



**ANALISIS SENTIMEN PENILAIAN APLIKASI WONDR *BY* BNI
MENGUNAKAN *MODIFIED K-NEAREST NEIGHBOR (MK-NN)***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Statistika**

Oleh

Arifatul Fathinah Essa

B2A223004

PROGRAM STUDI STATISTIKA

FAKULTAS SAINS DAN TEKNOLOGI PERTANIAN

UNIVERSITAS MUHAMMADIYAH SEMARANG

2024

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Segala puji dan Syukur penulis ucapkan kehadiran Allah Subhanhuwata'ala dan mengharapkan ridho yang telah melimpahkan Rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis Sentimen Penilaian Aplikasi wondr by BNI Menggunakan *Modified K-Nearest Neighbor (MK-NN)*”. Skripsi ini disusun sebagai salah satu persyaratan meraih gelar Sarjana Statistika pada Program Studi S1 Statistika Universitas Muhammadiyah Semarang. Shalawat dan salam disampaikan kepada junjungan alam Nabi Muhammad Shallallahu'Alaihi Wasallam, mudah-mudahan kita semua mendapatkan syafaat-Nya di yaumul akhir nanti, Aamiin

Penulis menyadari sepenuhnya, bahwa dalam penyelesaian proposal skripsi ini tidak terlepas dari bantuan dan bimbingan berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan terimakasih dan penghargaan yang setinggi-tingginya kepada:

1. Dr. Nurhidajah, M.Si., selaku Dekan Fakultas Sains dan Teknologi Pertanian Universitas Muhammadiyah Semarang.
2. Bapak M.Al Haris, M.Si., selaku Ketua Program Studi S1 Statistika Fakultas Sains dan Teknologi Pertanian Universitas Muhammadiyah Semarang yang telah memberikan motivasi dan inspirasinya untuk kami semua.
3. Ibu Tiani Wahyu Utami, M.Si., dan Ibu Prizka Rismawati Arum, S.Si., M.Si selaku dosen pembimbing 1 dan Dosen pembimbing 2 yang telah meluangkan banyak waktu untuk memberikan bimbingan, pengarahan, dan saran-saran dalam penyusunan skripsi ini
4. Bapak dan Ibu Dosen, Staf pengajar, dan Karyawan program Studi S1 Statistika Fakultas Sains dan Teknologi Pertanian Universitas Muhammadiyah Semarang yang telah memberikan bekal ilmu yang bermanfaat kepada penulis sehingga penulis dapat menyusun skripsi ini.

5. Kedua Orang tua yang selalu memberikan semangat dan doa demi kelancaran selama penelitian, sehingga skripsi ini dapat terselesaikan.
6. Semua pihak yang tidak dapat disebutkan satu persatu, yang telah menemani dan memberikan semangat sehingga dapat terselesaikannya skripsi ini.

Penulis menyadari akan segala keterbatasan dan kekurangan dari isi maupun tulisan ini. Oleh karena itu, kritik dan saran yang bersifat membangun dari semua pihak masih diterima dengan senang hati. Semoga hasil penelitian ini dapat memberikan manfaat dan kontribusi bagi pengembangan pembelajaran di bidang statistika.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Semarang, 2024
Penulis

Arifatul Fathinah Essa

ABSTRAK

Essa, Arifatul Fathinah, 2024, Analisis Sentimen Penilaian Aplikasi Wondr by BNI Menggunakan *Modified K-Nearest Neighbor (MK-NN)*, Skripsi, Program Studi Statistika, Universitas Muhammadiyah Semarang, Pembimbing I: Tiani Wahyu Utami, M.Si., Pembimbing II: Prizka Rismawati Arum, S.Si, M.Stat.

Perkembangan teknologi digital yang pesat telah mengubah berbagai aspek kehidupan, termasuk bidang keuangan melalui financial technology (fintech). PT Bank Negara Indonesia (BNI) telah meluncurkan aplikasi mobile banking kedua bernama *wondr by BNI*. Meskipun awalnya mendapat respon positif, rating aplikasi ini di Google Play Store menurun dari 4,7 menjadi 3,4 dalam dua bulan. Penelitian ini bertujuan untuk mengklasifikasikan sentimen pengguna aplikasi *wondr by BNI* menggunakan metode *Modified K-Nearest Neighbor (MK-NN)*. Data yang digunakan adalah komentar pengguna di Google Play Store yang di-scrape dalam periode 5 Juli hingga 2 September 2024. Dengan teknik *text mining*, analisis sentimen dilakukan menggunakan pembobotan kata TF-IDF dan metode MK-NN. Kata yang paling sering dibahas dalam sentimen positif adalah "aplikasi", "wondr", dan "transaksi", sedangkan dalam sentimen negatif adalah "masuk", "verifikasi wajah", dan "transaksi". Hasil penelitian menunjukkan akurasi klasifikasi sebesar 87,24%, dengan nilai *precision* 87,55% dan *recall* 95,84%. Hasil penelitian ini dapat membantu BNI dalam meningkatkan kualitas aplikasi dan meningkatkan kepuasan nasabah.

Kata kunci: *Cosine Distance*, Klasifikasi, *MK-NN*, *text mining*, *wondr by BNI*.

ABSTRACT

Essa, Arifatul Fathinah, 2024, Sentiment Analysis of Wondr by BNI Application Assessment Using Modified K-Nearest Neighbor (MK-NN), Thesis, Statistics Study Program, Universitas Muhammadiyah Semarang, Advisor I: Tiani Wahyu Utami, M.Si., Advisor II: Prizka Rismawati Arum, S.Si, M.Stat.

The rapid development of digital technology has changed various aspects of life, including finance through financial technology (fintech). PT Bank Negara Indonesia (BNI) has launched a second mobile banking application called wondr by BNI. Despite an initial positive response, the app's rating on the Google Play Store dropped from 4.7 to 3.4 in two months. This research aims to classify the sentiment of wondr by BNI app users using the Modified K-Nearest Neighbor (MK-NN) method. The data used are user comments on Google Play Store that were scraped in the period from July 5 to September 2, 2024. Using text mining techniques, sentiment analysis was performed using TF-IDF word weighting and MK-NN method. The most frequently discussed words in positive sentiment are “application”, “wondr”, and “transaction”, while in negative sentiment are “login”, “face verification”, and “transaction”. The results showed a classification accuracy of 87.24%, with a precision value of 87.55% and a recall of 95.84%. The results of this study can assist BNI in improving the quality of its application and enhancing customer satisfaction.

Keywords: *Classification, Cosine Distance, MK-NN, text mining, wondr by BNI.*

DAFTAR ISI

KATA PENGANTAR.....	i
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Batasan Penelitian	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Tinjauan Statistik.....	7
2.1.1 <i>Data Mining</i>	7
2.1.2 <i>Text mining</i>	8
2.1.3 Analisis sentimen	9
2.1.4 Pelabelan data.....	10
2.1.5 Pembobotan kata	10
2.1.6 <i>TF-IDF Weighting</i>	11
2.1.7 <i>Modified K-Nearest Neighbors (MK-NN)</i>	12
a. <i>Cosine Similarity</i>	13
b. Validitas.....	13
c. <i>Cosine Distance</i>	14
d. <i>Weight Voting</i>	15
2.2 Tinjauan Non Statistik.....	18
2.2.1 Google Play	18

2.2.2	Wondr by BNI.....	18
BAB III METODOLOGI PENELITIAN		20
3.1	Sumber Data	20
3.2	Variabel Penelitian	20
3.3	Langkah Penelitian	20
3.4	Diagram Alir Penelitian.....	22
BAB IV HASIL DAN PEMBAHASAN.....		23
4.1	<i>Scraping data</i>	23
4.2	<i>Pre-processing data</i>	24
4.3	Pelabelan data.....	26
4.4	<i>wordcloud</i>	27
4.5	Pembobotan kata	29
4.6	Klasifikasi <i>Modified K-Nearest Neighbor (MK-NN)</i>	31
a.	<i>Cosine similarity</i>	32
b.	Validitas.....	32
c.	<i>Cosine Distance</i>	34
d.	<i>Weight Voting</i>	34
e.	Evaluasi model	36
BAB V KESIMPULAN DAN SARAN		39
5.1	Kesimpulan.....	39
5.2	Saran	40
DAFTAR PUSTAKA		41
LAMPIRAN.....		45

