

**IMPLEMENTASI *TEXT MINING* UNTUK DETEKSI  
*TRENDING TOPICS* MENGGUNAKAN METODE BN-GRAM  
PADA STUDI KASUS *TWEET COVID-19***

**TUGAS AKHIR**



**Oleh:**

**ALVIN RAMADHAN  
NIM : 1611500032**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
2020**

**IMPLEMENTASI *TEXT MINING* UNTUK DETEKSI  
*TRENDING TOPICS* DATA *TWEET* MENGGUNAKAN  
METODE BN-GRAM PADA STUDI KASUS *TWEET COVID-19***

**Diajukan untuk memenuhi salah satu persyaratan  
memperoleh gelar Sarjana Komputer (S.Kom)**

**TUGAS AKHIR**



**Oleh :**

**ALVIN RAMADHAN  
NIM : 1611500032**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
2020**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

---

**PERSETUJUAN TUGAS AKHIR**

Nama : Alvin Ramadhan  
Nomor Induk Mahasiswa : 1611500032  
Program Studi : Teknik Informatika  
Bidang Peminatan : *Programming Expert*  
Jenjang Studi : Strata 1  
Judul : **IMPLEMENTASI *TEXT MINING* UNTUK  
DETEKSI *TRENDING TOPICS*  
MENGUNAKAN METODE BN-GRAM  
PADA STUDI KASUS *TWEET COVID-19***

Disetujui untuk dipertahankan dalam sidang Tugas Akhir periode semester Genap  
tahun ajaran 2019/2020

Tangerang Selatan, 01 Agustus 2020

Dosen Pembimbing

( Dr. Indra, S.Kom., M.T.I. )

## ABSTRAK

# IMPLEMENTASI *TEXT MINING* UNTUK DETEKSI *TRENDING TOPICS* MENGGUNAKAN METODE BN-GRAM PADA STUDI KASUS *TWEET COVID-19*

Oleh : Alvin Ramadhan (1611500032)

Twitter merupakan aplikasi situs jejaring sosial yang sangat populer dan berkembang pesat saat ini. Twitter banyak digunakan, karena *platform* ini dapat diakses dengan mudah melalui *smartphone* maupun website untuk berinteraksi dengan sesama penggunanya. Dalam sehari, *tweet* yang dihasilkan dalam aplikasi tersebut bisa mencapai lebih dari ribuan *tweet* dari berbagai negara, untuk itu seringkali berita populer yang dihasilkan Twitter juga menjadi acuan bagi penyedia layanan informasi atau media berita untuk mempublikasikannya menjadi sebuah informasi populer karena sebagian banyak *tweet* membicarakan sesuatu yang sedang terjadi pada waktu tertentu. Salah satu fitur dari aplikasi Twitter adalah “*Trending Topics*”, fitur ini mendeteksi tren berita pada waktu tertentu berdasarkan jumlah *hashtag* dalam *tweet* penggunaannya. Peneliti melakukan penelitian terkait *trending topic tweet* dengan melakukan pengolahan terhadap setiap kata pada *tweet* dengan tujuan informasi *trending topic* yang dihasilkan dapat menjadi acuan topik untuk berita pada hari setelahnya, selain itu juga diharapkan dapat menjadi landasan evaluasi kejadian yang terjadi pada hari tertentu dari waktu pendeteksian tren. Data penelitian ini bersumber dari *tweet* dengan kata kunci terkait *Covid-19* diantaranya yaitu “*covid-19*”, “*corona*”, “*kemenkes*”, “*bnpb*”, “*gugus tugas relawan*”. Metode yang digunakan dalam penelitian ini yaitu BN-Gram, didapat hasil pengujian menggunakan metode *Ground Truth* dengan data pembanding yaitu berita yang bersumber dari, kumparan.com, kompas.com, dan bnpb.go.id pada tanggal 24 s.d 25 Juli 2020 dan data pengujian yaitu *tweet* bertanggal 24 Juli 2020 yang menghasilkan *Topic Recall* :100%, *Keyword Precision*: 10%, dan *Keyword Recall*: 23%, dengan demikian disimpulkan bahwa metode BN-Gram dapat mendeteksi *trending topic* dengan hasil yang cukup baik dan akurat, sehingga dapat menjadi landasan dalam menyajikan berita yang aktual dan relevan.

**Kata Kunci :** *Trending Topic*, Twitter, BN-Gram, Pengujian

xii + 66 halaman : 31 gambar, 22 tabel

## **SURAT PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN PUBLIKASI**

Saya yang bertanda tangan dibawah ini :

Nama	: Alvin Ramadhan
Nim	: 1611500032
Program Studi	: Teknik Informatika
Bidang Peminatan	: <i>Programming Expert</i>
Jenjang Studi	: Strata 1
Fakultas	: Teknologi Informasi

Menyatakan bahwa TUGAS AKHIR yang berjudul:

.....  
.....  
.....

Merupakan:

1. Karya tulis saya sebagai laporan tugas akhir yang asli dan belum pernah diajukan untuk mendapatkan gelar akademik apapun, baik di Universitas Budi Luhur maupun di perguruan tinggi lainnya.
2. Karya tulis ini bukan saduran / terjemahan, dan murni gagasan, rumusan dan pelaksanaan penelitian / implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan pembimbing di organisasi tempat riset.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Saya menyerahkan hak milik atas karya tulis ini kepada Universitas Budi Luhur, dan oleh karenanya Universitas Budi Luhur berhak melakukan pengelolaan atas karya tulis ini sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh berdasarkan karya tulis ini, serta sanksi lainnya sesuai dengan norma di Universitas Budi Luhur dan Undang-Undang yang berlaku.

Tangerang Selatan, 01 Agustus 2020

Alvin Ramadhan

## KATA PENGANTAR

Puji serta syukur Alhamdulillah, penulis panjatkan kehadiran Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat dan karunia-Nya, sehingga pada akhirnya penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Adapun Tugas Akhir ini disusun untuk memenuhi persyaratan dalam menyelesaikan tingkat Pendidikan Strata 1 (S1) pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Budi Luhur dengan judul tugas akhir yang penulis ambil yaitu **“IMPLEMENTASI TEXT MINING UNTUK DETEKSI TRENDING TOPICS MENGGUNAKAN METODE BN-GRAM PADA STUDI KASUS TWEET COVID-19”**.

Penulis berharap laporan ini dapat memberikan manfaat kepada para pembaca umumnya dan kepada mahasiswa khususnya. Selain itu, diharapkan laporan ini juga dapat menjadi bahan perbandingan dalam melakukan karya penelitian selanjutnya. terselesaikannya penelitian ini tidak lepas dari bantuan berbagai pihak, rasa terimakasih yang mendalam ingin penulis sampaikan kepada mereka yang telah berjasa dalam membantu penyusunan tugas akhir ini, yaitu kepada:

1. Allah Subhanahu Wa Ta'ala, atas segala petunjuk, kemudahan, serta nikmat-Nya yang diberikan sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini dengan baik.
2. Segenap keluarga penulis, khususnya orang tua tercinta Bapak & Ibu, serta saudara khususnya Adik, yang telah membantu memberikan dukungan baik moral maupun material, dan selalu memberikan doa restu.
3. Bapak Dr. Ir. Wendi Usino, M.Sc., M.M., selaku Rektor Universitas Budi Luhur.
4. Bapak Dr. Deni Mahdiana, M.M., M.Kom, selaku Dekan Fakultas Teknologi Informasi Universitas Budi Luhur.
5. Bapak Dr. Hari Soetanto, S.Kom, M.Sc., selaku Dosen Penasehat Akademik.
6. Bapak Dr. Indra, S.Kom., M.T.I, selaku Kaprodi Teknik Informatika Fakultas Teknologi Informasi Universitas Budi Luhur sekaligus Dosen Pembimbing Tugas Akhir yang telah membantu penulis dalam memberikan saran-saran dalam penyelesaian laporan tugas akhir ini.
7. Ibu Painem, M.Kom., selaku Kepala Laboratorium ICT Universitas Budi Luhur yang selalu memberikan arahan dan ilmu selama penulis mengabdikan di LAB ICT.
8. Rekan-rekan Asisten Laboratorium ICT Terpadu Universitas Budi Luhur khususnya angkatan 2016, sebagai rekan kerja selama 3 tahun di LAB ICT.
9. Farah Nur Arafah, S.M. sebagai teman dekat yang selama ini telah menemani perjuangan penulis dari awal berkuliah di Universitas Budi Luhur sampai saat ini.

Tangerang Selatan, 01 Agustus 2020

Penulis

## DAFTAR TABEL


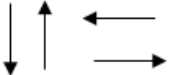


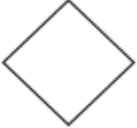




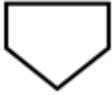
Tabel 2.1 Kombinasi Awalan Akhiran Yang Tidak Diijinkan .....	8
Tabel 2.2 Cara Menentukan Tipe Awalan Untuk Kata Yang Diawali “te-” .....	9
Tabel 2.3 Jenis Awalan Berdasarkan Tipe Awalannya .....	9
Tabel 2.4 Daftar <i>Contextual Feature</i> .....	13
Tabel 2.5 Daftar <i>Morphological Feature</i> .....	13
Tabel 2.6 Daftar <i>part-of-speech features</i> .....	14
Tabel 2.7 Contoh <i>rule</i> dari proses <i>feature assignment</i> .....	14
Tabel 2.8 Studi literatur .....	15
Tabel 3.1 Sampel data <i>tweet</i> .....	20
Tabel 3.2 Hasil Pelabelan <i>Feature Assigment</i> .....	29
Tabel 4.1 Sampel Data <i>Tweet</i> .....	36
Tabel 4.2 Pemetaan <i>time slot tweet</i> .....	36
Tabel 4.3 Ekstraksi Bigram .....	37
Tabel 4.4 Skor DF IDF <sub>t</sub> .....	38
Tabel 4.5 Sampel Perhitungan Skor DF IDF <sub>t</sub> .....	41
Tabel 4.6 Pemetaan N-Gram .....	41
Tabel 4.7 Pemetaan Jarak .....	42
Tabel 4.8 Sampel pemetaan n-gram .....	43
Tabel 4.9 Iterasi ke-1 .....	43
Tabel 4.10 Iterasi ke-2 .....	44
Tabel 4. 11 Tabel Pengujian .....	59
Tabel 4. 12 Tabel Perhitungan <i>Recall</i> .....	60

## DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Tahapan BN-Gram.....	11
Gambar 3.1 Tahapan Metode.....	21
Gambar 3.2 Metode Pengumpulan Data.....	22
Gambar 3.3 <i>Casefolding</i> .....	23
Gambar 3.4 Menghapus karakter selain a-z.....	24
Gambar 3.5 Menghapus 1 karakter.....	25
Gambar 3.6 Menghapus 1 karakter.....	26
Gambar 3.7 <i>Slangword</i> .....	26
Gambar 3.8 <i>Slangword</i> .....	27
Gambar 3.9 <i>Stemming</i> .....	28
Gambar 3.10 <i>Tokenization</i> .....	29
Gambar 3. 11 Tahapan Utama BN-Gram .....	31
Gambar 4. 1 <i>Flowchart</i> keseluruhan sistem .....	45
Gambar 4. 2 <i>Flowchart crawling tweet</i> .....	46
Gambar 4. 3 <i>Flowchart preprocessing tweet</i> .....	46
Gambar 4. 4 <i>Flowchart cleaning</i> .....	47
Gambar 4. 5 <i>Flowchart slangword</i> .....	48
Gambar 4. 6 <i>Flowchart</i> NER.....	49
Gambar 4. 7 <i>Flowchart Stopword</i> .....	50
Gambar 4. 8 <i>Flowchart Stemming</i> .....	51
Gambar 4. 9 <i>Flowchart</i> BN-Gram.....	52
Gambar 4. 10 <i>Flowchart</i> Perhitungan $DF-IDF_t$ .....	53
Gambar 4. 11 <i>Flowchart</i> Klasterisasi N-Gram.....	53
Gambar 4. 12 <i>Flowchart Topic Ranking</i> .....	54
Gambar 4. 13 Tampilan Layar Beranda.....	60
Gambar 4. 14 Tampilan Layar <i>Datasets</i> .....	61
Gambar 4. 15 Tampilan Pilih Tanggal Preprocessing .....	61
Gambar 4. 16 Tampilan Hasil Preprocessing .....	62
Gambar 4. 17 Tampilan Layar BN-Gram.....	62
Gambar 4. 18 Tampilan Layar Klasterisasi .....	63
Gambar 4. 19 Tampilan Layar Pengujian .....	63



## DAFTAR SIMBOL *FLOWCHART*

GAMBAR	NAMA	KETERANGAN
	<i>Terminal Point Symbol</i> / Simbol Titik Terminal	Simbol untuk pemulaan atau akhir dari suatu program
	<i>Flow Direction Symbol</i> / Simbol Arus	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain ( <i>connecting line</i> ).
	<i>Connector (On-page)</i>	Simbol keluar atau masuk prosedur atau proses dalam lembar atau halaman yang sama
	<i>Processing Symbol</i> / Simbol Proses	Simbol proses yang menunjukkan pengolahan yang dilakukan sistem
	<i>Decision Symbol</i> / Simbol Keputusan	Simbol kondisi yang akan menghasilkan <i>output true</i> atau <i>false</i>
	<i>Preparation Symbol</i> / Simbol Persiapan	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam <i>storage</i>
	<i>Input-Output</i> / Simbol Keluar-Masuk	Simbol yang menyatakan proses <i>input</i> atau <i>output</i> tanpa bergantung jenis peralatannya
	<i>Manual Input Symbol</i>	Simbol <i>input</i> data secara manual menggunakan <i>online keyboard</i> .
	<i>Predefined Process</i> / Simbol Proses Terdefinisi	Simbol yang Menggambarkan proses – proses yang masih dapat dijabarkan dalam Algoritme.
	<i>Connector (Off-page)</i>	Simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda

## DAFTAR ISI

ABSTRAK.....	iii
KATA PENGANTAR .....	v
DAFTAR TABEL.....	vi
DAFTAR GAMBAR .....	vii
DAFTAR SIMBOL <i>FLOWCHART</i> .....	viii
DAFTAR ISI.....	ix
BAB I.....	1
PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Batasan Masalah .....	3
1.4    Tujuan .....	3
1.5    Manfaat.....	4
1.6    Sistematika Penulisan.....	4
BAB II.....	5
LANDASAN TEORI.....	5
2.1 <i>Data Mining</i> .....	5
2.2 <i>Text Mining</i> .....	5
2.3 <i>Trending Topic</i> .....	5
2.4 <i>Preprocessing</i> .....	6
2.4.1 <i>Casefolding</i> .....	6
2.4.2    Menghapus karakter kecuali a sampai z .....	6
2.4.3    Menghapus teks dengan 1 karakter.....	6
2.4.4    Menghapus URL .....	6
2.4.5    Mengganti <i>slangword</i> .....	6
2.4.6    Menghapus <i>stopword</i> .....	7
2.4.7 <i>Stemming</i> .....	7
2.4.8 <i>Tokenization</i> .....	10
2.5    Ngram.....	10
2.6    BNgram.....	11

2.6.1	Perhitungan DF-IDF <sub>t</sub> .....	11
2.6.2	Klasterisasi Ngram.....	12
2.7	<i>Named Entity Recognition</i> (NER) .....	12
2.8	Pengujian.....	15
2.9	Studi Literatur .....	15
BAB III .....		20
METODOLOGI PENELITIAN.....		20
3.1	Data Penelitian .....	20
3.2	Penerapan Metode.....	21
3.2.1	Pengumpulan Data.....	22
3.2.2	<i>Preprocessing</i> .....	23
3.2.3	<i>Named Entity Recognition</i> (NER).....	29
3.2.4	BN-Gram.....	31
3.3	Rancangan Pengujian .....	33
3.3.1	<i>Topic Recall</i> (TR) .....	33
3.3.2	<i>Keyword Precision</i> (KP).....	33
3.3.3	<i>Keyword Recall</i> (KR) : .....	33
BAB IV .....		35
HASIL DAN PEMBAHASAN.....		35
4.1	Lingkungan Percobaan.....	35
4.1.1	Spesifikasi Perangkat Keras.....	35
4.1.2	Spesifikasi Perangkat Lunak .....	35
4.2	Implementasi Metode dan Langkah Pengujian .....	35
4.2.1	Tahap Pengumpulan Data .....	35
4.2.2	Tahapan BN-Gram.....	35
4.3	<i>Flowchart</i> Tahapan Metode .....	44
4.3.1	<i>Flowchart</i> Keseluruhan Sistem.....	44
4.3.2	<i>Flowchart Crawling</i> .....	45
4.3.3	<i>Flowchart Preprocessing</i> .....	46
4.3.4	<i>Flowchart Cleaning</i> .....	47
4.3.5	<i>Flowchart Slangword</i> .....	47

4.3.6	<i>Flowchart Labelling NER</i> .....	48
4.3.7	<i>Flowchart Stopword</i> .....	50
4.3.8	Flowchart Stemming .....	50
4.3.9	<i>Flowchart BN-Gram</i> .....	51
4.3.10	Flowchart Perhitungan DF-IDF <sub>t</sub> .....	52
4.3.11	<i>Flowchart Klasterisasi N-Gram</i> .....	53
4.3.12	Flowchart <i>Topic Ranking</i> .....	54
4.4	Algoritme Tahapan Metode .....	54
4.4.1	Algoritme Keseluruhan Sistem.....	54
4.4.2	Algoritme <i>Crawling</i> .....	55
4.4.3	Algoritme <i>Preprocessing</i> .....	55
4.4.4	Algoritme <i>Cleaning</i> .....	55
4.4.5	Algoritme <i>Slangword</i> .....	55
4.4.6	Algoritme NER .....	56
4.4.7	Algoritme <i>stopword</i> .....	56
4.4.8	Algoritme <i>Stemming</i> .....	57
4.4.9	Algoritme BN-Gram.....	57
4.4.10	Algoritme Perhitungan DF-IDF <sub>t</sub> .....	58
4.4.11	Algoritme Klasterisasi N-Gram .....	58
4.4.12	Algoritme <i>Topic Ranking</i> .....	58
4.5	Pengujian.....	59
4.6	Tampilan Layar Aplikasi.....	60
4.6.1	Tampilan layar Beranda .....	60
4.6.2	Tampilan layar <i>datasets</i> .....	60
4.6.3	Tampilan layar <i>preprocessing</i> .....	61
4.6.4	Tampilan layar BN-Gram .....	62
4.6.5	Tampilan layar hasil klasterisasi .....	62
4.6.6	Tampilan layar pengujian .....	63
BAB V	.....	64
PENUTUP	.....	64
5.1	Kesimpulan .....	64

5.2	Saran .....	64
DAFTAR PUSTAKA .....		65

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Seperti yang telah kita ketahui bersama, bahwa wabah pandemi virus Corona telah ditetapkan dalam status kedaruratan bencana nasional oleh BNPB (Badan Nasional Penanggulangan Bencana) pada Mei 2020 yang terlampir dalam Surat Edaran (SE) Nomor 6 Tahun 2020 tentang Status Keadaan Darurat Bencana Nonalam Covid-19. Wabah ini menjadi *trending issue* selama beberapa bulan terakhir karena telah merubah segala bentuk aktivitas masyarakat yang sebelumnya. Mengingat bahwa wabah *Covid-19* ini merupakan masalah dan tanggung jawab kita bersama sebagai warga negara, dimulai dari tindakan pencegahan melalui protokol kesehatan sampai kepada pencegahan melalui edukasi penyebaran informasi yang valid dan relevan untuk masyarakat.

Adanya wabah virus corona ini telah menimbulkan banyak sekali dampak serta perubahan yang terjadi pada masyarakat seluruh dunia khususnya di Indonesia. Pemerintah pun dalam hal ini tengah berusaha keras bersama lembaga terkait bahkan masyarakat untuk mengumpulkan dan memperbarui berbagai macam data yang mana data tersebut digunakan sebagai analisis pencegahan dampak buruk yang ditimbulkan dari pandemi wabah ini. Beberapa hasil penelitian sebagai upaya membantu pemerintah dalam membuat data analisis ini pun sudah dikembangkan oleh sebagian orang terkait *Covid-19*, diantaranya yaitu penelitian oleh Hanoatubun, (2020) tentang Dampak *Covid-19* Terhadap Perekonomian Indonesia, dan penelitian oleh (Sindi dkk., 2020) tentang Analisis Algoritma *K-Medoids Clustering* dalam Pengelompokan Penyebaran *Covid-19* di Indonesia.

Berdasarkan dua penelitian yang telah disebutkan di atas, disimpulkan bahwa hasil penelitian tersebut merupakan analisis yang dikembangkan untuk mengetahui perkembangan isu terkait wabah virus corona baik dalam hal dampak ekonomi, sosial, pendidikan, kesehatan dan lain-lain, analisis ini kemudian menjadi acuan bagi pemerintah ataupun lembaga terkait untuk kemudian melakukan sebuah kebijakan baru untuk masyarakat nantinya, maka dari itu peneliti bermaksud untuk melakukan pengembangan analisis berdasarkan data yang bersumber dari salah satu media sosial.

