



**ANALISIS SENTIMEN OPINI PUBLIK TERHADAP PELAYANAN BPJS
KESEHATAN MENGGUNAKAN METODE *IMPROVED K-NEAREST
NEIGHBOR***



TUGAS AKHIR

Program Studi

S1 SISTEM INFORMASI

UNIVERSITAS
Dinamika

Oleh:

GLORIA VALERY SIDABUTAR

18410100067

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**ANALISIS SENTIMEN OPINI PUBLIK TERHADAP PELAYANAN
BPJS KESEHATAN MENGGUNAKAN METODE
*IMPROVED K-NEAREST NEIGHBOR***

TUGAS AKHIR

**Diajukan sebagai salah syarat untuk menyelesaikan
Program Sarjana Komputer**



**UNIVERSITAS
Dinamika**

Oleh:

**Nama : Gloria Valery Sidabutar
NIM : 18410100067
Program Studi : Sistem Informasi**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2023

Tugas Akhir

ANALISIS SENTIMEN OPINI PUBLIK TERHADAP PELAYANAN BPJS KESEHATAN MENGGUNAKAN METODE *IMPROVED K-NEAREST NEIGHBOR*

Dipersiapkan dan disusun oleh

Gloria Valery Sidabutar

NIM: 18410100067

Telah diperiksa, dibahas, dan disetujui oleh Dewan Pembahas

Pada: Jumat, 25 Agustus 2023

Susunan Dewan Pembahas

Pembimbing:

I. Titik Lusiani, M.Kom.
NIDN: 0714077401

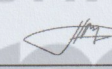
II. Tutut Wuriyanto, M.Kom.
NIDN: 0703056702

Pembahas:

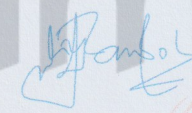
Julianto Lemantara, S.Kom., M.Eng
NIDN: 0722108601



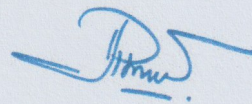
Date: 2023.08.28
08:28:32 +07'00'



Digitally signed by Julianto
Date: 2023.08.28 12:15:40
+07'00'



Tugas akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana,



Digitally signed by
Universitas Dinamika
Date: 2023.08.29
17:30:48 +07'00'

Tri Sagirani, S.Kom., M.MT.
NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



*Kupersembahkan untuk keluarga tercinta
dan orang-orang yang selalu mendukungku
dalam perjalanan penulisan tugas akhir ini*

UNIVERSITAS
Dinamika

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Gloria Valery Sidabutar**
NIM : **18410100067**
Program Studi : **S1 Sistem Informasi**
Fakultas : **Fakultas Teknologi dan Informatika**
Jenis Karya : **Tugas Akhir**
Judul Karya : **ANALISIS SENTIMEN OPINI PUBLIK TERHADAP PELAYANAN BPJS KESEHATAN MENGGUNAKAN METODE *IMPROVED K-NEAREST NEIGHBOR***

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 23 Agustus 2023



Gloria Valery Sidabutar
NIM : 18410100067

ABSTRAK

BPJS Kesehatan adalah Badan Penyelenggara Jaminan Sosial Kesehatan di Indonesia yang bertanggung jawab dalam menyelenggarakan program jaminan kesehatan untuk seluruh penduduk Indonesia. Opini Masyarakat tentang BPJS Kesehatan beragam pendapat dan sudut pandang yang dapat dipengaruhi oleh berbagai faktor, seperti pengalaman pribadi, persepsi masyarakat, dan kinerja BPJS Kesehatan itu sendiri. BPJS Kesehatan perlu melakukan evaluasi dan perbaikan sistem jaminan kesehatan ini agar dapat memberikan manfaat yang lebih baik bagi masyarakat secara keseluruhan.

Beragam opini dapat diolah agar memberikan informasi untuk pihak yang membutuhkan. Salah satu caranya pengolahannya adalah Analisis Sentimen. Analisis sentimen merupakan proses untuk mengekstraksi dan menganalisis sentimen atau sikap pengguna dalam teks yang diungkapkan di media sosial. Metode *Improved K-Nearest Neighbor* digunakan untuk mengklasifikasikan sentimen dari *tweet* yang terkait dengan pelayanan BPJS Kesehatan menjadi positif atau negatif dengan perbandingan data *training* dan data *testing* adalah 70:30. Metode ini memanfaatkan klasifikasi berbasis *k*-nearest neighbor yang ditingkatkan dengan mengubah nilai tetangga terdekat sesuai kelas masing-masing. Dengan menggunakan *Improved K-Nearest Neighbor* dapat meningkatkan nilai akurasi pada *K-Nearest Neighbor* yang cenderung mengikuti kelas dengan jumlah data terbanyak.

Melalui implementasi dan pengujian sistem, didapatkan hasil berupa pengaruh jumlah data *training*, proporsi keseimbangan kategori data *training*, dan nilai *k* terhadap akurasi analisis sentiment. Hasil *Accuracy Improved K-nearest Neighbor* sebesar 86% sedangkan *K-nearest Neighbor* sebesar 84,67%. Rata-rata tingkat ketepatan (*precision*) yang diperoleh oleh sistem adalah 76,02%, rata-rata tingkat kecocokan (*recall*) sebesar 74,30%, dan rata-rata *F-measure* sebesar 73,78%. Dengan demikian, dapat disimpulkan bahwa sistem telah berjalan dengan efektivitas yang baik.

Kata kunci: *Analisis Sentimen, BPJS Kesehatan, Improved K-Nearest Neighbor*

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul “Analisis Sentimen Opini Publik Terhadap Pelayanan Bpjs Kesehatan Menggunakan Metode *Improved K-Nearest Neighbor*”

Penyelesaian tugas akhir ini tidak terlepas dari bantuan dari berbagai pihak yang telah memberikan bantuan, dukungan dan doa. Dalam kesempatan ini, penulis ingin mengucapkan terima kasih yang setulus-tulusnya kepada :

1. Papa dan Mama yang telah mendoakan dan mendukung secara materi dan moral untuk penyelesaian tugas akhir ini.
2. Kakak Sisil, Abang Carlos dan Putri yang selalu memberikan motivasi dan semangat dalam pengerjaan tugas akhir ini.
3. Bapak Dr. Anjik Sumkaaji, S. Kom., M. Eng., selaku Kepala Program Studi S1 Sistem Informasi
4. Ibu Titik Lusiani, M.Kom selaku Dosen Pembimbing 1 yang memberikan arahan, bimbingan, dan masukan berharga sehingga penulis dapat mengembangkan ide dan penelitian tugas akhir ini dengan baik
5. Bapak Tutut Wurjianto, M.Kom. selaku Dosen Pembimbing 2 yang memberikan arahan, bimbingan, dan masukan berharga sehingga penulis dapat mengembangkan ide dan penelitian tugas akhir ini dengan baik
6. Bapak Julianto Lemantara, S.Kom., M.Eng selaku Dosen Pembahas yang telah memberi arahan dan saran dalam pengerjaan tugas akhir ini.
7. Ibu Dr.M.J. Dewiyani Sunarto selaku Dosen Wali yang selalu memberikan dukungan dan arahan selama masa perkuliahan hingga penyelesaian tugas akhir ini.
8. Teman-teman Cenayank Alifah, Salsa, dan Sahsa sebagai teman seperjuangan yang telah berbagi pengalaman dan pengetahuan dalam perjalanan penulis menyelesaikan Tugas Akhir ini.
9. Serta seluruh pihak yang telah membantu dan mendukung penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini tentu tidak lepas dari segala keterbatasan dan kekurangan. Oleh karena itu, segala kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini dapat memberikan manfaat dan kontribusi yang positif bagi perkembangan ilmu pengetahuan dan teknologi di masa yang akan datang.

Akhir kata, semoga hasil dari Tugas Akhir ini dapat memberikan manfaat bagi penulis pribadi, kampus, dan masyarakat luas. Dengan rasa rendah hati, penulis menyerahkan Tugas Akhir ini sebagai bentuk kontribusi penulis dalam pengembangan ilmu pengetahuan.

Sidoarjo, 22 Agustus 2023

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xv
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	4
1.5 Manfaat	4
BAB II. LANDASAN TEORI	5
2.1 Penelitian Terdahulu	5
2.2 Analisa Sentimen	6
2.2.1 Data	6
2.2.2 <i>Preprocessing</i>	6
2.2.3 <i>Feature Extraction</i>	7
2.2.4 <i>Data Split</i>	7
2.2.5 <i>Classification</i>	7
2.2.6 Evaluasi	7

2.3 Python	7
2.4 Text Mining	8
2.4.1 Text Preprocessing.....	9
2.5 Pembobotan kata <i>TF - ID</i>	10
2.6 <i>Improved K-Nearest Neighbor</i>	11
2.7 Evaluasi	13
2.7.1 <i>K-Fold Cross Validation</i>	13
2.7.2 <i>Confusion Matrix</i>	13

BAB III. METODELOGI PENELITIAN..... 15

3.1 Tahap Awal	15
3.1.1 Studi literatur.....	15
3.1.2 Persiapan Data.....	16
3.2 Tahap Analisis.....	17
3.2.1 <i>Preprocessing Data</i>	17
3.2.2 Pembobotan Kata <i>TF IDF</i>	21
3.2.3 Klasifikasi Sentimen	24
3.2.4 Evaluasi dan Validasi.....	30
3.2.5 Visualisasi	30
3.3 Tahap Akhir	30
3.3.1 Kesimpulan dan Saran.....	30

BAB IV. HASIL DAN PEMBAHASAN..... 32

4.1 Penambangan data.....	32
---------------------------	----

4.2 Pelabelan Data.....	32
4.3 <i>Pre-processing</i> Data.....	32
4.3.1 <i>Cleansing</i>	33
4.3.2 <i>Case Folding</i>	33
4.3.3 <i>Tokenizing</i>	34
4.3.4 <i>Filtering</i>	34
4.3.5 <i>Stemming</i>	35
4.4 Pembobotan Kata TF IDF	36
4.5 Pembagian <i>Data Training</i> dan <i>Data Testing</i>	37
4.6 <i>Cosine Similarity</i>	37
4.7 Klasifikasi <i>Improved K-Nearest Neighbor</i>	38
4.8 Evaluasi dan Validasi.....	40
4.9 Visualisasi	41
4.10 Pembahasan.....	42
4.10.1 Hasil Klasifikasi <i>Improved K-Nearest Neighbor</i>	42
4.10.2 Perbandingan Klasifikasi <i>Improved K-Nearest Neighbor</i> dan <i>K-Nearest Neighbor</i>	43
4.10.3 Rekomendasi Hasil Visualisasi	44
BAB V. KESIMPULAN DAN SARAN	47
5.1 Kesimpulan	47
5.2 Saran.....	47
DAFTAR PUSTAKA	49
LAMPIRAN	51

DAFTAR GAMBAR

	Halaman
Gambar 3.1 Tahapan Penelitian	15
Gambar 3.2 Contoh <i>Tweet</i>	16
Gambar 3.3 Diagram Alir <i>Preprocessing</i> Data.....	18
Gambar 3.4 Diagram Alir TF IDF	22
Gambar 3.5 Diagram Alir Klasifikasi Sentimen.....	25
Gambar 3.6 Diagram Alir <i>Cosine Similarity</i>	26
Gambar 3.7 Diagram Alir menghitung Probabilitas	27
Gambar 4.1 <i>Sourcecode Cleansing</i>	33
Gambar 4.2 <i>Sourcecode Case folding</i>	33
Gambar 4.3 <i>Sourcecode Tokenizing</i>	34
Gambar 4.4 <i>Sourcecode</i> deklarasi <i>stopwords</i>	34
Gambar 4.5 <i>Sourcecode Filtering</i>	35
Gambar 4.6 <i>Sourcecode Stemming</i>	36
Gambar 4.7 <i>Sourcecode TF IDF</i>	37
Gambar 4.8 <i>Sourcecode Cosine Similarity</i>	37
Gambar 4.9 <i>Sourcecode</i> Menghitung <i>k-value</i> positif.....	38
Gambar 4.10 <i>Sourcecode</i> Menghitung <i>k-value</i> negatif.....	38
Gambar 4.11 <i>Sourcecode Improve K-Nearest Neighbor</i>	39
Gambar 4.12 <i>Sourcecode</i> Perbandingan Hasil Kelas.....	40
Gambar 4.13 <i>Sourcecode</i> Menjalankan Algoritma.....	40
Gambar 4.14 <i>Word cloud</i> Positif.....	41
Gambar 4.15 <i>Word cloud</i> Negatif.....	42



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 2.1 Penelitian Terdahulu	5
Tabel 3.1 Hasil pengumpulan <i>tweets</i>	16
Tabel 3.2 Hasil Pelabelan <i>Tweets</i>	17
Tabel 3.3 Hasil <i>Cleansing</i>	19
Tabel 3.4 Hasil <i>Case folding</i>	19
Tabel 3.5 Hasil <i>Tokenizing</i>	20
Tabel 3.6 Hasil <i>Filtering</i>	20
Tabel 3.7 Hasil <i>Stemming</i>	20
Tabel 3.8 Hasil Perhitungan <i>Term Frequency</i>	22
Tabel 3.9 Hasil Perhitungan Dokumen <i>Frequency</i>	23
Tabel 3.10 Hasil Perhitungan IDF	23
Tabel 3.11 Hasil Perhitungan TF-IDF	24
Tabel 3.12 Hasil Perhitungan Skalar.....	28
Tabel 3.13 Hasil Perhitungan Panjang Vektor.....	28
Tabel 3.14 Jumlah Data <i>Training</i>	29
Tabel 3.15 <i>K-value</i> Baru	29
Tabel 4.1 Perbandingan Hasil Pemetaan <i>Confusion Matrix</i>	43
Tabel 4.2 Perbandingan Hasil Evaluasi <i>Confusion Matrix</i>	44
Tabel 4.3 Data <i>Tweet</i> Dengan Unsur Kata Negatif.....	44
Tabel 4.4 Data <i>Tweet</i> Dengan Unsur Kata Positif	45

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Hasil <i>Text Processing</i>	51
Lampiran 2 Hasil <i>Improved K-nearest Neighbor</i>	56
Lampiran 3 Hasil <i>Confusion Matrix K-Nearest Neighbor</i>	57
Lampiran 4 Hasil Rata-rata Evaluasi <i>10 Fold Validation Improved K-Nearest Neighbor</i>	58
Lampiran 5 <i>Tweets keywords Word Cloud</i>	59
Lampiran 6 Hasil Plagiasi	61
Lampiran 7 Kartu Bimbingan Tugas Akhir	74
Lampiran 8 Biodata Penulis	75



UNIVERSITAS
Dinamika

BAB I PENDAHULUAN

1.1 Latar Belakang

Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan didirikan pada Januari 2014 oleh pemerintah, berperan sebagai instansi publik yang bertugas menyelenggarakan program Jaminan Kesehatan Nasional (JKN) dengan mengikuti prinsip asuransi sosial dan prinsip ekuitas. Tujuan BPJS Kesehatan adalah memberikan manfaat pemeliharaan kesehatan dan perlindungan bagi peserta, sehingga kebutuhan dasar kesehatan dapat terpenuhi (BPJS Kesehatan, n.d.)