DAFTAR TABEL

Tabel	Halaman
2. 1 Perbedaan K-NN dan MK-NN	12
2. 2 <i>Confussion matrix</i>	16
3. 1 Atribut Penelitian	20
4. 1. Hasil <i>scraping</i> dari Google Play Store	23
4. 2 Hasil <i>cleaning data</i>	24
4. 3 Hasil <i>case folding</i>	24
4. 4 Hasil <i>tokenizing</i>	25
4. 5 Hasil <i>stopword removal</i>	25
4. 6 Hasil <i>stemming</i>	26
4. 7 Hasil pelabelan data	26
4.8. 5 komentar pertama	29
4. 9 Perhitungan TF dan IDF	29
4. 10 Hasil pembobotan TF-IDF 5 data pertama	30
4. 11 Pembagian data latih dan data uji	31
4. 12. 10 komentar pertama	31
4. 13 Hasil <i>cosine similarity</i>	32
4. 14. Hasil validitas antar data latih	33
4. 15 Hasil <i>cosine distance</i> data latih dan data uji	34
4. 16 Hasil <i>weight voting</i>	35
4. 17 Hasil akurasi, <i>precision</i> , dan <i>recall</i>	36
4. 18 Hasil <i>confusion matrix</i>	36

DAFTAR GAMBAR

Gambar	Halaman
1. 1 Perbandingan penilaian Wondr <i>by</i> BNI dan BNI Mobile Banking pada 2 September 2024	3
2. 1 bentuk <i>wordcloud</i>	17
2. 2 Logo Google Play	18
2. 3 Logo Wondr <i>by</i> BNI	18
3. 1 Diagram Alir Penelitian	22
4. 1 Diagram batang pelabelan sentimen	27
4. 2 <i>Wordcloud</i> sentimen positif	28
4. 3 <i>Wordcloud</i> sentimen negatif	28

BAB I

PENDAHULUAN

1.1 Latar Belakang

Zaman mengalami perubahan yang cepat dengan didukung inovasi teknologi yang membuat hampir semua aspek berjalan secara digital. digitalisasi hampir meliputi seluruh hal, seperti: bersosialisasi, berbelanja, membaca berita, bahkan pembuatan surat penting atau rekening bank. Perangkat elektronik juga terus berevolusi sehingga tercipta perangkat telepon seluler menggunakan fitur *fingerprint*, *face ID*, *Near Field Communication* (NFC), dan fitur futuristik lainnya yang menjadikan perkembangan pesat di bidang digital (Fitria et al., 2021).

Kemajuan di era digital juga diiringi dengan perkembangan internet yang sudah merambah ke seluruh dunia. Pengguna internet mengetahui segala hal yang ada di belahan dunia manapun melalui berita yang dipublikasi secara *online*, dan juga dapat memberikan bantuan berupa uang elektronik melalui perangkat seluler dengan bantuan internet. Penggunaan internet di Indonesia terus mengalami peningkatan setiap tahunnya, sesuai yang disampaikan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) (2024), Jumlah pengguna Internet di Indonesia pada awal tahun 2024 sebesar 221.563.479 jiwa dari total populasi penduduk Indonesia pada tahun 2023 sebanyak 278.696.200 jiwa. Dapat disimpulkan bahwa lebih dari 75% penduduk Indonesia sudah menggunakan internet.

Melalui perkembangan teknologi dan kemajuan era digitalisasi, bidang keuangan juga memberikan banyak kemudahan dalam berbagai transaksi. Perkembangan teknologi pada ranah keuangan atau saat ini dinamakan *financial technology* (*fintech*). Seperti transfer uang, pengecekan saldo, pembayaran tagihan listrik, dan lainnya bisa dilakukan via *mobile banking*. *Fintech* berperan penting dalam memaksimalkan penggunaan teknologi dalam mempercepat layanan keuangan yang tersedia termasuk pinjaman dana hingga pengelolaan aset (Safitri & Miftah Andriansyah, 2020)

Saat ini, bank berlomba-lomba dalam meluncurkan aplikasi *mobile banking* seperti yang tersedia di laman Google Play Store atau App Store. Seluruh bank terus berusaha dalam menciptakan perangkat terbaik untuk para konsumen maupun calon konsumen. Dikutip dari laman berita itrade.cgsi.co.id pada 5 Juli 2024 dalam ajang *Infobank 21st Banking Service Excellence Awards* (BSEA), kategori saluran digital terbaik, *phone banking* terbaik, dan *E-banking* terbaik dimenangkan oleh PT Bank Negara Indonesia (BNI) Tbk. Melalui penghargaan dibidang digital ini, BNI membuktikan mampu untuk bersaing di era *fintech*.

Menurut Fitria, dkk (2021) dalam penelitian pengaruh penggunaan *mobile banking* BNI terhadap kepuasan nasabah menunjukkan hasil bahwa para nasabah dan juga menjadi *user* dari aplikasi *mobile banking* BNI merasa puas terhadap fitur yang dapat digunakan dalam *mobile banking*. Selain layanan fitur pada aplikasi, *mobile banking* juga menjamin keamanan dan kemudahan bagi pengguna berusia muda maupun berusia senja. Penelitian tersebut juga disokong dengan penelitian yang dilakukan oleh Amalia & Hastriana (2022) pada 100 nasabah yang menjadi responden, diperoleh hasil bahwa keamanan dan kemudahan penggunaan *mobile banking* menjadi faktor kepuasan nasabah.

BNI meluncurkan *mobile banking* kedua yang diberi nama *wondr by BNI*. Aplikasi ini diluncurkan pada tanggal 5 Juli 2024. Pada tanggal 17 Juli 2024 rating yang diperoleh sebesar 4,7 dari 5 dan rating per tanggal 2 September 2024 mengalami penurunan menjadi 3,4 dari 5 di Google Play Store. Penurunan rating ini tentu dipengaruhi dengan jumlah *user* dan pengalaman *user* dalam menggunakan aplikasi. Dibandingkan dengan aplikasi *mobile banking* yang diluncurkan pertama yaitu BNI Mobile Banking yang memiliki penilaian dengan bintang 4,6 di Google Play, *wondr by BNI* memiliki penilaian lebih rendah dibandingkan aplikasi *mobile banking* pertama. Beberapa *user* berkomentar bahwa aplikasi *wondr by BNI* memiliki fitur yang kurang lengkap dibandingkan dengan aplikasi BNI Mobile Banking.

<p>wondr by BNI</p> <p>PT Bank Negara Indonesia (Persero) Tbk.</p> <p>wondr by BNI: analisa, transaksi, dan rencanakan keuanganmu dengan mudah</p> <p>  3,4★ 7,89 rb ulasan </p> <p> 1 jt+ Hasil download </p> <p>  Rating 3+ </p>	<p>BNI Mobile Banking</p> <p>PT Bank Negara Indonesia (Persero) Tbk.</p> <p>Aplikasi Resmi BNI Mobile Banking</p> <p>  4,6★ 1,05 jt ulasan </p> <p> 10 jt+ Hasil download </p> <p>  Rating 3+ </p>
--	--

Gambar 1. 1 Perbandingan Penilaian Wondr by BNI dan BNI Mobile Banking pada 2 September 2024

Berdasarkan Google Play, jumlah penilaian wondr by BNI pada tanggal 9 Juli 2024 berjumlah sekitar 1.600 penilaian dan per tanggal 2 September 2024 sudah mencapai sekitar 7.890 penilaian. Penilaian pada aplikasi dapat mempengaruhi keputusan seseorang untuk menjadi *user* aplikasi dan mempengaruhi citra brand. Penilaian yang diberikan oleh *user* bisa berupa suatu apresiasi atau juga berbentuk evaluasi. Seiring berjalannya waktu, penilaian pada aplikasi terus bertambah sehingga membuat pihak developer memerlukan waktu lebih lama jika harus memperhatikan setiap komentar dari setiap *user*, sehingga diperlukan suatu ilmu untuk mengetahui apa yang menjadi topik bahasan oleh *user* secara otomatis. Ilmu yang dikembangkan untuk penggalian dokumen teks ini disebut dengan *text mining* (Arizal, 2023).