Perkembangan media sosial sebagai sarana komunikasi dan sumber informasi sangat memiliki peranan penting bagi masyarakat terutama pada masa pandemic *Covid-19* ini, hal ini terjadi karena masyarakat sudah merasakan banyak manfaat yang dihasilkan melalui media sosial, salahsatunya media sosial dapat diakses dengan mudah kapan saja dan dimana saja, terlebih didalamnya, pengguna bisa saling berinteraksi dengan sesama penggunanya, arus penyebaran informasinya pun sangat cepat dan tidak terbatas. Salahsatu layanan jejaring sosial yang sangat banyak digunakan saat ini, serta sudah lama digunakan banyak orang untuk keperluan pribadi, komunitas maupun perusahaan yaitu Twitter, maka dari itu peneliti menggunakan sumber data berupa *tweet* dalam penelitian ini. Menurut Wiyadi, (2017) Twitter adalah layanan jejaring sosial dan mikroblog

daring yang memungkinkan penggunaanya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter, yang dikenal dengan sebutan kicauan (*tweet*). Twitter juga memiliki fitur *trending topics* yang selalu menampilkan cuitan populer berdasarkan *hashtag*, maka tak heran, media sosial Twitter kini juga dapat membantu dalam penyajian informasi salah satunya melalui topik-topik yang sedang populer dibicarakan atau biasa disebut *trending topics*.

*Trending topics* sendiri yaitu sebuah berita yang sedang terkenal dan populer, paling banyak dicari orang dalam waktu tertentu (Juditha & Christiany, 2018). Suatu topik akan menjadi *trending topics* ketika banyak orang yang membicarakannya. Menurut Cvijikj & Michahelles, (2011), *trending topics* terdiri dari tiga kategori yaitu *disruptive events*, topik populer dan rutinitas sehari-hari. *Disruptive events* adalah kejadian atau fenomena yang menarik perhatian global, seperti gempa bumi di Jepang. Topik populer biasanya berhubungan dengan peristiwa masa lampau, selebriti, produk, merk yang tetap populer selama periode waktu yang lebih lama, seperti Michael Jackson dan Coca cola. Rutinitas sehari-hari berhubungan dengan kata-kata umum, seperti ucapan selamat malam atau ucapan ulang tahun.

Status di Twitter berisi informasi yang tersembunyi, sehingga deteksi *trending topic* diperlukan untuk menyajikan informasi yang menjadi perbincangan warganet. *Trending topics* adalah salah satu fitur yang dimiliki oleh Twitter. Penelitian tentang *trending topics* di Twitter sebelumnya sudah pernah dilakukan oleh Mediayani et al., (2019). Hasil penelitiannya berupa pengelompokan lima topik yang menjadi tren di New York sebelum tahun baru menggunakan analisis data dengan pengelompokan K-Means. Dikarenakan masih kurangnya penelitian *trending topic* terkait *Covid-19* ini, menjadi peluang penulis untuk mengangkat topik tersebut

Menurut Indra et al., (2019), deteksi *trending topics* dalam *tweet* berbahasa Indonesia dipengaruhi oleh prapemrosesan dan jumlah data *tweet* yang dikumpulkan. Pada penelitiannya, deteksi *trending topics tweet* dalam bahasa Indonesia lebih akurat dengan metode BN-grams dibandingkan dengan Doc-p. BN-grams menghasilkan akurasi yang lebih tinggi dalam deteksi *trending topics* daripada Doc-p dalam semua *dataset* yang diuji.

Pada prosesnya, peran *trending topics* dalam menemukan informasi atau sebuah topik terbaru dan aktual dengan cepat sangatlah penting. Terutama bagi para pengamat media sosial maupun para pekerja seperti juru warta atau reporter media berita. Namun, ada peristiwa yang tidak seluruhnya tersampaikan kepada masyarakat karena kurangnya akses informasi dan metode *trending topics* yang dilakukan Twitter yaitu dengan cara penghitungan *hashtag* terbanyak, sehingga terkadang banyak *tweet* yang tidak berhubungan dengan tren teratas karena menggunakan *hashtag* tersebut, terutama hal-hal yang berhubungan dengan topik dalam penelitian ini, yaitu topik yang berkaitan dengan masa pandemi *Covid-19*. Untuk itu, dibuatlah sebuah *tools* yang dapat mendeteksi tren terkini yang sedang terjadi pada suatu waktu yang telah ditentukan dengan metode pengolahan kata. Pengumpulan data dilakukan secara *streaming*. Sehingga, informasi *tweet*

yang didapat dari pengumpulan data yang begitu besar tadi lebih relevan terhadap peristiwa yang sedang populer dibicarakan.

Penelitian ini akan membahas tentang mengenai *trending topics* pada *tweet* yang terkait dengan pandemi wabah *Covid-19*. Pengumpulan data *tweet* dilakukan dengan cara *streaming* dari Twitter, dengan kata kunci tertentu yang berkaitan dengan informasi *covid-19* diantaranya Covid19, Corona, Kemenkes, BNPB, Gugus Tugas Relawan yang mana kata kunci tersebut mewakili informasi yang terkait dengan wabah *Covid-19* di Indonesia. Informasi *trending topics* yang dihasilkan dalam penelitian ini kemudian divalidasi dengan data yang bersumber dari portal media berita daring seperti *kompas.com*. Untuk pendekatannya menggunakan metode BN-Gram. Program aplikasi untuk deteksi *trending topics* ini dibuat dengan menggunakan bahasa pemrograman Python dan PHP. Dengan ini diharapkan hasil yang didapat dari aplikasi yang dibuat dapat membantu menyajikan informasi berita dalam topik masa pandemi *Covid-19* agar lebih relevan dan sesuai untuk hari berikutnya setelah pendeteksian suatu tren.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan sebelumnya, maka dapat disimpulkan sebuah rumusan masalah yaitu:

- a. Bagaimana merancang model untuk mendeteksi *trending topics* pada kata kunci yang berkaitan dengan “*Covid-19*” dengan data *stream* Twitter?
- b. Bagaimana membangun aplikasi deteksi *trending topics* dari data *stream* Twitter berbasis web?
- c. Bagaimana metode untuk melakukan ekstraksi teks dalam menghasilkan keluaran *trending topics*?

## 1.3 Batasan Masalah

Adapun batasan atau ruang lingkup masalah antara lain sebagai berikut:

- a. Data yang akan digunakan adalah data *streaming* dari Twitter dan berbahasa Indonesia.
- b. Deteksi *trending topics* hanya dilakukan pada topik terkait *Covid-19*.
- c. Kata kunci yang digunakan seputar topik yang berhubungan dengan *covid-19*, diantaranya “*Covid-19*, Corona, Kemenkes, BNPB, Gugus Tugas Relawan”.
- d. Pengumpulan data dilakukan saat dimulainya masa pandemi *covid-19* secara bertahap mulai dari tanggal 20 Juli 2020 sampai 30 Juli 2020.

## 1.4 Tujuan

- a. Merancang sebuah model untuk mendeteksi *trending topics* dengan topik terkait *Covid-19*.
- b. Mengimplementasikan rancangan model yang dibuat ke dalam aplikasi berbasis web.
- c. Menggunakan metode BN-Gram untuk pengolahan data untuk menghasilkan sebuah keluaran *trending topics* dari data *Tweet*.



## 1.5 Manfaat

Adapun manfaat dari penelitian ini adalah sebagai acuan untuk topik berita pada hari esoknya, atau evaluasi peristiwa penting pada hari tertentu dari waktu pendeteksian tren yang bersumber dari data *stream* Twitter. Sehingga diharapkan, informasi yang disajikan dapat lebih relevan, aktual, cepat dan akurat bagi khalayak umum.

## 1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut:

### **BAB I: PENDAHULUAN**

Bagian ini berisi tentang latar belakang, rumusan masalah, batasan masalah, manfaat dan tujuan penelitian, dan juga membahas mengenai sistematika penulisan.

### **BAB II: LANDASAN TEORI**

Bagian ini berisi tentang algoritma dan metode yang akan dibahas, serta teori-teori yang berkaitan dengan penelitian ini, yaitu pengertian dan pemahaman Twitter, *text mining*, *data mining*, *preprocessing*, *trending topics*, *scraping*, n-gram, metode yang digunakan yaitu BN-grams serta studi literatur.

### **BAB III: METODOLOGI PENELITIAN**

Bagian ini berisi tentang sumber data penelitian, penerapan atau tahapan metode yang digunakan. Bab ini juga berisi tentang rancangan pengujian dari ekstraksi informasi yang didapat.

### **BAB IV: HASIL DAN PEMBAHASAN**

Bagian ini berisi mengenai lingkungan percobaan sistem yang dibuat, implementasi metode, *flowchart* tahapan metode, dan uraian algoritme pada proses, serta analisa pengujian sistem yang telah dibangun apakah data hasil pengelompokan yang didapat sudah sesuai dan relevan.

### **BAB V: PENUTUP**

Bagian ini berisi tentang kesimpulan yang dapat ditarik dari penelitian dan saran untuk pengembangan lebih lanjut mengenai topik terkait dalam penelitian berikutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Data Mining**

*Data mining* merupakan serangkaian proses untuk menggali informasi dengan melakukan analisa data untuk menemukan suatu pola dari kumpulan data tersebut. *Data mining* mampu menganalisa data yang besar menjadi ekstraksi berupa pola yang mempunyai arti bagi pendukung keputusan (Gunadi & Sensuse, 2012). Data mining juga bisa disebut knowledge discovery adalah proses pengambilan pola pada data yang akan di proses lalu output tersebut berupa informasi yang sangat penting. Proses yang dilakukan untuk mengekstrak pengetahuan dalam data mining adalah pengenalan pola, clustering, asosiasi, prediksi dan klasifikasi (Fitri dkk., 2018). *Data mining* memiliki variasi untuk menemukan pola dari ekstraksi sebuah kumpulan sekumpulan data tekstual yang disebut dengan *text mining*. *Text mining* memiliki fokus pada pengolahan data berupa kata atau teks.

#### **2.2 Text Mining**

Dalam jurnal Februariyanti, (2012), menurut Hearst *text mining* diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, dengan secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber . Sedangkan menurut Harlian *text mining* memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

*Text mining* merupakan bagian dari *data mining*, yang mana digunakan untuk mendapatkan informasi dari sebuah data atau dokumen berupa sekumpulan teks yang memiliki format yang terstruktur ataupun tidak terstruktur dengan jumlah yang besar. Dalam *text mining* memiliki tugas khusus yaitu klasifikasi dan klasterisasi. Sedangkan dalam penerapannya, *text mining* berfungsi untuk mencari pola dalam teks, menganalisa teks agar bisa menghasilkan keluaran berupa informasi yang bermanfaat pada tujuan tertentu. Dikarenakan data yang diproses pada *text mining* merupakan sebuah teks yang tidak terstruktur, maka diperlukan pemilihan teks sebelum dilakukan proses selanjutnya, pada tahap ini dikenal dengan prapemrosesan (*preprocessing*).

#### **2.3 Trending Topic**

*Trending topic* sendiri diartikan sebagai sebuah berita yang paling populer dan paling banyak dicari orang dalam waktu tertentu (Juditha, 2018). Suatu topik akan menjadi tren ketika banyak orang yang membicarakannya. Biasanya topik-topik yang sedang ramai menjadi perbincangan public di dunia maya seperti Twitter contohnya. Menurut TEJASREE et al., (2017), setidaknya ada 2 faktor penting yang dijadikan alat ukur terhadap topik yang sedang tren, diantaranya endogenitas dan eksogenitas, endogenitas ini

digunakan untuk menggambarkan efek turunan dari topik tertentu dalam suatu cakupan, sementara eksogenitas ini mewakili kekuatan pendorong eksternal ke suatu cakupan. Pertimbangkan faktor-faktor dari suatu topik yang mempengaruhi tren seperti popularitas, transmisi, cakupan potensial, serta reputasinya.

## 2.4 *Preprocessing*

Tahapan *preprocessing* atau praproses merupakan bagian yang sangat penting dalam menyiapkan data, hal ini dikarenakan struktur data yang dihasilkan pada tahap pengumpulan tidak beraturan, sehingga menyebabkan proses menjadi tidak berjalan dengan baik.

Merujuk pada penelitian sebelumnya yang dilakukan oleh Himalatha dalam jurnal Mujilahwati, (2016) maka pada penelitian ini akan dibahas beberapa tahapan *preprocessing* teks antara lain, *casefolding*, menghapus karakter selain a-z, menghapus teks dengan 1 karakter, menghapus *URL*, mengganti *slangword*, menghapus *stopword*, *stemming*, dan *tokenization*.

### 2.4.1 *Casefolding*

Pada proses ini bertujuan untuk mengubah semua karakter huruf menjadi huruf kecil (*lowercase*), hal ini dilakukan untuk menyamakan arti dari suatu kata yang sama, apabila penulisan besar kecilnya huruf tidak sama.

### 2.4.2 Menghapus karakter kecuali a sampai z

Pada proses ini dilakukan penghapusan untuk seluruh karakter berupa simbol dan angka, atau menyisakan hanya karakter angka, termasuk menghapus hashtag (#) dan mention (@), hal ini dilakukan karena simbol dan angka dianggap tidak terlalu penting, tetapi jika ini diperlukan, maka proses ini dihilangkan.

### 2.4.3 Menghapus teks dengan 1 karakter

Pada proses ini dilakukan penghapusan kata dengan jumlah hanya 1 karakter saja, penghapusan karakter ini bertujuan untuk mengurangi kata yang dianggap tidak memiliki arti.

### 2.4.4 Menghapus URL

Munculnya sebuah *URL* dari tweet, membuat data tidak efektif dan tidak memiliki arti. Untuk itu perlu adanya penghapusan *URL* atau alamat web. Kemunculan alamat web atau *URL* ini disebabkan karena banyaknya pengguna mempromosikan sesuatu pada sebuah situs supaya pengguna yang lain dapat langsung mengakses halaman web yang dimaksud.

### 2.4.5 Mengganti *slangword*

Teks yang tidak terstruktur membuat sebuah teks terkadang tidak sesuai dengan ejaan bahasa Indonesia yang baku (EYD) pada konteks ini, kata yang tidak baku disebut dengan *slangword*, untuk mendapatkan informasi dari teks agar maksimal, kata-kata tidak baku,

baik kata gaul, singkatan atau yang lain sebanyak mungkin ditampung ke dalam kamus *slangword*, untuk kemudian dilakukan *replace* supaya menjadi kata dengan bahasa Indonesia yang baku sesuai EYD.

#### 2.4.6 Menghapus *stopword*

*Stopword* merupakan salahsatu kata yang diabaikan dalam pemrosesan, Ringkasnya *stopword* adalah kata hubung atau kata sambung dalam sebuah kalimat, seperti “di”, “pada”, “karena”, “sebuah”, “oleh”, dll. Sebelum melakukan proses penghapusan *stopword*, kumpulkan daftar atau kamus *stopword* yang diberi nama *stoplist*. Kemudian lakukan perbandingan antara sebuah teks dengan *stoplist*. Jika terdapat kata-kata yang terdapat dalam *stoplist*, maka kata tersebut dihilangkan. Untuk *stoplist* dalam bahasa Indonesia, datanya bersumber dari Tala, (2003).

#### 2.4.7 *Stemming*

*Stemming* adalah proses pemotongan (pembuangan) imbuhan (*affix*), baik *prefix* maupun *suffix*, dari sebuah *term* untuk mendapatkan kata dasar (*root* atau *stem*) dari kata yang berimbuhan (Wahyudi dkk., 2017). Dalam penelitian ini proses *stemming* menggunakan *library* Sastrawi, algoritma *stemming* Nazief dan Adriani tahun 1996, algoritma ini dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan akhiran (*confixes*). Berikut contoh *stemming* pada Tabel 2.1 yang menjelaskan perubahan pada kata yang memiliki imbuhan, kemudian dilakukan proses *stemming* sehingga menghasilkan sebuah kata dasar.

Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

- a. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah *root word*. Maka algoritma berhenti.
- b. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
- c. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah c1.
  - 1) Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah c2.
  - 2) Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah d.

- d. Hapus *Derivation Prefix*. Jika pada langkah c ada sufiks yang dihapus maka pergi ke langkah d1, jika tidak pergi ke langkah d2.
  - 1) Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah d2.
  - 2) For  $i = 1$  to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah e, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
- e. Melakukan *Recoding*.
- f. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

- a. Jika awalannya adalah: “di-”, “ke-”, atau “se-” maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
- b. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
- c. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” maka berhenti.
- d. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada Tabel 2.2 Hapus awalan jika ditemukan.
- e. Pada Tabel 2.3 merupakan daftar awalan berdasarkan tipe awalan dan awalan yang harus dihapus.

**Tabel 2.1 Kombinasi Awalan Akhiran Yang Tidak Diijinkan**

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

**Tabel 2.2 Cara Menentukan Tipe Awalan Untuk Kata Yang Diawali “te-”**

Following CharactersR				Tipe Awalan
Set 1	Set 2	Set 3	Set 4	
“-r-“	“-r-“	-	-	none
“-r-“	Vowel	-	-	ter-luluh
“-r-“	not (vowel or “-r-”)	“-er-“	vowel	ter
“-r-“	not (vowel or “-r-”)	“-er-“	not vowel	ter-
“-r-“	not (vowel or “-r-”)	not “-er-“	-	ter
not (vowel or “-r-”)	“-er-“	vowel	-	None
not (vowel or “-r-”)	“-er-“	not vowel	-	Te

**Tabel 2.3 Jenis Awalan Berdasarkan Tipe Awalannya**

Tipe Awalan	Awalan yang harus dihapus
di-	di-
ke-	ke-
se-	se-
te-	te-
ter-	ter-
ter-luluh	Ter

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan dibawah ini:

a. Aturan untuk reduplikasi.

- 1) Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, contoh : “buku-buku” *root word*-nya adalah “buku”.
- 2) Kata lain, misalnya “bolak-balik”, “berbalas-balasan, dan ”seolah-olah”. Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata “berbalas-balasan”, “berbalas” dan “balasan” memiliki *root word* yang sama yaitu “balas”, maka *root word* “berbalas-balasan” adalah “balas”. Sebaliknya, pada kata “bolak-balik”,

“bolak” dan “balik” memiliki *root word* yang berbeda, maka *root word*-nya adalah “bolak-balik”

- b. Tambahkan bentuk awalan dan akhiran serta aturannya.
  - 1) Untuk tipe awalan “mem-“, kata yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”.
  - 2) Tipe awalan “meng-“, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng-”.

#### 2.4.8 Tokenization

Pada proses ini dilakukan pemisahan sebuah kalimat ke dalam bentuk pencahan kata. Kata-kata yang telah dipisahkan akan dimasukkan ke dalam sebuah index array untuk digunakan pada tahap berikutnya, dalam penelitian ini proses *tokenization* termasuk ke dalam bagian N-Gram .

### 2.5 Ngram

Ngram adalah salahsatu proses yang digunakan dalam metode pengolahan bahasa, Menurut (Fahma, 2018), N-gram merupakan urutan sebanyak N huruf dalam sebuah kata atau string. Salah satu keuntungan dari metode N-gram ini adalah bahwa bahasa bersifat independen, sebagai contoh pada kalimat **“menggunakan masker adalah salahsatu pencegahan penularan virus corona”**.

Jika dimasukkan ke dalam  $N=1$ , maka menjadi **“menggunakan, masker, adalah, salahsatu, pencegahan, penularan, virus, corona”**. Adapun  $N=1$  disebut unigram,  $N=2$  disebut bigram,  $N=3$  disebut tigram, dan  $N=4$  disebut quadgram, lihat Persamaan (2.1) berikut.

$$NgramsK = A - (N - 1) \dots (2.1)$$

dimana :

$A$  = jumlah kata dalam 1 kalimat.

$N$  = ukuran n-gram : unigram ( 1-gram), bigram (2gram), trigram (3-gram).

$NgramsK$  = jumlah n-gram dalam kalimat K.

Misalnya kalimat K adalah **“menggunakan masker adalah salahsatu pencegahan penularan virus corona”** diubah ke dalam bentuk trigram (3-gram). Jumlah  $A = 8$ ,  $N = 3$ , maka:

$$NgramsK = 8 - (3 - 1) = 8 - 2 = 6$$

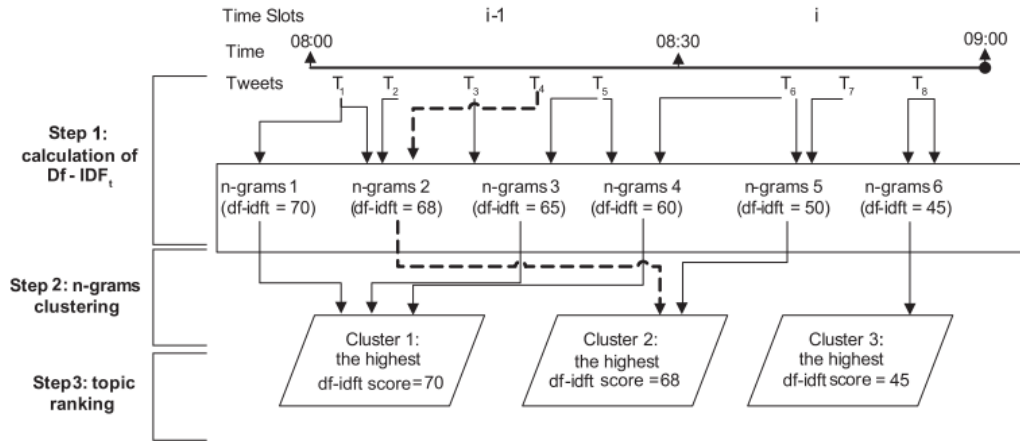
Jadi, jumlah n-gram dalam kalimat K adalah 6 yang terdiri dari:

- 1. menggunakan masker adalah
- 2. masker adalah salahsatu
- 3. adalah salahsatu pencegahan
- 4. salahsatu pencegahan penularan
- 5. pencegahan penularan virus

## 6. penularan virus corona

### 2.6 BNgram

BNgram merupakan pengembangan lebih lanjut dari studi penelitian yang dilakukan oleh (Aiello dkk., 2013), di mana klaster yang dibentuk dikembangkan menjadi *trending topics* di Twitter (Indra dkk., 2019). BNgram terdiri dari tiga tahapan antara lain yaitu perhitungan  $DF-IDF_t$  untuk mendapatkan skor setiap n-grams, klasterisasi n-grams berdasarkan jarak antar n-grams dan perangkingan topik yang didapat dari klasterisasi sebelumnya untuk mendapatkan *trending topics*. Untuk tahapan BN-Gram dalam penelitian (Indra dkk., 2019) dijelaskan dalam Gambar 2.1 berikut.



