan koneksi dan streaming data dari media sosial Twitter.

Service yang telah berhasil dibuat tersebut bekerja secara *real time* dan berkelanjutan selama periode pengumpulan data, yaitu antara tanggal 26 Mei 2021 hingga 6 Juni 2021. *Service* ini bekerja mengumpulkan data *tweet* yang mengandung kata kunci yang telah ditentukan.

Proses *streaming* ini menghasilkan data sebanyak 31.787 baris yang kemudian akan dilakukan praposes untuk membersihkan kolom *tweet_text* dari *stopwords* yang tidak relevan untuk kebutuhan proses analisis selanjutnya. Dalam proses *streaming*, terdapat data yang tidak konsisten dimana data ini menjadi salah satu data yang penting dalam pemetaan sentimen kedepannya. Dalam objek data yang didapatkan dari data *stream*, 90% dari data tidak memiliki informasi *geo-location* (dinonaktifkan). Untuk mengatasi masalah ini, apabila *tweet* tersebut tidak memiliki informasi *geo-location*, maka data yang disimpan di dalam basis data mengambil informasi berupa isian *user location* yang terdapat pada *user information* akun Twitter.

B. Pemrosesan Teks

Pemrosesan teks dilakukan pada data *stream* Twitter yang disimpan di dalam basis data. Pemrosesan secara *batch* dilakukan dalam kurun waktu tertentu dengan melakukan *service* pengambilan data Twitter yang ada pada baris dan kemudian melakukan proses: *case folding*, *remove punctuation*, *remove username*, *remove hashtag*, *clean number*, *clearn on character*, *remove URL*, *remove re-tweet*, *convert number*, *remove stop word*, *convert word*, dan *convert emotion*.

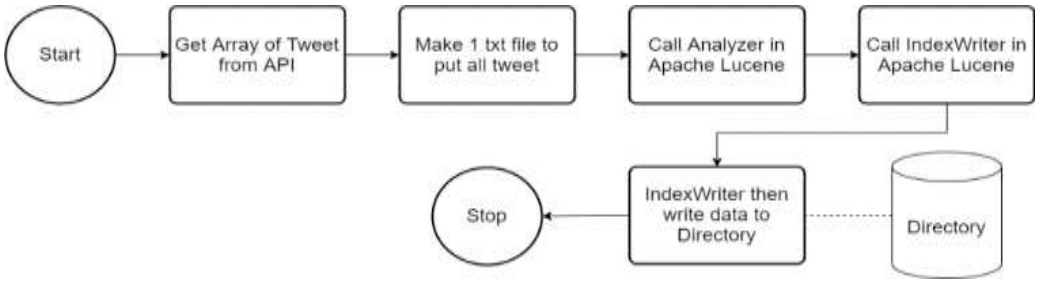
Dalam proses ini setiap kata akan dirubah sesuai dengan tahapan-tahapan sebagaimana disebutkan dalam paragraf sebelumnya. Di bawah ini adalah contoh perubahan tweet sebelum dan setelah pemrosesan teks.

Sebelum:

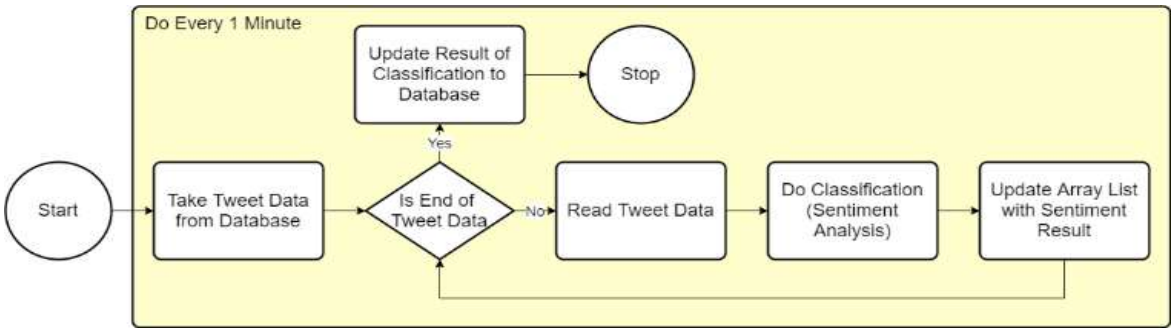
```
Batuk-batuk mulu ampun, deg-degan banget tapi hamdallah dah antigen 2x pun ya negatif huhu takut bgt :")


Ok susah nak batuk susah nak bersin sebab perut aku sakit. Sakit sangat ye muscle pain ni.
```