Text mining adalah sebuah teknik untuk menggali informasi dari sebuah text dengan menemukan pola dari sekumpulan data. Analisis sentimen merupakan bagian dari *text mining* yang melakukan proses pemahaman, penarikan, dan pengolahan data dalam bentuk teks untuk mendapatkan sentimen yang terkandung pada tiap opini (Hidayat T.F.T, Garno, G, 2021). Klasifikasi dapat dilakukan dengan banyak cara untuk analisis sentimen diantaranya, *decision tree*, *naive bayes classifier*, *support vector machine (SVM)* dan *k-nearest neighbor (K-NN)*. Salah satu cara klasifikasi sederhana yang sering digunakan adalah *k-nearest neighbor* dengan cara kerjanya mengklasifikasikan berdasarkan jarak (Ikhsan & Adytia, 2024).

K-Nearest Neighbor (K-NN) dimodifikasi dengan penambahan validasi pada data latih dan pembobotan terbesar dari tetangga terdekatnya dengan nama *Modified K-Nearest Neighbor (MK-NN)*. Perhitungan validasi dan pembobotan terbesar pada *MK-NN* mampu mengatasi kelemahan klasifikasi berdasarkan jarak

terdekat pada metode *K-NN*. Tujuan dari penambahan prosedur ini adalah untuk meningkatkan akurasi pada klasifikasi dengan *K-Nearest Neighbor* (Amelia, 2022).

Penelitian implementasi metode *MK-NN* dalam analisis sentimen yang dilakukan oleh Sofiah, dkk (2023) tentang kenaikan harga bahan bakar minyak di Twitter menunjukkan hasil akurasi sebesar 83,33%. Analisis sentimen dengan penerapan *MKNN* juga dilakukan oleh Paramitha, dkk (2020) tentang ulasan pengguna MRT Jakarta, dengan nilai *K* yang digunakan adalah 3 dan memperoleh akurasi sebesar 86%. Implementasi *MKNN* juga dilakukan oleh Halim, dkk (2023) dengan topik Prabowo Subianto bakal calon presiden 2024 pada aplikasi Twitter. Diperoleh nilai akurasi sebesar 93,3% pada nilai *K*=3.

Dari uraian latar belakang masalah dan penelitian sebelumnya, dapat diterapkan teknik klasifikasi penilaian dari *user wondr by BNI* dapat menggunakan metode algoritma *Modified K-Nearest Neighbor (MK-NN)*. Menurut Rizki (2022), Algoritma *Modified K-Nearest Neighbor (MK-NN)* memiliki tingkat akurasi yang menjanjikan dalam klasifikasi teks dengan menambahkan pembobotan menggunakan *weight voting* sehingga diperoleh nilai akurasi yang baik dibandingkan dengan *K-Nearest Neighbors (KNN)* saja. Berdasarkan alasan tersebut, maka *Modified K-Nearest Neighbors (MK-NN)* digunakan untuk mengklasifikasikan penilaian pada aplikasi *wondr by BNI*.

1.2 Rumusan Masalah

Berlandaskan uraian latar belakang masalah diatas, maka terbentuk rumusan masalah sebagai berikut:

1. Bagaimana gambaran umum data penilaian aplikasi *wondr by BNI* berdasarkan Google Play Store?
2. Bagaimana akurasi *Modified K-Nearest Neighbors (MK-NN)* dalam mengklasifikasikan teks mengenai penilaian aplikasi *wondr by BNI* berdasarkan Google Play Store?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui gambaran umum data penilaian aplikasi wondr by BNI berdasarkan Google Play Store.
2. Mengetahui akurasi *Modified K-Nearest Neighbors* (MK-NN) dalam mengklasifikasikan teks mengenai penilaian aplikasi wondr by BNI berdasarkan Google Play Store.

1.4 Manfaat Penelitian

1. Manfaat Teoritis

Bagi Pembaca:

- Memperluas pemahaman tentang penerapan statistika dalam pengklasifikasian *text mining* menggunakan *Modified K-Nearest Neighbors* (MK-NN)
- Memungkinkan pembaca untuk memperdalam pengetahuan mereka tentang *text mining* menggunakan *Modified K-Nearest Neighbors* (MK-NN)

Bagi Universitas Muhammadiyah Semarang:

- Menjadi referensi yang berguna dalam mengembangkan pengetahuan tentang aplikasi statistika dalam konteks klasifikasi *text mining*.

2. Manfaat Praktis:

Bagi *developer*:

- Memberikan panduan praktis dalam mengetahui pendapat pengguna aplikasi baik bentuk pendapat positif dan negatif, sehingga dapat digunakan sebagai referensi untuk menjaga kualitas aplikasi.
- Membantu investor dalam mengevaluasi kendala atau hambatan yang terjadi dalam aplikasi, sehingga dapat melakukan pembenahan aplikasi untuk menjadi lebih baik.

1.5 Batasan Penelitian

Batasan penelitian yang digunakan dalam penelitian ini adalah:

1. Data yang digunakan merupakan data penilaian *user* aplikasi *wondr by BNI* yang terdapat di Google Play Store.
2. Analisis data akan dilakukan dengan bantuan perangkat lunak *Python*.
3. Jarak yang digunakan adalah *cosine distance*.
4. Data komentar yang digunakan dari 5 Juli sampai 2 september 2024.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Statistik

2.1.1 *Data Mining*

Data mining atau biasa disebut *knowledge discovery database (KDD)* merupakan tahap pengumpulan dan pemberian informasi yang bermanfaat dari sekumpulan data yang besar. *Data mining* juga digunakan untuk memberikan penguraian pada ilmu pengetahuan pada *database* yang melibatkan ilmu statistika, kecerdasan buatan dan *machine learning* untuk mengidentifikasi informasi yang bermanfaat untuk ilmu pengetahuan (Turban Efraim, Jay E. Aronson, 2005).

Menurut Han Jiawei dan Jian Pei (2006), *KDD* terdiri dari beberapa langkah, yaitu:

1. *Data cleaning*

Merupakan proses penghapusan data yang tidak konsisten atau menghilangkan *noise* pada data.

2. *Data Integration*

Merupakan proses penggabungan data apabila memiliki sumber data dalam sistem data mining tersebut.

3. *Data selection*

Merupakan proses pengambilan data yang relevan yang akan digunakan dalam proses data mining.

4. *Data transformation*

Merupakan proses data ditransformasi menjadi bentuk yang sesuai untuk dilakukan kalkulasi data mining.

5. *Data mining*

Merupakan proses utama yang melibatkan metode untuk menghasilkan pola data.

6. *Pattern evaluation*

Merupakan proses untuk menguji kebenaran dari pola data yang mewakili *knowledge* yang ada di dalam data.

7. *Knowledge representation*

Merupakan proses visualisasi dan teknik yang menyajikan *knowledge* yang dapat digunakan untuk menampilkan hasil analisis dari metode *data mining* kepada pengguna.

2.1.2 *Text mining*

Text mining adalah ilmu untuk menganalisis teks dalam jumlah besar dengan menggunakan metode statistik untuk menemukan pola dan tren yang tersembunyi (Luqyana, 2018). *Text mining* mampu menghasilkan informasi melalui alur proses, pengelompokan, dan analisis data yang tidak terstruktur dalam jumlah besar. Proses mempelajari struktur data teks dalam *text mining* adalah dengan terlebih dahulu menentukan fitur yang mewakili setiap kata pada dokumen. Menurut Purnaramadhan (2021), dokumen merupakan satuan data teks terkecil dalam *text mining*. Dokumen ini dapat berupa berbagai format teks. Kumpulan dari dokumen-dokumen ini disebut dengan *corpus*.