**Gambar 2.1 Ilustrasi Tahapan BN-Gram**

Pada Gambar 2.1 terdapat tiga *step*, yaitu *calculation of  $DF-IDF_t$* , *n-grams clustering*, dan *topic ranking*, tahapan tersebut merupakan tahapan utama BN-gram. Pada gambar tersebut, terdapat dua *time slot*, yaitu 08.00-08.30 dilabelkan dengan  $i-1$  dan 08.30-09.00 dilabelkan dengan  $i$ , kemudian *tweet* pada masing-masing *time slot* dilakukan ekstraksi n-gram, dan dihitung nilai  $DF-IDF_t$ , setelah *tweet* diekstraksi, kemudian dilakukan proses klasterisasi pada setiap n-gram, setelah itu ditentukan *topic ranking*, klaster yang berisi nilai  $DF-IDF_t$  tertinggi yaitu dengan nilai skor 70, maka klaster tersebut dideteksi sebagai *trending topic*.

#### 2.6.1 Perhitungan $DF-IDF_t$

$$df - idf_t = \frac{df_{i+1}}{\log\left(\frac{\sum_{j=i}^t df_{i-j}}{t} + 1\right) + 1} \cdot boost \dots (2.2)$$

penjabaran pada Persamaan (2.2) adalah sebagai berikut :

$df$  = jumlah kemunculan n-gram dalam beberapa *tweet* pada time slot  $i$ .

$df_{i-j}$  = jumlah kemunculan n-gram dalam beberapa *tweet* pada time slot  $i-j$

$t$  = jumlah semua *time slot*.

**boost** = skor penilaian terhadap n-gram yang mengandung nama orang, lokasi atau organisasi pada tahap NER. Jika sebuah n-gram mengandung



salahsatu dari ketiga kategori tersebut, maka nilai boost 1,5 selain itu 1.

#### 2.6.2 Klasterisasi Ngram

Pada BN-gram klasterisasi hirarki yang digunakan adalah metode *group average linkage* berisi dua tahapan. Tahap pertama menghitung jarak n-grams. Hal ini disimbolkan dengan  $D_{\text{man}}$  yaitu jarak di antara ngrams x dan n-grams y seperti persamaan berikut :

$$d(g_1, g_2) = 1 - \frac{A}{\min(B, C)} \dots (2.3)$$

Penjabaran pada Persamaan (2.3) adalah sebagai berikut :

$A$  = jumlah *tweet* yang sama pada ngram  $g_1, g_2$ ,

$B$  = Total *tweet* pada N-Gram ( $g_1$ ),

$C$  = Total *tweet* pada N-Gram ( $g_2$ ),

### 2.7 Named Entity Recognition (NER)

*Named Entity Recognition* (NER) merupakan turunan dari ekstraksi informasi, bertujuan untuk memudahkan mencari informasi dengan cara pemberian label nama entitas pada setiap kata dalam sebuah teks (Setiyoaji dkk., 2017). Label entitas yang memiliki makna biasanya adalah kata yang merupakan nama orang, lokasi, organisasi, jumlah uang, persentase dan tanggal. NER adalah langkah pertama menuju ekstraksi informasi terstruktur dari teks tidak terstruktur (Budi dkk., 2005). Dalam penelitian ini, NER menggunakan 3 label pada prediksi teks, diantaranya adalah nama orang (*person*), lokasi (*location*) dan organisasi (*organization*), hal ini bertujuan untuk mendapatkan nilai *boost* pada tahapan penghitungan DF-IDFt untuk setiap n-gram sebelumnya.

Merujuk pada penelitian Budi et al. (2005), proses deteksi kata yang merupakan NER, dapat diidentifikasi dengan mengikuti aturan kaidah bahasa atau disebut *feature assigment*, aturan bahasa ini dipetakan menjadi beberapa bagian, diantaranya, membuat *list of contextual features*, *list of morphological features*, dan *list of part-of-speech feature*.

Identifikasi *contextual features*, menjadi kunci utama dalam melakukan pencarian NER, *contextual feratues* ini terdapat 3 titik pencarian kata, yaitu *prefix*, *middle*, *suffix*. *Prefix* merupakan kata awalan yang menunjukkan suatu kata terdeteksi NER, misalkan “ke”, “di”, kedua awalan ini memungkinkan untuk kata setelahnya adalah kata *location*, contoh “di Jakarta”, “ke Indonesia”. *Middle* merupakan kata tengah yang menunjukkan deteksi NER, dalam *list contextual features* ini, *middle* hanya digunakan dalam deteksi *person*, contoh “bin”, “van”, yang memungkinkan kata setelah dan sebelumnya merupakan *person*. Sedangkan *suffix* yaitu akhiran kata, misalnya, “S.Kom” merupakan *suffix* dari *person*, “Utara” merupakan *suffix* dari *location*, “Company” merupakan *suffix* dari *organization*.

Selain itu juga pendeteksian huruf besar sangat mempengaruhi, dalam penulisan nama, lokasi, organisasi tentunya sebuah kata dimulai dari huruf

kapital (*Uppercase*), untuk itu hal ini dapat pula dimasukkan ke dalam sebuah aturan deteksi kata yang merupakan NER.

Sebagai contoh pada kalimat, “Presiden Habibie bertemu dengan Prof. Amien di Jakarta kemarin”, pada kalimat tersebut, *term* yang merupakan NER yaitu, “Habibie”, ”Amien”, “Jakarta”. Dimana Habibie dan Amien merupakan *person* dideteksi dari awalan huruf besar pada kata “Habibie”, serta kata sebelumnya terdapat *contextual features prefix* berupa *person title* yaitu pada kata “Presiden”. Pada kata Jakarta merupakan *location*, dideteksi dengan adanya preposisi yang termasuk ke dalam *contextual features* pada kata “di”, dalam kalimat ini, aturan deteksi bisa saja terjadi pada kalimat lain, maka disimpulkan aturan deteksi ini dapat digunakan.

**Tabel 2.4 Daftar Contextual Feature**

<i>Feature Name</i>	<i>Explanation</i>	<i>Example</i>
PPRE	<i>Person prefix</i>	Dr., pak, K.H.,
PMID	<i>Person middle</i>	bin, van
PSUF	<i>Person suffix</i>	Skom, SH
PTIT	<i>Person title</i>	Menristek, Mendagri
OPRE	<i>Organization prefix</i>	PT., Universitas
OSUF	<i>Organization suffix</i>	Ltd., company
OPOS	<i>Position in organization</i>	Ketua
OCON	<i>Other organization contextual</i>	Muktamar, Rakernas
LPRE	<i>Location prefix</i>	Kota, Propinsi
LSUF	<i>Location suffix</i>	Utara, City
LLDR	<i>Location leader</i>	Gubernur, Walikota
POLP	<i>Preposition that's usually followed by person name</i>	oleh, untuk
LOPP	<i>Preposition that's usually followed by location name</i>	di, ke, dari
DAY	<i>Day</i>	Senin, Selasa
MONTH	<i>Month</i>	Januari, Febuari

Pada Tabel 2.4 di atas merupakan contoh sebagian daftar *contextual feature* dari setiap label baik *person*, *location*, maupun *organization*. Tabel ini berisi *prefix middle* dan *suffix* dari setiap kemungkinan kata yang terdeteksi sebagai NER.

**Tabel 2.5 Daftar Morphological Feature**

<i>Feature Name</i>	<i>Explanation</i>	<i>Example</i>
<i>TitleCase</i>	<i>Begin with uppercase letter and followed by all lowercase letter</i>	Soedirman
<i>UpperCase</i>	<i>All uppercase letter</i>	KPU
<i>LowerCase</i>	<i>All lowercase letter</i>	Menuntut
<i>MixedCase</i>	<i>Uppercase and lowercase letter</i>	LelP
<i>CapStart</i>	<i>Begin with uppercase letter</i>	LelP, Muhammad
<i>ChartDigit</i>	<i>Letter and number</i>	P3K
<i>Digit</i>	<i>All number</i>	2004
<i>DigitSlash</i>	<i>Number with slash</i>	17/5
<i>Numeric</i>	<i>Number dot or comma</i>	20,5; 17.500,00

<i>Feature Name</i>	<i>Explanation</i>	<i>Example</i>
<i>NumStr</i>	<i>Number in word</i>	satu, tujuh, lima
<i>Roman</i>	<i>Roman number</i>	VII, XI
<i>TimeForm</i>	<i>Number in time format</i>	17:05, 19.30

Pada Tabel 2.5 di atas merupakan penguraian lebih lanjut pada setiap kata yang akan dilabelkan, dalam hal ini *morphological feature* mendeteksi setiap kunci pada karakter tertentu berdasarkan keberadaan huruf besar dan kecil, serta karakter numerik.

**Tabel 2.6 Daftar *part-of-speech features***

<i>Feature Name</i>	<i>Explanation</i>	<i>Example</i>
ART	Article	si,sang
ADJ	Adjective	indah, baik
ADV	Adverb	telah, kemarin
AUX	Auxiliary verb	harus
C	Conjunction	dan, atau, lalu
DEF	Definition	merupakan
NOUN	Noun	rumah, gedung
NOUNP	Personal noun	ayah, ibu
NUM	Number	satu, dua
MODAL	Modal	akan
OOV	Out of dictionary	-
PAR	Particle	kah, pun
PREP	Preposition	di, ke, dari
PRO	Pronominal	saya, beliau
VACT	Active verb	menuduh
VPAS	Passive verb	dituduh
VERB	Verb	pergi, tidur

Pada Tabel 2.6 merupakan contoh dari pemberian tag *part of speech*, pada hal ini biasa disebut sebagai *Pos Tagging*. *Pos Tagging* ini berfungsi untuk menentukan kelompok kelas kata yang nantinya dapat dideteksi menjadi sebuah *rule* baru dalam melakukan pelabelan. *Rule* dari semua pengelompokkan *feature assignment* dapat dilihat pada Tabel 2.7 di bawah ini.

**Tabel 2.7 Contoh *rule* dari proses *feature assignment***

<i>Token string</i>	<i>Token kind</i>	<i>Contextual feature</i>	<i>Morphological features</i>	<i>Part-of-speech features</i>
Ketua	WORD	OPOS	<i>TitleCase, CapStart</i>	NOUN
MPR	WORD		<i>UpperCase, CapStart</i>	OOV
,	OPUNC			
Amien	WORD		<i>TitleCase, CapStart</i>	OOV
Rais	WORD		<i>TitleCase, CapStart</i>	Noun
Pergi	WORD		<i>LowerCase</i>	VERB
Ke	WORD		<i>LowerCase</i>	PREP

<i>Token string</i>	<i>Token kind</i>	<i>Contextual feature</i>	<i>Morphological features</i>	<i>Part-of-speech features</i>
Bandung	WORD		<i>TitleCase, CapStart</i>	NOUN
Kemarin	WORD		<i>LowerCase</i>	NOUN, ADV
(	SPUNC			
24/4	NUM		<i>DigitSlash</i>	
)	EPUNC			
,	OPUNC			

Berdasarkan Tabel 2.7 di atas, maka dapat menangkap sebuah aturan baru yang mendeteksi kata sebelum ataupun sesudahnya melalui proses *feature assignment*, aturan tersebut adalah untuk pelabelan *organization*,

**IF**

Token[i].Kind="WORD" and Token[i].OPOS and  
Token[i+1].Kind="WORD" and Token[i+1].UpperCase and  
Token[i+1].OOV

**THEN** Token[i+1] = "**ORGANIZATION**".

Dengan demikian, proses pendeteksian NER berbahasa Indonesia dilakukan tahap demi tahap dan sangat bergantung dengan pendeteksian kata sebelum dan sesudahnya sesuai dengan aturan menggunakan *feature assignment*.

## 2.8 Pengujian

Dalam mengevaluasi performa dari metode yang diusulkan, perlu adanya sebuah pengujian dengan membandingkan hasil dengan data asli yang sudah valid. Dalam penelitian ini, dilakukan pengujian secara *recall* yaitu menentukan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi sebenarnya.

## 2.9 Studi Literatur

Berdasarkan landasan teori yang telah dijelaskan, terdapat penelitian yang sudah ada sebelumnya, yang dirangkum dalam Tabel 2.8 berikut:

**Tabel 2.8 Studi literatur**

No	Penulis	Judul	Jurnal	Deskripsi
1	Palupi, Pahlevi, (2020)	Analisis Sentimen Opini Publik Mengenai Covid-19 pada Twitter menggunakan Metode Naïve Bayes dan KNN	Inti Nusa Mandiri, Tahun 2020, ISSN 2685-807X	Melakukan suatu analisis sentimen untuk memberikan sebuah pandangan baru mengenai isu Covid-19, dengan data sekunder yang bersumber dari Twitter sebanyak 1098 opini dengan kata kunci Covid-19. Didapatkan hasil pengujian dengan akurasi tertinggi yaitu

No	Penulis	Judul	Jurnal	Deskripsi
				menggunakan metode Naive Bayes sebesar 63.21% sedangkan metode KNN sebesar 58.10%, dan kecenderungan opini masyarakat di Twitter condong ke positif dengan jumlah opini positif sebesar 610 sedangkan negatif 488.
2	Indra, Winarko, Pulungan, (2019)	<i>Trending topics detection of Indonesian tweets using BN-grams and Doc-p</i>	<i>Journal of King Saud University - Computer and Information Sciences</i> , Tahun 2019, ISSN 22131248	Perbandingan dua metode yaitu BNgram dan Doc-p, Untuk presisi kata kunci dengan data sekunder yang bersumber dari <i>tweet</i> . Doc-p memiliki kualitas lebih baik dibandingkan BNgram. Tetapi, untuk hasil akurasi yang tinggi dalam pendeteksian trending topics BN-gram lebih baik dibandingkan Doc-P. Hasil yang maksimal ini didapatkan dengan mengekstraksi tweet ke dalam bentuk trigram.
3	Ningtias, Sudiar, Latiar, (2020)	Tren Topik Pemberitaan PASCA Pemilihan Presiden pada Portal Berita Online	Info Bibliotheca: Jurnal Perpustakaan dan Ilmu Informasi, Tahun 2020, ISSN 2714-805X	Melakukan analisis tren topik pasca pilpres dengan data sekunder yang bersumber dari portal berita online detik.com dan tribunnews.com 2019. Hasil riset dari 2 portal berita online pascapresiden, 17 April hingga 22 Mei yakni portal detik.com paling banyak menerbitkan berita Prabowo-Sandi sebanyak 652 berita, sedangkan pasangan Jokowi-Amin 586 berita. Portal Tribunnews.com menerbitkan topik terbanyak tentang pasangan Jokowi-Amin sebanyak 537 berita, sedangkan pasangan Prabowo-Sandi mendapat 536 berita.
4	Mujilawati, (2016)	<i>Pre-Processing Text Mining Pada Data Twitter</i>	<i>Seminar Nasional Teknologi Informasi dan</i>	Membahas teknik penanganan data prampemrosesan data komentar dari Twitter. Hasil ekstraksi ini

No	Penulis	Judul	Jurnal	Deskripsi
			<i>Komunikasi</i> , Tahun 2016, ISSN 2089-9815	kemudian diujikan pada pengklasifikasian layanan sebuah perusahaan telekomunikasi serta didapatkan hasil akurasi mencapai 93,11% dengan 450 data uji.
5	Juditha, (2018)	<i>FENOMENA TRENDING TOPIC DI TWITTER: ANALISIS WACANA TWIT #SAVEHAJILULUNG</i>	<i>Jurnal Penelitian Komunikasi dan Pembangunan</i> , Tahun 2018, ISSN 1411-139X	Melakukan analisis wacana dengan data sekunder <i>tweet</i> #SaveHajiLulung yang menjadi <i>trending topics</i> . Hasil yang didapat menunjukkan makna yang ditekankan terhadap <i>tweet</i> tersebut mengandung unsur parodi, cenderung hiperbola (melebih-lebihkan) dan repetisi/alterasi (mengulang-ulangi kalimat atau kata tertentu).
6	Saputra, (2017)	<i>Implementasi teknik crawling untuk pengumpulan data dari media sosial twitter</i>	<i>Jurnal Dinamika Dotcom</i> , Tahun 2017, ISSN 2086-2652	Implementasi teknik crawling untuk mengumpulkan data sekunder yang berasal dari Twitter dengan bahasa pemrograman Java dan R yang menghasilkan perbedaan jumlah <i>tweet</i> yang berhasil dikumpulkan dari masing-masing aplikasi, untuk jumlah data pencarian yang banyak. Untuk pencarian <i>tweet</i> sebanyak 10.000 pada aplikasi Java hanya diperoleh hasil sebanyak 7.850. Sedangkan untuk aplikasi R, diperoleh 10.000 <i>tweet</i> sesuai dengan jumlah pencarian yang diinginkan hal ini dikarenakan pada aplikasi yang dikembangkan dengan Java mengalami <i>rate limit exceeded</i> .

No	Penulis	Judul	Jurnal	Deskripsi
7	Munarko, (2016)	Analisa Model <i>Named Entity Recognition</i> Tweet Bahasa Indonesia	Seminar Nasional Teknologi dan Rekayasa (SENTRA), Tahun 2016, ISSN 2527-6050	Melakukan analisis pengenalan entitas bernama terhadap data sekunder yang bersumber dari Twitter untuk merancang model menggunakan Algoritma Conditional Random Field (CRF). Didapat hasil dengan 1000 data pengujian yaitu klasifikasi dengan nilai rata-rata precision 92,8% dan recall 83,9%. Namun nilai rata-rata ini berpotensi untuk dinaikkan dengan memanfaatkan POS Tagger, dimana dalam penelitian ini digunakan algoritma Hidden Markov Model (HMM).
8	Simarangkir, (2017)	Studi Perbandingan Algoritma - Algoritma <i>Stemming</i> Untuk Text Bahasa Indonesia	Jurnal Inkofar, Tahun 2017, ISSN 2581-2920	Membuat perbandingan hasil uji <i>stemming</i> pada Algoritma Nazief Adriani, Algoritma Arifin, dan Setiono, Algoritma Vega dan Algoritma Tala pada data sekunder yang diambil dari artikel berita media elektronik, dengan pengujian data pembandingan dari Kamus Besar Bahasa Indonesia. Didapat hasil yang menunjukkan akurasi tertinggi terdapat pada pengujian dengan Algoritma Nazief Adriani.
9	Annisa, Munarko dan Azhar, (2016)	Peringkasan Tweet Berdasarkan Trending Topic Twitter Dengan Pembobotan TF-IDF	Jurnal Kinetik, Tahun 2016, ISSN 2503-2267	Analisa <i>trending topic</i> menggunakan algoritma TF-IDF dan <i>Single Linkage Agglomerative Hierarchical Clustering</i> dengan data sekunder 50 <i>tweet</i> berbahasa Indonesia, sedangkan untuk pengujian digunakan 30 data <i>trending topic</i> , dengan hasil keluaran yaitu <i>trending topic</i> “WordCancerDay”, “Chris Martin” dan “HitzSirkusPagi”.
10	Abdurrahman, (2019)	<i>Clustering</i> Data Kredit Bank	JUSTINDO (Jurnal Sistem dan	Klasterisasi terhadap nasabah disuatu bank dengan algoritma

No	Penulis	Judul	Jurnal	Deskripsi
		Menggunakan Algoritma Agglomerative Hierarchical Clustering Average Linkage	Teknologi Informasi Indonesia), Tahun 2019, ISSN 2502-5724	<i>agglomerative hierarchical clustering average linkage</i> , dengan data primer dalam penelitian yaitu data nasabah sebanyak 1000 <i>instances</i> , yang kemudian dijadikan sebagai data <i>training</i> sebanyak 25 %, 50 %, dan 75 %, sedangkan untuk data <i>testing</i> digunakan keseluruhan data, dan diperoleh hasil 3 <i>cluster</i> yakni cluster-1 berjumlah 806 (98%), cluster-2 berjumlah 5 (1%), cluster-3 berjumlah 9 (1%). Pada cluster-4 dan cluster-5 masing-masing hanya beranggotakan 1 (0%), sehingga dalam hal ini <i>cluster</i> tidak terbentuk.