Berdasarkan data dari detik.com sampai dengan 30 Juni 2022, jumlah peserta BPJS Kesehatan mencapai 226,3 juta penduduk Indonesia (Kusnandar, 2022). Hal ini menunjukkan banyak masyarakat Indonesia yang menggunakan BPJS Kesehatan. Tentunya, kondisi ini perlu diikuti dengan peningkatan pelayanan. Dari pemantauan di media massa ditemukan berbagai opini dan keluhan masyarakat terkait pelayanan bpjs kesehatan. Misalnya masyarakat mengeluhkan antrian panjang hingga 5-6 jam (Azizah, 2021). Contoh lainnya masyarakat yang mengalami kesulitan dalam meminta rujukan rumah sakit (Maya, 2018). Namun ada juga tanggapan positif seperti biaya pelayanan kesehatan tidak menjadi bebannya lagi berkat bantuan bpjs kesehatan (CNN Indonesia, 2021). Saat ini opini tentang layanan BPJS masih beragam padahal opini atau tanggapan masyarakat diperlukan untuk memperbaiki layanan pada BPJS Kesehatan.

Masyarakat banyak menggunakan media sosial untuk menyampaikan opini mereka, baik itu keluhan, berbagi informasi ataupun dukungan. Salah satu media sosial yang banyak digunakan masyarakat Indonesia adalah *twitter*. Jumlah pengguna aktif *twitter* di Indonesia hingga bulan Januari 2022 sekitar 18,45 juta pengguna (Annur, 2022). Opini yang diberikan oleh masyarakat dapat digunakan sebagai bahan evaluasi oleh pemerintah selaku lembaga yang menaungi langsung BPJS Kesehatan untuk melakukan perbaikan dan juga meningkatkan kualitas kinerja dimasa depan. Hasil opini pada sosial media memiliki banyak tata bahasa, sehingga tidak mempunyai kemampuan untuk mengagregasi informasi menjadi sebuah kesimpulan. Untuk itu perlu adanya sistem filterasi opini sehingga opini

dapat diidentifikasi secara objektif dan dapat ditarik kesimpulan. Salah satu caranya adalah menggunakan analisis sentimen.

Analisis sentimen merupakan keilmuan yang berfokus pada sentimen, emosi, perilaku, dan penilaian terhadap berbagai entitas seperti pelayanan, produk, individu, kejadian, dan topik (D'souza & Sonawane, 2019). Analisis sentimen bertujuan untuk memahami pendapat atau sikap publik terkait topik, produk atau peristiwa tertentu. Hasilnya menjadi acuan dalam upaya meningkatkan pelayanan atau kualitas produk (Cambria, Das, Bandyopadhyay, & Feraco, 2019)

Penilaian opini tentang layanan BPJS Kesehatan diambil dari media sosial yaitu *Twitter*. *Twitter* sebagai layanan mikroblogging yang memungkinkan pengguna memberikan informasi tentang suatu peristiwa yang sedang terjadi. Pengguna *twitter* memposting pesan yang disebut "*tweet*" dan muncul pada halaman *twitter* sesuai urutan waktu pengiriman *tweet*. Pengambilan data dari *twitter* cukup mudah dilakukan karena *twitter* sudah menyediakan *Application Programming Interface* (API) (Russell, 2014). Setelah data diambil dilanjutkan dengan pengolahan data agar terklasifikasi positif atau negatif. Klasifikasi dilakukan menggunakan metode *Improved K-Nearest Neighbor* yang merupakan modifikasi dari *K-Nearest Neighbor*. *K-Nearest Neighbor* memiliki beberapa keunggulan seperti struktur yang sederhana, sehingga mudah untuk diimplementasikan dan cenderung tidak berpengaruh pada data *Noise* (Rui-Jun, Dan-Feng, & Feng, 2019). Algoritma *Improved K-Nearest Neighbor* merupakan pengembangan dari metode *K-Nearest Neighbor* (KNN) yang memiliki kelemahan pada tingkat akurasi yang rendah karena distribusi sampel yang tidak merata sehingga klasifikasi cenderung mengikuti sampel dominan. Kelemahan ini diatasi dengan memodifikasi nilai *k-values* metode oleh *Improved K-Nearest Neighbor*. Algoritma *Improved K-Nearest Neighbor* menunjukkan kestabilan dalam proses kategorisasi dengan variasi nilai *k* yang berbeda (Baoli, Shiwen, & Qin, 2003). Hasil penelitian pada 2015 menunjukkan *Improved K-Nearest Neighbor* unggul dibandingkan penggunaan metode *Support Vector machine* (SVM) dan *K-Nearest Neighbor* tradisional. Hasil *F1 Measure* pada metode *Improved K-Nearest Neighbor* adalah 89,28%. Sedangkan metode SVM sebesar 88,4% dan KNN sebesar 85,54% (Iswarya & Radha, 2015). Penelitian lain juga menunjukkan

Improved K-Nearest Neighbor memberikan hasil akurasi yang lebih unggul senilai 92% sedangkan pada *K-Nearest Neighbor* 88% (Zakia Nathania & Abdurrachma Bachtiar, 2018)

Berdasarkan uraian diatas penelitian ini bertujuan melakukan analisis sentimen opini publik terhadap pelayanan BPJS Kesehatan dengan mengimplementasikan metode *Improved K-Nearest Neighbor* untuk meningkatkan efisiensi proses klasifikasi. Hasil analisis sentimen digunakan untuk mengidentifikasi arah kualitas pelayanan BPJS Kesehatan saat ini. Hasil dominan positif mengindikasikan kualitas yang baik dan terus ditingkatkan, sebaliknya kecenderungan negatif yang mengisyaratkan adanya aspek-aspek yang perlu diperbaiki dalam pelayanannya. Hasil penelitian akan divisualisasikan menggunakan *word cloud*. *Word cloud* akan menampilkan kata-kata yang mengarah pada *tweet* positif ataupun negatif, sehingga menggambarkan topik atau masalah pada tiap kelas yang dapat dijadikan acuan peningkatan ataupun perbaikan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah adalah bagaimana melakukan analisis sentimen menggunakan metode *Improved K-Nearest Neighbor* sehingga mendapatkan presentase tanggapan masyarakat terhadap pelayanan BPJS Kesehatan.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Sumber data pada penelitian ini berasal dari tweet masyarakat di media sosial twitter sejak bulan Januari 2022 sampai dengan April 2022.
- b. Proses klasifikasi dilakukan pada tweet berbahasa Indonesia menggunakan dua kelas yaitu positif dan negatif
- c. Data yang digunakan berasal dari tweet dengan *keyword* “@BPJSKesehatanRI”, “layanan BPJS Kesehatan”, “pelayanan BPJS Kesehatan, #BPJSKesehatan”.
- d. Pengolahan data menggunakan bantuan library *Scikit-Learn*, *Natural Language Toolkit (NLTK)*, *Pandas*, *Numpy*, *Matplotlib*, *Word cloud*, *Swifter* dan *Sastrawi* pada python.

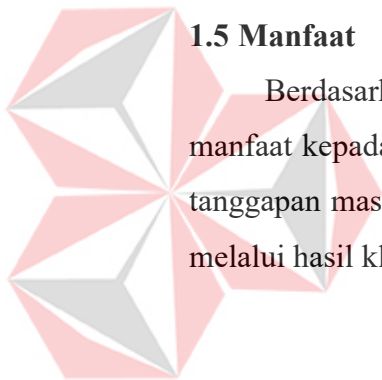
- e. Output dari penelitian ini meliputi hasil klasifikasi positif dan negatif, *word cloud* untuk memvisualisasikan kata-kata yang sering muncul dalam analisis sentimen dan *pie chart* untuk mengvisualisasikan persentase porsi kelas positif dan negatif.
- f. Pelabelan data *tweets* dilakukan secara manual oleh sukarelawan.
- g. Prosedur analisis sentimen yang ditetapkan dapat dilihat pada landasan teori metode *Improved K-Nearest Neighbor*

1.4 Tujuan

Tujuan pada penelitian ini adalah menghasilkan klasifikasi opini masyarakat terhadap pelayanan BPJS Kesehatan kedalam 2 kelas yaitu positif dan negatif menggunakan metode *Improved K-Nearest Neighbor*.

1.5 Manfaat

Berdasarkan analisis sentimen yang dilakukan, diharapkan memberikan manfaat kepada pihak pemegang keputusan di BPJS Kesehatan untuk memahami tanggapan masyarakat terhadap pelayanan BPJS Kesehatan, yang telah dilakukan melalui hasil klasifikasi sentimen.



UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu dijadikan referensi untuk pengembangan dan penguatan teori dalam penelitian ini. Penelitian sebelumnya menemukan topik yang serupa, namun menghadirkan solusi yang berbeda dalam penyelesaiannya. Informasi terkait penelitian-penelitian tersebut disajikan pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Nama Peneliti	Judul Penelitian	Hasil Penelitian
Willa Oktinas (2017)	Analisis Sentimen Pada Acara Televisi Menggunakan <i>Improved K-Nearest Neighbor</i>	Hasil akurasi tertinggi dengan nilai $k=10$ sebesar 90%. Menggunakan pembagian dataset 70:30 dengan data <i>training</i> 2700 dan data <i>testing</i> 300. Menggunakan 3 kelas klasifikasi yaitu positif, negatif dan netral.

Perbedaan :

- Evaluasi menggunakan *10-fold cross validation*, sedangkan pada penelitian ini ditambahkan *confusion matrix* untuk perhitungan *accuracy*, *precision*, *f-measure*, dan *recall*

Nama Peneliti	Judul Penelitian	Hasil Penelitian
Indriya, Dewi, Onantya, Indriati, Putra, Pandu, Adikara (2019)	Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan BM25 Dan <i>Improved K-Nearest Neighbor</i>	Hasil dari penelitian ini adalah nilai <i>k-values</i> terbaik sebesar 10, dengan nilai <i>precision</i> sebesar 0,946, <i>recall</i> sebesar 0,934, <i>f-measure</i> sebesar 0,939, dan <i>accuracy</i> sebesar 0,942.

Perbedaan:

- Pembobotan kata yang digunakan menggunakan BM25, sedangkan pada penelitian ini menggunakan pembobotan kata TF-IDF
- Pada penelitian ini menambahkan visualisasi *pie chart* dan *word cloud* yang belum ada pada penelitian diatas.

Nama Peneliti	Judul Penelitian	Hasil Penelitian
Faisal Faturrohman, Budhi Irawan, S.Si., M.T., Casi Setianingsih, S.T., M.T. (2020)	Analisis Sentimen Pada BPJS Kesehatan Menggunakan <i>Recurrent Neural Network</i>	Hasil akurasi yang didapatkan sebesar 86,67%, <i>precision</i> sebesar 87%, <i>recall</i> sebesar 86,66%, dan F1 score sebesar 86,63%

Perbedaan:

- Algoritma yang digunakan adalah *Recurrent Neural Network*, sedangkan penulis menggunakan algoritma *Improved K-Nearest Neighbor*

Nama Peneliti	Judul Penelitian	Hasil Penelitian
Nur Rahma Windiarti (2018)	Klasifikasi Opini Netizen Berbahasa Indonesia Berbasis <i>Twitter</i> menggunakan metode <i>Improved K-Nearest Neighbor</i>	Hasil akurasi tertinggi yang pada pengtestingan ke-3 dengan 89%. Dengan jumlah data <i>training</i> 119 Positif dan 97 Negatif

Perbedaan:

- Evaluasi menggunakan *confusion matrix*, sedangkan pada penelitian ini menambahkan *k-fold cross validation*.
-

2.2 Analisa Sentimen

Analisis sentimen adalah studi komputasi tentang pendapat, sentimen, evaluasi, sikap, suasana hati, dan emosi orang. Analisis sentimen dapat digunakan untuk menentukan pendapat pelanggan terhadap suatu produk. Interaksi manusia dan komputer diproses melalui analisis sentimen dengan tujuan mengurai emosi yang terkandung dalam opini pelanggan sehingga memberikan informasi berharga bagi perkembangan bisnis (Cambria et al., 2019). Analisis sentimen mengklasifikasikan opini ke dalam kategori sentimen positif maupun negatif agar menjadi acuan untuk perbaikan suatu sistem, produk, layanan, dan sebagainya.

Proses klasifikasi sentimen kedalam sebuah kelas menggunakan bantuan *machine learning*. *Machine learning* digunakan untuk menganalisis dataset opini pelanggan dan merentangkan ciri-ciri unik yang mendefinisikan sentimen tertentu. Dengan mempelajari pola-pola ini, mesin mampu dengan akurat mengklasifikasikan sentimen dalam konteks yang sesuai.

Adapun langkah-langkah dalam melakukan analisis sentimen menggunakan *machine learning* adalah sebagai berikut (Dwi Antonio, Efendi, & Mawengkang, 2022):

2.2.1 Data

Data berupa teks seperti komentar, *tweet*, atau buku yang sudah dilabeli oleh annator yaitu orang yang bertanggung jawab dalam pemberian label positif dan negatif pada setiap data.

2.2.2 Preprocessing

Preprocessing merupakan proses membersihkan data agar mengurangi *noisy* data dan juga mengatasi hilangnya informasi.

2.2.3 Feature Extraction

Data teks perlu diekstrak menjadi data numerik agar bisa diolah dengan komputer. Terdapat tiga teknik ekstraksi fitur yaitu *Bag of Word (TF IDF)*, *Word Embedding (Glove, Word2vec, FastText)*, *Character Embedding*.