Pre-processing adalah tahap awal dari *data mining* yang melibatkan transformasi data mentah menjadi format yang mudah dimengerti. Proses ini dilakukan untuk menggali, mengolah, dan mengatur informasi dan untuk menganalisis hubungan tekstural dari data terstruktur dan data tidak terstruktur (Nugroho, 2016).

Tahapan dari *pre-processing* meliputi:

1. *Cleaning*

Cleaning merupakan proses pembersihan data dari komponen-komponen yang tidak diperlukan dalam analisis sentimen. Hal yang tidak diperlukan dalam analisis sentimen seperti *url*, tanda baca, digit dan tagar.

2. *Case folding*

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil. Hal ini dilakukan untuk menstandarisasi bentuk kata sehingga memudahkan proses pencarian dan analisis teks.

3. *Tokenizing*

Tokenizing adalah proses memecah teks menjadi unit-unit terkecil yang disebut dengan *token* dengan menggunakan spasi sebagai pembatas antar kata.

4. *Stopword removal*

Stopword removal adalah proses eliminasi kata-kata yang sering muncul namun tidak memiliki nilai tambah dalam analisis teks, seperti kata *hubung*.

5. *Stemming*

Stemming bertujuan untuk menyederhanakan kata menjadi bentuk dasarnya agar lebih mudah untuk dianalisis. Proses ini sering dilakukan setelah tahap *stopword removal*.

2.1.3 Analisis sentimen

Menurut Brahimi, dkk (2021) analisis sentimen adalah proses mengekstraksi, mengolah, dan memahami data berupa teks tidak terstruktur secara otomatis untuk mendapatkan informasi sentimen pada kalimat opini atau pendapat. Dewi dan Sulastri (2022) menjelaskan bahwa analisis sentimen digunakan sebagai gambaran umum media sosial untuk mengetahui pengetahuan perasaan lebih cenderung opini positif atau opini negatif. Maka dapat disimpulkan bahwa analisis sentimen merupakan analisis yang mengolah informasi berupa teks yang berisikan tanggapan, pendapat, atau komentar masyarakat terkait suatu topik permasalahan untuk mengetahui kecenderungan opini masyarakat.

Analisis sentimen juga disebut dengan *opinion mining* yang berguna dalam pengelolaan bahasa alami, komputasi linguistik, dan *text mining* (Luqyana, 2018). Analisis sentimen digunakan untuk menentukan perilaku atau opini terhadap topik tertentu. Perilaku tersebut dapat mengindikasikan penilaian dan kondisi kecenderungan. Analisis sentimen dapat diklasifikasikan ke dalam kelas sentimen bersifat positif atau negatif.

2.1.4 Pelabelan data

Data diberi label berdasarkan bintang atau skor yang diberikan oleh *user*. Label yang diberikan adalah positif dan negatif menggunakan fungsi *if* pada *Python*. Pelabelan memiliki nilai acuan yaitu jika atribut skor bernilai 1, 2 dan 3 maka label sentimen tersebut adalah sentimen negatif. Jika atribut skor bernilai 4 dan 5 maka label sentimen tersebut adalah sentimen positif (Maulana et al., 2023).

Penelitian oleh Nguyen, dkk (2018) menyatakan bahwa skor 3 sebagai kelompok negatif. Alasannya karena sulit untuk mengenali suatu kata yang bersifat netral termasuk kalimat positif ataupun negatif. Selain itu, distribusi skor pada aplikasi Wondr by BNI senilai 3,4 sehingga masuk akal jika skor 3 dikelompokkan sebagai sentimen negatif. Nguyen, dkk juga menyebutkan bahwa dalam praktiknya, ketika perusahaan mengumpulkan ulasan dari pelanggan mereka, mereka biasanya menganggap tanggapan yang netral sebagai negatif.

2.1.5 Pembobotan kata

Pembobotan kata adalah hal yang dilakukan setelah melakukan *pre-processing*. *Term* atau kata yang diperoleh dari *pre-processing* dijadikan indeks untuk mewakili dokumen. Karena setiap kata dalam dokumen memiliki bobot yang berbeda dan seluruh kata juga mencirikan indikator yang disebut *term weight* atau pembobotan kata (Zafikri, 2008).

Menurut Zafikri (2008) pembobotan kata memiliki komponen seperti berikut:

1. *Term Frequency (TF)*

Didasarkan pada seringnya kata muncul dalam suatu dokumen. banyaknya kemunculan suatu kata (*term frequency*) dikalkulasi dalam pemberian bobot terhadap suatu kata. Jika kemunculan *term* dalam dokumen berjumlah besar (TF tinggi), maka semakin besar juga bobotnya dalam dokumen atau akan memberikan nilai kesesuaian yang semakin tinggi. Rumus TF disajikan pada persamaan (2.1).

$$W_{tf_{t,d}} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

Keterangan:

$W_{tf_{t,d}}$ = Frekuensi kemunculan kata t dalam kalimat d

t = term/kata

d = dokumen

tf = *Term frequency*

2. Inverse Document Frequency (IDF)

IDF yaitu mengurangi dominasi kata-kata yang sering muncul pada dokumen. Hal ini diperlukan karena *term* yang sering muncul di berbagai dokumen, dapat dianggap sebagai *term* umum (*common term*) sehingga nilainya tidak penting. Faktor kejarangan muncul kata (*term scarcity*) dalam koleksi dokumen harus diperhatikan.

Metode TF-IDF merupakan metode pembobotan *term* yang banyak digunakan sebagai metode pembandingan terhadap metode pembobotan baru. Pada metode ini, perhitungan bobot *term t* dalam sebuah dokumen dilakukan dengan mengalikan nilai TF dan IDF. Rumus perhitungan IDF pada persamaan (2.2).

$$idf_t = \log_{10} N / df_t \quad (2.2)$$

Keterangan:

df_t = banyaknya dokumen yang memuat t

N = jumlah total dokumen

2.1.6 TF-IDF Weighting

Bobot yang terdapat pada TF-IDF *weighting* merupakan hasil perkalian dari $tf_{t,d}$ dan idf_t , rumus TF-IDF *weighting* dijelaskan pada persamaan (2.3).

$$W_{t,d} = W_{tf_{t,d}} \times idf_t \quad (2.3)$$

Keterangan :

$W_{t,d}$ = Bobot TF-IDF

$W_{tf_{t,d}}$ = Frekuensi kemunculan kata t dalam kalimat d

idf_t = nilai idf pada kata t

2.1.7 *Modified K-Nearest Neighbors* (MK-NN)

Algoritma *K-Nearest Neighbors* (*KNN*) adalah algoritma yang sederhana, yang mengklasifikasikan data baru berdasarkan tetangga terdekatnya (Arafat, 2021). *Modified K-Nearest Neighbors* (*MK-NN*) adalah versi pengembangan dari algoritma *KNN* yang dirancang untuk meningkatkan nilai akurasi. Algoritma *MK-NN* menambah dua tahapan klasifikasi, yaitu proses validasi data latih dan pemberian pembobotan terbesar (*weight voting*) pada tetangga terdekat (Amelia, 2022). Proses validasi dan pemberian bobot terbesar yang ada pada *MK-NN* mampu mengatasi kelemahan klasifikasi yang hanya didasarkan pada jarak terdekat dalam metode *K-NN*. Dengan kedua tahapan tersebut diharapkan metode *MK-NN* dapat meningkatkan akurasi yang kurang optimal pada metode *K-NN*. Berikut beberapa hal yang membedakan *KNN* dan *MK-NN*:

Tabel 2. 1 Perbedaan K-NN dan MK-NN

K-Nearest Neighbors	Modified K-Nearest Neighbor
Algoritma klasifikasi sederhana	dimodifikasi dari <i>MK-NN</i>
Mengklasifikasikan berdasarkan mayoritas tetangga terdekat	Klasifikasi dilakukan berdasarkan bobot dari tetangga terdekat
Bobot tetangga sama (<i>uniform</i>)	Perbedaan bobot antar tetangga, berdasarkan jarak atau <i>similarity</i>
Tidak memperhatikan seberapa dekat tetangga	Memperhatikan kedekatan tetangga dan kemiripan tetangga

Untuk memprediksi label pada data uji diperlukan melihat label pada tetangga terdekat. Label yang digunakan pada penelitian ini adalah 2 kelas, yaitu positif dan negatif. Jika nilai K yang digunakan dengan jumlah besar, sampel yang

memiliki label lebih banyak dapat mengalahkan sampel yang memiliki label yang lebih sedikit, sehingga menyebabkan bias dalam hasil klasifikasi (Mucwherino et al., 2009).