### BAB III METODOLOGI PENELITIAN

#### 3.1 Data Penelitian

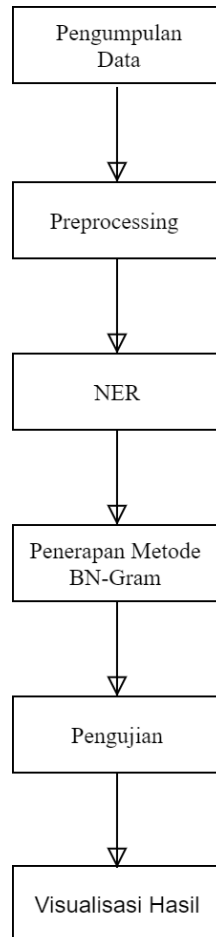
Dataset atau data yang digunakan dalam penelitian ini bersumber dari data *tweet* Twitter pada tanggal 20 sampai 30 Juli 2020 yang dilakukan proses *crawling* menggunakan *library* Tweepy dari bahasa pemrograman Python, *library* ini berfungsi untuk mengambil data *tweet* pada Twitter dengan akses menggunakan API Key yang didapatkan dari akun *developer* Twitter. Dataset ini diambil pada rentang waktu tertentu dan dengan kata kunci diantaranya : *Covid19*, *Corona*, *Kemenkes*, *BNPB*, *Gugus Tugas Relawan* yang merupakan kata kunci keterkaitan *tweet* dengan topik yang diambil, yaitu masa pandemi *Covid-19*. Berikut adalah contoh *record* dari data *tweet* hasil *crawling* dapat dilihat pada Tabel 3.1 berikut:

**Tabel 3.1 Sampel data *tweet***

Tweet ID	Username	Text	Timestamp
123412935378 2378496	TaQur	Negara lain yg gk percaya kredibilitas standar pengujian Corona Virus di Indonesia kl sampai Indonesia 0 kasus ... Wisatawan jd enggan datang kl ada travel warning dr negaranya ...	2020-03-01 14:52:21
123412999953 3228032	Dian Fajriyah	Tp corona blm ada vaksin nya.. dan masyarakat indonesia masih blm sadar pentingnya germas krn memang udh kebiasaan dari sananya <a href="https://twitter.com/DORKyungsoo4/status/1233397983732891652">https://twitter.com/DORKyungsoo4/status/1233397983732891652</a>	2020-03-01 14:53:34
123412981846 0938241	Kafir Garis Lurus	Kadang gua mikir. Lebih baik corona mewabah di indonesia, dan seluruh bumi hingga seluruh negara saling mengisolasi diri. Kota kota di lockdown sampai sebulan, pabrik ditutup kendaraan dibatasi. Kita mengkarantina sebulan di rumah disubsidi pemerinth. Diksh jahe ama supermi.	2020-03-01 14:54:30
123413028843 6862976	adityasl	btw mau ngumpulin orang yang gapercaya Indonesia blm kena corona wkwkwk	2020-03-01 14:56:04

### 3.2 Penerapan Metode

Dalam membangun sebuah sistem deteksi *trending topic* Twitter menggunakan metode BN-Gram, terdapat beberapa tahapan tertentu yang menjadi rancangan utama, dimana rancangan ini merepresentasikan proses tahapan awal hingga akhir sistem berjalan, yang terdapat pada Gambar 3.1 berikut.



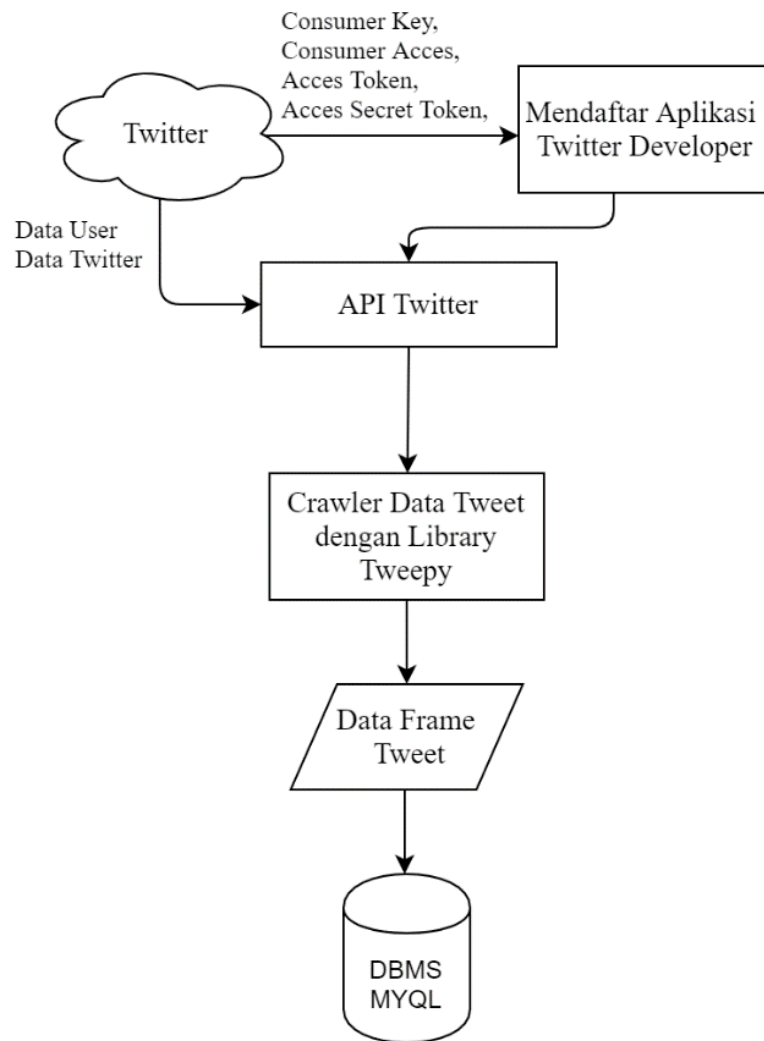
**Gambar 3.1 Tahapan Metode**

Pada Gambar 3.1, proses *crawling* dilakukan untuk mendapatkan *dataset*, dalam penelitian ini *dataset* menggunakan *tweet*. Selanjutnya, *tweet* yang sudah dalam bentuk *excel*, kemudian diimpor ke dalam database untuk dilakukan *preprocessing*, pada tahap *preprocessing* dilakukan sebagai pemilihan kata yang layak untuk diproses ke dalam tahapan utama BN-Gram. Kemudian *tweet* bersih yang sudah dilakukan *preprocessing*, masuk ke tahap implementasi BN-Gram, yaitu tahapan utamanya adalah, pemetaan N-Gram, penghitungan DF-IDFt, klusterisasi, dan perangkungan. Pada tahap perangkungan, kluster dengan skor DF-IDFt tertinggi akan berada pada peringkat atas, dan termasuk ke dalam identifikasi *trending topic*,

selanjutnya hasil *trending topic* akan diujikan dengan pengujian *Ground Truth*, untuk mengetahui nilai keakuratan atau presisi dari *trending topic* usulan.

### 3.2.1 Pengumpulan Data

Pada tahapan ini dilakukan proses *crawling* pada data stream Twitter, prosesnya adalah mendapatkan API Key Twitter melalui akun *developer* Twitter untuk mendapatkan akses data *tweet* saat *library* Tweepy dijalankan. Data *tweet* yang berhasil dikumpulkan kemudian disimpan dalam sebuah *dataframe* yang sudah dikoneksikan dengan *database* MySQL, kemudian *tweet* yang telah disimpan dalam dataframe dimasukkan ke dalam tabel *database* MySQL. Untuk ilustrasi penerapan metode pengumpulan data dapat dilihat pada Gambar 3.2 di bawah ini.



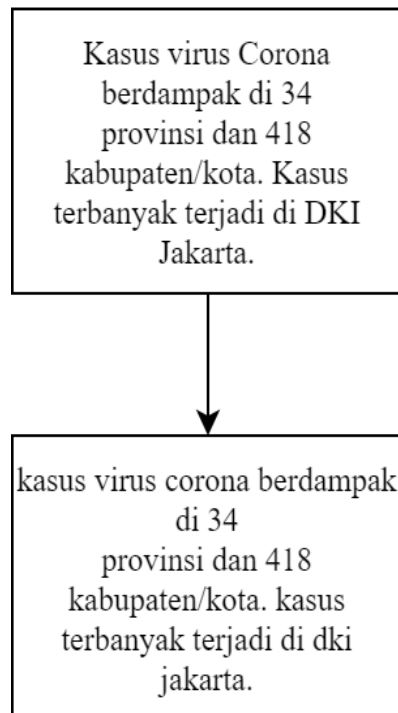
**Gambar 3.2 Metode Pengumpulan Data**

### 3.2.2 Preprocessing

Pada tahapan *preprocessing*, dilakukan beberapa proses yang bertujuan untuk menghasilkan *tweet* bersih supaya dapat memudahkan pada proses tahapan selanjutnya, proses penerapan *preprocessing* pada penelitian ini diantaranya yaitu:

#### a. Casefolding

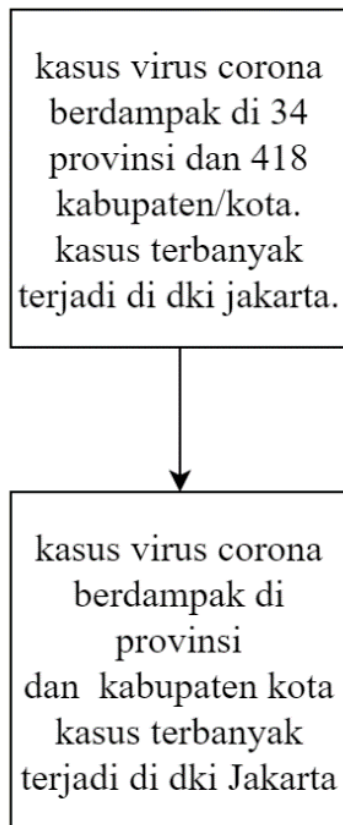
Pada Gambar 3.3 dilakukan penyetaraan kata yang mengandung huruf besar untuk diubah menjadi huruf kecil, misalnya Kasus menjadi kasus, Corona menjadi corona, dan seterusnya.



**Gambar 3.3 Casefolding**

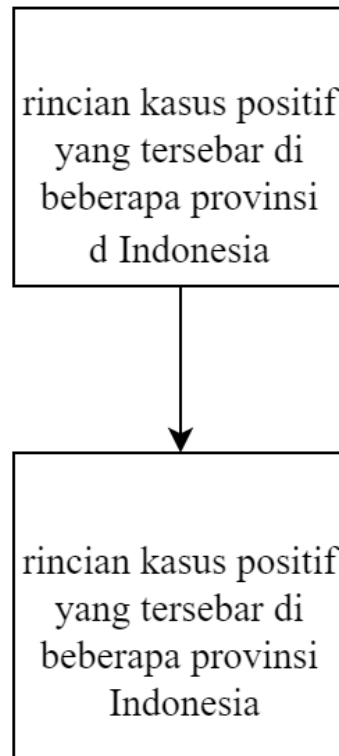
#### b. Menghapus karakter selain a-z.

Pada Gambar 3.4 menjelaskan bahwa karakter selain a sampai z atau karakter selain huruf, karakter tersebut dihilangkan, seperti contoh 34, 418, / (garis miring), dan seterusnya sehingga hanya menyisakan karakter huruf saja.



**Gambar 3.4 Menghapus karakter selain a-z**

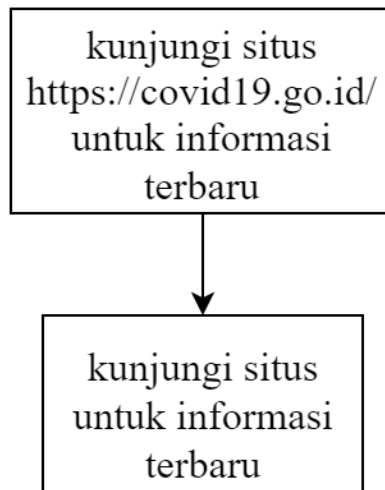
- c. Menghapus teks dengan 1 karakter.  
Pada Gambar 3.5 terdapat karakter dengan jumlah 1 huruf yaitu "d", karakter ini dianggap tidak memiliki arti maka dari itu karakter "d" dihilangkan.



**Gambar 3.5 Menghapus 1 karakter**

d. Menghapus *URL*

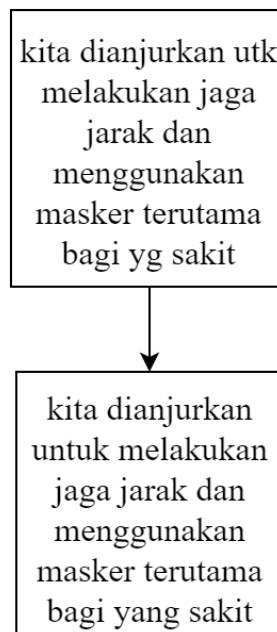
Pada Gambar 3.6 terdapat *link* <https://covid19.go.id>, pada tahap ini link dihilangkan karena dianggap tidak memiliki makna dan seringkali disisipkan dalam *tweet* dengan tujuan mempromosikan situs tersebut.



**Gambar 3.6 Menghapus 1 karakter**

e. Mengganti *slangword*

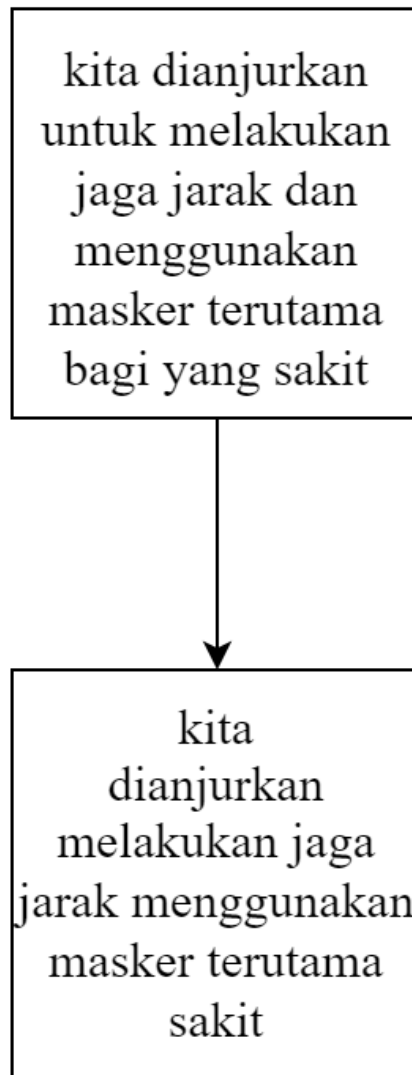
Pada Gambar 3.7 terdapat kata tidak baku yaitu “utk” dan “yg”, kata yang tidak baku tersebut biasanya berupa singkatan atau bahasa yang kekinian, untuk itu supaya kata-kata dalam teks setara dengan EYD, maka kata tersebut digantikan dengan kata baku yang seharusnya yaitu menjadi “untuk” dan “yang”, pergantian kata-kata ini berdasarkan kamus yang terdapat dalam *library slangword*.



**Gambar 3.7 Slangword**

f. Menghapus *stopword*

Dalam penelitian Tala, (2003), terdapat kumpulan kata-kata yang termasuk ke dalam *stoplist* yaitu kata umum yang dianggap tidak terlalu memiliki makna yang penting dan kemunculan kata ini sangat tinggi frekuensinya, seperti pada contoh teks di bawah ini yaitu pada kata “untuk”, “dan”, “bagi”, kata tersebut dihilangkan karena masuk ke dalam daftar *stopword*.

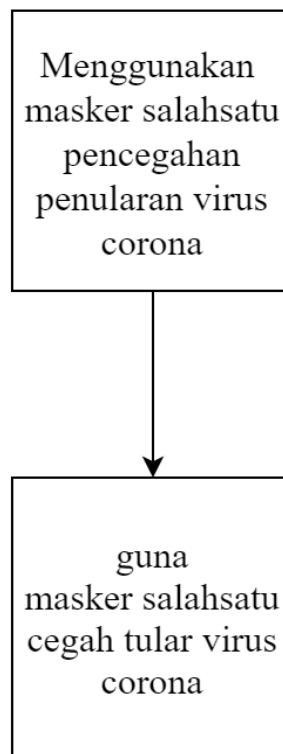


**Gambar 3.8 Slangword**

g. *Stemming*

Pada proses ini, *stemming* menggunakan algoritma dari Nazief dan Adriani mengubah kata menjadi kata dasar sebagai contoh pada teks di bawah ini yaitu, “menggunakan” diubah menjadi kata dasar yaitu “guna”, “pencegahan” menjadi “cegah” dan seterusnya, lihan Gambar 3.9.

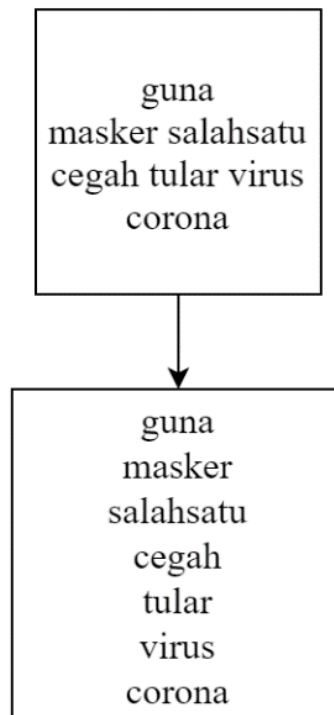




**Gambar 3.9 Stemming**

h. *Tokenization*

Pada Gambar 3.10, proses tokenisasi dilakukan pemisahan sebuah kalimat ke dalam bentuk pecahan per satu kata. Kata-kata yang telah dipisahkan kemudian dimasukkan ke dalam sebuah index array untuk digunakan pada tahap berikutnya yaitu proses pelabelan NER (*Named Entity Recognition*).



**Gambar 3.10 Tokenization**

### 3.2.3 Named Entity Recognition (NER)

Dalam proses NER ini, dilakukan pelabelan terhadap token atau kata yang termasuk ke dalam 3 entitas, yaitu nama orang (*person*), lokasi (*location*), dan organisasi (*organization*), hal ini diperlukan untuk mendapatkan nilai *boost* yang berbeda dari kata biasa pada perhitungan BN-Gram.

Berdasarkan teori yang telah dijelaskan dalam sub bab (2.9), bahwa penerapan NER berbahasa Indonesia dilakukan dengan proses *feature assignment* yang kemudian hasil akhirnya dalam bentuk sebuah *rule* atau aturan pelabelan kata, misalnya pada contoh kalimat “**Presiden Jokowi pergi ke Jakarta guna melakukan perjalanan dinas bersama kepala BNPB**”. Pada Tabel 3.2 di bawah ini berisi ekstraksi informasi pada kalimat tersebut jika dilakukan proses *feature assignment*.

**Tabel 3.2 Hasil Pelabelan Feature Assignment**

<i>Token String</i>	<i>Contextual Feature</i>	<i>Morphological Features</i>	<i>Part-of-Speech Features</i>	<i>Label</i>
Presiden	OPOS	<i>TitleCase</i>	<i>NOUN</i>	
Jokowi	-	<i>TitleCase</i>	<i>OOV</i>	<i>PERSON</i>
pergi	-	<i>LowerCase</i>	<i>VERB</i>	
ke	LOPP	<i>LowerCase</i>	<i>PREP</i>	

<i>Token String</i>	<i>Contextual Feature</i>	<i>Morphological Features</i>	<i>Part-of-Speech Features</i>	<i>Label</i>
Jakarta	-	<i>TitleCase</i>	<i>NOUN</i>	<i>LOCATION</i>
guna	-	<i>LowerCase</i>	<i>NOUN</i>	
melakukan	-	<i>LowerCase</i>	<i>VERB</i>	
perjalanan	-	<i>LowerCase</i>	<i>NOUN</i>	
dinas	-	<i>LowerCase</i>	<i>NOUN</i>	
bersama	-	<i>LowerCase</i>	<i>ADV, VERB</i>	
Kepala	OPOS	<i>TitleCase</i>	<i>NOUN</i>	
BNPB	-	<i>UpperCase</i>	<i>OOV</i>	<i>ORGANIZATION</i>

Pada kata “Jokowi” dalam kalimat diatas dideteksi sebagai *PERSON*, aturannya yaitu pertama kata sebelumnya adalah “Presiden” yang merupakan sebuah *prefix* dari penamaan seseorang termasuk ke dalam *PERSON TITLE* dalam kamus *contextual feature*. Setelah itu kata “Jokowi” dalam *Part of Speech* termasuk ke ala *OOV (Out of Vocabulary)*, artinya kata tersebut tidak ada dalam kamus bahasa, kemudian kata “Jokowi” pun diawali dengan huruf besar dan masuk ke dalam aturan *Morphological Feature* yaitu *Title Case*. Dengan demikian didapatlah aturan seperti berikut.

```

if (Token[i] = Title Case AND Token[i] = OOV)
{
    if (Token[i-1] = PTIT)
        Then Token[i] = “ORGANIZATION”
}

```

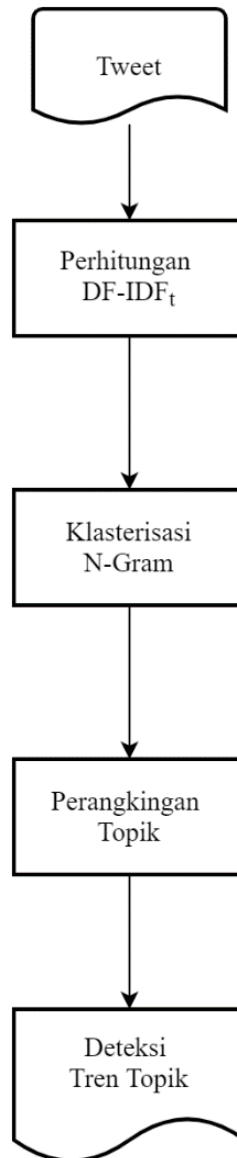
Dengan demikian pula pada kata “Bandung” dilabelkan dengan label “*LOCATION*” dan “BNPB” dilabelkan dengan “*ORGANIZATION*”, pelabelan NER didapat dari pendeteksian token sebelum atau sesudahnya dengan melihat kamus *Contextual Feature*, kemudian dikuatkan dengan *Part of Speech* dan pola *Morphological Feature*. Hasil pelabelan NER pada kalimat “**Presiden Jokowi pergi ke Jakarta guna melakukan perjalanan dinas bersama kepala BNPB**” adalah sebagai berikut.

Presiden Jokowi[**PERSON**] pergi ke Jakarta[**LOCATION**] guna melakukan perjalanan dinas bersama kepala BNPB[**ORGANIZATION**].

Pada proses pelabelan NER, menggunakan kosa-kata *Part of Speech* berbahasa Indonesia, *Part of Speech* ini untuk menguatkan aturan deteksi, disimpulkan bahwa kata yang termasuk NER merupakan kata yang memiliki jenis *Noun* (kata benda), dan *OOV* (tidak terdapat dalam kosa kata).