2.2.4 Data Split

Data *split* adalah proses membagi dataset menjadi data *training* dan data *testing* dengan perbandingan tertentu.

2.2.5 Classification

Analisis sentimen *machine learning* memakai algoritma dalam mengklasifikasi data. Algoritma yang bisa digunakan adalah *Support Vector Classification*, *Naive Bayes*, *K- Nearest Neighbor*, *Maximum Entropy*, *Ensemble Classifier*.

2.2.6 Evaluasi

Tahap Evaluasi diperlukan untuk mengetahui seberapa efektif model yang telah dibangun. Evaluasi model dilakukan menggunakan data *testing* untuk melihat seberapa jauh kemampuan model untuk mengatasi masalah sebuah pada masalah baru. Kriteria yang digunakan dalam evaluasi model adalah akurasi, presisi, sensitivitas dan spesifisitas.

2.3 Python

Python merupakan salah bahasa pemrograman yang dapat digunakan dalam *data mining*. Keunggulan *python* dalam *data mining* adalah sebagai berikut (Nagpal & Gabrani, 2019):

1. Sintaks yang ringkas dan struktur yang sederhana

Python memiliki dengan sintaks yang ringkas, penggunaan *keyword* yang lebih sedikit, dan struktur yang sederhana. Hal ini memudahkan proses pengembangan solusi dan memahami kode.

2. Open Source Libraries

Python memiliki sejumlah *open source library* yang memudahkan dalam pengembangan sistem. Penggunaan *library* secara signifikan menghemat waktu pengembangan dengan menyediakan fungsionalitas yang sudah siap pakai. Beberapa *library* yang bisa digunakan adalah:

- a. *Scikit-learn* digunakan untuk machine learning dan analisis data.
- b. *Numpy* digunakan untuk perhitungan numerik.
- c. *Pandas* digunakan untuk manipulasi dan analisis data.
- d. *NLTK* digunakan untuk pemrosesan bahasa alami secara umum
- e. *Sastrawi* digunakan untuk pemrosesan bahasa alami dalam bahasa Indonesia
- f. *Matplotlib* digunakan untuk visualisasi data.
- g. *Word cloud* digunakan untuk membuat *word cloud* dari teks.

3. Fleksibilitas Struktur Data

Python memiliki berbagai struktur data seperti *sets*, *list*, *dictionaries*, *tupel*, dan *strings*. Setiap struktur memiliki fungsi tersendiri sesuai kebutuhan data mining. Contohnya *list* yang digunakan menyimpan sejumlah element yang dapat diindeks, diiris dan dipisah. *Dictionaries* memetakan pasangan *key* dan *value*, dimana *key* digunakan sebagai *indeks* untuk mengakses *value* yang terkait. Variasi struktur data memudahkan untuk memilih struktur data yang cocok untuk representasi dan manipulasi data yang berbeda-beda sesuai kebutuhan analisis sentimen.

Dengan sintaks sederhana, *open source libraries* dan fleksibilitas struktur data yang dimiliki, *Python* menjadi pilihan yang tepat untuk melakukan analisis sentimen. *Python* membantu proses analisis sentimen menjadi lebih efisien dan efektif.

2.4 Text Mining

Text mining merupakan ilmu untuk memproses pengambilan informasi dari sebuah teks yang diperoleh dari peramalan pola dan kecenderungan melalui pola statistik (Athira Luqyana, Cholissodin, & Perdana, 2018). *Text mining* digunakan

untuk menganalisis informasi menarik dan relevan secara efektif dan efisien dari sejumlah besar data yang tidak terstruktur. Penerapan *text mining* dalam berbagai area yaitu , *filters email*, analisis saran produk, deteksi penipuan, dan analisis media sosial. *Text mining* juga digunakan untuk penambangan opini meliputi ekstraksi fitur, analisis sentimen, analisis prediktif, dan analisis *trend* (Talib, Kashif, Ayesha, & Fatima, 2016).

2.4.1 Text Preprocessing

Text Preprocessing adalah proses mengekstraksi dokumen sesuai dengan format yang baku agar dapat diolah lebih lanjut. *Text processing* dilakukan dengan langkah-langkah sebagai berikut (Dwi Antonio et al., 2022) :

1. Cleansing

Cleansing adalah proses ini membersihkan dokumen dari tanda baca, *mention*, *hashtag*, link, email dan karakter lainnya yang kurang penting. *Cleansing* bertujuan untuk mengurangi noise.

a. Regular Expression

Regular Expressions adalah pola notasi untuk memanipulasi dan mencocokkan sebuah kata yang ingin dicari. *Regular expression* memiliki karakter khusus yang berguna untuk mencocokkan karakter dengan pola tertentu. Regular expression bermanfaat dalam pemograman seperti memeriksa validitas alamat email, mencari kata tertentu pada sebuah dokumen, membersihkan data dan mengubah data yang tidak teratur. *Regular expression* terdapat dalam berbagai bahasan seperti *Javascript* dan *Python*, namun memiliki implementasi berbeda pada tiap bahasa.

2. Case Folding

Case folding adalah proses mengubah seluruh huruf pada dokumen menjadi huruf kecil. *Case folding* bertujuan untuk menghindari perbedaan tulisan yang dapat mengganggu konsistensi dan akurasi dalam pengolahan dokumen.

3. *Tokenizing*

Tokenizing adalah proses memotong atau memecah kalimat menjadi beberapa bagian atau kata yang menyusunnya. Proses ini juga menghapus tanda baca, angka, karakter, simbol dan *whitespace*. *Tokenizing* bertujuan untuk mempermudah dalam proses analisis data.

4. *Filtering*

Filtering adalah proses menghapus kata yang tidak memiliki arti penting seperti kata keterangan (adverbia), kata ganti (pronomina), kata seru (interjeksi), kata depan (preposisi). *Filtering* memilih kata-kata yang penting untuk menyederhanakan dokumen agar meningkatkan efisiensi dan kualitas proses analisis.

5. *Stemming*

Stemming adalah proses mengubah kata berimbuhan menjadi menjadi kata dasar. Proses ini mencocokkan daftar kata berimbuhan dengan dokumen kemudian mengubahnya menjadi kata dasar. *Stemming* bertujuan untuk mengurangi variasi kata dalam dokumen agar meningkatkan kemampuan sistem untuk mengenali kata-kata yang memiliki arti yang sama.

2.5 Pembobotan kata *TF - ID*

Pembobotan kata adalah proses menghitung bobot tiap kata dalam dokumen untuk mengetahui seberapa penting suatu kata dalam sebuah dokumen. Pembobotan kata dilakukan menggunakan metode *Term Frequency - Inverse Document Frequency* (TF-IDF) dengan menggabungkan frekuensi suatu kata muncul dalam sebuah dokumen dengan frekuensi suatu kata muncul dalam seluruh dokumen. *Inverse document frequency* merupakan logaritma pembagian dari jumlah total dokumen dalam kumpulan dokumen dengan jumlah dokumen yang mengandung kata tertentu. Bobot kata yang hanya muncul dalam sejumlah kecil dokumen akan memiliki nilai IDF yang tinggi, sementara kata yang muncul dalam semua dokumen akan memiliki nilai IDF 0. Pembobotan kata TF - IDF dilakukan

dengan langkah-langkah sebagai berikut (Mee, Homapour, Chiclana, & Engel, 2021)

1. Menghitung *term frequency* (tf) yaitu jumlah kata muncul dalam sebuah dokumen.
2. Menghitung total *document frequency* (df) yaitu total *document frequency* (df) kata pada seluruh dokumen
3. Menghitung *inverse document frequency* (idf) yaitu mengukur seberapa umum kata muncul pada seluruh dokumen. Hasil idf akan memberikan nilai kebalikan dari df. Perhitungan idf menggunakan rumus sebagai berikut:

$$idf = \log \left(\frac{N}{df} \right) \quad (1)$$

4. Menghitung nilai bobot TF-IDF dengan menggabungkan hasil perhitungan TF dan IDF. Hasil ini menunjukkan seberapa penting term dalam suatu dokumen dan secara keseluruhan dokumen. menggunakan rumus sebagai berikut:

$$W(t, d) = tf(t, d) \times idf \quad (2)$$

Keterangan:

N = jumlah seluruh dokumen pada kumpulan dokumen

df = jumlah total dokumen yang terdapat di dalamnya kata i

$W(t, d)$ = bobot kata i pada dokumen d

$tf(t, d)$ = frekuensi kemunculan kata t pada dokumen d

idf = nilai idf term ke- i

2.6 Improved K-Nearest Neighbor

K-Nearest Neighbor merupakan salah satu algoritma *supervised learning* yang memanfaatkan mayoritas *k-values* dari tetangga terdekat untuk mengklasifikasikan data. Sedangkan, *Improved K-Nearest Neighbor* merupakan modifikasi dari metode *K-Nearest Neighbor*. Perbedaan utama antara keduanya terletak pada pengaturan nilai *k-values* sebagai parameter jumlah tetangga terdekat. Pada metode *K-Nearest Neighbor*, nilai *k-values* untuk setiap kategori memiliki nilai yang sama. Sementara itu, *Improved K-Nearest Neighbor* memiliki nilai *k-values* yang berbeda pada masing-masing kelas sehingga ketika nilai k semakin tinggi, hasil kategori tidak dipengaruhi oleh kategori dengan jumlah data *training*

yang lebih besar. Fenomena ini berpengaruh terhadap tingkat akurasi analisis sentiment (Baoli et al., 2003)

Tahapan klasifikasi dokumen menggunakan algoritma *Improved K-Nearest Neighbor* adalah sebagai berikut:

1. Melakukan *text pre processing* pada dokumen *training* dan dokumen *testing*
2. Melakukan perhitungan bobot kata menggunakan TF - IDF pada dokumen *training* dan dokumen *testing*
3. Menghitung kemiripan dokumen *testing* terhadap seluruh dokumen *training* atau menggunakan *cosine similarity* menggunakan persamaan:

$$\text{CosSim}(q, dj) = \frac{\sum_{i=1}^m W_{i,q} \cdot W_{i,j}}{\sqrt{\sum_{i=1}^m W_{i,q}^2} \cdot \sqrt{\sum_{i=1}^m W_{i,j}^2}} \quad (3)$$

Keterangan :

q = dokumen testing

dj = dokumen training ,

$W_{i,q} \cdot W_{i,j}$ = nilai bobot yang diberikan pada setiap kata pada dokumen

4. Mengurutkan hasil cosine similarity dari terbesar.
5. Menghitung k-value baru menggunakan persamaan sebagai berikut

$$n = \left\lceil \frac{k \times N(Cm)}{\max\{n(Cm) \mid j=1 \dots Nc\}} \right\rceil \quad (4)$$

Keterangan:

n = Nilai *k-values* baru

k = Nilai *k-values* awal

$N(Cm)$ = Jumlah data *training*

$\max\{n(Cm) \mid j = 1 \dots Nc\}$ = Jumlah data *training* terbanyak pada seluruh kategori

6. Menghitung probabilitas dokumen *testing* menjadi kelas dokumen *training* terhadap setiap kelas

$$p(x, Cm) = \operatorname{argmax}_m \frac{\sum_{dj \in \text{top } n \text{ KNN}(Cm)} \text{sim}(x, dj) y(dj, Cm)}{\sum_{dj \in \text{top } n \text{ KNN}(Cm)} \text{sim}(x, dj)} \quad (5)$$

Keterangan:

$p(x, Cm)$ = Probabilitas data x menjadi anggota kelas

$Cm(x, dj)$ = *cosine similarity* antara data dengan dokumen x dj

top n KNN = Top n tetangga

$y(djCm)$ = Fungsi atribut yang memenuhi kategori akan bernilai 1

7. Membandingkan hasil nilai probabilitas pada setiap kelas. Dokumen *testing* akan masuk ke dalam kelas yang memiliki probabilitas terbesar.

2.7 Evaluasi

Evaluasi dilakukan untuk mengukur untuk menentukan seberapa tepat klasifikasi yang telah dibuat. Evaluasi pada penelitian ini menggunakan *confusion matrix* dan 10 fold cross validation.

2.7.1 K-Fold Cross Validation

Cross validation adalah cara untuk menguji kinerja model menggunakan dataset yang berbeda dari pelatihan model. Model klasifikasi akan dilatih menggunakan data *training* dan data *testing* yang belum digunakan. *Cross validation* bertujuan mengevaluasi kemampuan model dalam mengklasifikasikan data baru yang tidak digunakan dalam estimasi model dan mengidentifikasi masalah seperti bias pemilihan atau *overfitting* (Nti, 2021). Metode ini bekerja dengan membagi dataset menjadi dua bagian yaitu dataset *training* dan dataset *testing*. Dataset *training* digunakan untuk melatih model, sementara dataset *testing* digunakan untuk menguji kinerja model. Pengujian validasi dilakukan dengan membagi dataset menjadi K partisi. Proses validasi akan dilakukan sebanyak nilai K. Setiap iterasi, salah satu subset akan menjadi set pengujian, sementara K subset lainnya digunakan sebagai set pelatihan. Setelah K iterasi selesai akan diukur model dengan menghitung metrik evaluasi (Alwasi'a, 2020)

2.7.2 Confusion Matrix

Confusion matrix adalah tabel yang mencatat hasil dari kinerja klasifikasi. *Confusion matrix* memetakan hasil klasifikasi kedalam empat komponen yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Tabel *Confusion matrix* disajikan pada Tabel 2.1

Tabel 2.1 *Confusion Matrix*

		<i>Predict Class</i>	
		<i>Positive</i>	<i>Negative</i>
<i>True Class</i>	<i>Positive</i>	<i>True Positive</i>	<i>False Negative</i>
	<i>Negative</i>	<i>False Positive</i>	<i>True Negative</i>

(Sumber : Dwi Antonio et al., 2022)

Kondisi *True Positive* adalah data terprediksi kelas positif dan kelas sebenarnya adalah positif. Kondisi *False Positive* adalah data terprediksi kelas *negative* namun kelas sebenarnya adalah *positive*. Kondisi *True Negative* adalah data terprediksi *negative* dan kelas sebenarnya adalah *negative*. Kondisi *False Negative* adalah data diprediksi *negative* dan kelas sebenarnya adalah *positive* (Dwi Antonio et al., 2022)

Pemetaan *confusion matrix* digunakan untuk menghitung *accuracy*, *precision*, dan *recall*. Hasilnya untuk mengetahui optimisasi analisis sentimen (D'souza & Sonawane, 2019).