Penelitian yang dilakukan oleh Isnain, dkk (2021) menggunakan K yang bernilai 1 hingga 10. Penggunaan nilai K ganjil atau genap tidak menghasilkan bias pada MK-NN karena nilai K bukan digunakan untuk penentu label dari data uji, melainkan untuk perhitungan validitas pada data latih yang akan menjadi informasi tambahan dalam perhitungan *weight voting*. Maka digunakan nilai K=1 hingga K=10 dalam penelitian ini.

a. *Cosine Similarity*

Cosine similarity merupakan metode yang digunakan untuk menghitung derajat kemiripan antara dokumen. Nilai *cosine similarity* dihitung berdasarkan besarnya nilai fungsi *cosine* terhadap sudut yang dibentuk oleh dua vektor, yakni dalam kasus ini adalah sebuah representasi dokumen-dokumen antar data latih dan data uji. Rumus *cosine similarity* disajikan pada persamaan (2.4).

$$CosSim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (W_{ij} \cdot W_{iq})}{\sqrt{\sum_{i=1}^t W_{ij}^2 \cdot \sum_{i=1}^t W_{iq}^2}} \quad (2.4)$$

Keterangan:

\vec{d}_j = data latih ke-i

q = tetangga data latih ke-i

W_{ij} = nilai pembobotan kata pada dokumen latih

W_{iq} = nilai pembobotan kata pada tetangga dokumen latih

b. Validitas

Parvin (2008) berpendapat validitas digunakan untuk menghitung jumlah titik dengan label yang sama pada seluruh data dalam data latih. Validitas setiap data tergantung pada tetangga terdekatnya. Setelah proses validasi dilakukan, data tersebut digunakan sebagai informasi tambahan. Persamaan (2.5) digunakan untuk menghitung validitas.

$$validity(x) = \frac{1}{K} \sum_{i=1}^K S(lbl(x), lbl(Ni(x))) \quad (2.5)$$

Keterangan:

Validity = Validitas antar data latih

K = Jumlah tetangga terdekat

i = Nilai terbaik bernilai 1

lbl(x) = Label kelas x

lbl(Ni(x)) = Label kelas titik terdekat dengan x

S merupakan perhitungan antara titik a dan data ke-b dari tetangga paling dekat (Maharani et al., 2023). Untuk menghitung nilai S, dapat diperoleh oleh persamaan berikut:

$$S(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$$

Keterangan:

S = similarity

a = label data ke-i pada data latih

b = label tetangga terdekat data ke-i pada data latih

c. *Cosine Distance*

Dua titik yang satu pada data latih dan satunya lagi pada data uji akan dihitung jarak antara keduanya dijelaskan pada persamaan (2.6).

Cosine distance:

$$1 - \text{CosSim}(d_j, q) = 1 - \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = 1 - \frac{\sum_{i=1}^t (W_{ij} \cdot W_{iq})}{\sqrt{\sum_{i=1}^t W_{ij}^2 \cdot \sum_{i=1}^t W_{iq}^2}} \quad (2.6)$$

Keterangan:

d_j = data uji

q = data latih ke-i

W_{ij} = nilai pembobotan pada dokumen uji

W_{i,q} = nilai pembobotan kata pada tetangga dokumen latih

d. *Weight Voting*

Langkah pertama dalam pembobotan voting adalah pembobotan setiap tetangga dihitung dengan menggunakan $1/(d_c + 0,5)$, selanjutnya nilai tersebut dikalikan dengan validitas masing-masing data berdasarkan *cosine distance*. Metode ini memberikan bobot yang lebih tinggi pada data yang memiliki validitas yang lebih tinggi, sehingga menghasilkan output yang lebih akurat dan representatif. Selain itu, teknik ini juga memungkinkan perbaikan terhadap data yang memiliki masalah validitas dengan cara menyesuaikan bobot berdasarkan jarak tetangganya (Amelia, 2022). Rumus untuk menghitung *weight voting* adalah pada persamaan (2.7).

$$W(i) = \text{validity}(i) \times \frac{1}{d_c + 0,5} \quad (2.7)$$

Keterangan :

W	= bobot antara data latih dengan data latih ke- i
i	= jumlah data latih
validity	= validitas data latih
d_c	= jarak antara data uji dengan latih

Nilai *regular smoothing* sebesar 0,5 digunakan untuk mengatasi masalah probabilitas nol. Masalah ini muncul ketika ada kata atau fitur yang tidak ada di data latih namun muncul di data uji, atau sebaliknya (A. Prasetyo, Rino, Indriati, P. Adikara, 2018).

2.1.8 Evaluasi

a. *Confusion Matrix*

Confusion matrix merupakan suatu metode evaluasi yang digunakan untuk menilai performa model klasifikasi. Matriks ini menyajikan data dalam bentuk tabel yang menunjukkan seberapa akurat model dalam mengklasifikasikan data uji, baik yang diklasifikasikan dengan benar ataupun salah (Erdiansyah et al., 2022).

Confusion matrix terdiri dari empat istilah yang menggambarkan hasil klasifikasi, yaitu *True Positive* (TP) yang berarti data positif yang diprediksi dengan benar, *True Negative* (TN) yang berarti data negatif yang diprediksi dengan benar ketika hasil prediksi sesuai dengan kondisi sebenarnya. Sebaliknya, *False Positive* (FP) adalah data negatif yang salah diklasifikasikan sebagai data positif, dan *False Negative* (FN) adalah data positif yang diprediksi sebagai data negatif, atau ketika hasil prediksi tidak sesuai dengan keadaan yang sebenarnya terjadi (Romadhon & Kurniawan, 2021).

Tabel 2. 2 *Confusion matrix*

Hasil Prediksi	Hasil Aktual	
	Positif	Negatif
Positif	TP	FP
Negatif	FN	TN

Berdasarkan nilai TP, TN, FP, dan FN pada *confusion matrix*, dapat diperoleh beberapa parameter seperti akurasi, presisi, dan *recall* yang digunakan untuk mengevaluasi kinerja klasifikasi dari algoritma tersebut.

b. *Precision*

Precision menunjukkan ketika model memprediksi positif, seberapa sering prediksi itu benar (Saputro & Sari, 2020). Sehingga *precision* dapat digunakan sebagai gambaran tingkat keakuratan prediksi positif. *Precision* dapat dihitung dengan rumus pada persamaan (2.8).

$$Precision (P) = \frac{TP}{TP + FP} \quad (2.8)$$

c. *Recall*

Recall menunjukkan ketika kelas aktualnya positif, seberapa sering model memprediksi positif (Saputro & Sari, 2020). *Recall* dapat digunakan sebagai penilaian kinerja model dalam mengklasifikasikan data positif dari seluruh data aktual. *Recall* dapat dihitung dengan persamaan (2.9).

$$Recall (R) = \frac{TP}{TP + FN} \quad (2.9)$$

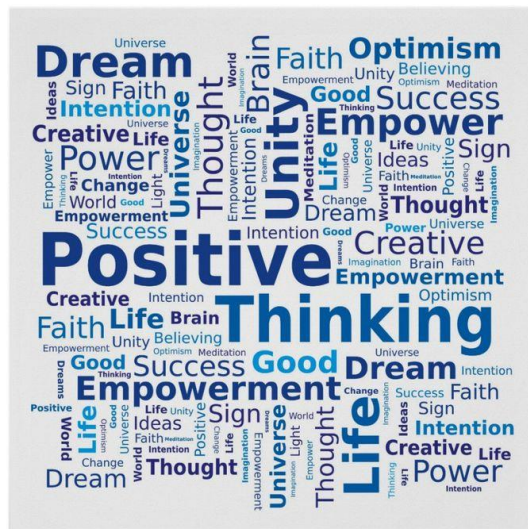
d. Akurasi

Akurasi mengukur seberapa dekat nilai prediksi dengan nilai sederhana. Akurasi digunakan untuk menilai seberapa banyak prediksi yang sesuai dengan kelas aktual. Semakin tinggi nilai akurasi, semakin baik performa klasifikasi yang dihasilkan. Perhitungan akurasi disajikan pada persamaan (2.10)

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100 \quad (2.10)$$

e. Wordcloud

Wordcloud menyoroti istilah yang sering digunakan berdasarkan frekuensinya dan dapat menjelaskan pertanyaan penelitian dengan cepat dan mudah (Feldman & Sanger, 2012). Semakin besar ukuran *term* atau kata dalam visual, maka semakin umum kata tersebut dalam dokumen. Sedangkan jika *term* memiliki ukuran yang kecil, artinya *term* tersebut tidak cukup banyak muncul dari analisis.