#### 3.2.4 BN-Gram

Dalam menentukan hasil deteksi *trending topic* dengan metode BN-Gram, terdapat 3 tahapan utama, yaitu perhitungan  $DF-IDF_t$ , klasterisasi n-gram, dan perangkingan topik, adapun tahapan utama BN-Gram diilustrasikan dalam Gambar 3.11 berikut.



**Gambar 3. 11 Tahapan Utama BN-Gram**

Prosesnya adalah, sebelum dilakukan tahapan BN-Gram, *tweet* dari hasil prapemrosesan diekstraksi ke dalam bentuk n-grams. Dalam penelitian ini n-gram yang digunakan adalah bigram atau  $N=2$ , yaitu memetakan token menjadi 2 pasang kata. Sebagai contoh pada kalimat ***K = Presiden Jokowi pergi ke Jakarta guna melakukan perjalanan dinas bersama kepala BNPB***, diubah ke dalam bentuk bigram (2-gram) dan dimasukkan ke dalam persamaan (2.1) maka menjadi:

Jumlah  $A = 12$ ,  $N = 2$ , maka:

$$NgramsK = 12 - (2 - 1) = 12 - 1 = 11$$

Jadi, jumlah n-gram dalam kalimat *K* adalah 6 yang terdiri dari:

1. **Presiden - Jokowi**
2. **Jokowi - pergi**
3. **pergi - ke**
4. **ke - jakarta**
5. **Jakarta - guna**
6. **guna - melakukan**
7. **melakukan - perjalanan**
8. **perjalanan - dinas**
9. **dinas - bersama**
10. **bersama - kepala**
11. **kepala – BNPB**

Hasil dari ekstraksi n-gram tersebut merupakan sebuah *keyword* yang diberikan indeks dan diimplementasikan menggunakan *Lucene*. Pemberian indeks ditujukan untuk memudahkan dalam pencarian n-grams. Kemudian untuk selanjutnya setiap ngrams dilakukan pemetaan pada rentang waktu yang berbeda, lalu dilanjutkan ketahap penghitungan  $DF-IDF_t$  pada setiap ngrams dengan persamaan (2.2).

Selanjutnya, jika sudah mendapatkan skor  $DF-IDF_t$  pada masing-masing n-gram di setiap rentang waktu, dilanjutkan dengan klasterisasi ngram, Pada BN-Gram klasterisasi hirarki yang digunakan adalah metode *group average linkage* berisi dua tahapan. Tahap pertama menghitung jarak n-gram. Hal ini disimbolkan dengan  $D_{man}$  yaitu jarak di antara n-gram  $x$  dan n-gram  $y$  seperti yang telah dijelaskan pada persamaan (2.3).

Setelah mendapatkan nilai jarak kedekatan antar ngram, maka Tahap selanjutnya dilakukan pemilihan jarak dua kelompok berdasarkan rata-rata dari dua kelompok. Sebelumnya, terlebih dahulu dilakukan pemilihan dari jarak dua kelompok terkecil berdasarkan matrik jarak kedekatan n-grams sampai dengan nilai kesamaan lebih dari atau sama dengan nilai *threshold* 0.50, lalu dilakukan penghapusan baris dan kolom yang bersesuaian, kemudian n-grams digabungkan menjadi sebuah klaster baru, dan lakukan iterasi tersebut sampai klaster terakhir yang tidak dapat terbentuk.

Tahap terakhir adalah perangkingan dengan menentukan skor dari masing-masing klaster yang terbentuk dari penghitungan jarak berdasarkan skor  $DF-IDF_t$  pada setiap n-gram. Klaster yang berisi n-gram dengan skor  $DF-IDF_t$  tertinggi maka klaster tersebut diidentifikasi sebagai *trending topics*.

### 3.3 Rancangan Pengujian

Pengujian dilakukan untuk mengetahui tingkat akurasi dan presisi dari metode usulan. Pada penelitian ini pengujian dilakukan dengan cara membandingkan hasil deteksi *trending topic* usulan dengan sekumpulan topik dalam *ground truth*. Adapun *ground truth* yang dimaksud adalah data pembandingan yang sudah valid yaitu data tren topik yang bersumber dari media-media informasi terpercaya, seperti pada *headline* berita kompas.com, kumparan.com dan lain-lain.

Pengujian *trending topic* pada penelitian ini menggunakan tiga pengukuran yaitu :

#### 3.3.1 Topic Recall (TR)

*Topic Recall* adalah Perbandingan antara identifikasi *trending topic* dengan topik yang ada pada *ground truth* (Persamaan (3.4)).

#### 3.3.2 Keyword Precision (KP)

*Keyword Precision* perbandingan antara keyword *trending topic* yang konsisten dengan kata kunci *ground truth* dibandingkan dengan total keseluruhan kata kunci pada *trending topic* metode usulan (Persamaan (3.5)).

#### 3.3.3 Keyword Recall (KR) :

*Keyword Recall* adalah perbandingan antara kata kunci *trending topic* yang konsisten dengan kata kunci *ground truth* dibandingkan dengan total keseluruhan kata kunci pada *ground truth* (Persamaan (3.6))

Adapun pengukuran TR, KP dan KR didefinisikan dalam persamaan sebagai berikut:

$$TR = \frac{|GT \cap BT|}{|GT|} \quad \dots\dots\dots(3.4)$$

$$KP = \frac{|KGT \cap KBT|}{|KBT|} \quad \dots\dots\dots(3.5)$$

$$KR = \frac{|KGT \cap KBT|}{|KGT|} \quad \dots\dots\dots(3.6)$$

dengan,

- a. *GT (Ground Truth Topic)* adalah sekumpulan topik pada suatu *ground truth*.
- b. *BT (Trending Topic)* adalah sekumpulan *trending topic* metode usulan.
- c. *KGT (Keyword Ground Truth Topic)* adalah sekumpulan kata kunci pada *ground truth*.
- d. *KBT (Keyword Trending Topic)* adalah sekumpulan kata kunci pada *trending topic* metode usulan.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Lingkungan Percobaan

Agar aplikasi yang telah dikembangkan dapat berjalan dengan semestinya, dibutuhkanlah perangkat dengan spesifikasi tertentu, adapun dalam penelitian ini menggunakan spesifikasi perangkat diantaranya:

#### 4.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang mendukung aplikasi ini berjalan dengan baik sebagai berikut:

- a. *Processor* : Intel (R) Core (TM) i5-5200U CPU @ 2.20GHz
- b. *RAM* : 12 GB DDR3
- c. *Harddisk* : 500 GB
- d. *VGA* : NVIDIA GeForce 930M

#### 4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang mendukung aplikasi ini berjalan dengan baik sebagai berikut:

- a. Sistem Operasi : Windows 10 Pro 64-bit
- b. *IDE* : Visual Studio Code v.1.37.1, Spyder3
- c. *DBMS* : Mysql Database
- d. *Web Server* : Apache (XAMPP v3.2.3, PHP versi 7.3.4)
- e. *Browser* : Google Chrome, Mozilla Firefox

### 4.2 Implementasi Metode dan Langkah Pengujian

#### 4.2.1 Tahap Pengumpulan Data

Tahap pengumpulan data merupakan tahap yang pertama kali dilakukan sebelum tahap prapemrosesan data. Pada penelitian ini, data yang dikumpulkan yaitu *tweet* yang berisi kata kunci yang berkaitan dengan topik pembahasan *Covid-19*. Kata kunci yang digunakan diantaranya yaitu: Corona, *Covid-19*, Kemenkes, BNPB, Gugus Tugas Relawan.

*Tweet* yang berhasil dikumpulkan dengan menggunakan *library* Tweepy berupa teks dengan informasi seperti, *tweet id*, teks *tweet*, *timestamp*, *username*, *link* dll. yang kemudian diimpor ke dalam *database* MYSQL untuk kemudian dilakukan prapemrosesan data.

#### 4.2.2 Tahapan BN-Gram

Setelah *tweet* melalui tahap *preprocessing*, kemudian masuk ke tahap perancangan metode BN-Gram, berikut penjabaran dari metode BN-Gram.



**Tabel 4.1 Sampel Data *Tweet***

<i>Tweet(T<sub>i</sub>)</i>	<i>Timestamp</i>	<i>Tweet</i>
1	2020-03-01 14:55:21	corona belum ada vaksin masyarakat indonesia masih belum sadar pentingnya hidup sehat
2	2020-03-01 14:60:01	corona indonesia masih nol karena memang tidak ada atau dikiranya masuk angin biasa
3	2020-03-01 14:67:30	untuk apa beli vaksin virus corona indonesia tidak terdampak
4	2020-03-01 14:74:00	kurang edukasi corona masyarakat indonesia banyak termakan hoax

Data *tweet* pada Tabel 4.1 terdapat tiga kolom yaitu *tweet (i)* yang berarti daftar *tweet* ke sekian, *timestamp* yang berarti waktu *tweet* dipublikasikan, dan *text* yang merupakan isi *tweet* bersih. Selanjutnya *tweet* dipetakan berdasarkan *time slot*, pada contoh ini *time slot tweet* yaitu rentang waktu 10 menit.

**Tabel 4.2 Pemetaan *time slot tweet***

<i>Time</i>	<i>Time Slot</i>	<i>Index Tweet (T<sub>n</sub>)</i>	<i>Tweet</i>
2020-03-01 — 14:54:30 s.d 14:64:30	1	T <sub>1</sub>	corona belum ada vaksin masyarakat indonesia masih belum sadar pentingnya hidup sehat
		T <sub>2</sub>	corona indonesia masih nol karena memang tidak ada atau dikiranya masuk angin biasa
2020-03-01 — 14:64:30 s.d 14:74:30	2	T <sub>1</sub>	untuk apa beli vaksin virus corona indonesia tidak terdampak
		T <sub>2</sub>	kurang edukasi corona masyarakat indonesia banyak termakan hoax

Pada Tabel 4.2 kolom *time* merupakan pemetaan waktu berdasarkan *time slot* yaitu 10 menit, waktu tersebut dimulai dari *timestamp* pertama kali *tweet* dipublikasikan dan diakhiri sampai *tweet timestamp* paling akhir. *Tweet* yang dipublikasikan berdasarkan rentang *time* akan dimasukkan ke dalam *slot time* nya masing-masing, kemudian pada kolom *index tweet (T<sub>n</sub>)* merupakan pemetaan *index tweet* pada masing-masing *time slot*. *Tweet* yang sudah dipetakan,

kemudian dilakukan ekstraksi n-gram, pada contoh ini menggunakan dua n-gram (bigram), ekstraksi n-gram dapat dilihat pada kolom *Bigram* Tabel 4.3.

**Tabel 4.3 Ekstraksi Bigram**

<i>Time</i>	<i>Time Slot</i>	<i>Index Tweet (T<sub>n</sub>)</i>	<i>Bigram</i>
2020-03-01 —— <b>14:54:30</b> s.d <b>14:64:30</b>	1	<b>T<sub>1</sub></b>	{ corona belum ; belum ada ; ada vaksin ; vaksin masyarakat ; masyarakat indonesia ; indonesia masih ; masih belum ; belum sadar ; sadar pentingnya ; pentingnya hidup ; hidup sehat ; }
		<b>T<sub>2</sub></b>	{ kasus corona ; corona indonesia ; indonesia masih ; masih nol ; nol karena ; karena memang ; memang tidak ; tidak ada ; ada atau ; atau dikiranya ; dikiranya masuk ; masuk angin ; angin biasa ; }
2020-03-01 —— <b>14:64:30</b> s.d <b>14:74:30</b>	2	<b>T<sub>1</sub></b>	{ untuk apa ; apa beli ; beli vaksin ; vaksin virus ; virus corona ; corona indonesia ; indonesia tidak ; tidak berdampak ; }
		<b>T<sub>2</sub></b>	{ kurang edukasi ; edukasi corona ; corona masyarakat ; masyarakat indonesia ; }

<i>Time</i>	<i>Time Slot</i>	<i>Index Tweet (T<sub>n</sub>)</i>	<i>Bigram</i>
			indonesia banyak ; banyak termakan ; termakan hoax ; }

Selanjutnya tweet yang sudah diekstraksi dalam bentuk bigram diproses ke dalam tiga tahapan utama BNgram.

a. Penghitungan DF-IDF<sub>t</sub>

Pada tahap DF-IDF<sub>t</sub> dilakukan perangkingan BN-Gram dengan persamaan (3.2). Perangkingan bigram menggunakan penghitungan skor DF-IDF<sub>t</sub> pada setiap *time slot* ke-i. Perhitungan DF-IDF<sub>t</sub> dibuat berdasarkan frekuensi kemunculan bigram pada rentang waktu tertentu dibandingkan dengan logaritma dari rata-rata frekuensi jumlah keseluruhan frekuensi bigram pada satu rentang waktu sebelumnya. Ilustrasi dari proses perangkingan bigram sesuai persamaan (2.2) dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Skor DF-IDF<sub>t</sub>**

<i>Time</i>	<i>Time Slot Ke-i</i>	<i>Bigram</i>	<i>df</i>	<i>t</i>	<i>df - idf<sub>t</sub></i>
2020-03-01 —— 14:54:30 until 14:64:30	1	corona belum ;	1	2	2
		belum ada ;	1		2
		ada vaksin ;	1		2
		vaksin masyarakat ;	1		2
		masyarakat indonesia ;	1		3
		indonesia masih ;	2		4.5
		masih belum ;	1		2
		belum sadar ;	1		2
		sadar pentingnya ;	1		2

<i>Time</i>	<i>Time Slot Ke-i</i>	<b>Bigram</b>	<i>df</i>	<i>t</i>	<i>df</i> – <i>idf<sub>t</sub></i>
		pentingnya hidup ;	1		2
		hidup sehat ;	1		2
		corona indonesia ;	1		3
		indonesia masih ;	2		4.5
		masih nol ;	1		2
		nol karena ;	1		2
		karena memang ;	1		2
		memang tidak ;	1		2
		tidak ada ;	1		2
		ada atau ;	1		2
		atau dikiranya ;	1		2
		dikiranya masuk ;	1		2
		masuk angin ;	1		2
		angin biasa ;	1		2
2020-03-01 — 14:64:30 until 14:74:30	2	untuk apa ;	1		2
		apa beli ;	1		2
		beli vaksin ;	1		2
		vaksin virus ;	1		2

<i>Time</i>	<i>Time Slot Ke-i</i>	<b>Bigram</b>	<i>df</i>	<i>t</i>	$df - idf_t$
		virus corona ;	1		2
		corona indonesia ;	1		2.13
		indonesia tidak ;	1		3
		tidak terdampak ;	1		2
		kurang edukasi ;	1		2
		edukasi corona ;	1		2
		corona masyarakat;	1		2
		masyarakat indonesia ;	1		2.13
		indonesia banyak ;	1		3
		banyak termakan ;	1		2
		termakan hoax ;	1		2

Pada Tabel 4.4, kolom *t* merupakan jumlah terbentuknya *time slot* yaitu 2 *time slot* kemudian kolom *df* adalah jumlah kemunculan n-gram pada *time slot tweet* tersebut dan kolom  $df - idf_t$  merupakan hasil perhitungan n-gram berdasarkan persamaan (2.2). bigram yang menunjukkan “masyarakat-indonesia”, “Indonesia-masih”, dan “corona-indonesia” memiliki skor yang berbeda dari dominan skor pada bigram lainnya, hal ini dikarenakan ada perbedaan frekuensi kemunculan **pada *time slot ke-i***, dan nilai *boost* pada bigram tersebut, nilai *boost* pada bigram “masyarakat-indonesia” dan “corona-indonesia” dan “indonesia-masih” adalah 1.5, karena pada kata “indonesia” merupakan kata yang menunjukkan NER (*Named Entity Recognition*), yaitu nama lokasi. Kemudian pada bigram “Indonesia-masih” memiliki nilai ***df* = 2**, ini dikarenakan jumlah kemunculan bigram tersebut adalah 2 kali pada *time slot* yang sama dan nilai pada bigram “masyarakat-indonesia” dan “corona-indonesia” memiliki nilai  **$df_{i-j} = 1$**

karena bigram tersebut muncul satu kali pada satu rentang waktu sebelumnya, untuk contoh perhitungannya terdapat dalam Tabel 4.5.

**Tabel 4.5 Sampel Perhitungan Skor DF IDF<sub>t</sub>**

<b>Term (N-gram)</b>	<b>Rumus</b>	<b>Hasil</b>
masyarakat indonesia	$df - idf_t = \frac{1 + 1}{\log\left(\frac{1}{2} + 1\right) + 1} \cdot 1,5$	<b>2.13</b>
indonesia masih	$df - idf_t = \frac{2 + 1}{\log\left(\frac{0}{2} + 1\right) + 1} \cdot 1,5$	<b>4,5</b>
vaksin virus	$df - idf_t = \frac{1 + 1}{\log\left(\frac{0}{2} + 1\right) + 1} \cdot 1$	<b>2</b>
corona indonesia	$df - idf_t = \frac{1 + 1}{\log\left(\frac{1}{2} + 1\right) + 1} \cdot 1,5$	<b>2.13</b>

b. Klasterisasi N-Gram

Pada BN-Gram klasterisasi hirarki yang digunakan yaitu metode *group average linkage* yang berisi dua tahapan. Tahap pertama adalah menghitung *distance* atau jarak n-gram, hal ini disimbolkan dengan  $D_{man}$  yaitu jarak di antara n-gram  $x$  dan n-gram  $y$ . Penjabaran dari Persamaan (2.3) dapat diimplementasikan dengan memetakan n-gram ke dalam setiap *tweet* yang dapat dilihat pada Tabel 4.6.

**Tabel 4.6 Pemetaan N-Gram**

<b>Time Slot</b>	<b>Tweets</b>	corona -indonesia (n-gram 1)	masyarakat – Indonesia (n-gram 2)
1	$T_1$	0	1
	$T_2$	1	0
2	$T_1$	1	0
	$T_2$	0	1
<b>Total</b>		<b>2</b>	<b>2</b>

Dari Tabel 4.6 menghasilkan pemetaan dengan jumlah n-gram 2, yaitu “corona - indonesia” dan “masyarakat - indonesia”, n-gram tersebut dihasilkan dari bigram pada Tabel 4.4 yang jumlah kemunculannya lebih dari satu pada *tweet time slot* ke-2, baik dalam *time slot* yang sama, maupun *time slot* ke-1 maka

didapatlah 2 n-gram tersebut. Kemudian, pada masing-masing n-gram terdapat biner 1 dan biner 0, biner 1 menunjukkan bahwa dalam *tweet* ke- $n$  terdapat n-gram pada kolom tersebut, begitu juga sebaliknya, biner 0 menunjukkan *tweet* tidak mengandung n-gram yang dimaksud. Sebagai contoh *tweet* ke-1( $T_1$ ), n-gram “corona - indonesia” nilai biner 0, artinya *tweet* ke-1 tidak terdapat n-gram tersebut, sedangkan *tweet* ke-2, n-gram “corona - indonesia” nilai biner 1 karena *tweet* tersebut mengandung n-gram tersebut, dan begitu juga seterusnya. Dari hasil pemetaan tiap-tiap n-gram, tahap selanjutnya adalah menghitung jarak kedekatan n-gram, dan hasil penghitungan jarak n-gram terdapat pada Tabel 4.7.

**Tabel 4.7 Pemetaan Jarak**

$D_{man}$	n-gram 1	n-gram 2
n-gram 1	0	0
n-gram 2	0	0

Berdasarkan persamaan (2.3) jarak kedekatan tiap n-gram menghasilkan nilai pada Tabel 4.7 dapat diilustrasikan sebagai berikut:

$$d(1,2) = 1 - \frac{0}{\min(2,2)} = 0$$

Pada  $d(1,2)$  jarak kedekatan antara n-grams 1 dan ngrams 2,  $A$  bernilai nol karena jumlah *tweet* yang mengandung n-gram 1 dan n-gram 2 tidak ada. Perhitungan  $d(x,y)$  juga berlaku untuk perhitungan jarak kedekatan antar n-gram  $x$  dengan n-gram  $y$  yang bersesuaian. Untuk n-gram dengan nilai  $x$  dan  $y$  sama, maka nilainya otomatis 0. Dengan demikian pada Tabel 4.7 mengidentifikasikan bahwa setiap n-gram memiliki jarak kedekatan yang bernilai 0, hal ini terjadi karena sampel *tweet* yang diilustrasikan sangat sedikit, sehingga tingkat frekuensi kemunculan sebuah n-gram dalam setiap *tweet* pun juga sedikit.

Tahap kedua, yaitu pemilihan jarak dua kelompok berdasarkan rata-rata dari dua kelompok. Sebelum menghitung rata-rata dari dua kelompok n-gram, terlebih dahulu dilakukan pemilihan dari jarak dua kelompok yang terkecil berdasarkan matrik dari Tabel 4.7 sampai dengan nilai kesamaan kurang dari atau sama dengan nilai *threshold* 0,50. Terpilih n-gram 1 dan 2 dengan nilai 0, sehingga kedua n-gram ini digabungkan menjadi sebuah klaster dan dilabelkan dalam kelompok (12). Dengan demikian proses klasterisasi pada n-gram Tabel 4.7 selesai dikarenakan jumlah klaster n-gram yang terbentuk hanya klaster (12).

Untuk melihat proses klasterisasi n-gram yang lebih detail, disediakan sampel pemetaan jarak n-gram pada Tabel 4.8 berikut.