Accuracy untuk mengukur tingkat keakuratan model dalam mengklasifikasikan dengan benar. *Accuracy* dirumuskan pada persamaan 6.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Precision untuk mengukur tingkat ketepatan dari informasi aktual dengan prediksi model. *Precision* dirumuskan pada persamaan 7.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

Recall untuk mengukur tingkat keberhasilan model dalam menampilkan informasi dalam bentuk analisis sentimen. *Recall* dirumuskan pada persamaan 8

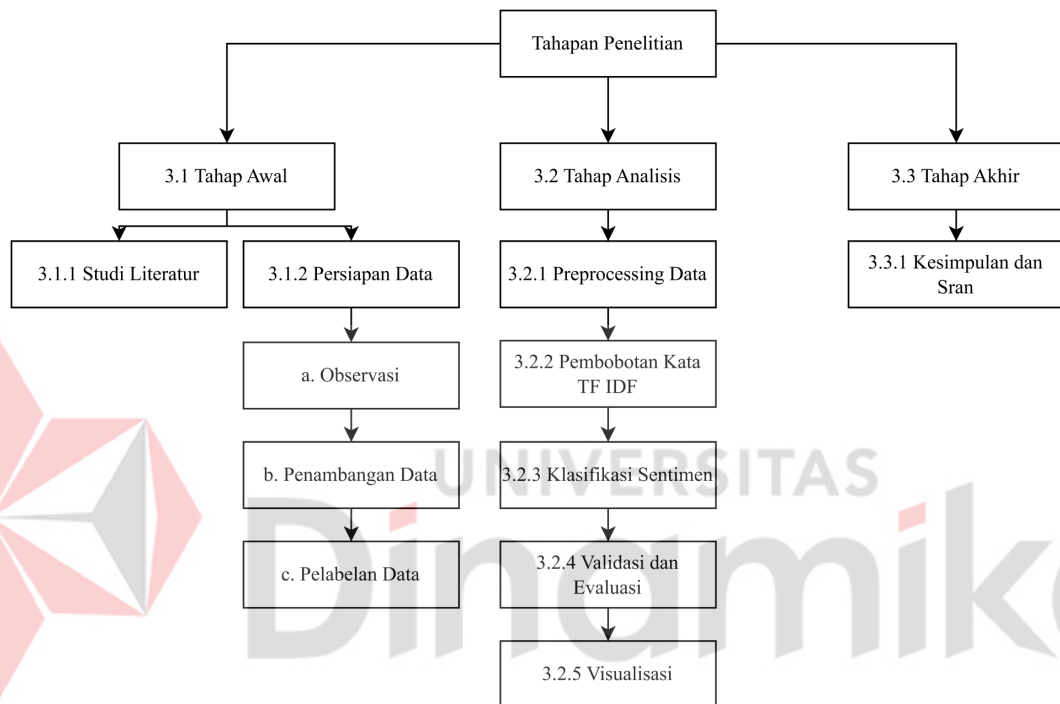
$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

F1-Measure untuk mengukur pengaruh antara *precision* dan *recall*. *F1-Measure* dirumuskan pada persamaan 9

$$\text{F1 - Measure} = \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}} \quad (9)$$

BAB III METODELOGI PENELITIAN

Metode penelitian melibatkan serangkaian tahapan yang dilakukan secara sistematis untuk memastikan bahwa pengerjaan penelitian berjalan dengan baik dan teratur. Penelitian ini memiliki tiga tahapan utama yaitu tahap awal, tahap analisis dan tahap akhir. Tahapan penelitian secara umum ditampilkan pada Gambar 3.1



Gambar 3.1 Tahapan Penelitian

3.1 Tahap Awal

Tahap awal adalah langkah pertama yang dilakukan untuk melakukan analisis. Proses dasar yang dilakukan meliputi studi literature dan persiapan Data. Persiapan data dibagi menjadi tiga bagian yaitu observasi, penambangan data dan pelabelan data.

3.1.1 Studi literatur

Studi literatur adalah tahap mendapatkan dasar dan acuan teori yang membantu dalam penyelesaian masalah. Dilakukan dengan kajian pada jurnal terdahulu serta referensi teori-teori yang berkaitan dengan penyelesaian masalah.

3.1.2 Persiapan Data

Tahapan ini dilakukan pengumpulan data dari media sosial *twitter* yang disebut *crawling*. Data yang diambil menggunakan *tweet* yang mengandung *keyword* “@BPJSKesehatanRI”, “layanan BPJS Kesehatan”, “#BPJSKesehatan”. Tahapan pengumpulan data adalah sebagai berikut:

1. Observasi

Observasi adalah proses mengamati dan memahami objek yang akan diteliti. *Tweets* diamati dengan *keyword* yang telah ditentukan pada berbagai akun pengirim, dengan tujuan untuk menentukan isi *tweet* yang akan diproses dan diolah. Contoh *tweet* ditampilkan pada Gambar 3.2



Gambar 3.2 Contoh *Tweet*

2. Pengumpulan Data

Pengumpulan data menggunakan teknik *scrapping* dengan bantuan *library twint* pada *python*. Keunggulan menggunakan *library twint* dibandingkan dengan teknik *crawling* API adalah *twint* tidak memiliki batasan waktu terbit *tweet*, sehingga memungkinkan pengambilan *tweet* yang lebih lama. Hasil pengumpulan *tweets* disajikan pada Tabel 3.1.

Tabel 3.1 Hasil pengumpulan *tweets*

No	Input <i>Cleansing</i>	Output <i>Cleansing</i>
----	------------------------	-------------------------

1	rahyangnsntra	Gitu aja sih ceritaku pakai BPJS Kesehatan. Pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya. Terima kasih @BPJSKesehatanRI
2	tvblora	@CNNIndonesia Pelayanan BPJS kesehatan kurang optimal, serahkan swasta aja agar pelayanannya bagus
3	rabdulah	@Sofianpalupi pengalaman mertua saya harus masuk ICU 1 minggu. Diitung2 bayar hampir 100 jutaan. Ada BPJS, free. Tks @BPJSKesehatanRI .

3. Pelabelan Data

Tahapan ini dilakukan pelabelan dengan memberikan atribut label positif dan negatif pada data *tweets*. Pelabelan dilakukan oleh sukarelawan dengan tujuan memberikan label secara objektif. Pemberian label positif dan negatif diberikan pada *tweet* yang mengandung makna sentimen kelas tersebut. Pelabelan Data disajikan pada Tabel 3.2

Tabel 3.2 Hasil Pelabelan *Tweets*

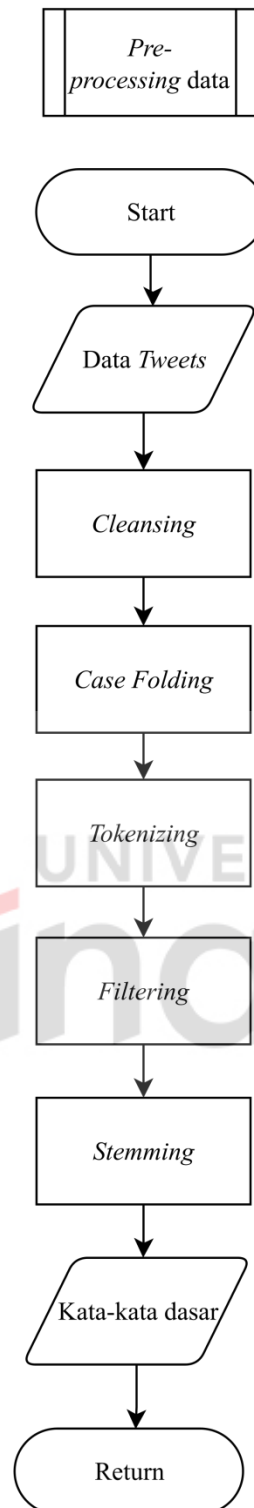
No	Username	<i>Tweet</i>	Label
1	rahyangnsntra	Gitu aja sih ceritaku pakai BPJS Kesehatan. Pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya. Terima kasih @BPJSKesehatanRI	negatif
2	tvblora	@CNNIndonesia Pelayanan BPJS kesehatan kurang optimal, serahkan swasta aja agar pelayanannya bagus	positif
3	rabdulah	@Sofianpalupi pengalaman mertua saya harus masuk ICU 1 minggu. Diitung2 bayar hampir 100 jutaan. Ada BPJS, free. Tks @BPJSKesehatanRI. Semoga kedepan semakin baik, menjadi best practice welfare state policy di Indonesia	?

3.2 Tahap Analisis

Tahapan analisis adalah proses menganalisis teks atau data yang berkaitan dengan opini, ulasan, atau komentar untuk menentukan perasaan atau sentimen yang terkandung di dalamnya.

3.2.1 *Preprocessing* Data

Data yang diambil merupakan data berformat tidak terstruktur, sehingga perlu dibersihkan untuk mendapat data berformat kata dasar tersusun. *Preprocessing* memiliki 6 tahapan yaitu *cleansing*, *case folding*, *tokenizing*, *filtering* dan *stemming*. *Tools* yang digunakan adalah *jupyter notebook*. Alur proses *preprocessing* data ditampilkan pada Gambar 3.3



Gambar 3.3 Diagram Alir *Preprocessing* Data

1. *Cleansing*

Pada proses ini data *tweets* dibersihkan menjadi format teks yang diinginkan. Proses ini meliputi penghapusan link, mention, *retweet*, angka, tanda baca dengan bantuan *Regular expression*. *Regular expression* berfungsi membersihkan dan

mengubah data yang tidak teratur dengan mencocokkan karakter pada pola tertentu. Contoh hasil cleansing dapat dilihat pada Tabel 3.3

Tabel 3.3 Hasil *Cleansing*

No	Input <i>Cleansing</i>	Output <i>Cleansing</i>
1	@CNNIndonesia Pelayanan BPJS kesehatan kurang optimal, serahkan swasta aja agar pelayanannya bagus	Pelayanan BPJS kesehatan kurang optimal serahkan swasta aja agar pelayanannya bagus
2	Gitu aja sih ceritaku pakai BPJS Kesehatan. Pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya. Terima kasih @BPJSKesehatanRI	Gitu aja sih ceritaku pakai BPJS Kesehatan Pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya Terima kasih
3	@Sofianpalupi pengalaman mertua saya harus masuk ICU 1 minggu. Diitung2 bayar hampir 100 jutaan. Ada BPJS, free. Tks @BPJSKesehatanRI . Semoga ke depan semakin baik, menjadi best practice welfare state policy di Indonesia	pengalaman mertua saya harus masuk ICU minggu Diitung bayar hampir jutaan Ada BPJS free Tks Semoga ke depan semakin baik menjadi best practice welfare state policy di Indonesia

2. Case Folding

Proses ini melakukan penyeragaman penggunaan huruf menjadi huruf kecil. Input dari proses ini berasal dari hasil *cleansing*. Data akan ditransformasi menjadi huruf kecil menggunakan *function lower.str()*. Hasil *Case folding* ditampilkan pada Tabel 3.4.

Tabel 3.4 Hasil *Case folding*

No	Input <i>Case folding</i>	Output <i>Case folding</i>
1	Gitu aja sih ceritaku pakai BPJS Kesehatan Pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya Terima kasih	gitu aja sih ceritaku pakai bpjs kesehatan pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya terima kasih
2	Pelayanan BPJS kesehatan kurang optimal serahkan swasta aja agar pelayanannya bagus	pelayanan bpjs kesehatan kurang optimal serahkan swasta aja agar pelayanannya bagus
3	pengalaman mertua saya harus masuk ICU minggu Diitung bayar hampir jutaan Ada BPJS free Tks Semoga ke depan semakin baik menjadi best practice welfare state policy di Indonesia	pengalaman mertua saya harus masuk icu minggu diitung bayar hampir jutaan ada bpjs free tks semoga ke depan semakin baik menjadi best practice welfare state policy di indonesia

3. Tokenizing

Proses ini memotong atau memecah kalimat menjadi beberapa bagian atau kata yang menyusunnya. *Tokenizing* meliputi proses penghapusan spasi, penghapusan tanda baca dan memecah string kedalam token. Data menggunakan

function .apply() dan *function word_tokenize()*. Hasil *tokenizing* ditampilkan pada Tabel 3.5.

Tabel 3.5 Hasil *Tokenizing*

No	<i>Input Tokenizing</i>	<i>Output Tokenizing</i>
1	segitu saja sih ceritaku pakai bpjs kesehatan pengalaman pertama memakai fasilitas ini dan gue sangat puas dengan segala macam layanan dan fasilitasnya terima kasih	[segitu, saja, sih, ceritaku, pakai, bpjs, kesehatan, pengalaman, pertama, memakai, fasilitas, ini, dan, gue, sangat, puas, dengan, segala, macam, layanan, dan, fasilitasnya, terima, kasih]
2	pelayanan bpjs kesehatan buruk serahkan swasta saja agar pelayanannya bagus	[pelayanan, bpjs, kesehatan, buruk, serahkan, swasta, saja, agar, pelayanannya, bagus]
3	pengalaman mertua saya harus masuk icu minggu ditung bayar hampir jutaan ada bpjs free tks semoga ke depan semakin baik menjadi best practice welfare state policy di indonesia	[pengalaman, mertua, saya, harus, masuk, icu, minggu, ditung, bayar, hampir, jutaan, ada, bpjs, free, terima, kasih, semoga, ke, depan, semakin, baik, menjadi, best, practice, welfare, state, policy, di, indonesia]

4. Filtering

Filtering adalah tahapan menghapus kata-kata yang tidak memiliki makna.

Proses ini menggunakan kamus *stopword* bahasa Indonesia dari *library Natural Language Toolkit* (NLTK). Hasil *filtering* disajikan pada Tabel 3.6.