Setelah:

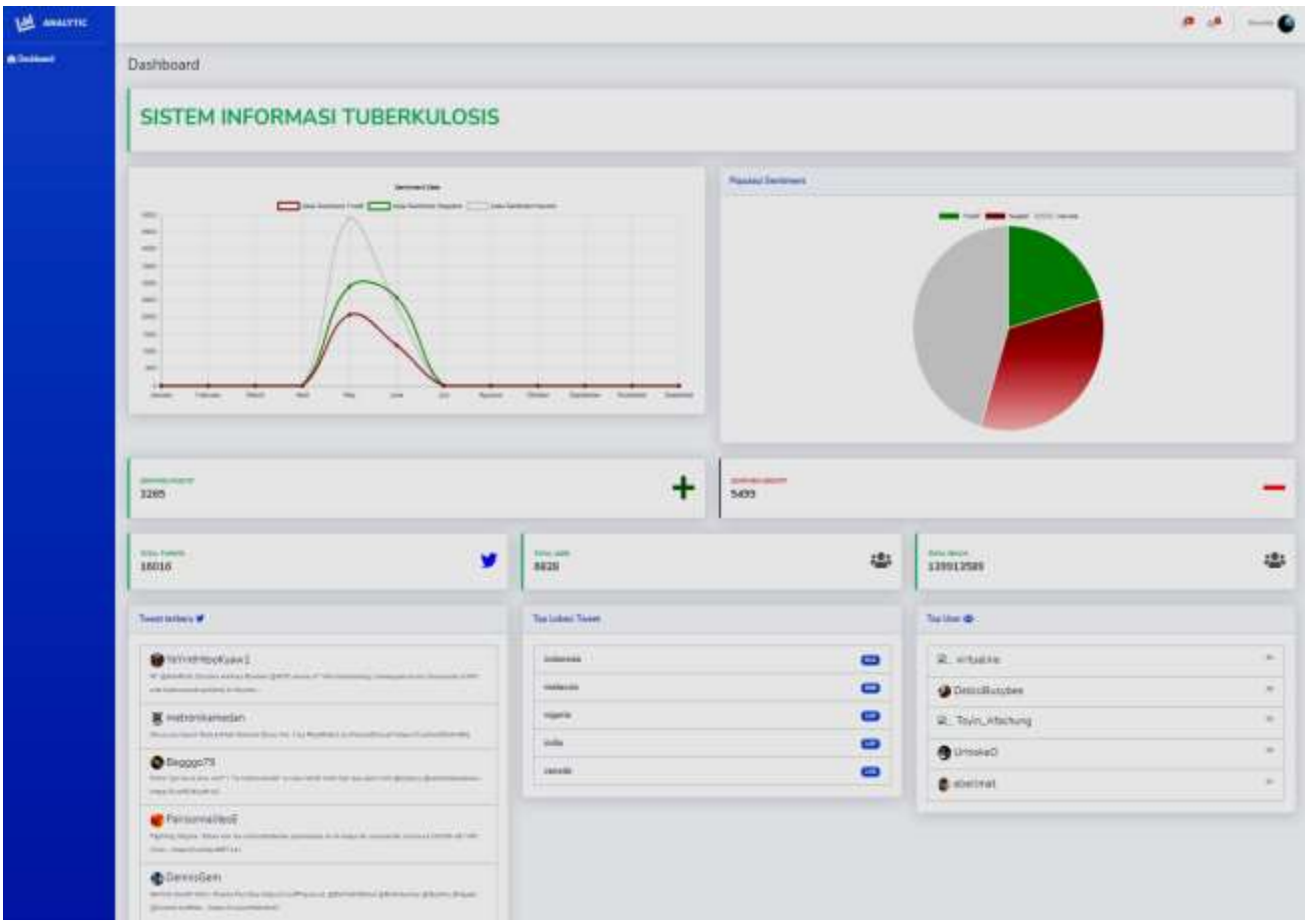
```
batuk batuk mulu ampun deg degan banget tapi hamdallah dah antigen 2x pun ya negatif huhu takut bgt


ok susah nak batuk susah nak bersin sebab perut aku sakit sakit sangat ye muscle pain ni
```