**Tabel 4.8 Sampel pemetaan n-gram**

$D_{man}$	n-gram 1	n-gram 2	n-gram 3	n-gram 4
n-gram 1	0	0	0.5	0
n-gram 2	0	0	0	0
n-gram 3	0.5	0	0	0
n-gram 4	0	0	0	0

Pada Tabel 4.8 terdapat data sampel untuk pemetaan n-gram yang akan dilakukan proses klasterisasi. Tahapannya yaitu memilih jarak terkecil antara dua n-gram dengan nilai kesamaan kurang dari atau sama dengan nilai *threshold* 0,50, sama seperti pada Tabel 4.7, terpilih n-gram 1 dan 2, kemudian 2 n-gram ini digabungkan menjadi n-gram (12) Tahap berikutnya adalah menghitung jarak antar n-gram (12) dengan kelompok lain yang tersisa, yaitu n-gram 3 dan n-gram 4.

$$D_{(12)3} = average \{d_{13}; d_{23}\} = average \{0.5; 0\} = \mathbf{0.25}$$

$$D_{(12)4} = average \{d_{14}; d_{24}\} = average \{0; 0\} = \mathbf{0}$$

Setelah mendapatkan nilai rata-rata, kemudian hapus baris dan kolom matriks yang bersesuaian dengan n-gram 1 dan 2, kemudian tambahkan baris dan kolom dari klaster (12) yang sebelumnya terbentuk, lihat Tabel 4.9.

**Tabel 4.9 Iterasi ke-1**

$D_{man}$	n-gram 12	n-gram 3	n-gram 4
n-gram 12	0	0.25	0
n-gram 3	0.25	0	0
n-gram 4	0	0	0

Selanjutnya lakukan iterasi kembali dengan memilih kembali jarak terkecil seperti sebelumnya, kemudian digabungkan kembali menjadi sebuah klaster. Klaster n-gram selanjutnya yaitu n-grams 12 dengan ngrams 4 dengan nilai 0, maka gabungkan n-gram menjadi klaster (124), dan lakukan penghitungan rata-rata.

$$D_{(124)3} = average \{d_{13}; d_{23}; d_{4,3}\} = average \{0.5; 0; 0\} = \mathbf{0.1666}$$



Hapus kembali matriks yang bersesuaian dengan n-gram 12 dan n-gram 3, maka matriks akan seperti pada Tabel 4.10.

**Tabel 4.10 Iterasi ke-2**

$D_{man}$	n-gram 124	n-gram 4
n-gram 124	0	0.1666
n-gram 3	0.1666	0

Pada Tabel 4.10 berada pada iterasi ke-2 dan iterasi ini merupakan iterasi paling terakhir, karena hanya menyisakan 2 kluster terakhir, untuk kemudian dilakukan *topic ranking*.

Jadi kelompok (124) dan kelompok (3) merupakan hasil akhir pengelompokan n-gram berdasarkan perhitungan jarak antar n-gram menggunakan *group average linkage* hingga terbentuk suatu kluster. Hasil pembentukan kluster menghasilkan satu kluster yang dituliskan dengan  $K_1 = \{n_1, n_2, n_4\}$ , sedangkan n-gram 3 tidak termasuk ke dalam kluster, dikarenakan n-gram tunggal yang tidak tergabung dari n-gram lainnya.

c. Perangkingan Topik

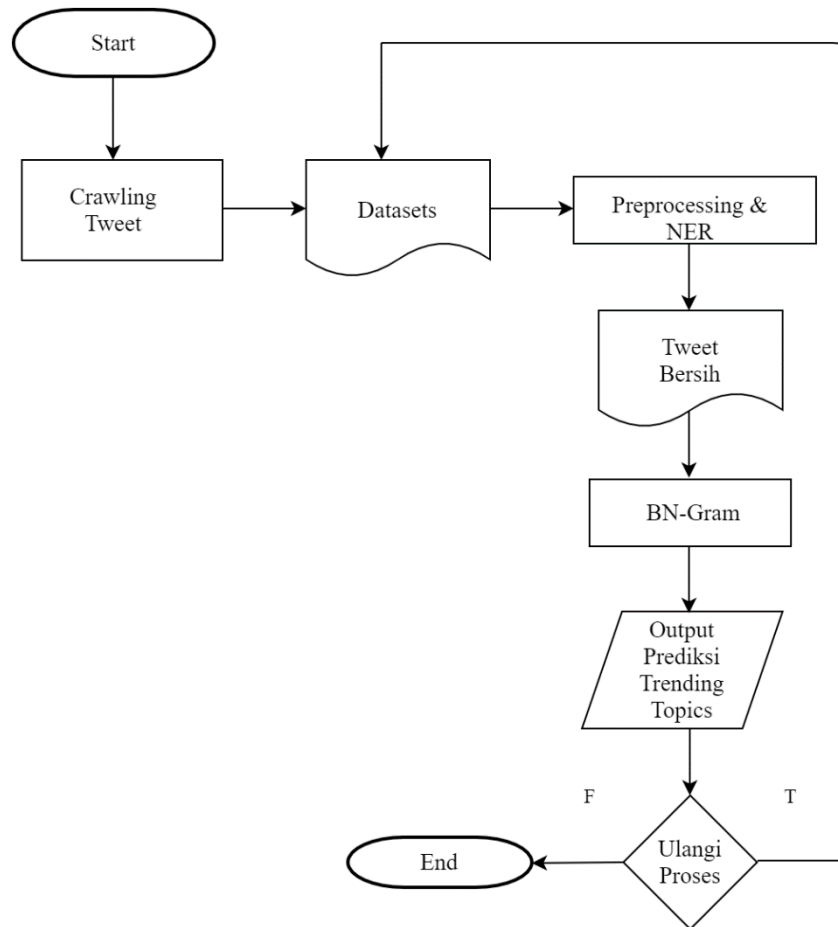
Kluster dilakukan perangkingan berdasarkan skor  $DF-IDF_t$  setiap n-gram. Kluster yang berisi n-gram dengan skor  $DF-IDF_t$  tertinggi maka kluster tersebut diidentifikasi sebagai *trending topics*. Berdasarkan Tabel 4.4 n-grams 1 dan n-grams 2 pada Tabel 4.7 memiliki skor tertinggi dibandingkan n-grams pada time slots ke-1 yaitu dengan skor  $DF-IDF_t = 2.13$ . Oleh karena itu, kluster yang berisi n-grams 1, n-grams 2 yaitu “corona-indonesia” dan “masyarakat-indonesia” menjadi *topic ranking*.

### 4.3 Flowchart Tahapan Metode

*Flowchart* adalah suatu bagan atau simbol-simbol yang menggambarkan alur kerja atau urutan proses pada suatu program. Berikut adalah penjabaran *flowchart* pada tahapan metode yang digunakan:

#### 4.3.1 Flowchart Keseluruhan Sistem

Pada *flowchart* ini menjelaskan tahapan mengenai berjalannya sistem, mulai dari tahapan pengumpulan data, hingga mendapatkan hasil tren topik.

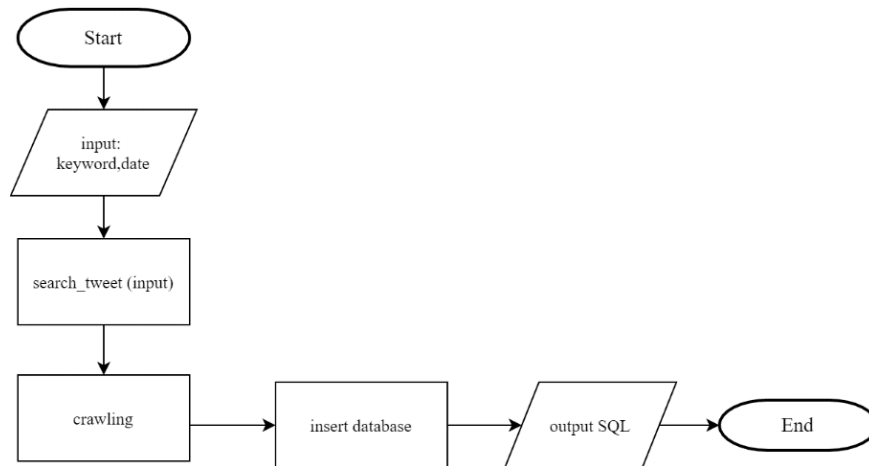


**Gambar 4. 1 Flowchart keseluruhan sistem**

Pada gambar 4.1 menjelaskan proses keseluruhan sistem yang dibuat dengan tahap awal yaitu proses pengumpulan data twitter atau *crawling tweet*, kemudian sebelum masuk ke tahapan BN-Gram, *tweet* yang merupakan dataset ini dilakukan tahapan pembersihan data atau *preprocessing* dan dilakukan *labelling* NER, setelah dataset menjadi *tweet* bersih, selanjutnya dilakukan proses tahapan utama BN-Gram, hingga menghasilkan *output*, berupa tren topik. Proses yang sama akan berulang jika dimasukkan dataset yang berbeda.

#### 4.3.2 Flowchart Crawling

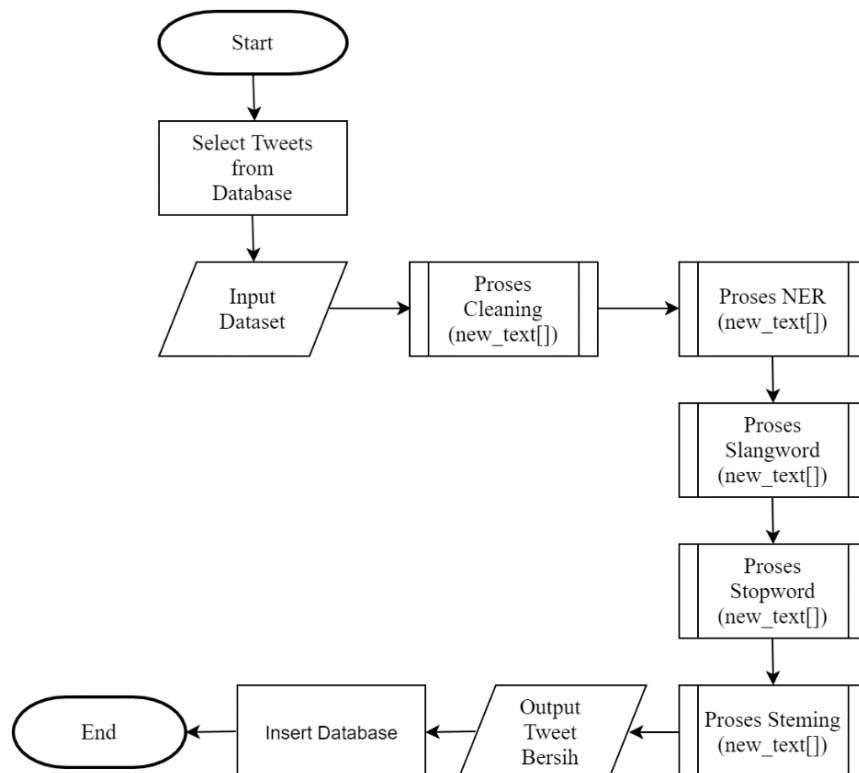
Pada *flowchart* ini, menjelaskan tahapan proses pengumpulan data atau *crawling* data *tweet* dimulai dari memasukkan kata kunci dan tanggal *tweet*, kemudian dilakukan pencarian *tweet* berdasarkan parameter inputan pada fungsi *search\_tweet(input)*. Selanjutnya jika proses pencarian selesai, maka *tweet* tersebut dilakukan *crawling* dan hasil pengumpulan data akan disimpan dalam *data frame* kemudian dimasukkan ke dalam *database*.



**Gambar 4. 2 Flowchart crawling tweet**

#### 4.3.3 Flowchart Preprocessing

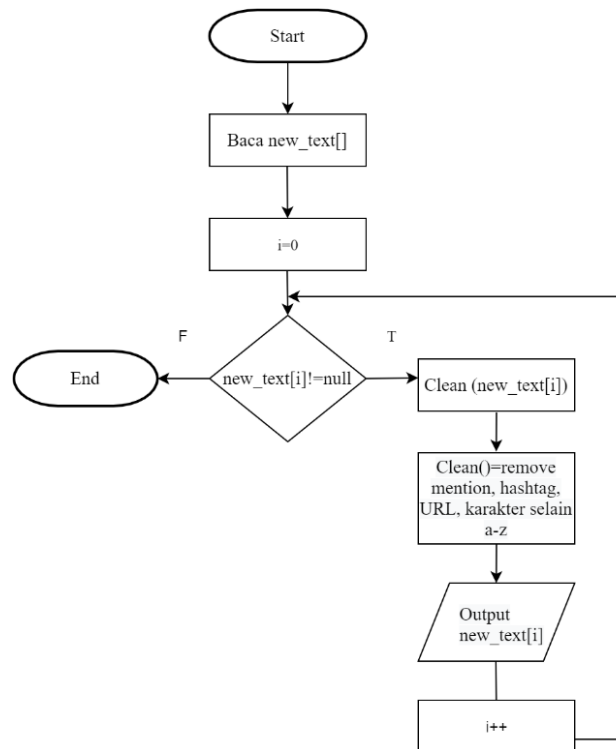
Pada flowchart ini, menjelaskan tahapan *preprocessing* data *tweet*. Data *tweet* hasil *crawling* yang sudah diimpor ke dalam *database* di pilih untuk kemudian dilakukan sub proses *cleaning*, *labelling* NER, *slangword*, *stopword*, dan *stemming* setelah itu *tweet* bersih disimpan ke dalam array `new_text[]`, dan dimasukkan kembali ke dalam *database*.



**Gambar 4. 3 Flowchart preprocessing tweet**

#### 4.3.4 Flowchart Cleaning

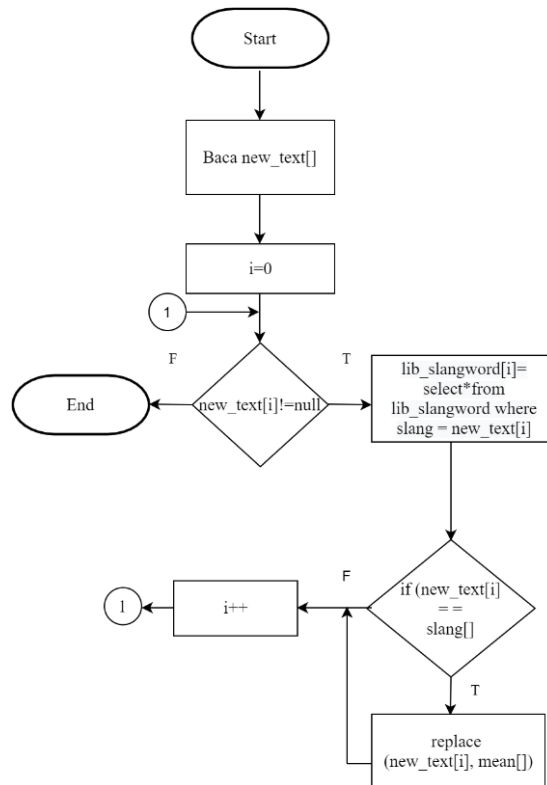
Pada *flowchart* ini, menjelaskan sub proses tahapan dari *preprocessing* yaitu *cleaning*. Pada tahapan ini, *tweet* yang sudah tersimpan pada array `new_text[]` satu per satu dilakukan penghilangan pada mention “@”, hashtag “#”, URL, dan karakter selain a-z dengan fungsi `clean()`. Proses ini dijalankan sampai *tweet* habis atau *tweet* paling terakhir.



**Gambar 4. 4 Flowchart cleaning**

#### 4.3.5 Flowchart Slangword

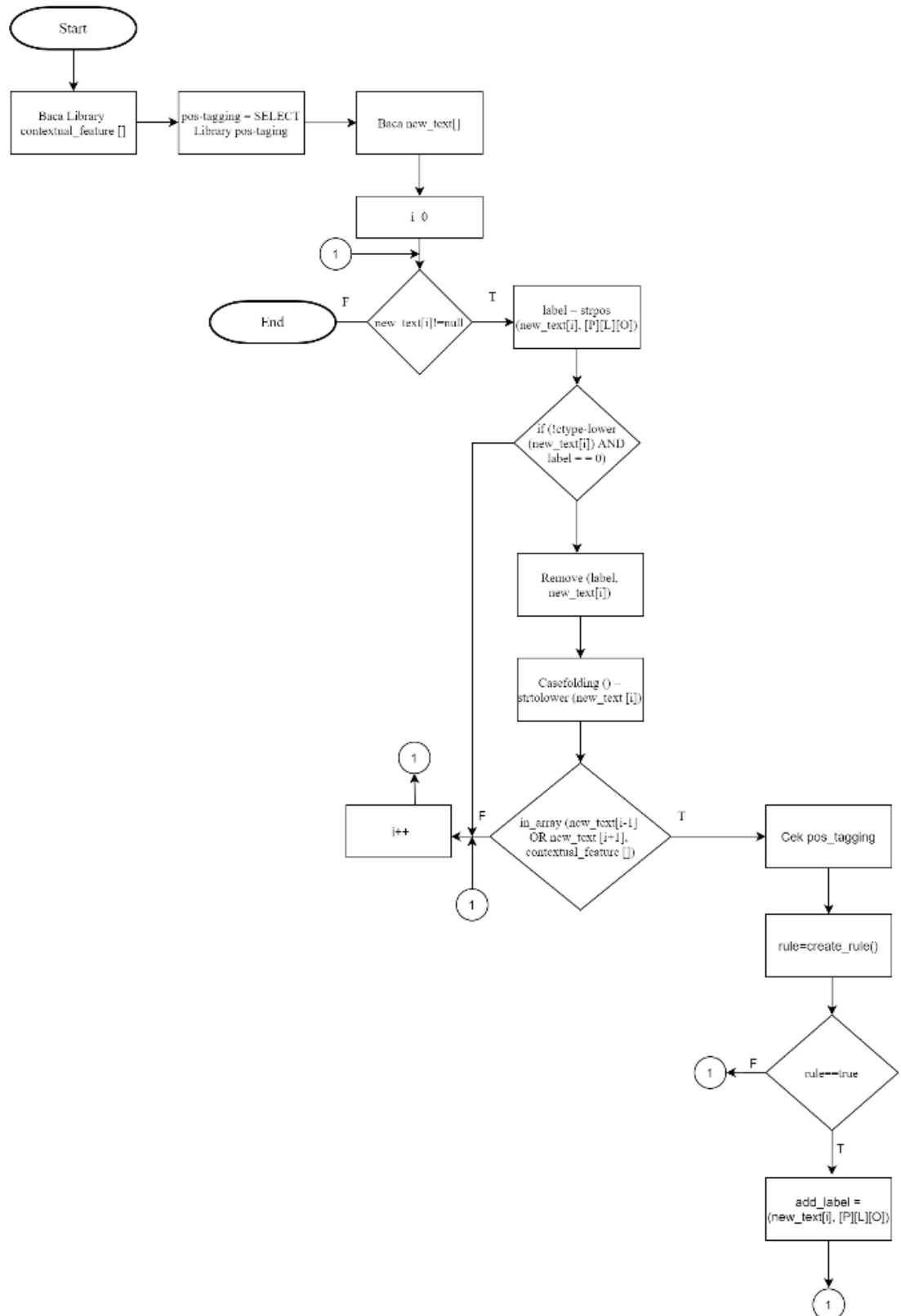
Pada *flowchart* ini, menjelaskan sub proses tahapan dari *preprocessing* yaitu *slangword*. Proses *slangword* diawali dengan membaca *tweet* pada array `new_text[]` dan memanggil *library slangword* yang sama pada array `new_text[i]`, jika ada yang sama maka dilakukan perintah `replace(new_text[i], mean)` yang artinya mengganti *tweet* tersebut dengan arti pada kamus *library slangword*. Proses ini berulang sampai *tweet* habis.



**Gambar 4. 5 Flowchart slangword**

#### 4.3.6 Flowchart Labelling NER

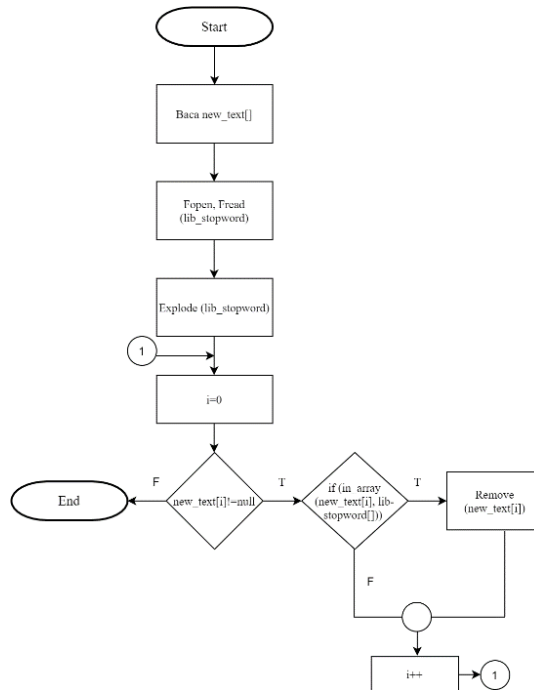
Pada *flowchart* ini, menjelaskan sub proses tahapan dari *preprocessing* yaitu *labelling* kata yang mengandung nama atau NER. Proses ini diawali dengan membaca *library* contextual feature, kemudian tweet dalam array `new_text[i]` dilakukan pencarian label dan deteksi huruf besar (*uppercase*), jika label ada, maka label dihapus dahulu dengan fungsi `remove(label, new_text[i])`, lalu dilakukan `casefolding()` untuk mengubah *tweet* menjadi *lowercase*. Selanjutnya mencari *contextual feature* pada array `new_text[i-1]` dan `new_text[i+i]`, jika ada maka dilanjutkan dengan mengecek *POS Tagging* lalu membuat *rule* pada token tersebut, jika *rule* benar, maka ditambahkan label pada fungsi `add_label = (new_text[i], [P],[L],[O])`, dengan P adalah *person*, L adalah *location* dan O adalah *organization*.