Tabel 3.6 Hasil *Filtering*

No	<i>Input Filtering</i>	<i>Output Filtering</i>
1	[segitu, saja, sih, ceritaku, pakai, bpjs, kesehatan, pengalaman, pertama, memakai, fasilitas, ini, dan, gue, sangat, puas, dengan, segala, macam, layanan, dan, fasilitasnya, terima, kasih]	[ceritaku, pakai, bpjs, kesehatan, pengalaman, pertama, memakai, fasilitas, gue, sangat, puas, segala, macam, layanan, fasilitasnya, terima, kasih]
2	[pelayanan, bpjs, kesehatan, buruk, serahkan, swasta, saja, agar, pelayanannya, bagus]	[pelayanan, bpjs, kesehatan, buruk, serahkan, swasta, pelayanannya, bagus]
3	[pengalaman, mertua, saya, harus, masuk, icu, minggu, dihitung, bayar, hampir, jutaan, ada, bpjs, free, terima, kasih, semoga, ke, depan, semakin, baik, menjadi, best, practice, welfare, state, policy, di, indonesia]	[pengalaman, masuk, icu, minggu, dihitung, bayar, hampir, jutaan, bpjs, free, terima, kasih, semoga, ke, depan, semakin, baik, menjadi, best, practice, welfare, state, policy, di, indonesia]

5. Stemming

Proses ini mengembalikan kata ke bentuk dasarnya dengan *library swifter* dan *library sastrawi* sebagai pedoman penggunaan kata Bahasa Indonesia. Hasil *Stemming* disajikan pada Tabel 3.7.

Tabel 3.7 Hasil *Stemming*

Kode	<i>Input Filtering</i>	<i>Output Filtering</i>
------	------------------------	-------------------------

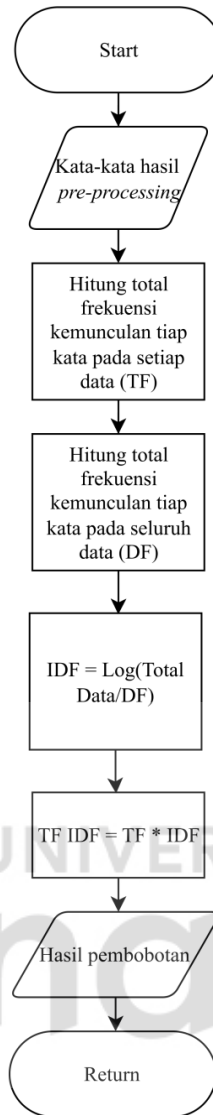
1	[ceritaku, pakai, bpjs, kesehatan, pengalaman, pertama, memakai, fasilitas, gue, sangat, puas, segala, macam, layanan, fasilitasnya, terima, kasih]	[cerita, pakai, bpjs, sehat, pengalaman, pertama, pakai, fasilitas, puas, macam, layanan, fasilitas, terima, kasih]
2	[pelayanan, bpjs, kesehatan, buruk, serahkan, swasta, pelayanannya, bagus]	[layanan, bpjs, sehat, buruk, serah, swasta, layanan, bagus]
3	[pengalaman, masuk, icu, minggu, dihitung, bayar, hampir, jutaan, bpjs, free, terima, kasih, semoga, ke, depan, semakin, baik, menjadi, best, practice, welfare, state, policy, di, indonesia]	[pengalaman, masuk, icu, hitung, bayar, juta, bpjs, free, terima, kasih, makin, baik, best, practice, welfare, state, policy, indonesia]

3.2.2 Pembobotan Kata TF IDF

Pembobotan kata adalah proses memberikan bobot pada tiap kata didalam dokumen. Proses ini menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) yang tersedia di *library Scikit-Learn Python*. Pembobotan dilakukan pada setiap kata diseluruh dokumen. Hasil dari pembobotan ini digunakan untuk klasifikasi menggunakan algoritma *Improved K-Nearest Neighbor* untuk klasifikasi dokumen. Gambar 3.4 menyajikan diagram alir TF-IDF.



UNIVERSITAS
Dinamika



Gambar 3.4 Diagram Alir TF IDF

1. Proses I: Menghitung kemunculan kata setiap data. Hasil perhitungan dapat dilihat pada Tabel 3.8

Tabel 3.8 Hasil Perhitungan *Term Frequency*

Kata	<i>Term Frequency</i>		
	D1	D2	D3
cerita	1	0	0
pakai	1	0	0
bpjs	1	1	1
sehat	1	1	0
pengalaman	1	0	1
pertama	1	0	0
pakai	1	0	0
fasilitas	1	0	0
puas	1	0	0
macam	1	0	0
layanan	1	1	0
terima	1	0	0
kasih	1	0	0

buruk	0	1	0
serah	0	1	0
swasta	0	1	0
bagus	0	1	1

2. Proses II: Menghitung total frekuensi tiap kata pada seluruh dokumen (DF).

Hasil perhitungan DF disajikan pada Tabel 3.9.

Tabel 3.9 Hasil Perhitungan Dokumen *Frequency*

Kata	D1	D2	D3	DF
cerita	1	0	0	1
pakai	1	0	0	1
bpjs	1	1	1	3
sehat	1	1	0	2
pengalaman	1	0	1	2
pertama	1	0	0	1
pakai	1	0	0	1
fasilitas	1	0	0	1
puas	1	0	0	1
macam	1	0	0	1
layanan	1	1	0	2
terima	1	0	0	1
kasih	1	0	0	1
buruk	0	1	0	1
serah	0	1	0	1
swasta	0	1	0	1
bagus	0	1	1	1

3. Proses III: Menghitung IDF menggunakan persamaan (1) berdasarkan data pada

Tabel 3.9. Hasil perhitungan IDF disajikan pada Tabel 3.10.

Tabel 3.10 Hasil Perhitungan IDF

Kata	D1	D2	D3	DF
cerita	1	0	0	1
pakai	1	0	0	1
bpjs	1	1	1	3
sehat	1	1	0	2
pengalaman	1	0	1	2
pertama	1	0	0	1
pakai	1	0	0	1
fasilitas	1	0	0	1
puas	1	0	0	1
macam	1	0	0	1
layanan	1	1	0	2
terima	1	0	0	1
kasih	1	0	0	1
buruk	0	1	0	1
serah	0	1	0	1
swasta	0	1	0	1
bagus	0	1	1	1

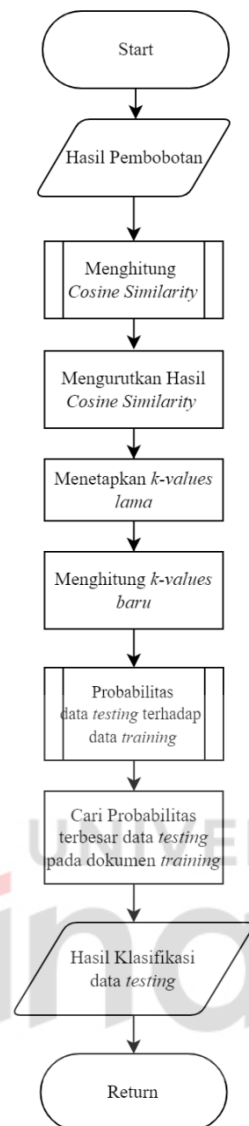
4. Proses IV: Menghitung TF-IDF menggunakan persamaan (2). Seluruh hasil TF pada Tabel 3.8 dikalikan dengan hasil IDF pada Tabel 3.10. Hasil perhitungan TF-IDF ditampilkan pada Tabel 3.11

Tabel 3.11 Hasil Perhitungan TF-IDF

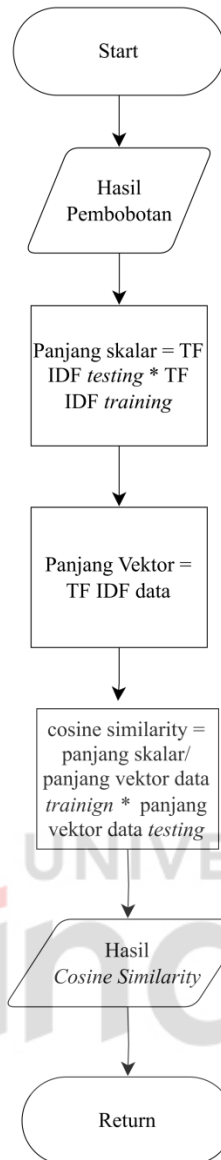
D1	D2	D3	DF
0,4771	0	0	0,4771
0,4771	0	0	0,4771
0	0	0	0
0,1761	0,1761	0	0,1761
0,1761	0	0,1761	0,1761
0,4771	0	0	0,4771
0,4771	0	0	0,4771
0,4771	0	0	0,4771
0,4771	0	0	0,4771
0,4771	0	0	0,4771
0,1761	0,1761	0	0,1761
0,4771	0	0	0,4771
0,4771	0	0	0,4771
0	0,4771	0	0
0	0,4771	0	0
0	0,4771	0	0
0	0,4771	0	0

3.2.3 Klasifikasi Sentimen

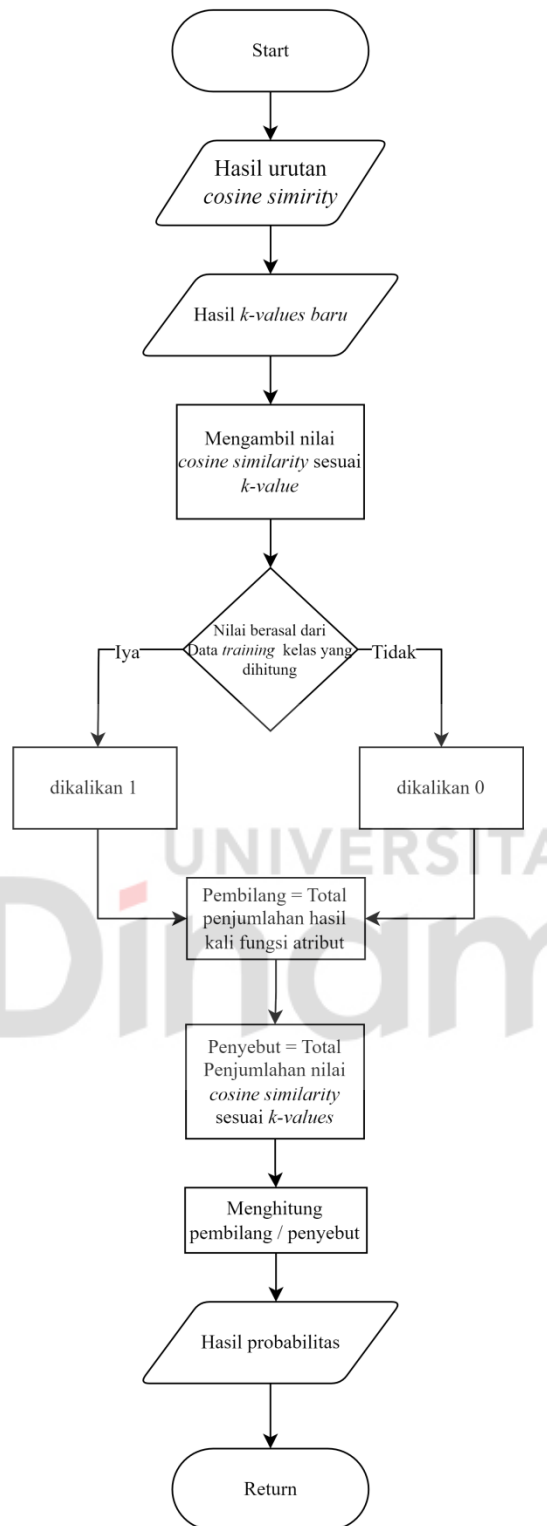
Tahap ini mengklasifikasikan data *tweets* menggunakan algoritma *Improved K-Nearest Neighbor*. Algoritma ini merupakan modifikasi dari metode *K-Nearest Neighbor*, di mana *k-value* pada setiap kategori memiliki nilai yang berbeda. Proses klasifikasi ditampilkan pada Gambar 3.5. Detail proses lanjutan klasifikasi sentimen di tampilkan pada Gambar 3.6 dan Gambar 3.7.



Gambar 3.5 Diagram Alir Klasifikasi Sentimen



Gambar 3.6 Diagram Alir *Cosine Similarity*



Gambar 3.7 Diagram Alir menghitung Probabilitas

1. Proses I : Menghitung *Cosine similarity* data *training* dan data *testing*. Gambar 3.6 menjelaskan detail menghitung *cosine similarity*. Proses perhitungan *cosine similarity* adalah sebagai berikut :

a. *Cosine Similarity*

1) Menghitung panjang skalar dengan mengalikan bobot tiap kata pada data *testing* dengan bobot data *training* kemudian menghitung total tiap dokumen. Hasil perhitungan skalar disajikan pada Tabel 3.12.

Tabel 3.12 Hasil Perhitungan Skalar

Kata	D3*D1	D3*D2
cerita	0	0
pakai	0	0
bpjs	0	0
sehat	0	0
pengalaman	0,031	0
pertama	0	0
pakai	0	0
fasilitas	0	0
puas	0	0
macam	0	0
layanan	0	0
terima	0,031	0
kasih	0,031	0
buruk	0	0
serah	0	0
swasta	0	0
bagus	0	0
Jumlah	0,093	0

2) Menghitung panjang vektor tiap data dengan menghitung kuadrat hasil pembobotan data *training* dan *testing* yang diakar. Perhitungan panjang vektor menggunakan rumus persamaan (3). Hasil perhitungan panjang vektor ditampilkan pada Tabel 3.1.

Tabel 3.13 Hasil Perhitungan Panjang Vektor

Kata	D1	D2	D3
cerita	0,6907	0	0
pakai	0,6907	0	0
bpjs	0	0	0
sehat	0,4196	0,4196	0
pengalaman	0,4196	0	0,4196
pertama	0,6907	0	0
pakai	0,6907	0	0
fasilitas	0,6907	0	0
puas	0,6907	0	0
macam	0,6907	0	0
layanan	0,4196	0,4196	0
terima	0,4196	0	0,4196
kasih	0,4196	0	0,4196
buruk	0	0,6907	0

serah	0	0,6907	0
swasta	0	0,6907	0
bagus	0	0,6907	0
Total jumlah	6,9329	3,602	1,2588
Hasil	1,3222	0,9861	0,305

3) Menghitung kemiripan data *testing* dan data *training*, pada contoh diatas dilakukan perhitungan kemiripan D3 terhadap data D1 dan D2 menggunakan persamaan (3) .

$$\text{Cos}(D3,D1) = 0,093 / (0,305 * 1,3222) = 0,034$$

$$\text{Cos}(D3,D2) = 0 / (0,305 * 0,9861) = 0$$

Hasil *similarity* data D3 terhadap D1 adalah 0,0343 sedangkan D3 terhadap D2 adalah 0.