Gambar 4. Ilustrasi penggunaan Lucene untuk proses *indexing*



Gambar 5. Ilustrasi proses klasifikasi analisis sentimen



Gambar 6. Luaran informasi dalam bentuk *dashboard* dari hasil implementasi *microservices* dan analisis sentimen. Keterangan detil dapat dilihat dalam Gambar 7-16.

C. Indexing dan Pembobotan TF-IDF

Setelah praposes data selesai, maka proses selanjutnya adalah proses *indexing*, perhitungan bobot dengan TF-IDF kemudian klasifikasi. Proses *indexing* dan TF&-DF dilakukan bila terdapat *trigger* dari *user*, sedangkan proses klasifikasi dilakukan pada interval waktu tertentu yaitu setiap satu menit sekali. Ketiga proses ini dibuat dalam bahasa pemrograman Java menggunakan Maven. Proses *indexing* dan pembobotan TF-IDF dilakukan bila terdapat *trigger* dari *user* [8, 9]. Pada proses ini, *indexing* dilakukan dengan menggunakan *library* Apache Lucene. Ilustrasi penggunaan Lucene untuk proses *indexing* ini dapat dilihat pada Gambar 4.

D. Klasifikasi Analisis Sentimen

Proses klasifikasi dilakukan dalam rentang waktu tertentu, yaitu setiap satu menit sekali. Dalam proses klasifikasi ini, setiap *tweet* dikelompokkan ke dalam sentimen positif, negatif atau netral [10, 11]. Untuk proses ini, digunakan model sentimen yang telah dibangun dalam Sengon *Project* [12]. Sengon *Project* adalah sebuah upaya untuk membentuk dataset dan model analisis sentimen berbasiskan Bahasa Indonesia yang ditulis dalam bahasa pemrograman Java. Proyek ini dibangun menggunakan *maven* dan mengandung beberapa *library* yaitu OpenNLP, Apache Lucene dan Language Detector. Sengon Project ini menggunakan *lexicon classification* untuk metode klasifikasinya.

Proses klasifikasi dimulai dengan mengambil data dari basis data. Data yang diambil adalah data yang belum pernah diproses sebelumnya (memiliki nilai sentimen *null*) dan telah dilakukan prapemrosesan teks. Data yang diambil adalah maksimal lima puluh data untuk menghindari terjadinya *deadlock*. Data yang telah diambil tersebut kemudian disimpan dalam *array list* yang akan dilakukan proses *looping* untuk dilakukannya klasifikasi dan setiap hasil klasifikasi yang diterima akan dilakukan *update* terhadap *array list* tersebut yang kemudian secara bersamaan akan dilakukan update hasil sentimen ke basis data.

E. API Gateway Services

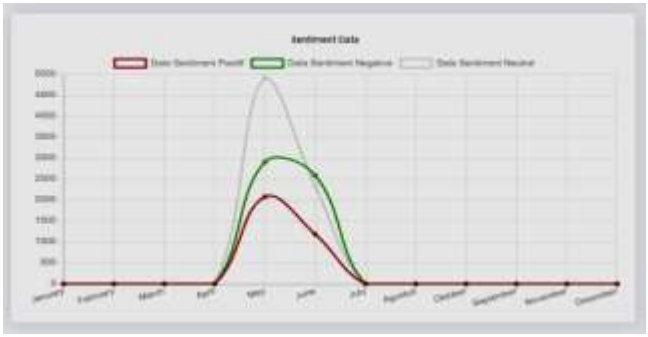
API *Gateway Service* dibangun dengan menggunakan *framework* Laravel sebagai *base framework*. *Service* yang dibuat ini memiliki fungsi untuk melakukan pemanggilan data ke dalam dataset yang telah dilakukan proses prapemrosesan teks, pembobotan TD-IDF, dan juga klasifikasi analisis sentimen. *Service* yang telah dibuat akan mengirimkan respons dengan format JSON yang pada akhirnya oleh *service frontend* akan ditampilkan kepada pengguna.