**Gambar 4. 6 Flowchart NER**

#### 4.3.7 Flowchart Stopword

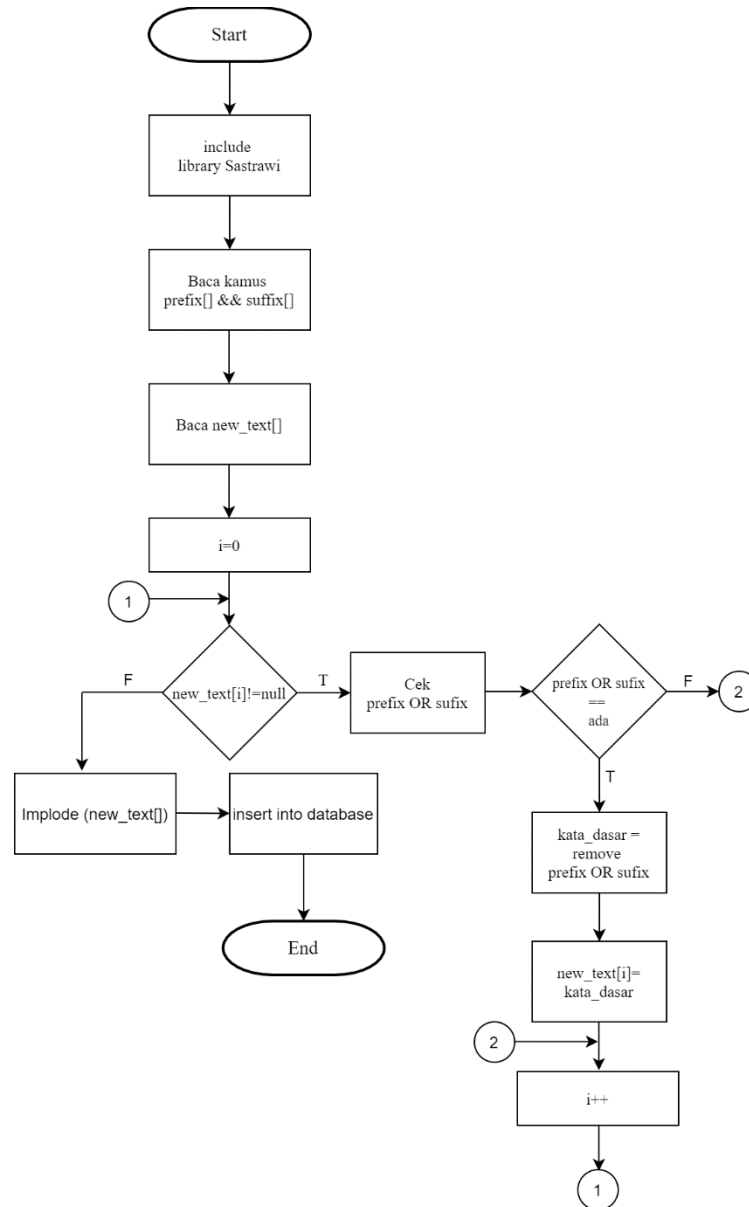
Pada *flowchart* ini, menjelaskan sub proses tahapan dari *preprocessing* yaitu *stopword*, proses ini diawali dengan membaca *library stopwords* yang berisi kamus *stopword*. Kemudian dilakukan pencarian dengan fungsi `in_array (new_text[i], lib-stopword[])`, artinya jika sebuah kata dalam array `new_text[i]` terdapat dalam kamus *stopword*, maka kata tersebut dihapus menggunakan fungsi `remove(new_text[i])`. Proses ini berulang sampai *tweet* habis.



**Gambar 4. 7 Flowchart Stopword**

#### 4.3.8 Flowchart Stemming

Pada *flowchart* ini, menjelaskan sub proses tahapan dari *preprocessing* yaitu *stemming*, proses ini diawali dengan memasukkan *library* Sastrawi untuk fungsi *stemming*, kemudian membaca *tweet* pada array `new_text[]` dan kamus *prefix/suffix*. Pada setiap text dilakukan proses cek *prefix/suffix*, jika ada maka hapus dan `new_text[i]` menjadi sebuah kata dasar. Proses tersebut dijalankan sampai *tweet* habis, kemudian dilakukan penggabungan menjadi kalimat kembali dengan fungsi `implode(new_text[])`, lalu hasil *tweet* bersih diimpor ke dalam *database*.



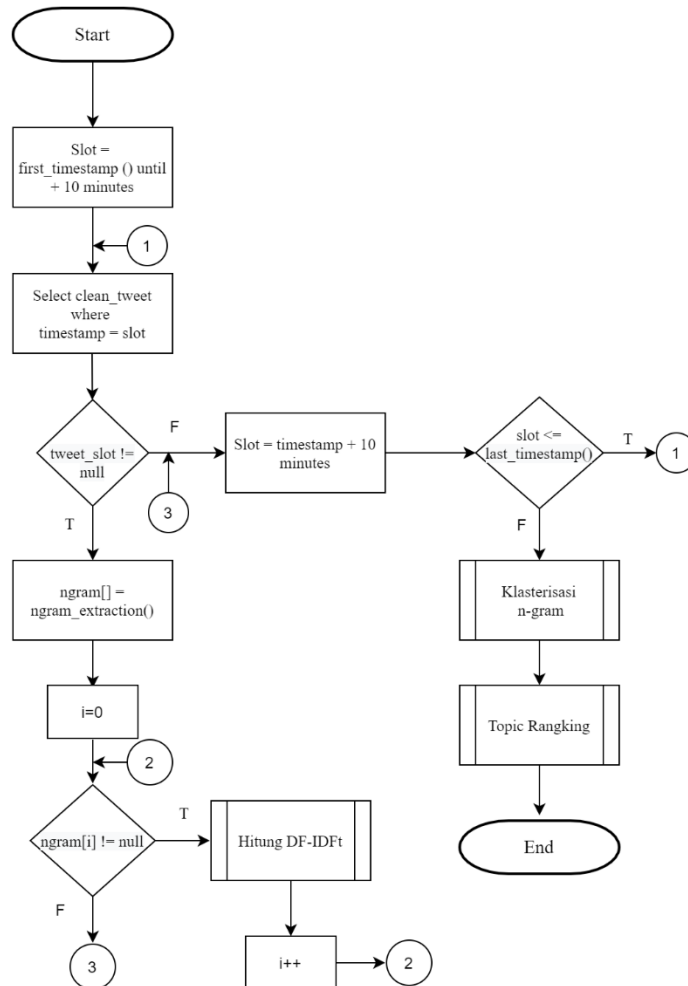
**Gambar 4. 8 Flowchart Stemming**

#### 4.3.9 Flowchart BN-Gram

Pada *Flowchart* ini menjelaskan tahapan utama pada BN-Gram, yaitu penghitungan DF-IDFt, klasterisasi n-gram, dan perangkingan n-gram. Pada kasus ini *tweet* dipetakan berdasarkan slot waktu 10 menit dihitung dari *record tweet* yang memiliki *timestamp* paling awal, kemudian jika dalam satu slot tersebut *tweet* tersedia, maka *tweet* dilakukan ekstraksi n-gram atau pemecahan per kata, proses ini diberi nama *ngram\_extraction()*. Setelah itu, n-gram dilakukan perhitungan DF-IDFt sebagaimana telah dijelaskan pada persamaan 3.2, lalu jika slot waktu pada *tweet* sudah mencapai *timestamp* paling akhir, proses selanjutnya adalah mengklasterkan n-gram.



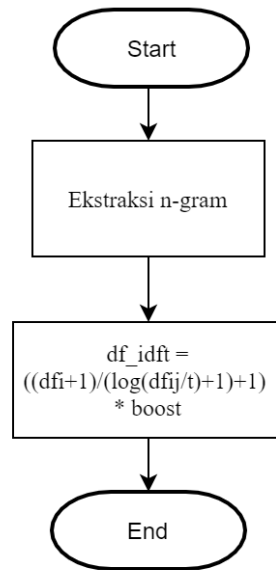
Pada fungsi Klasterisasi = merge(ngram[]) dilakukan pengabungan ngram dimana jarak antar n-gram adalah  $\leq 0.5$ , fungsi ini dilakukan iterasi sampai menghasilkan 2 klaster terakhir, setelah itu proses terakhir adalah perangkingan, pada fungsi Klastering(skor df-idft) dilakukan pembobotan n-gram berdasarkan skor df-idft pada slot tersebut, skor tertinggi maka klaster tersebut merupakan hasil *trending topic*.



**Gambar 4. 9 Flowchart BN-Gram**

#### 4.3.10 Flowchart Perhitungan DF-IDF<sub>t</sub>

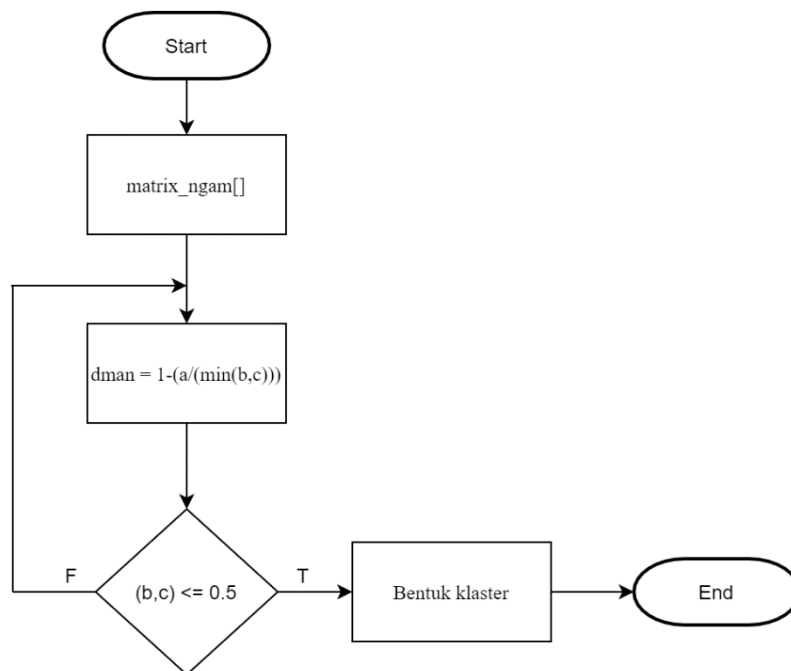
Pada *flowchart* ini menjelaskan sub tahapan dari *Flowchart* BN-Gram pada Gambar 4.9 yaitu sub proses Hitung DF-IDF<sub>t</sub>.



**Gambar 4. 10 Flowchart Perhitungan DF-IDF<sub>t</sub>**

#### 4.3.11 Flowchart Klasterisasi N-Gram

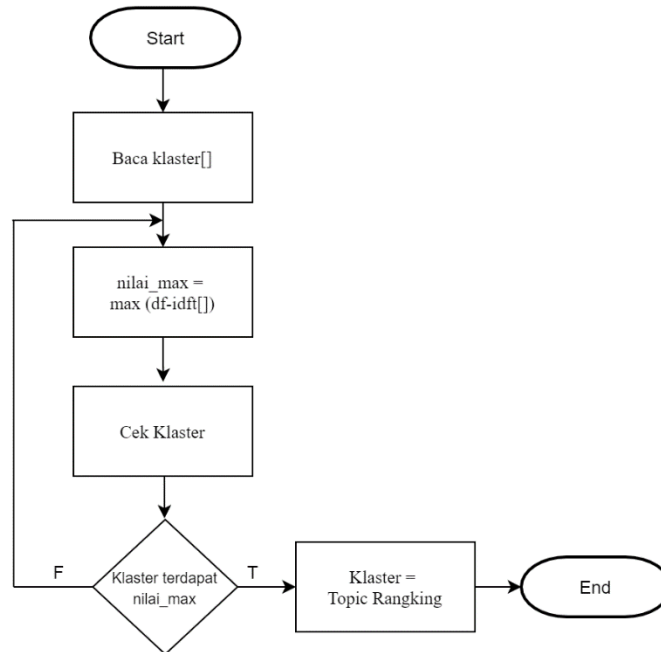
Pada *flowchart* ini menjelaskan sub tahapan dari *Flowchart* BN-Gram pada Gambar 4.9 yaitu sub proses Klasterisasi N-Gram.



**Gambar 4. 11 Flowchart Klasterisasi N-Gram**

#### 4.3.12 Flowchart *Topic Ranking*

Pada *flowchart* ini menjelaskan sub tahapan dari *Flowchart* BN-Gram pada Gambar 4.9 yaitu sub proses *Topic Ranking*.



**Gambar 4. 12** *Flowchart Topic Ranking*

#### 4.4 Algoritme Tahapan Metode

Algoritme adalah urutan atau alur tahapan proses yang dijabarkan dalam bentuk tulisan, algoritme ini merupakan representasi dari *flowchart* yang telah dijelaskan sebelumnya.

##### 4.4.1 Algoritme Keseluruhan Sistem

Pada algoritme ini dijelaskan tentang proses sistem secara keseluruhan pada metode yang digunakan.

1. Proses *crawling tweet*
2. Baca dokumen *datasets*
3. Proses *preprocessing & NER*
4. Hasil dokumen *tweet* bersih
5. Proses BN-Gram
6. Output prediksi tren topik
7. *If* (ulangi proses)
8. *Kembali ke nomor 2*
9. *End if*
10. *End*

#### 4.4.2 Algoritme *Crawling*

Pada algoritme ini dijelaskan tentang proses tahapan pada *crawling tweet*.

1. *Input* : keyword, date
2. Mencari *tweet*
3. Proses *Crawling tweet*
4. Simpan ke dalam database
5. *Output* : *SQL*
6. *End*

#### 4.4.3 Algoritme *Preprocessing*

Pada algoritme ini dijelaskan tentang proses tahapan pada preprocessing secara keseluruhan.

1. Ambil *tweet* dari database
2. *Input datasets*
3. Lakukan proses *cleaning()*
4. Lakukan proses *NER()*
5. Lakukan proses *slangword()*
6. Lakukan proses *stopword()*
7. Lakukan proses *stemming()*
8. *Output tweet* bersih
9. Proses *insert* ke database
10. *End*

#### 4.4.4 Algoritme *Cleaning*

Pada algoritme ini dijelaskan tentang proses sub tahapan pada preprocessing yaitu *cleaning tweet*.

1. Baca array *new\_text[]*
2. *i*=0
3. *if* (*new\_text[i]* ada)
4.     hapus *mention*, *hashtag*, *URL*, karakter selain *a-z*
5.     *Output new\_text[i]*
6.     *i++*
7. Kembali ke nomor 3
8. *End if*
9. *End*

#### 4.4.5 Algoritme *Slangword*

Pada algoritme ini dijelaskan tentang proses sub tahapan pada preprocessing yaitu *slangword*.

1. Baca array *new\_text[]*
2. *i*=0

```

3.  if (new_text[i] ada)
4.      select lib_slangword where slang= new_text[i]
5.          if (new_text == slang)
6.              replace(new_text[i],mean)
7.              i++
8.              Kembali ke nomor 3
9.          Endif
10.      Kembali ke nomor 7
11.  Endif
12.  End

```

#### 4.4.6 Algoritme NER

Pada algoritme ini dijelaskan tentang proses sub tahapan pada *preprocessing* yaitu ekstraksi NER (*Named Entity Recognition*).

```

1.  Baca library contextual_feature[]
2.  Select Library POS Tagging
3.  Baca array new_text[]
4.  i=0
5.  if (new_text[i] ada)
6.      label = strpos(new_text[i],[P][L][O])
7.      if (new_text[i] diawali uppercase AND ada label)
8.          Remove(label, new_text[i])
9.          strtolower(new_text[i])
10.         if (new_text[i+1] AND new_text[i-1]
            terdapat contextual_feature[])
11.             Cek POS Tagging
12.             Buat Rule()
13.             If (Rule() == True)
14.                 Tambahkan label P,L,O
15.             Endif
16.             i++
17.             kembali ke nomor 5
18.         Endif
19.     kembali ke nomor 10
20. Endif
21. kembali ke nomor 10
22. Endif
23. End

```

#### 4.4.7 Algoritme *stopword*

Pada algoritme ini dijelaskan tentang proses sub tahapan pada *preprocessing* yaitu *stopword*.

```

1.  Baca array new_text[]
2.  Baca array library stopwords[]
3.  i=0

```

```

4. if (new_text[i] ada)
5.   if (new_text[i] terdapat dalam stopword[])
6.     remove(new_text[i])
7.     i++
8.     Kembali ke nomor 4
9.   Endif
10.  Kembali ke nomor 7
11. Endif
12. End

```

#### 4.4.8 Algoritme *Stemming*

Pada algoritme ini dijelaskan tentang proses sub tahapan pada *preprocessing* yaitu *stemming*.

```

1. Include library Sastrawi
2. Baca kamus prefix dan suffix
3. Baca array new_text[]
4. i=0
5. if (new_text[i] ada)
6.   Cek prefix OR suffix
7.   If(prefix OR suffix == ada)
8.     Kata dasar = remove (prefix OR suffix)
9.     new_text[i] = Kata dasar
10.  Endif
11.  i++
12.  Kembali ke nomor 5
13. Endif
14. Implode(new_text[])
15. Insert ke database
16. End

```

#### 4.4.9 Algoritme BN-Gram

Pada algoritme ini dijelaskan tentang proses tahapan utama pada metode BN--Gram yaitu penghitungan DF-IDFt, Klasterisasi, dan Perangkingan.

```

1. Slot=tweet waktu awal sampai tweet ditambah waktu 10 menit
2. Ambil tweet dimana timestamp = slot
3. If (tweet tersedia)
4.   Ekstraksi n-gram = n-gram[]
5.   i=0
6.   if (ngram[i] ada)
7.     Hitung DF-IDFt
8.     i++
9.     Kembali ke nomor 6
10.  Endif

```

```

11. Endif
12. Slot + 10 menit
13. If (slot <= timestamp akhir)
14.     kembali ke nomor 2
15. Endif
16. Klastering n-gram
17. Perangkingan DF-IDFt n-gram pada klaster
18. End

```

#### 4.4.10 Algoritme Perhitungan DF-IDF<sub>t</sub>

Pada algoritme ini dijelaskan tentang proses sub tahapan pada BN-Gram yaitu sub proses Hitung DF-IDF<sub>t</sub>.

```

1. Baca array n-gram[]
2. hitung = ((dfi+1)/(log(dfij/t)+1)+1) * boost
3. End

```

#### 4.4.11 Algoritme Klasterisasi N-Gram

Pada algoritme ini dijelaskan tentang proses sub tahapan pada BN-Gram yaitu sub proses Klasterisasi N-Gram.

```

1. Baca array matrix n-gram[]
2. hitung = 1-(a/(min(b,c)))
3. if (b,c <= 0.5)
4.     bentuk klaster
5. Endif
6. Kembali ke nomor 2
7. End

```

#### 4.4.12 Algoritme *Topic Ranking*

Pada algoritme ini dijelaskan tentang proses sub tahapan pada BN-Gram yaitu sub proses *Topic Ranking*.

```

1. Baca array Klaster[]
2. nilai tertinggi = max(df_idft[])
3. Cek Klaster
4. if(Klaster ada nilai tertinggi)
5.     Klaster = Topic Rangking
6. Endif
7. Kembali ke nomor 3
8. End

```

#### 4.5 Pengujian

Pengujian merupakan salah satu hal yang perlu dilakukan dalam setiap pengembangan sistem untuk mengevaluasi, menganalisa dan mengetahui tingkat akurasi atau kesamaan hasil yang telah dicapai oleh sistem yang telah dirancang. Pada penelitian ini, dilakukan pengujian *ground truth* secara manual terhadap *precision* dan *recall* pada implementasi metode BN-gram. Rumus yang digunakan yaitu telah dijabarkan dalam persamaan (3.4), persamaan (3.5) dan persamaan (3.6), untuk hasil pengujian dapat dilihat pada Tabel 4.11.

**Tabel 4. 11 Tabel Pengujian**

<i>Ground Truth</i>	<i>Trending Topics usulan</i>
bpbd kabupaten bolaang mongondow melaporkan banjir dan longsor terjadi di tiga kecamatan bencana tersebut terjadi pada pukul 03.30 waktu setempat	langkah darurat tsunami situasi covid kerjasama,kerjasama bnpb bmkkg kesiapsiagaan hadap picu banjir hujan intensitas informasi manfaat manfaat kesiapsiagaan hadap bencana bencana sama sama kolateral kolateral covid
pemko batam ikut keputusan pusat soal pembubaran gugus tugas covid	kemenkes sempurna sempurna teknis update covid tanggulang covid bubar tim covid nasional kembang covid covid indonesia
sampaikan hasil swab test jokowi alhamdulillah negatif covid	fadil nilai tangan covid pimpin rapat gugus tool swab rapid habis mayan hasil mah blakangan phk usaha bangkrut gugus tugas covid panglima tni restrukturisasi gugus presiden jokowi restrukturisasi tugas cepat tangan corona komite covid jatim upaya gugus jatim jalur guna masker biasa cuci gitu pakai masker

Pada Tabel 4.12, pengujian dilakukan pada *tweet* bertanggal 24 Juli 2020, dengan topik *Ground Truth* didapat dari berbagai sumber media informasi, diantaranya *kompas.com*, *kumparan.com*, dan *bnpb.co.id* pada tanggal 24 dan 25 Juli 2020. Untuk hasil perhitungan pengujian terdapat pada Tabel 4.12 berikut.