Gambar 2. 1 Bentuk *Wordcloud*

2.2 Tinjauan Non Statistik

2.2.1 Google Play

Google Play adalah platform distribusi digital yang dikembangkan oleh Google dan berfungsi sebagai toko aplikasi resmi untuk sistem operasi Android. Melalui platform ini, *user* dapat menjelajahi dan mengunduh aplikasi yang dirilis oleh *developer* melalui Google. Google Play juga berfungsi sebagai toko media digital yang menawarkan produk-produk seperti buku, permainan, aplikasi, ataupun pemutar media berbasis *cloud* (Praptiwi, 2018).



Gambar 2. 2 Logo Google Play

2.2.2 Wondr by BNI

Aplikasi Wondr by BNI yang diluncurkan oleh PT Bank Negara Indonesia (BNI) Tbk pada tanggal 5 Juli 2024 yang memberikan solusi bagi masyarakat Indonesia untuk dapat melakukan pengelolaan keuangan yang lebih terencana sesuai kebutuhan finansial masing-masing melalui fitur 3 dimensi keuangan (*transaksi, insight dan growth*).



Gambar 2. 3 Logo Wondr by BNI

Dengan aplikasi Wondr *by* BNI yang memiliki 3 dimensi keuangan yang mencerminkan konsep keuangan masa lalu, masa kini, dan masa depan. 3 dimensi tersebut dirincikan sebagai berikut:

1. *Transaksi*. Dengan menyediakan fitur transfer bank domestik, pembayaran tagihan dan penjadwalan transfer yang dapat diselesaikan dalam tiga langkah, aplikasi ini memfasilitasi pengelolaan transaksi harian nasabah secara *real-time*, sehingga memenuhi kebutuhan keuangan mereka secara cepat dan efisien.
2. *Insight*. Dengan adanya fitur *insight*, nasabah dapat memperoleh pemahaman yang lebih rinci tentang keuangan mereka melalui analisis data historis yang disajikan dalam bentuk laporan keuangan harian. Fitur ini memungkinkan nasabah untuk mengelola keuangan lebih optimal dan membuat keputusan finansial yang lebih cerdas.
3. *Growth*. Aspek pertumbuhan dalam layanan ini menekankan pada perencanaan keuangan jangka panjang. Dengan menyediakan pilihan produk investasi dan tabungan yang beragam, nasabah dapat mencapai tujuan finansial mereka secara efektif.

BAB III

METODOLOGI PENELITIAN

3.1 Sumber Data

Penelitian ini menggunakan data sekunder. Data penelitian ini diperoleh melalui teknik *scraping* pada penilaian aplikasi Wondr by BNI di Google Play. Periode pengumpulan data dilakukan dari 5 Juli hingga 2 September 2024. Seluruh *user* yang aktif memberikan komentar pada periode tersebut menjadi sampel dalam penelitian ini sebanyak 4.264 komentar.

3.2 Variabel Penelitian

Data yang diperoleh melalui teknik *scraping* dengan *Python* di Google Collaboratory, yang mencakup tiga atribut, yaitu tanggal penilaian, skor, dan komentar. Lalu dilakukan pembagian data menjadi dua kelompok: data latih (80%) dan data uji (20%). Pembagian data ini mengikuti proporsi yang umum digunakan dalam penelitian sebelumnya. Dari variabel penelitian, hanya menggunakan atribut komentar dan atribut skor digunakan untuk pelabelan kelas sentimen.

Tabel 3. 1 Atribut Penelitian

No	Indikator	Jenis data	Keterangan
1	Tanggal	<i>Date</i>	Tanggal dibuatnya penilaian
2	Skor	<i>Scale</i>	Tingkat kepuasan <i>user</i>
3	Komentar	<i>String</i>	Isi penilaian <i>user</i>

3.3 Langkah Penelitian

Penelitian ini menggunakan 2 perangkat lunak untuk membantu proses analisis, yaitu *Python* dengan bantuan Google Collaboratory dan *Microsoft Excel* sebagai tempat menyimpan data. Adapun langkah analisis yang dilakukan adalah sebagai berikut:

1. Mengumpulkan data ulasan dengan menggunakan *library google play scraper* pada Google Colaboratory;
2. Melakukan *pre-processing data*;
 - a. *Cleaning data*, untuk pembersihan data dari komponen yang tidak diperlukan seperti emoji, tanda baca, digit dan tagar.
 - b. *Case folding*, untuk mengubah seluruh teks menjadi huruf kecil.
 - c. *Tokenizing*, untuk memisahkan kalimat menjadi kata.
 - d. *Stopword removal*, untuk menghapus kata-kata yang tidak diperlukan seperti: di, ke, dan, atau.
 - e. *Stemming*, untuk mencari kata dasar dari hasil kata *stopword removal*.
3. Melakukan pelabelan data berdasarkan skor yang diberikan *user*;
4. Memvisualisasikan hasil sentimen dengan *wordcloud*;
5. Melakukan pembobotan kata dengan TF IDF pada persamaan (2.3);
6. Membagi data latih sebanyak 80% dan data uji sebanyak 20%;
7. Melakukan Analisis *Modified K-Nearest Neighbor*;

Pada data latih dilakukan:

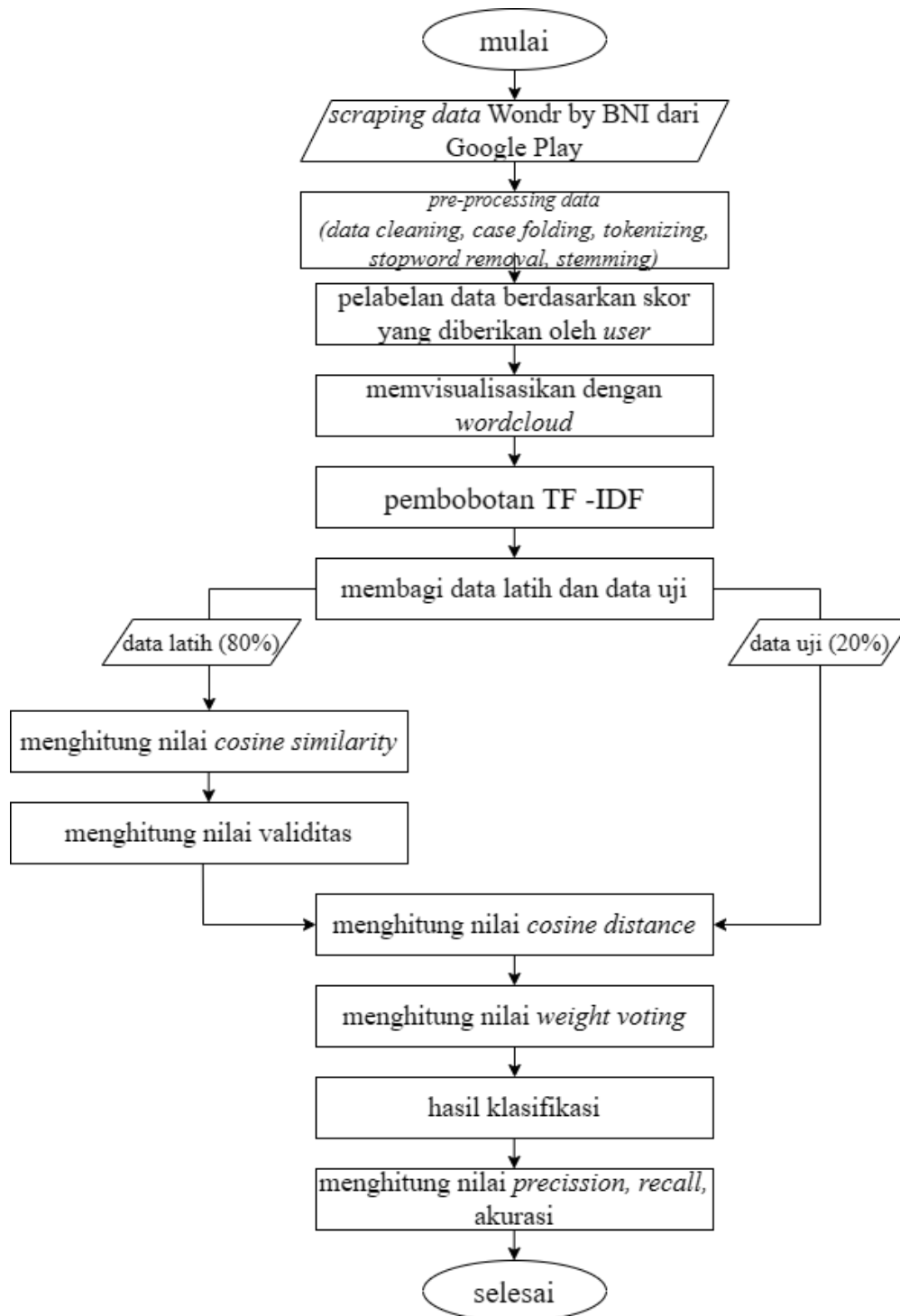
- a. Menghitung *cosine similarity* pada data latih dengan persamaan (2.4).
- b. Mengurutkan hasil *cosine similarity* antar data latih.
- c. Menghitung nilai validitas data dengan persamaan (2.5).

Lalu dilanjutkan kalkulasi data latih yang melibatkan data uji:

- d. Menghitung *cosine distance* data latih terhadap data uji dengan persamaan (2.6).
 - e. Menghitung nilai *weight voting* dengan persamaan (2.7).
 - f. Memperoleh hasil klasifikasi dalam bentuk *confusion matrix*.
 - g. Mengevaluasi hasil klasifikasi dengan *precision* yang dihitung menggunakan persamaan (2.8), *recall* yang dihitung dengan persamaan (2.9), dan akurasi yang dihitung dengan persamaan (2.10).
8. Menginterpretasikan hasil klasifikasi dan membuat kesimpulan.

3.4 Diagram Alir Penelitian

Diagram alir penelitian ini disajikan pada Gambar 3.1



Gambar 3. 1 Diagram Alir Penelitian

BAB IV HASIL DAN PEMBAHASAN

4.1 Scraping data

Hasil *scraping data* dari komentar wondr by BNI pada Google Play dalam rentang 5 Juli – 2 September 2024 diperoleh data komentar sebanyak 4.264 data. Dengan menggunakan *package “google-play-scraper”* dan memasukkan domain BNI yaitu “id.bni.wondr” maka diperoleh beberapa kolom dari hasil *scraping* yaitu: *reviewID*, *username*, *userImage*, *content*, *score*, *thumbsUpCount*, *reviewCreatedVersion*, *at*, *replyContent*, *repliedAt*, dan *AppVersion*. Pada penelitian ini hanya diperlukan *At* (tanggal), *score* (bintang), dan *content* (ulasan). Hasil *scraping* disajikan dalam tabel 4.1.

Tabel 4. 1. Hasil *scraping* dari Google Play Store

<i>score</i>	<i>at</i>	<i>content</i>
5	7/5/2024 0:03	The best
3	7/5/2024 0:14	Tolong ditambahkan fitur tarik tunai dan setor cashless seperti bni mobile
5	7/5/2024 0:42	Transaksi lebih cepat dan aman, menu mudah di pahami Menyalaaa ðŸ”¥ðŸ”¥ðŸ”¥
5	7/5/2024 1:00	Good
5	7/5/2024 1:04	Tampilan baru dg warna yang cantik, menu aplikasi yg lengkap dan mudah di akses
5	7/5/2024 1:13	Bagus
3	7/5/2024 1:13	Saya kasih 3 dulu dikarenakan pengiriman otp kode email cukup lama sehingga waktunya habis
5	7/5/2024 1:14	kerenn banget
5	7/5/2024 1:16	Lebih mudah,lebih cepat ðŸ”¥ðŸ”¥ðŸ”¥
5	7/5/2024 1:21	Keren dari yang sebelumnya dan lebih mudah dan praktis no ribet...
5	7/5/2024 0:03	The best
3	7/5/2024 0:14	Tolong ditambahkan fitur tarik tunai dan setor cashless seperti bni mobile
5	7/5/2024 0:42	Transaksi lebih cepat dan aman, menu mudah di pahami Menyalaaa ðŸ”¥ðŸ”¥ðŸ”¥
5	7/5/2024 1:00	Good
5	7/5/2024 1:04	Tampilan baru dg warna yang cantik, menu aplikasi yg lengkap dan mudah di akses
⋮	⋮	⋮
5	9/2/2024 13:23	Mantap

4.2 Pre-processing data

Tahap *pre-processing data* perlu dilakukan sebelum melakukan pembobotan pada dokumen. *Pre-processing* perlu dilakukan karena data yang diperoleh dari *scraping* Google play terdapat komponen yang tidak bisa diberikan nilai pembobotan. Maka tahap *pre-processing* ini berfungsi sebagai penyeragaman bentuk kata dan membersihkan kata dari komponen yang tidak diperlukan. Tahapan *pre-processing* adalah sebagai berikut:

a. *Cleaning data*

Cleaning data digunakan untuk menghapus komponen-komponen yang tidak diperlukan, seperti *url*, emoji, tagar, dan tanda baca. Hasil *cleaning data* disajikan dalam Tabel 4.2.

Tabel 4. 2 Hasil *cleaning data*

Komentar	Hasil <i>cleaning data</i>
Makin mudah dan cepat bertransaksi dengan wondr 🖐	Makin mudah dan cepat bertransaksi dengan wondr
Wondr is wonderful... ☐ 🖐	Wondr is wonderful

b. *Case folding*

Case folding digunakan untuk mengubah seluruh huruf kapital menjadi huruf kecil. Hal ini dilakukan untuk menstandarisasi data agar memudahkan proses pencarian kata dasar dan analisis teks. Hasil *case folding* disajikan dalam Tabel 4.3.

Tabel 4. 3 Hasil *case folding*

Komentar	Hasil <i>case folding</i>
Makin mudah dan cepat bertransaksi dengan wondr	makin mudah dan cepat bertransaksi dengan wondr
Wondr is wonderful	wondr is wonderful

c. *Tokenizing*

Tokenizing digunakan untuk memisahkan kalimat menjadi potongan-potongan kata yang dinamakan *token*. *Token* akan mempermudah perhitungan kata dalam komentar untuk pembobotan kata. Hasil *tokenizing* disajikan dalam Tabel 4.4.

Tabel 4. 4 Hasil *tokenizing*

Komentar	Hasil <i>tokenizing</i>
makin mudah dan cepat bertransaksi dengan wondr wondr is wonderful	['makin', 'mudah', 'dan', 'cepat', 'bertransaksi', 'dengan', 'wondr', 'wondr', 'is', 'wonderful']

d. *Stopword removal*

Stopword removal digunakan untuk menghapus kata yang sering muncul, namun tidak memiliki arti. Seperti konjungsi. Hasil *stopword removal* disajikan dalam Tabel 4.5.