F. Frontend Services

Frontend Services dibangun dengan menggunakan *framework* VueJS. Tugas *framework* ini adalah untuk melakukan *rendering script* di sisi *client* sehingga proses *load* suatu aplikasi menjadi lebih cepat karena sebagian prosesnya dilakukan di sisi *browser*. Hasil dari *frontend service*, berupa *dashboard*, dapat dilihat dalam Gambar 6.

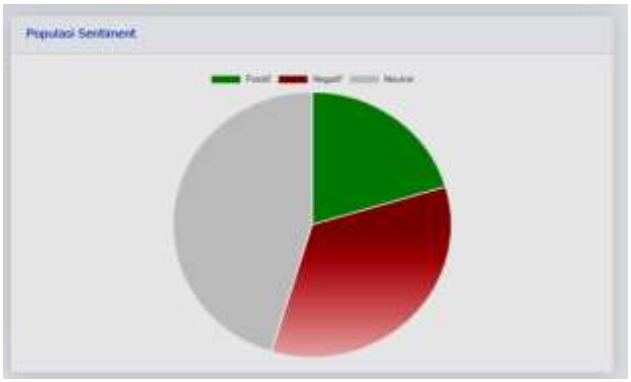
Melalui pengolahan *frontend services* ini, pengguna dapat melihat informasi:

- Grafik Sentimen berdasarkan bulan
  Grafik pada Gambar 7 menampilkan kurva jumlah *tweet* yang mengandung kata-kata yang telah difilter, sesuai kebutuhan kata kunci (kueri) yang akan dianalisis dan telah ditentukan sebelumnya.



Gambar 7. Kurva jumlah *tweet*

- *Pie Chart* Sebaran Sentimen
  Grafik pada Gambar 8 menampilkan persentase sentimen positif, negatif dan juga netral dalam bentuk *pie chart* sehingga pengguna lebih mudah untuk melihat persentase sebaran sentimen yang terhimpun pada *tweet* saat ini, yang telah disimpan dalam basis data.

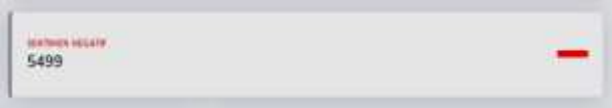


Gambar 8. Sebaran sentimen pada *tweet*

- Jumlah Sentimen Positif
  Bagian *dashboard* pada Gambar 9 menampilkan jumlah sentimen positif dari *tweet* yang ada.

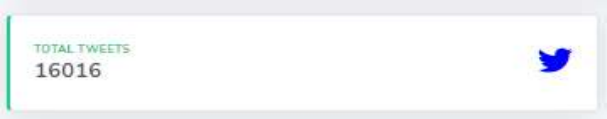
Gambar 9. Jumlah sentimen positif

- Jumlah Sentimen Negatif
  Bagian *dashboard* pada Gambar 10 menampilkan jumlah sentimen negatif dari *tweet* yang ada.



Gambar 10. Jumlah sentimen negatif

- Total *Tweet* terkait TBC
  Bagian *dashboard* pada Gambar 11, menampilkan jumlah total *tweet* yang mengandung kata (kueri) terkait TBC yang telah difilter.

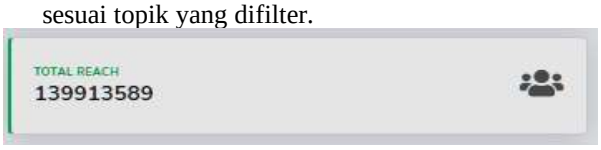


Gambar 11. Jumlah kata terkait kueri TBC (lihat Tabel 1)

- Total *Pengguna* yang Aktif
  Bagian *dashboard* pada Gambar 12, menampilkan jumlah total *tweet* dari pengguna yang paling sering mencuitkan tentang topik terkait TBC yang telah difilter.