2. Proses II : Mengurutkan hasil *cosine similarity*

3. Proses III: Menetapkan k-value lama.

4. Proses IV: Menghitung k-value baru pada masing-masing kelas menggunakan persamaan (4). Tabel 3.14 menyajikan hasil jumlah total data *training* dan Tabel 3.15 menyajikan hasil k-values baru.

Tabel 3.14 Jumlah Data *Training*

Data Training		
Positif	Negatif	Jumlah
1	1	2

$$n_{\text{positif}} = 1 * 1 / 1 = 1$$

$$n_{\text{negatif}} = 1 * 1 / 1 = 1$$

Tabel 3.15 *K-value* Baru

Nilai K	<i>n(k-value</i> baru)	
	Positif	Negatif
1	1	1

5. Proses V: Menghitung probabilitas kemiripan data *testing* terhadap data *training* pada suatu kelas menggunakan persamaan (5). Perhitungan probabilitas diuraikan pada Gambar 3.7. Nilai *cosinus similarity* akan dikalikan dengan fungsi atribut tiap kelas. Jika nilai berasal dari kelas data *training* yang sedang dihitung maka akan dikalikan dengan 1. Jika nilai bukan berasal dari data *training* yang dihitung maka dikalikan dengan 0. Kemudian dibagi dengan total *top N* tiap kelas.

$$P_{\text{positif}} = (0,0343 * 1) / 0,0343 = 1$$

$$P_{\text{negatif}} = (0,0343 * 0) / 0,0343 = 0$$

6. Proses VI: Membandingkan hasil probabilitas kemiripan dokumen tiap kelas. Hasil maksimum dari perhitungan merupakan kelas dari dokumen *testing*, sehingga dapat diklasifikasikan kategori dari kelas dokumen 3 adalah kelas positif.

3.2.4 Evaluasi dan Validasi

Tahap ini mengukur akurasi dan kualitas klasifikasi terhadap metode *Improved K-Nearest Neighbor*. Tujuannya untuk membandingkan hasil akurasi *Improved K-Nearest Neighbor* dengan hasil yang ada. Proses validasi menggunakan *10-folds cross-validation* dengan membagi data *training* menjadi 10 bagian. Selanjutnya *confusion matrix* digunakan untuk mengukur *accuracy*, *precision*, *recall* dan *f-Measure* pada *Improved K-Nearest Neighbor*. Pengukuran menggunakan persamaan (6), (7), (8), (9).

3.2.5 Visualisasi

Hasil analisis disajikan melalui visualisasi berupa *word cloud* dan diagram lingkaran (*pie chart*). *Word cloud* menampilkan kata-kata yang sering muncul dalam data *tweets*, dengan ukuran huruf yang berbeda untuk menggambarkan frekuensi kemunculan kata pada data. Ukuran huruf terbesar melambangkan kata dengan frekuensi terbanyak. Diagram *Pie Chart* digunakan untuk menampilkan perbandingan persentase masing-masing kelas.

3.3 Tahap Akhir

3.3.1 Kesimpulan dan Saran

Tahap akhir penelitian menyajikan kesimpulan dan saran mengenai hasil analisis sentimen yang menggunakan metode *Improved K-Nearest Neighbor*. Kesimpulan memberikan jawaban terhadap pertanyaan penelitian serta gambaran keseluruhan tentang hasil penelitian ini. Output dari analisis ini berupa klasifikasi data menjadi dua kelas, yaitu kelas positif dan negatif, berdasarkan opini publik terhadap pelayanan BPJS Kesehatan yang diambil dari *tweets Twitter*.

Saran bertujuan memberi masukan terhadap penelitian analisis sentimen opini publik terhadap pelayanan BPJS Kesehatan. Penelitian ini dapat menjadi referensi bagi penelitian lainnya yang berfokus pada topik serupa.



UNIVERSITAS
Dinamika

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas implementasi dan hasil dari penelitian berdasarkan tahapan-tahapan yang telah disusun dalam bab III. Implementasi melalui serangkaian tahapan yang meliputi mengumpulkan data dari *twitter*, pelabelan manual, *pre-processing*, pembobotan kata, pembagian data, pengklasifikasian data dengan metode *Improved K-Nearest Neighbor* dan melakukan visualisasi berupa *word cloud* dan diagram pie chart.

4.1 Penambahan data

Tahap pertama adalah penambahan data dari *Twitter*. Penambahan data dilakukan dengan bantuan *library twint python* menggunakan *tools jupyter notebook*. Jumlah data yang diperoleh sebanyak 500 *tweets* dari bulan Januari 2022 sampai April 2022. Data yang telah diambil, disimpan dalam format ekstensi “.csv”

4.2 Pelabelan Data

Pelabelan data dilakukan oleh sukarelawan dengan memberikan label kelas ke 500 data yang terbagi menjadi label positif dan negatif. Proses pemberian label dilakukan berdasarkan makna yang terkandung pada kalimat *tweet* jika data memiliki makna positif, maka akan diberi label positif. Sedangkan jika data bermakna negatif, akan diberi label negatif. Dalam implementasi *sourcecode* sistem, label positif dilambangkan dengan angka "1", sedangkan label negatif dilambangkan dengan angka “2”.

4.3 *Pre-processing* Data

Pre-processing adalah serangkaian langkah atau teknik yang diterapkan pada data *tweet* sebelum data tersebut dapat digunakan untuk analisis. *Pre-processing* membantu memastikan bahwa data yang digunakan dalam analisis memiliki kualitas yang baik, relevan, dan siap digunakan. Hasil *Pre-processing* data disajikan

pada Lampiran 1. *Pre-processing* data dilakukan dengan langkah-langkah sebagai berikut:

4.3.1 *Cleansing*

Cleansing dilakukan dengan menghapus *link*, mention, *retweet*, angka, tanda baca. *Cleansing* menggunakan bantuan *Regular expression*. Implementasi *sourcecode cleansing* disajikan pada Gambar 4.1.

```
def cleansing_tweet(text):
    # menghapus tab, new line, ans back slice
    text = text.replace('\t', " ").replace('\n', " ").replace('\r', "")
    #menghapus double huruf
    text = re.sub(r'\b\w*?(\w)(\1{2,})\w*\b', '', text)
    #menghapus emoticon
    text = text.encode('ascii', 'replace').decode('ascii')
    # menghapus mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\w+\S+)", " ", text).split())
    # menghapus URL
    text = text.replace("http://", " ").replace("https://", " ")
    # menghapus angka
    text = re.sub(r"\d+", "", text)
    # menghapus tanda baca
    text = text.translate(str.maketrans("", "", string.punctuation))
    # menghapus karakter spasi (whitespace) dari awal (leading) dan akhir (trailing)
    text = text.strip()
    # menghapus multiple whitespace menjadi single whitespace
    text = re.sub('\s+', ' ', text)
    # menghapus satu huruf
    text = re.sub(r"\b[a-zA-Z]\b", "", text)
    return text
```

Gambar 4.1 *Sourcecode Cleansing*

4.3.2 *Case Folding*

Case folding adalah proses transformasi yang mengubah seluruh huruf pada data menjadi huruf kecil, sehingga menghasilkan bentuk teks yang seragam. *Case folding* menggunakan *function str.lower()* untuk mengonversi semua karakter dalam sebuah *string* menjadi huruf kecil. Implementasi *sourcecode case folding* ditampilkan pada Gambar 4.2

```
# case folding
df['tweet'] = df['tweet'].str.lower()
```

Gambar 4.2 *Sourcecode Case folding*

4.3.3 Tokenizing

Tokenizing berfungsi pemecahan dokumen menjadi kata-kata berdasarkan spasi. Tujuan *tokenizing* untuk mempermudah pemrosesan teks dan analisis. Implementasi *sourcecode tokenizing* ditampilkan pada Gambar 4.3

```
def token_text(text):
# Tokenize the text
tokens = nltk.word_tokenize(text)
tokens = [token for token in tokens if token not in list_stopwords]
return tokens

df['tweet'] = df['tweet'].apply(token_text)
```

Gambar 4.3 *Sourcecode Tokenizing*

Sourcecode pada Gambar 4.3 menampilkan penggunaan *function* “*word_tokenize()*” dalam *library* NLTK untuk melakukan pemotongan *string*. Setiap elemen pada kolom *tweet* diproses menggunakan *function* “*token_text()*” dengan bantuan *function* *apply()*.

4.3.4 Filtering

Filtering bertujuan untuk menghilangkan elemen-elemen yang tidak relevan sehingga informasi lebih mudah diinterpretasikan. Implementasi *sourcecode* untuk deklarasi *variabel stopwords* ditampilkan pada Gambar 4.4

```
#membaca file stopwords tambahan
with open('stopwords.txt', 'r') as file:
    new_stopwords = [line.strip() for line in file]

#declare stopword
list_stopwords = stopwords.words("indonesian")
list_stopwords.remove("tidak")

#menambahkan stopwords tambahan kedalam list_stopwords
list_stopwords.extend(new_stopwords)
list_stopwords = set(list_stopwords)
```

Gambar 4.4 *Soucecode deklarasi stopwords*

Setiap token kata akan dibandingkan dengan kamus *stopwords* yang tersedia di variabel *list_stopwords*. Method “*.extend(new_stopwords)*” digunakan untuk menambahkan kata-kata yang tidak ada dalam kamus sastra ke dalam daftar *stopwords*. Jika ada kata yang termasuk dalam *stopwords*, maka kata tersebut akan dihapus, begitu juga sebaliknya. Setelah *stopwords* dideklarasikan selanjutnya

diterapkan pada data *tweet*. Implementasi *sourcecode* untuk menjalankan *filtering* ditampilkan pada Gambar 4.5

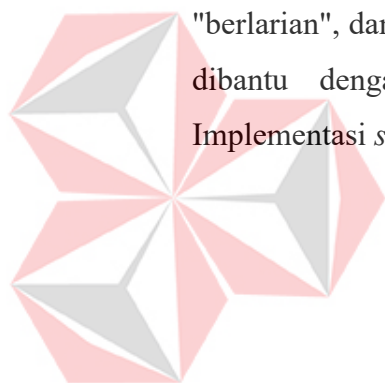
```
def token_text(text):
    # Tokenize the text
    tokens = nltk.word_tokenize(text)
    tokens = [token for token in tokens if token not in list_stopwords]
    return tokens

df['tweet'] = df['tweet'].apply(token_text)
```

Gambar 4.5 *Sourcecode Filtering*

4.3.5 *Stemming*

Stemming mengubah kata-kata menjadi kata dasar menggunakan *library* Sastrawi. Sastrawi merupakan *library stemming python* dalam bahasa Indonesia. Kata-kata dalam data akan dikembalikan ke bentuk dasarnya dengan menghilangkan imbuhan dan awalan. Misalnya, kata-kata seperti "berlari", "berlarian", dan "berlarilyah" akan diubah menjadi kata dasar "lari". *Stemming* juga dibantu dengan *package swifter* untuk mempercepat proses *stemming*. Implementasi *sourcecode stemming* ditampilkan pada Gambar 4.6



UNIVERSITAS
Dinamika

```

#Perform stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()

#create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

#stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)
term_dict = {}

#proses mencatat tiap kata dimasukan kedalam term_dict
for document in df['tweet']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ''

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":",term_dict[term])

#menggantikan kata yang ada dalam tweet dengan term_dict stem
def get_stemmed_term(document):
    result = []
    for term in document:
        # cek apakah kata sudah ada didalam dictionary
        if term not in result:
            result.append(term_dict.get(term))
    return result

df['tweet'] = df['tweet'].swifter.apply(get_stemmed_term)

```

Gambar 4.6 Sourcecode Stemming

Stemming dilakukan dengan mendefinisikan *function stemmed_wrapper()* yang berfungsi mencetak kata-kata dasar pada data *tweets*, kemudian dilakukan pengecekan setiap kata pada data *tweets* dengan kata pada kamus *stemming*. Kamus *stemming* dideklarasikan melalui *stemmed_wrapper()*. Jika kata tersebut memiliki kata dasar maka akan dikembalikan ke bentuk dasar dan disimpan pada variabel *dictionary term_dict*. Sedangkan, jika kata tidak memiliki kata dasar pada kamus akan disimpan pada *array result*. *Array result* menampung hasil dari *term_dict* dan kata yang tidak memiliki kata dasar.

4.4 Pembobotan Kata TF IDF

Pembobotan kata dilakukan untuk mencari kata-kata yang paling relevan dan penting dalam sebuah dokumen. Tahap ini menggunakan bantuan *library* Python *TfidfVectorizer()* untuk melakukan pembobotan kata. Hasil dari metode TF-IDF

adalah nilai bobot kata untuk setiap kata dalam dokumen. *Sourcecode* proses pembobotan TF IDF dapat dilihat pada Gambar 4.7

```
# Deklarasi objek TfidfVectorizer
vectorizer = TfidfVectorizer()

# Menghitung nilai TF-IDF pada data tweet
data_tf_idf = vectorizer.fit_transform(data_gabung)

# Mengubah hasil matrix menjadi array
tfidf = pd.DataFrame(data_tf_idf.toarray(), columns=vectorizer.get_feature_names())
```

Gambar 4.7 *Sourcecode TF IDF*

Sourcecode pada Gambar 4.7 menampilkan “*TfidfVectorizer()*” dideklarasikan sebagai objek “vektorisasi” yang digunakan untuk menghitung nilai TF IDF yang diberikan melalui *method* “*fit_transform()*”. Hasil TF IDF diubah menjadi bentuk *array* dengan *method* “*toarray()*” agar lebih mudah dikelola pada tahap selanjutnya.

4.5 Pembagian *Data Training* dan *Data Testing*

Pembagian data *training* dan data *testing* dilakukan dengan perbandingan 70:30 pada 500 data. Data *training* digunakan untuk melatih model atau algoritma. Sedangkan data *testing* digunakan untuk menguji kinerja model atau algoritma yang digunakan. Sebanyak 350 data digunakan sebagai data *training* dan 150 data sebagai data *testing*. Pembagian data digunakan pada *function* “*train_test_split()*” dari *library sklearn*.