**Tabel 4. 12 Tabel Perhitungan *Recall***

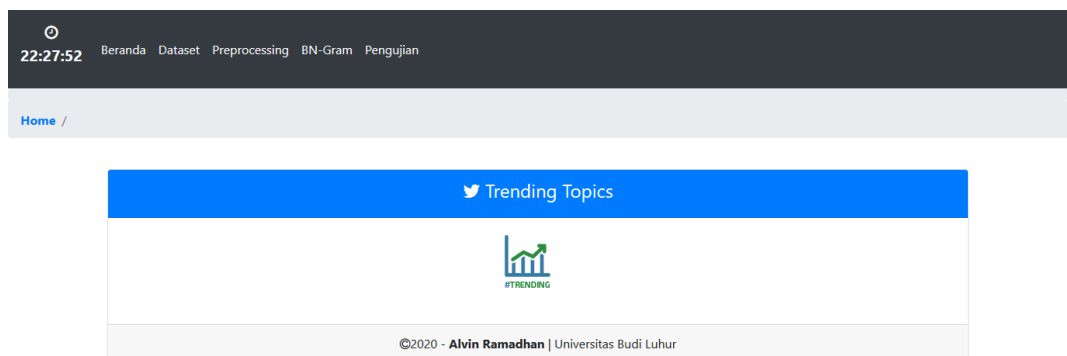
Pengujian <i>Ground Truth</i>		
<i>Topic Recall</i> (TR)	$\frac{ 3 }{ 3 }$	1.0 (100%)
<i>Keyword Precision</i> (KP)	$\frac{ 9 }{ 89 }$	0.10 (10%)
<i>Keyword Recall</i> (KR)	$\frac{ 9 }{ 37 }$	0.24 (24%)

Berdasarkan hasil pengujian, didapat akurasi yang cukup baik dari sisi kesamaan topik yaitu sebesar 100%, namun dilihat dari kesamaan kata dalam sekumpulan topik mendapatkan hasil yang sedikit yaitu KP sebesar 10% dan KR 24%, hal ini dipengaruhi dalam pemilihan topik pada *Ground Truth* atau topik pembanding dari media berita elektronik, secara garis besar *terms* yang dihasilkan sudah cukup untuk mendapatkan deteksi *trending topic* pada waktu tertentu.

## 4.6 Tampilan Layar Aplikasi

### 4.6.1 Tampilan layar Beranda

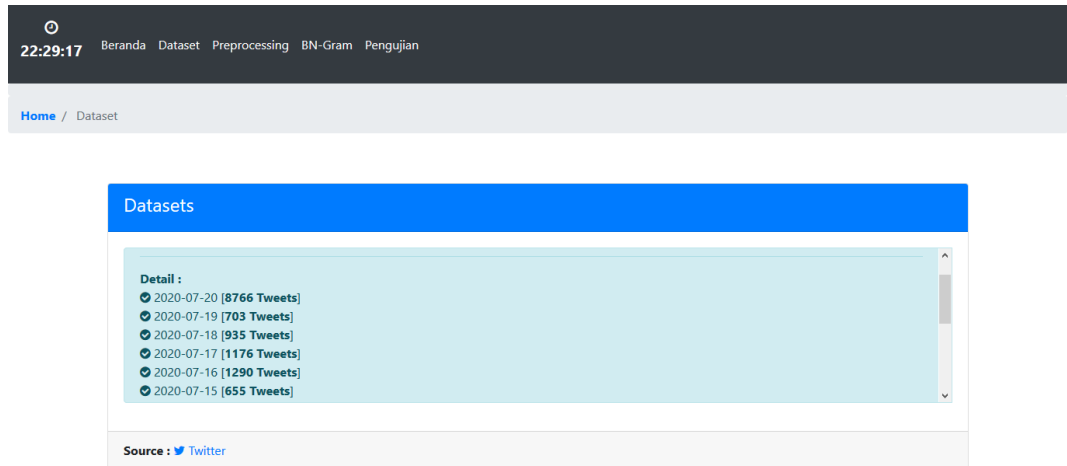
Tampilan layar pada Gambar 4.13 merupakan halaman awal saat mengakses aplikasi.



**Gambar 4. 13 Tampilan Layar Beranda**

### 4.6.2 Tampilan layar *datasets*

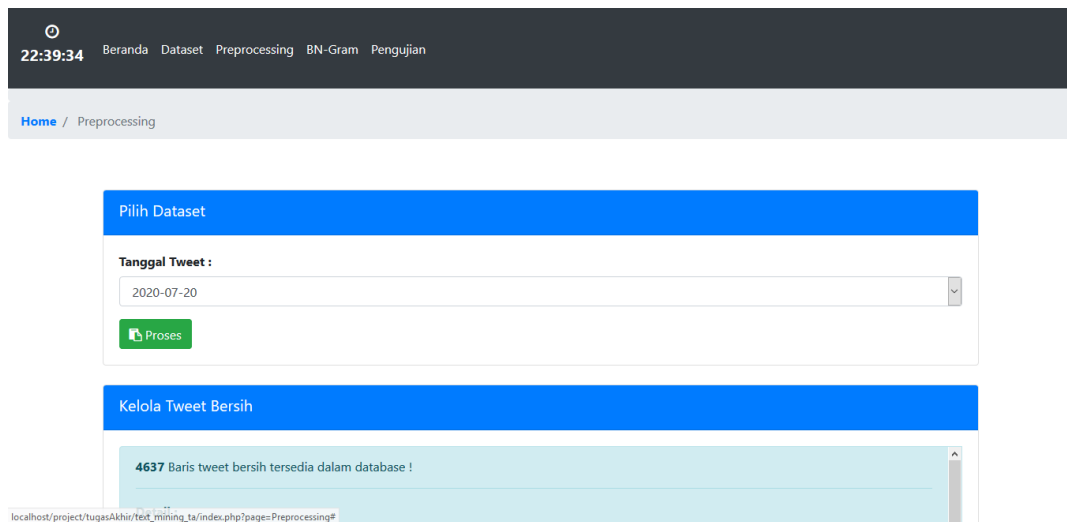
Tampilan layar pada Gambar 4.14 ini merupakan tampilan menu *datasets* yang berisi rincian jumlah *tweet* hasil *crawling*.



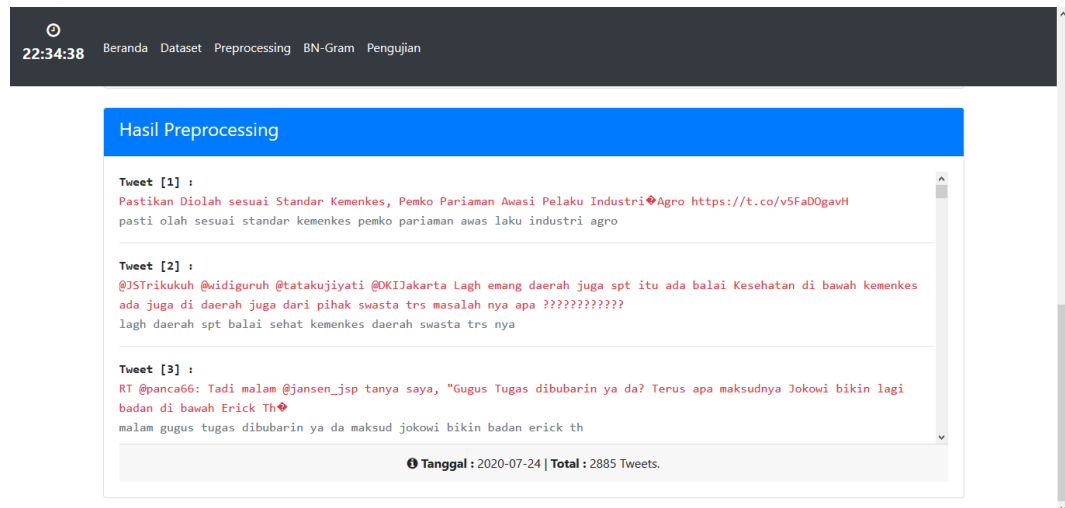
**Gambar 4. 14 Tampilan Layar *Datasets***

#### 4.6.3 Tampilan layar *preprocessing*

Tampilan layar pada Gambar 4.15 dan Gambar 4.16 ini merupakan menu *preprocessing*, halaman ini berisi proses *preprocessing* dari *datasets* yang telah dimasukkan kedalam *database* serta menampilkan *tweet* bersih hasil *preprocessing* pada *tweet* tanggal tertentu.



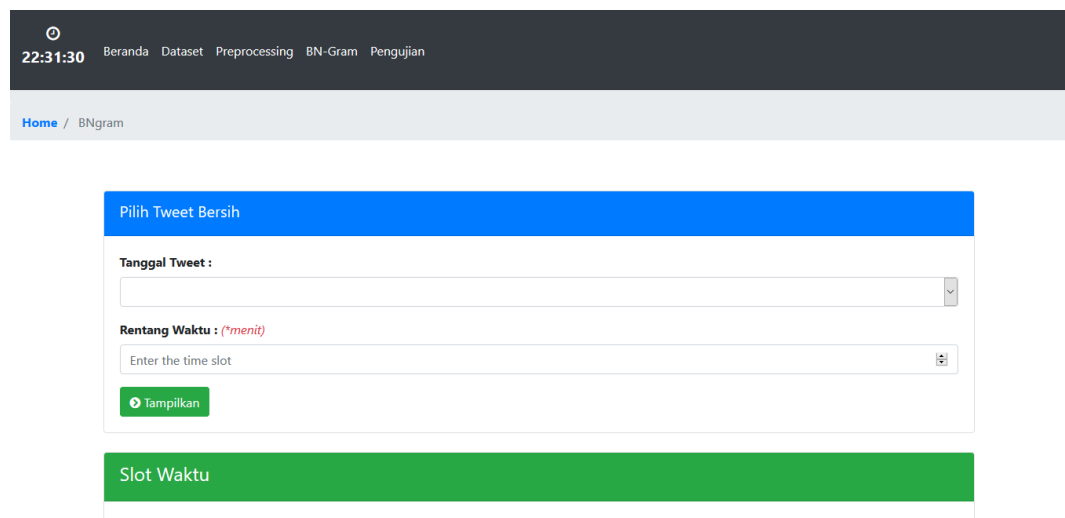
**Gambar 4. 15 Tampilan Pilih Tanggal Preprocessing**



**Gambar 4. 16 Tampilan Hasil Preprocessing**

#### 4.6.4 Tampilan layar BN-Gram

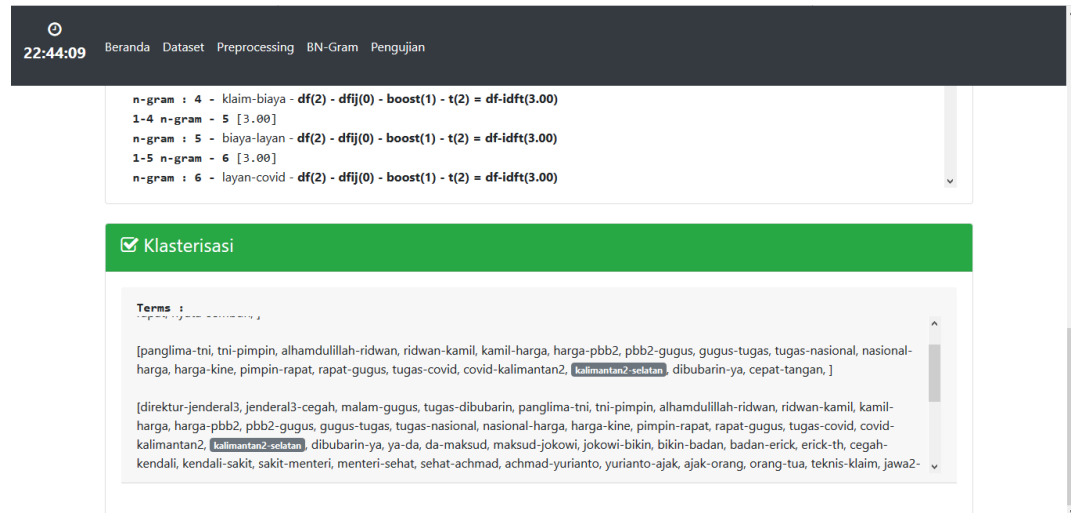
Tampilan layar pada Gambar 4.17 ini merupakan menu proses utama BN-Gram yang berisi form *input* tanggal dan *time slot tweet* tertentu yang akan diproses untuk dihasilkan *trending topic* usulan.



**Gambar 4. 17 Tampilan Layar BN-Gram**

#### 4.6.5 Tampilan layar hasil klusterisasi

Tampilan layar pada Gambar 4.18 ini merupakan hasil dari proses perhitungan  $df-idf_t$  sampai hasil klusterisasi *n-gram* disertai visualisasi *topic ranking*.



**Gambar 4. 18 Tampilan Layar Klasterisasi**

#### 4.6.6 Tampilan layar pengujian

Tampilan layar pada Gambar 4.19 ini merupakan hasil dari perhitungan pengujian *Ground Truth* dibandingkan dengan hasil tren topik usulan.

Hasil Pengujian	
Ground Truth	Trending Topics
bpbdb kabupaten bolaang mongondow melaporkan banjir dan longsor terjadi di tiga kecamatan bencana tersebut terjadi pada pukul waktu setempat	langkah darurat tsunami situasi covid kerjasama kerjasama bnpb bmkg kesiapsiagaan hadap picu banjir hujan intensitas informasi manfaat manfaat kesiapsiagaan hadap bencana bencana sama sama kolateral kolateral covid
pemko batam ikut keputusan pusat soal pembubaran gugus tugas covid	kemenkes sempurna sempurna teknis update covid tanggulang covid bubar tim covid nasional kembang covid covid indonesia
sampaikan hasil swab test jokowi alhamdulillah negatif covid	fadil nilai tangan covid pimpin rapat gugus tool swab rapid habis mayan hasil mah blakangan phk usaha bangkrut gugus tugas covid panglima tni restrukturisasi gugus presiden jokowi restrukturisasi tugas cepat tangan corona komite covid jatim upaya gugus jatim jalur guna masker biasa cuci gihu pakai masker
Topic Recall: 1	
Keyword Precision: 0.1	
Keyword Recall: 0.24	

**Gambar 4. 19 Tampilan Layar Pengujian**

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil evaluasi dari aplikasi deteksi *trending topic* pada data *tweet* dengan kata kunci terkait *Covid-19*, maka dapat disimpulkan bahwa:

- a. Aplikasi ini dapat mendeteksi sebuah *trending topic* terkait *Covid-19* dengan sumber data yaitu Twitter.
- b. Penggunaan metode deteksi *trending topic* dapat berjalan dengan baik dan maksimal dengan menghasilkan pengujian *Topic Recall* sebesar 100%, *Keyword Precision* 10% dan *Keyword Recall* 23%.
- c. *Preprocessing* yang baik juga menjadi penentu terbentuknya istilah-istilah yang sesuai pada hasil deteksi.

#### **5.2 Saran**

Adapun saran yang dapat peneliti berikan sebagai pengembangan lebih lanjut untuk aplikasi ini agar dapat berjalan dengan sempurna dengan fungsi yang lebih baik adalah sebagai berikut:

- a. Algoritma pada aplikasi dikembangkan dengan *library* atau *function* yang lebih ringkas sehingga pemrosesan data dapat berjalan lebih cepat pada *platform website*.
- b. Menambah aturan pendeteksian pada NER sehingga, deteksi dapat lebih akurat.
- c. Pengembangan aplikasi dapat dipadukan dengan metode lain, agar lebih baik hasil yang didapat.
- d. Menambah daftar kamus pada tahapan-tahapan *preprocessing* data, sehingga *tweet* yang diproses benar-benar terseleksi dengan baik.
- e. Membuat hasil deteksi berupa kalimat yang sesuai dengan aturan SPOK agar dapat dimengerti secara langsung oleh pengguna umum.
- f. Menambah kata kunci pengumpulan *tweet* sehingga dapat mendeteksi hasil tren topik yang lain.

## DAFTAR PUSTAKA

- Abdurrahman, G. (2019). Clustering Data Kredit Bank Menggunakan Algoritma Agglomerative Hierarchical Clustering Average Linkage. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, 4(1), 13. <https://doi.org/10.32528/justindo.v4i1.2418>
- Aiello, L. M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Goker, A., Kompatsiaris, I., & Jaimes, A. (2013). Sensing trending topics in twitter. *IEEE Transactions on Multimedia*, 15(6), 1268–1282. <https://doi.org/10.1109/TMM.2013.2265080>
- Budi, I., Bressan, S., Wahyudi, G., Hasibuan, Z. A., & Nazief, B. A. A. (2005). Named Entity Recognition for the Indonesian language: Combining contextual, morphological and part-of-speech features into a knowledge engineering approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3735 LNAI, 57–69. [https://doi.org/10.1007/11563983\\_7](https://doi.org/10.1007/11563983_7)
- Cvijikj, I. P., & Michahelles, F. (2011). Monitoring trends on Facebook. *Proceedings - IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, DASC 2011*, 895–902. <https://doi.org/10.1109/DASC.2011.150>
- Fahma, A. I. (2018). Identifikasi Kesalahan Penulisan Kata ( Typographical Error ) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(1), 53–62.
- Februariyanti, H. (2012). Klasifikasi Dokumen Berita Teks Bahasa Indonesia menggunakan Ontologi. *Teknologi Informasi DINAMIK*, 17(1), 14–23. <http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/1612/594>
- Fitri, S., Nurjanah, N., & Astuti, W. (2018). Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa (Studi Kasus: Umtas). *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 9(1), 633–640. <https://doi.org/10.24176/simet.v9i1.2002>
- Gunadi, G., & Sensuse, D. I. (2012). Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma Apriori Dan Frequent Pattern Growth ( Fp-Growth ) : *Telematika*, 4(1), 118–132.
- Hanoatubun, S. (2020). DAMPAK COVID – 19 TERHADAP PEREKONOMIAN INDONESIA. *Journal of Education, Psychology and Counseling*, 2(1), 146–153. <https://ummaspul.e-journal.id/Edupsycouns/article/view/423/240>

- Indra, Winarko, E., & Pulungan, R. (2019). Trending topics detection of Indonesian tweets using BN-grams and Doc-p. *Journal of King Saud University - Computer and Information Sciences*, 31(2), 266–274. <https://doi.org/10.1016/j.jksuci.2018.01.005>
- Juditha, C. (2018). Fenomena Trending Topic Di Twitter: Analisis Wacana Twit #Savehajilulung. *Jurnal Penelitian Komunikasi dan Pembangunan*, 16(2), 138. <https://doi.org/10.31346/jpkp.v16i2.1353>
- Mediayani, M., Wibisono, Y., Riza, L. S., & Rosales-Pérez, A. (2019). Determining trending topics in twitter with a data-streaming method in R. *Indonesian Journal of Science and Technology*, 4(1), 148–157. <https://doi.org/10.17509/ijost.v4i1.15807>
- Mujilawati, S. (2016). Pre-Processing Text Mining Pada Data Twitter. *Seminar Nasional Teknologi Informasi dan Komunikasi, 2016*(Sentika), 2089–9815.
- Munarko, Y. (2016). *Analisa Model Named Entity Recognition Tweet Bahasa Indonesia*. 191–197.
- Munarko, Y., & Azhar, Y. (2016). Peringkasan Tweet Berdasarkan Trending Topic Twitter Dengan Pembobotan TF-IDF dan. *Jurnal Kinetik*, 1(1), 9–16.
- Ningtias, P., Sudiar, N., & Latiar, H. (2020). Tren Topik Pemberitaan PASCA Pemilihan Presiden pada Portal Berita Online. *Info Bibliotheca: Jurnal Perpustakaan dan Ilmu Informasi*, 1(2), 113–128. <https://doi.org/10.24036/ib.v1i2.74>
- Palupi, E. S., Pahlevi, S. M., Bina, U., Informatika, S., Magister, P., & Komputer, I. (2020). Inti nusa mandiri. *Inti Nusa Mandiri*, 14(2), 133–138. <https://doi.org/https://doi.org/10.33480/inti.v14i2.1178> VOL.
- Saputra, P. (2017). Implementasi teknik crawling untuk pengumpulan data dari media sosial twitter. *Jurnal Dinamika Dotcom*, 8, 160–168.
- Setiyoaji, A., Muflikhah, L., & Fauzi, M. A. (2017). Named Entity Recognition Menggunakan Hidden Markov Model dan Algoritma Viterbi pada Teks Tanaman Obat. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1(12), 1858–1864. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/673>
- Simarangkir, H. (2017). *Studi Perbandingan Algoritma - Algoritma Stemming Untuk Text Bahasa Indonesia*. 1, 6–8. <https://doi.org/10.16309/j.cnki.issn.1007-1776.2003.03.004>
- Sindi, S., Ratnasari, W., Ningse, O., Sihombing, I. A., Zer, F. I. R. H., Hartama, D., & Kunci, K. (2020). *Analisis algoritma k-medoids clustering dalam pengelompokan penyebaran covid-19 di indonesia*. 4(1), 166–173. <http://www.jurnal.una.ac.id/index.php/jurti/article/view/1296/1112>

- Tala, F. Z. (2003). A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. *M.Sc. Thesis, Appendix D*, pp, 39–46.
- TEJASREE, S., SHAIK, N., & TELLA, P. (2017). Perception of Trend Topic in Twitter: a Case Study. *i-manager's Journal on Software Engineering*, 11(4), 12. <https://doi.org/10.26634/jse.11.4.13816>
- Wahyudi, D., Susyanto, T., & Nugroho, D. (2017). Implementasi Dan Analisis Algoritma Stemming Nazief & Adriani Dan Porter Pada Dokumen Berbahasa Indonesia. *Jurnal Ilmiah SINUS*, 15(2). <https://doi.org/10.30646/sinus.v15i2.305>
- Wiyadi, Y. P. (2017). *Pengaruh Tokoh Ahok Pada Media Sosial Menjadi*. November, 1–2.