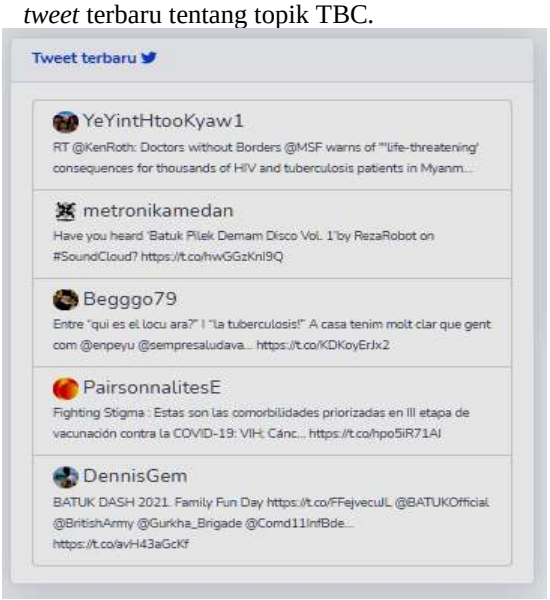
Gambar 12. Total jumlah tweet dari pengguna Twitter yang sering mencuitkan pesan terkait TBC

- Total *Reach* Pengguna
  Bagian *dashboard* pada Gambar 13, menampilkan total jumlah *reach* dari pengguna Twitter yang berhasil dikoneksikan dengan *microservices*, sesuai topik yang difilter.



Gambar 13. Total Jumlah *Reach* Pengguna Twitter

- *Tweet* Terbaru
  Bagian *dashboard* pada Gambar 14, menampilkan *tweet* terbaru tentang topik TBC.

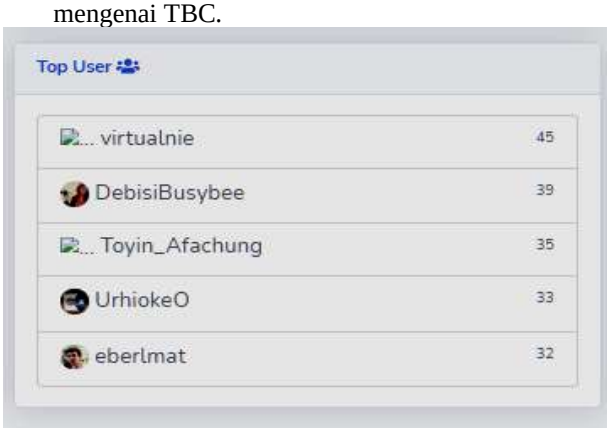


Gambar 14. Tweet Terbaru tentang topik TBC

- Top Lokasi *Tweet*
  Bagian dashboard pada Gambar 15, menampilkan kata kunci lokasi yang paling sering melakukan *tweet* terkait TBC. Lokasi inilah yang akan memberikan indikasi mengenai wilayah yang berpotensi terjadinya penyebaran TBC.

Gambar 15. Sebaran Lokasi *Tweet* Terkait TBC

- Top Pengguna
  Bagian dashboard pada Gambar 16, menampilkan pengguna teratas dalam mengirimkan *tweet* mengenai TBC.



Gambar 16. Daftar Pengguna Twitter dengan Tweet Terbanyak Terkait TBC

G. Diskusi dan Pembahasan

Aplikasi berbasis microservice untuk pelacakan penyakit TBC yang telah dikembangkan ini memiliki keunggulan antara lain:

1. Aplikasi barbasis microservices telah dapat menyediakan informasi tambahan terkait TBC. Melalui *dashboard*, pengguna dapat mengatahui berbagai informasi yang muncul dengan kata kunci terkait TBC dan sebarannya.
2. Aplikasi ini memiliki potensi pemanfaatan yang sangat besar dimana setiap komponen dan juga fungsinya bisa dilakukan optimasi dan digunakan secara independen yang dapat meningkatkan akurasi dari sentiment terhadap suatu cuitan/*tweet*.