4.6 *Cosine Similarity*

Tahap ini dilakukan untuk mengukur kesamaan antara dua dokumen atau teks berdasarkan representasi vektor kata-kata yang muncul dalam dokumen. Penerapan *cosine similarity* digunakan dengan memanfaatkan *function cosine_similarity()* pada *library sklearn*. Hasil perhitungan *cosine similarity* memiliki rentang antara 0 hingga 1, di mana semakin besar nilai tersebut menunjukkan semakin tinggi kesamaan antara kedua dokumen. Hasil ini akan digunakan sebagai acuan klasifikasi *Improved K-Nearest Neighbor*. *Sourcecode Cosine Similarity* dapat dilihat pada Gambar 4.8

```
cosin = cosine_similarity(x_train,x_test)
cosin = pd.DataFrame(cosin)
```

Gambar 4.8 *Sourcecode Cosine Similarity*

Sourcecode pada Gambar 4.8 menampilkan *function* “*cosine_similarity*” untuk menghitung tingkat kemiripan kosinus antara duaset data yaitu “*x_train*” sebagai data *training* dan “*x_test*” sebagai data *testing*. Hasil perhitungan kemudian diubah menjadi “*DataFrame*” menggunakan *library pandas*. Hal ini dilakukan agar lebih mudah dikelola dan dianalisis pada tahap selanjutnya.

4.7 Klasifikasi *Improved K-Nearest Neighbor*

Tahap ini melakukan proses klasifikasi dokumen kedalam kelas positif atau negatif. Klasifikasi dilakukan dengan mencari *k-value* baru menggunakan persamaan 4, mengurutkan hasil *cosine similarity* dan menghitung probabilitas menggunakan persamaan 5. Implementasi *sourcecode* menghitung *k-value* baru ditampilkan pada Gambar 4.9 dan Gambar 4.10

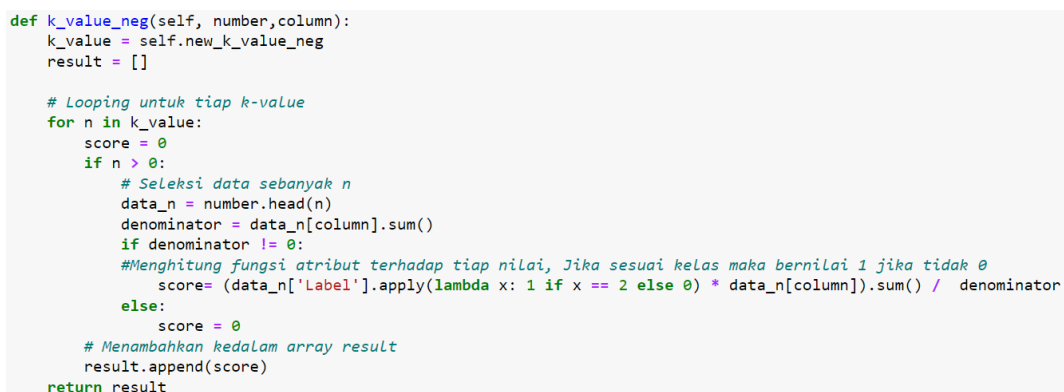


```
def k_value_pos(self, number, column):
    k_value = self.new_k_value_pos
    result = []

    # Looping untuk tiap k-value
    for n in k_value:
        score = 0
        if n > 0:
            # Seleksi data sebanyak n
            data_n = number.head(n)
            denominator = data_n[column].sum()

            if denominator != 0:
                # Menghitung fungsi atribut terhadap tiap nilai, Jika sesuai kelas maka bernilai 1 jika tidak 0
                score = (data_n['Label'].apply(lambda x: 1 if x == 1 else 0) * data_n[column]).sum() / denominator
            else:
                score = 0
            # Menambahkan kedalam array
            result.append(score)
    return result
```

Gambar 4.9 *Sourcecode* Menghitung *k-value* positif



```
def k_value_neg(self, number, column):
    k_value = self.new_k_value_neg
    result = []

    # Looping untuk tiap k-value
    for n in k_value:
        score = 0
        if n > 0:
            # Seleksi data sebanyak n
            data_n = number.head(n)
            denominator = data_n[column].sum()
            if denominator != 0:
                # Menghitung fungsi atribut terhadap tiap nilai, Jika sesuai kelas maka bernilai 1 jika tidak 0
                score = (data_n['Label'].apply(lambda x: 2 if x == 2 else 0) * data_n[column]).sum() / denominator
            else:
                score = 0
            # Menambahkan kedalam array result
            result.append(score)
    return result
```

Gambar 4.10 *Sourcecode* Menghitung *k-value* negatif

Sourcecode pada Gambar 4.9 menampilkan implementasi *function* perhitungan probabilitas pada kelas positif. Sedangkan Gambar 4.10 menampilkan

function perhitungan probabilitas pada kelas negatif. Variabel “*score*” menampung hasil perkalian nilai *cosine similarity* dengan *function* atribut tiap kelas. Hasil iterasi tiap kelas dimasukan kedalam array “*result*”.

```
def hitung_probabilitas(self, data):
    final_result = None
    for column in data:
        if column != 'index' and column != 'Label':
            sorted_data = data.sort_values(column, ascending=False)
            cpos = self.k_value_pos(sorted_data, column)
            cneg = self.k_value_neg(sorted_data, column)

            cpos = pd.DataFrame(cpos, columns=[f'{column}_cpos'])
            cneg = pd.DataFrame(cneg, columns=[f'{column}_cneg'])

            result = pd.concat([cpos, cneg], axis=1)

            if final_result is None:
                final_result = result
            else:
                final_result = pd.concat([final_result, result], axis=1)
    return final_result
```

Gambar 4.11 Sourcecode Improve K-Nearest Neighbor

Gambar 4.11 menampilkan *Sourcecode function* “*hitung_probabilitas()*”. Proses perhitungan dimulai dengan mengurutkan hasil *cosine similarity* yang telah ditampung dengan variabel “*data*” dengan bantuan *function* “*sorted_values()*”. Kemudian menjalankan *function* “*k_value_pos*” dan “*k_value_neg*” untuk menghitung probabilitas pada tiap kelasnya. Hasil perhitungan probabilitas kelas positif dan negatif digabungkan menggunakan *method* “*concat()*” dan disimpan pada variabel “*final_result*”. Hasil penggabungan dibandingkan menggunakan *function* “*compare_result*”. *Sourcecode function* “*compare_result*” ditampilkan pada Gambar 4.12.

```

def compare_result(self,row):
    compare_row = []
    for i in range(0, len(row), 2):
        pos_col = row.index[i]
        neg_col = row.index[i+1]
        if row[pos_col] > row[neg_col]:
            comparison = 1
        elif row[pos_col] < row[neg_col]:
            comparison = 2
        compare_row.append(comparison)

    return compare_row

```

Gambar 4.12 *Sourcecode* Perbandingan Hasil Kelas

Function “*compare_result*” bertujuan membandingkan hasil tiap baris pada kolom positif dan negatif. Hasil perbandingan ditampung kedalam *array* “*compare_row*”. Jika hasil kolom positif lebih besar terhadap kelas negatif maka hasilnya adalah kelas positif yang dilambangkan dengan angka “1”, Sedangkan jika hasil kolom negatif lebih besar terhadap kelas kelas maka hasilnya adalah kelas negatif yang dilambangkan dengan angka “2”. Seluruh *function* klasifikasi sentimen dijalankan pada Gambar 4.13

```

calculator = ImproveKnn(new_k_value_pos, new_k_value_neg)

# Menghitung probabilitas menggunakan data
final_result = calculator.hitung_probabilitas(cosin)
result_compare = final_result.apply(calculator.compare_result, axis=1).apply(pd.Series)
result_compare = result_compare.rename(index=dict(zip(final_result.index, k_lama)), columns=col_map)
result_compare.index.name = 'K Value'

```

Gambar 4.13 *Sourcecode* Menjalankan Algoritma

Gambar 4.13 menampilkan *Sourcecode* untuk menjalankan *function* “*hitung_probabilitas*” dan *function* “*compare_result*”. *Function* “*compare_result*” dijalankan dengan *method* “*apply()*”. Hasil klasifikasi sentimen menggunakan metode *Improved K-Nearest Neighbor* disajikan pada Lampiran 2. Perhitungan menggunakan *K-Nearest Neighbor* tanpa *Improved* disajikan pada Lampiran 3

4.8 Evaluasi dan Validasi

Tahap validasi menggunakan metode *Cross Validation* untuk mengevaluasi kinerja algoritma *Improved K-Nearest Neighbor* secara objektif. Pada tahap ini, data dibagi menjadi subset data *training* dan subset data *testing*. Data tersebut dibagi menjadi K subset dengan ukuran yang sama, di mana nilai K adalah 10.

precision sebesar 84.92%, *recall* sebesar 83.00% dan *F1-measure* sebesar 83.83% yang memberikan wawasan tentang performa model dalam mengenali sentimen positif dan negatif. Hasil *confusion matrix* Improved K-Nearest Neighbor dapat dilihat Lampiran 2.

Evaluasi pada model menggunakan *10-fold validation* menghasilkan rata-rata *accuracy* sebesar 79.40%, *precision* sebesar 76.02%, *recall* sebesar 74.30% dan *F1-measure* sebesar 73.78%. *K-value* optimal pada perhitungan rata-rata *10-fold validation* adalah 29. Hasil *confusion matrix* Improved K-Nearest Neighbor dapat dilihat Lampiran 4.

4.10.2 Perbandingan Klasifikasi *Improved K-Nearest Neighbor* dan *K-Nearest Neighbor*

Klasifikasi menggunakan *Improved K-Nearest Neighbor* dan *K-Nearest Neighbor* tradisional memberikan hasil berbeda pada setiap metode. Perbandingan hasil *confusion matrix* dengan rasio data 70:30 ditampilkan pada Tabel 4.1.

Tabel 4.1 Perbandingan Hasil Pemetaan *Confusion Matrix*

<i>K-value</i>	<i>Improved K-Nearest Neighbor</i>				<i>K-Nearest Neighbor</i>				
	<i>True Positif</i>	<i>False Negati</i> <i>f</i>	<i>True Negati</i> <i>f</i>	<i>False Positif</i>	<i>K-value</i>	<i>True Positif</i>	<i>False Negati</i> <i>f</i>	<i>True Negati</i> <i>f</i>	<i>False Positif</i>
15	37	13	89	11	19	32	18	93	7
16	37	13	91	9	20	34	16	90	10
17	37	13	91	9	21	32	18	93	7
18	37	13	91	9	22	31	19	93	7
19	36	14	91	9	23	31	19	93	7
20	38	12	91	9	24	36	14	91	9
21	37	13	89	11	25	32	18	91	9

Berdasarkan hasil perbandingan pemetaan *confusion matrix* dengan *k-value optimal*, menunjukkan bahwa *K-Nearest Neighbor* dominan mengklasifikasikan kelas negatif sedangkan *Improved K-Nearest Neighbor* cenderung seimbang. Hal ini disebabkan Nilai *k-value* tetap untuk seluruh kelas pada metode *K-Nearest Neighbor* tradisional sehingga cenderung mengikuti kelas mayoritas dominan. Kelas dominan pada klasifikasi ini adalah kelas negatif dengan jumlah 337 *tweets* sedangkan kelas positif 163 *tweets*.

Tabel 4.2 Perbandingan Hasil Evaluasi *Confusion Matrix*

<i>Improved K-Nearest Neighbor</i>					<i>K-Nearest Neighbor</i>				
<i>K</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-measure</i>	<i>K</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-measure</i>
15	84.00%	82.17%	81.50%	81.81%	19	83.33%	82.92%	78.50%	80.03%
16	85.33%	83.97%	82.50%	83.15%	20	82.67%	81.09%	79.00%	79.86%
17	85.33%	83.97%	82.50%	83.15%	21	83.33%	82.92%	78.50%	80.03%
18	85.33%	83.97%	82.50%	83.15%	22	82.67%	81.09%	79.00%	79.86%
19	84.67%	83.33%	81.50%	82.28%	23	82.67%	82.31%	77.50%	79.10%
20	84.67%	83.33%	81.50%	82.28%	24	84.67%	83.33%	81.50%	82.28%
21	86.00%	84.60%	83.50%	84.00%	25	82.00%	80.77%	77.50%	78.71%

Tabel 4.2 menunjukkan hasil evaluasi tiap metode. *K-value* optimal pada *Improved K-Nearest Neighbor* adalah 21 dengan nilai *accuracy* sebesar 86%, *precision* sebesar 84,67%, *recall* sebesar 83,5% dan *f1-measure* sebesar 84%. *K-value* optimal pada *K-Nearest Neighbor* adalah 24 dengan nilai *accuracy* sebesar 84,67%, *precision* sebesar 83,33%, *recall* sebesar 81,50% dan *f1-measure* sebesar 82,28%. Hasil Perbandingan metode *Improved K-Nearest Neighbor* dan *K-Nearest Neighbor* dapat dilihat pada Lampiran 2 dan Lampiran 3.

Berdasarkan uraian diatas menunjukkan *Improved K-Nearest Neighbor* lebih unggul dalam mengenali pola yang lebih kompleks dan tidak bergantung pada mayoritas kelas dominan, sehingga dapat digunakan untuk klasifikasi sentimen.

4.10.3 Rekomendasi Hasil Visualisasi

Hasil analisis sentimen pada 500 data menunjukkan bahwa kelas negatif lebih dominan dengan persentase 67,4% dan kelas positif sebesar 32,6%. Hasil sentimen setiap kelas kemudian dijadikan *word cloud* untuk menampilkan frekuensi kata terbanyak. Tabel 4.3 menampilkan hasil *tweets* berdasarkan unsur kata *word cloud top 3* sentimen negatif.