3. Aplikasi ini dibangun dengan menggunakan arsitektur *microservices*, arsitektur ini mendukung peningkatan skala menjadi lebih besar seiring dengan meningkatnya jumlah informasi yang diberikan dan jumlah informasi yang akan diolah.
4. Dengan arsitektur yang ada, sangat memungkinkan untuk melakukan penambahan sumber data tanpa harus merombak seluruh aplikasi. Untuk menambahkan sumber data, hanya diperlukan menambahkan *service* baru yang nantinya dapat langsung diintegrasikan dengan *service* sebelumnya yang sudah ada.

IV. KESIMPULAN

Proses pengolahan informasi terkait penyakit TBC pada media sosial sangat membantu dalam penambahan informasi terkait penyebaran dan juga sentimen masyarakat. Dengan menggunakan fitur-fitur yang ada di dalam aplikasi ini, pengguna dapat melihat sebaran data, jumlah sentimen positif atau negatif. Dengan adanya informasi terkait sentimen yang ada, pengguna dapat menambahkan dan melakukan identifikasi lebih dalam terhadap sebaran yang diinformasikan melalui *dashboard*.

Seiring dengan berkembangnya informasi yang ada pada media sosial, maka semakin banyak data yang perlu diolah dan juga diproses. Oleh karena itu diperlukan proses pembelajaran dengan variasi data yang telah masuk memanfaatkan metode *machine learning*, misalnya terkait dengan kata-kata non-formal (*slang*) dan nuansa bahasa daerah (muatan lokal).

REFERENSI

[1] (2018) Info DATIN Pusat Data dan Informasi Kementrian Kesehatan Republik Indonesia. [Online]. Available: https://pusdatin.kemkes.go.id/resources/download/pusdatin/infodatin/infodatin-tuberkulosis-2018.pdf.

[2] (2021) Sistem Informasi Tuberkulosis. [Online]. Available: http://sitb.id/sitb/about.

[3] N.K. Adi and D. Sebastian, "Pembentukan Dataset Topik Kata Bahasa Indonesia pada Twitter Menggunakan TF-IDF & Cosine SImilarity," *Jurnal Teknik Informatika dan Sistem Informasi*., vol. 4, pp. 376-386, Dec. 2018.

[4] J. Xiang, S. A. Chun, Z. Wei, and J. Geller, "Twitter sentiment classification for measuring public health concerns." *Social Network Analysis and Mining, Springer*., vol. 11, pp. 1-25, May. 2015.

[5] G. Aditya, V. Doma, S. Kendre, and L. Bhagwat, "Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach," *IEEE International Advance Computing Converence*., Sep. 2018.

[6] O. Bridianne, S. Wang, P. J. Batterham, A.L. Calear, C. Paris, and H. Christensen, "Detecting suicidality on Twitter," *Internet Interventions, Elsevier*, vol. 2, pp. 183-188, May. 2015.

[7] M. Demchenko (2020) Microservices vs. Monolithic: Which Architecture Suits Best for Your Project?. [Online]. Available: https://ncube.com/blog/microservices-vs-monolithic-which-architecture-suits-best-for-your-project.

[8] Lucene – Indexing Process. [Online]. Available: https://www.tutorialspoint.com/lucene/lucene_indexing_process.htm

[9] Class TFIDF Similarity. [Online]. Available: https://lucene.apache.org/core/7_6_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

[10] E. Adityawan, "Analisis sentimen dengan klasifikasi naïve bayes pada pesan Twitter menggunakan data seimbang," Skripsi,. Bogor (ID): Institut Pertanian Bogor, 2014.

[11] M. Guerini, L. Gatti, and M. Turchi, "Sentiment analysis: How to derive prior polarities from SentiWordNet," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Washington, Amerika Serikat (US): The Association for Computational Linguistics, pp. 1259-1269, 2013.

[12] (2017) Sengon Project. [Online]. Available: https://github.com/masasdani/sengon.