Tabel 4.3 Data *Tweet* Dengan Unsur Kata Negatif

Data Ke-	<i>Tweet</i>	<i>Keyword</i>
----------	--------------	----------------

92	@BPJSKesehatanRI Yasalam.. kan permasalahannya dari awal begitu min ya Allah ya Rabb :(itu kendalanya gabisa diubah. Appsnya error mulu. Dahlah min. Capek	
369	Helppp nangis bgt pengen crita @BPJSKesehatanRI. Pertama kali ubah faskes 1 lwt JKN mobile. Sblumnya belum 1 bulan aku udh ke faskes 1 yg baru (tapi ditolak). Oke di suruh nunggu 1 bulan. Sebulan setelah berlalu aku ke faskes baru itu, tapi aku daftar lewat online di Apk JKN-	Ubah
224	@BPJSKesehatanRI mau update NIK melalui pandawa udah mau 2 minggu nyobain pada saat upload KK selalu eror, datang langsung ke BPJS tetap diarahkan melalui Pandawa, tlp 165 tetap di suruh melalui pandawa, email juga ga dibalas, jadi harus gimana min?	Update
84	@BPJSKesehatanRI ini benar benar kendala atau akal akaln saja ya masalahnya sdr saya kok update no dpt jwbn dm yg sama	
217	@BPJSKesehatanRI Kami mau merubah status keanggotaan menjadi BPJS Mandiri, sudah lewat Pandawa, tetap saja gagal. Bahkan ke kantor cabang juga ditolak, disuruh lewat Apps. Tapi lewat Apps juga ngga bisa.	Gagal
227	@BPJSKesehatanRI min ini kenapa gagal terus ya mau ngerubah faskes dokter gigi	

Berdasarkan kumpulan *tweet top 3 word cloud* sentimen negatif pada Tabel

4.1, topik yang sering dibicarakan adalah sebagai berikut :

1. Masalah teknis dan aplikasi BPJS Kesehatan: pengguna sering mengalami kendala teknis dan error saat mengakses platform BPJS Kesehatan.
2. Keterbatasan respon dan bantuan: pengguna mengeluhkan respon dan jawaban yang diberikan oleh pihak BPJS saat mengalami kendala sehingga pengguna merasa kebingungan terkait langkah yang harus diambil berikutnya.
3. Kendala perubahan fasilitas kesehatan (faskes): pengguna mengeluhkan proses yang rumit dan membutuhkan waktu yang lama dalam perubahan faskes atau memperbaharui data peserta.

Berdasarkan topik sentimen negatif, BPJS dapat mempertimbangkan untuk meningkatkan kualitas pelayanan terutama dalam hal teknis aplikasi dan proses perubahan faskes serta memberikan respon dan bantuan yang lebih baik untuk meningkatkan kepuasan pengguna.

Sedangkan untuk kelas positif hasil *tweet* berdasarkan unsur kata *word cloud top 3* sentimen positif ditampilkan pada Tabel 4.4

Tabel 4.4 Data *Tweet* Dengan Unsur Kata Positif

Data Ke-	<i>Tweet</i>	<i>Keyword</i>
149	Finally bandage free! Setelah 1,5 bulan pemulihan luka & kulit abis operasi, dokter bedah memperbolehkan u/ lepas perban, tinggal oles salep aja sampe abis. Alhamdulillah, makasih dokter dan segenap nakes serta admin yg bertugas di JMC Hospital dan terima kasih BPJS Kesehatan!	<i>Free</i>

100	@Sofianpalupi pengalaman mertua saya harus masuk ICU 1 minggu. Diitung2 bayar hampir 100 jutaan. Ada BPJS, free. Tks @BPJSKesehatanRI . Semoga ke depan semakin baik, menjadi best practice welfare state policy di Indonesia dan bisa menghilangkan hal-hal	
77	@ikiule @BPJSKesehatanRI Iyo mbak, lahirane jago cesar nek g bpjs iso kenek 30jtan dewe, untung gawe bpjs wkwk dadi free	
367	Byk ngobrol ama mereka. Istrinya seminggu 2x msti cuci darah & itu dilakukan udah 10 thn. Sebulan bs 8x cuci darah. untung ada @BPJSKesehatanRI jd free. Tp mrka msti beli vitamin yg sebulan hbs 5-6 jta. Semangat ya Mas dan Mbak. Semoga jualannya laris manis	Untung
446	Kalau saya pribadi MRI full Di cover BPJS , saya jga selama ini cek ini itu pake BPJS. Kalau pakai uang pribadi jujur, gk sanggup bisa babak belur, bahkan jual motor pun blm tentu cukup, karena nya saya sangat terbantu sekali dengan manfaat dari @BPJSKesehatanRI Terima Kasih BPJS	
393	@kabocha_neko Asuransi tuh yg jelas ada limitnya sih, dan biasanya ga cover full, paling cuma nanggung 80-90%nya aja. Emang ga seribet bpjs sih. Tp bpjs tuh bagus kalo misal lu sakit serius. Kalo sakit ringan mah emang mendingan asuransi kesehatan	Full

Berdasarkan kumpulan *tweet top 3 word cloud* sentimen positif pada Tabel 4.2, topik yang sering dibicarakan adalah sebagai berikut :

1. Manfaat pengobatan gratis: mayoritas tweet berbicara tentang manfaat berobat gratis untuk perawatan medis tertentu seperti rawat inap ICU, tindakan MRI atau tindakan medis yang kompleks.
2. Pengaruh terhadap keuangan pribadi: pengguna merasa terbantu oleh BPJS Kesehatan dalam melindungi keuangan pribadi pengguna dari beban biaya perawatan medis yang besar dan tidak terduga.
3. Manfaat biaya sepenuhnya (*full*): pengguna merasa terbantu karena BPJS Kesehatan menanggung biaya perawatan medis secara penuh tanpa ada potongan biaya dibandingkan dengan beberapa jenis asuransi kesehatan swasta.

Berdasarkan topik sentimen positif, menunjukkan bahwa layanan BPJS Kesehatan telah memberikan dampak yang positif dalam memfasilitasi akses kesehatan. BPJS Kesehatan dapat menggunakan apresiasi dari pengguna untuk meningkatkan kesadaran dan partisipasi masyarakat dalam menggunakan BPJS Kesehatan.

BAB V

KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan hasil analisis dan pengujian tugas akhir ini, beserta saran-saran untuk pengembangan penelitian serupa di masa mendatang.

5.1 Kesimpulan

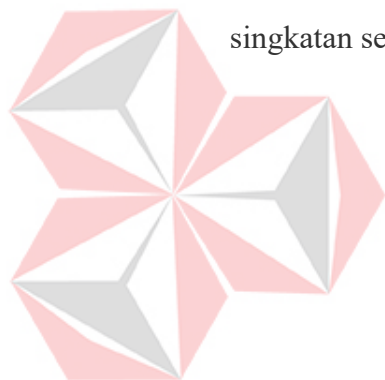
Kesimpulan dari analisis sentimen opini publik terhadap pelayanan BPJS Kesehatan menggunakan metode *Improved K-Nearest Neighbor* adalah sebagai berikut:

1. Hasil Analisis sentimen pada 500 data menghasilkan kelas positif sebesar 32,6% dan kelas negatif sebesar 67,4% yang menunjukkan BPJS Kesehatan perlu melakukan peningkatan pelayanan.
2. Hasil klasifikasi menggunakan *Improved K-Nearest Neighbor* mendapatkan akurasi sebesar 86% lebih tinggi dibandingkan dengan *K-Nearest Neighbor* tanpa *improved* sebesar 84,67% yang menunjukkan *Improved K-Nearest Neighbor* memberikan peningkatan akurasi dan lebih unggul dalam mengenali pola yang lebih kompleks serta tidak bergantung pada mayoritas kelas dominan.
3. Kata yang sering muncul dalam sentimen negatif adalah “Ubah”, “Update”, “Manual” dan “Gagal” dengan topik yang dibicarakan seputar kendala dalam teknis aplikasi, proses perubahan faskes, dan mendapatkan bantuan yang efektif dari pihak BPJS Kesehatan. Oleh karena itu, layanan BPJS Kesehatan mungkin perlu melakukan perbaikan dalam hal komunikasi, responsifitas, dan kemudahan akses teknis agar dapat memenuhi harapan penggunaanya. Sedangkan untuk sentimen positif adalah "Free", "Untung" dan "Full" dengan dengan topik yang dibicarakan seputar pengobatan gratis dan perlindungan keuangan secara penuh dari BPJS Kesehatan. Pengalaman positif ini menunjukkan bahwa layanan BPJS Kesehatan telah memberikan dampak yang positif dan terus ditingkatkan untuk meningkatkan partisipasi masyarakat dalam menggunakan BPJS Kesehatan.

5.2 Saran

Saran yang diberikan untuk pengembangan lebih lanjut adalah sebagai berikut :

1. Analisis sentimen dapat ditingkatkan akurasinya untuk meningkatkan keyakinan pengguna pada hasil prediksi yang disajikan oleh model atau sistem. Peningkatan akurasi dapat dilakukan dengan menggunakan sistem pengecekan sinonim kata dan pengecekan kemiripan berdasarkan makna. Frekuensi kemunculan kata akan dihubungkan dengan makna yang lebih tepat, memungkinkan hasil yang lebih optimal dalam mengukur kemiripan dan sentimen. Selain itu, peningkatan akurasi dapat dilakukan dengan penambahan jumlah data sehingga model memiliki banyak variasi rujukan kata.
2. Data dari twitter memiliki banyak gaya bahasa tidak baku yang tidak dapat diubah yang berpengaruh pada waktu proses pengolahan data. Kata-kata yang tidak bermakna tapi menggunakan singkatan tidak hilang pada proses *filtering* sehingga pada proses *stemming* memerlukan waktu yang lebih lama. Oleh karena itu diperlukan kamus khusus untuk mengidentifikasi kata-kata gaul atau singkatan sehingga dapat diolah pada *preprocessin* data.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Alwasi'a, A. (2020). *ANALISIS SENTIMEN PADA REVIEW APLIKASI BERITA ONLINE MENGGUNAKAN METODE MAXIMUM ENTROPY (Studi Kasus: Review Detikcom Pada Google Play 2019)*.
- Annur, C. M. (2022). Pengguna Twitter Indonesia Masuk Daftar Terbanyak di Dunia, Urutan Berapa? Retrieved May 4, 2022, from katadata website: <https://databoks.katadata.co.id/datapublish/2022/03/23/pengguna-twitter-indonesia-masuk-daftar-terbanyak-di-dunia-urutan-berapa>
- Athira Luqyana, W., Cholissodin, I., & Perdana, R. S. (2018). Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(11), 4704–4713. Retrieved from <http://j-ptiik.ub.ac.id>
- Azizah, K. N. (2021). 7 Tahun BPJS Kesehatan, Peserta Masih Keluhkan Antrean Berjam-jam. Retrieved January 14, 2022, from health.detik.com website: <https://health.detik.com/berita-detikhealth/d-5486380/7-tahun-bpjs-kesehatan-peserta-masih-keluhkan-antrean-berjam-jam>
- Baoli, L., Shiwen, Y., & Qin, L. (2003). An Improved k -Nearest Neighbor Algorithm. *Proceedings of the 20th International Conference on Computer Processing of Oriental Languages*, (July).
- BPJS Kesehatan. (n.d.). No Title. Retrieved April 10, 2021, from <https://www.bpjs-kesehatan.go.id/#/profil>
- Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2019). *Socio-Affective Computing 5 A Practical Guide to Sentiment Analysis*. Retrieved from <http://www.springer.com/series/13199>
- CNN Indonesia. (2021). Kisah Pengidap Tumor Asal Sulsel Lancar Operasi Berkat JKN-KIS. Retrieved January 15, 2022, from [cnnindonesia website: https://www.cnnindonesia.com/ekonomi/20211109163719-83-718750/kisah-pengidap-tumor-asal-sulsel-lancar-operasi-berkat-jkn-kis](https://www.cnnindonesia.com/ekonomi/20211109163719-83-718750/kisah-pengidap-tumor-asal-sulsel-lancar-operasi-berkat-jkn-kis)
- D'souza, S. R., & Sonawane, K. (2019). Sentiment analysis based on multiple reviews by using machine learning approaches. *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, (Iccmc), 188–193. <https://doi.org/10.1109/ICCMC.2019.8819813>
- Dwi Antonio, V., Efendi, S., & Mawengkang, H. (2022). Sentiment analysis for covid-19 in Indonesia on Twitter with TF-IDF featured extraction and stochastic gradient descent. *Int. J. Nonlinear Anal. Appl*, 13(1), 2008–6822. Retrieved from <http://dx.doi.org/10.22075/ijnaa.2021.5735>
- Iswarya, P., & Radha, V. (2015). Ensemble learning approach in improved K Nearest Neighbor algorithm for Text categorization. *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*. <https://doi.org/10.1109/ICIIECS.2015.7193250>
- Kusnandar, V. B. (2022). Peserta JKN Capai 241,79 Juta Jiwa per Juni 2022. Retrieved December 3, 2022, from katadata website: <https://databoks.katadata.co.id/datapublish/2022/07/05/peserta-jkn-capai-24179-juta-jiwa-per-juni-2022>
- Maya, C. (2018). Rujukan Online Banyak Tuai Keluhan, Ini tanggapan BPJS Kesehatan. Retrieved January 14, 2022, from balipost website:

<https://www.balipost.com/news/2018/11/07/60604/Rujukan-Online-Banyak-Tuai-Keluhan,...html>

- Mee, A., Homapour, E., Chiclana, F., & Engel, O. (2021). Sentiment analysis using TF-IDF weighting of UK MPs' tweets on Brexit[Formula presented]. *Knowledge-Based Systems*, 228, 107238. <https://doi.org/10.1016/j.knosys.2021.107238>
- Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics , Scientific and Technical Applications. *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 140–145.
- Nti, I. K. (2021). *Performance of Machine Learning Algorithms with Different K Values in K-fold Cross- Validation*. (December), 61–71. <https://doi.org/10.5815/ijitcs.2021.06.05>
- Rui-Jun, Y., Dan-Feng, D., & Feng, Y. (2019). Application of improved KNN algorithm in air quality assessment. *ACM International Conference Proceeding Series*, 108–112. <https://doi.org/10.1145/3341069.3342976>
- Russell, M. a. (2014). *Mining the Social Web (sampler)*.
- Talib, R., Kashif, M., Ayesha, S., & Fatima, F. (2016). Text Mining: Techniques, Applications and Issues. *International Journal of Advanced Computer Science and Applications*, 7(11), 414–418. <https://doi.org/10.14569/ijacsa.2016.071153>
- Zakia Nathania, D., & Abdurrachma Bachtiar, F. (2018). *Klasifikasi Spam Pada Twitter Menggunakan Metode Improved K-Nearest Neighbor*. 2(10), 3948–3956. Retrieved from <http://j-ptiik.ub.ac.id>



UNIVERSITAS
Dinamika