Tabel 4. 5 Hasil *stopword removal*

Komentar	Hasil <i>stopword removal</i>
['makin', 'mudah', 'dan', 'cepat', 'bertransaksi', 'dengan', 'wondr', 'wondr', 'is', 'wonderful']	['mudah', 'cepat', 'bertransaksi', 'wondr', 'wondr', 'is', 'wonderful']

e. *Stemming*

Stemming digunakan untuk mencari kata dasar dari suatu kata agar mempermudah proses pembobotan kata. Hasil *stemming* disajikan dalam Tabel 4.6.

Tabel 4. 6 Hasil *stemming*

Komentar	Hasil <i>stemming</i>
['makin', 'mudah', 'dan', 'cepat', 'bertransaksi', 'dengan', 'wondr', 'wondr', 'is', 'wonderful']	['mudah', 'cepat', 'transaksi', 'wondr', 'wondr', 'is', 'wonderful']

Setelah melakukan *pre-processing data*, jumlah data yang tersisa untuk dilanjutkan ke tahap berikutnya adalah sebanyak 4.191 data.

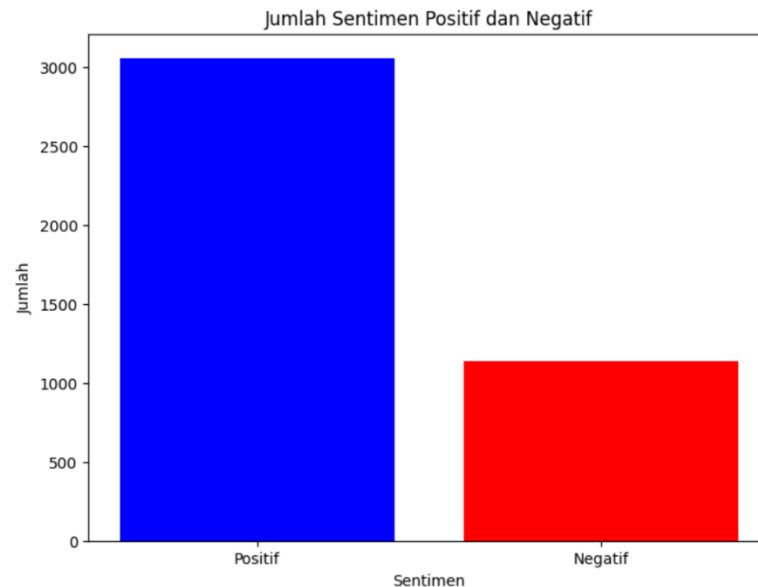
4.3 Pelabelan data

Pelabelan data menggunakan skor atau bintang yang diberikan oleh *user* dengan *library If else* pada *Python*. Skor 1, 2, dan 3 merupakan label sentimen negatif dan skor 4, dan 5 merupakan label sentimen positif. Hasil pelabelan data komentar disajikan dalam Tabel 4.7.

Tabel 4. 7 Hasil pelabelan data

No.	Komentar	Skor	Sentimen
1	mantap	5	Positif
2	woowsuka bangetwondr by bni mantap kerennsangat simpel transaksi apa praktis wondrfull	5	Positif
3	bayar qris wondr cepat banget cuman 3 step selesai	5	Positif
...
4.189	transaksi cepat aman menu mudah paham menyalaaa	5	Positif
4.190	tolong fitur tarik tunai setor cashless bni mobile	3	Negatif
4.191	the best	5	Positif

Hasil pelabelan sentimen positif dan negatif disajikan dalam diagram batang pada Gambar 4.1.



Gambar 4. 1 Diagram batang pelabelan sentimen

Pada hasil diagram batang diperoleh bahwa sentimen positif lebih banyak dari sentimen negatif. Sentimen positif berjumlah 3.056 data dan sentimen negatif sebanyak 1.135 data.

4.4 wordcloud

Setelah setiap data diberi pelabelan sentimen positif dan sentimen negatif, maka dapat dilakukan *wordcloud* untuk melihat kata apa saja yang sering dibahas oleh *user*. Semakin besar ukuran kata yang muncul pada *wordcloud* menandakan frekuensi pembahasan kata itu cukup sering. Sebaliknya, jika kata yang muncul hanya berukuran kecil menandakan frekuensi pembahasan kata tersebut cukup jarang. Berikut hasil *wordcloud* dari sentimen positif disajikan dalam Gambar 4.2.

4.5 Pembobotan kata

Pembobotan kata menggunakan metode TF-IDF dengan data yang tersisa setelah *pre-processing data* sebanyak 4.191 komentar. Berikut hasil 5 komentar pertama untuk mewakili pembobotan TF-IDF dari 4.191 komentar yang disajikan dalam tabel 4.8.

Tabel 4.8. 5 komentar pertama

D	Komentar	Label
D1	tampil warna cantik menu aplikasi lengkap mudah akses	Positif
D2	good	Positif
D3	transaksi cepat aman menu mudah paham	Positif
D4	tolong fitur tarik tunai setor cashless bni mobile	Negatif
D5	best	Positif

Pertama, dilakukan perhitungan TF yaitu dengan menghitung berapa banyak suatu kata terdapat dari semua ulasan. Lalu dilanjutkan dengan menghitung IDF seperti persamaan (2.2). Dicontohkan kata “tampil” dalam perhitungan IDF.

$$idf_{\text{tampil}} = \log\left(\frac{1}{5}\right) = 0,698$$

Hasil perhitungan TF dan IDF disajikan dalam tabel 4.9.

Tabel 4. 9 Perhitungan TF dan IDF

Term	TF					IDF	
	D1	D2	D3	D4	D5	df	log(n/df)
tampil	1					1	0.69897
warna	1					1	0.69897
cantik	1					1	0.69897
menu	1		1			2	0.39794
aplikasi	1					1	0.69897
lengkap	1					1	0.69897
mudah	1					1	0.69897
akses	1					1	0.69897
good		1				1	0.69897
transaksi			1			1	0.69897
cepat			1			1	0.69897
aman			1			1	0.69897
paham			1			1	0.69897
tolong				1		1	0.69897

fitur	1	1	0.69897
tarik	1	1	0.69897
tunai	1	1	0.69897
setor	1	1	0.69897
cashless	1	1	0.69897
bni	1	1	0.69897
mobile	1	1	0.69897
best		1	0.69897

Pada kolom TF yang terdiri atas 5 buah dokumen (D) sudah memuat distribusi frekuensi kata dari masing-masing dokumen. Apabila suatu kata terdapat dalam suatu dokumen maka akan diberi angka 1, dan jika tidak ada maka akan diberi angka 0. Setelah TF diperoleh, selanjutnya menghitung IDF dengan melogaritmakan total dokumen dibagi df. Tahapan selanjutnya adalah menghitung nilai bobot pada persamaan (2.3) dan diperoleh hasil pembobotan pada tabel 4.10.

Tabel 4. 10 Hasil pembobotan TF-IDF 5 data pertama

	D1	D2	D3	D4	D5
0.698970004	0	0	0	0	0
0.698970004	0	0	0	0	0
0.698970004	0	0	0	0	0
0.397940009	0	0.39794	0	0	0
0.698970004	0	0	0	0	0
0.698970004	0	0	0	0	0
0.698970004	0	0	0	0	0
0.698970004	0	0	0	0	0
0	0.69897	0	0	0	0
0	0	0.69897	0	0	0
0	0	0.69897	0	0	0
0	0	0.69897	0	0	0
0	0	0.69897	0	0	0
0	0	0	0.69897	0	0
0	0	0	0.69897	0	0
0	0	0	0.69897	0	0
0	0	0	0.69897	0	0
0	0	0	0.69897	0	0
0	0	0	0.69897	0	0
0	0	0	0.69897	0	0
0	0	0	0	0.69897	0

Dari hasil pembobotan TF IDF jika suatu kata hanya terdapat pada 1 dokumen maka nilainya adalah sebesar 0,698 dan jika suatu kata terdapat pada 2 dokumen maka nilainya adalah sebesar 0,397 dan jika suatu kata tidak terdapat pada suatu dokumen maka nilainya adalah 0.

4.6 Klasifikasi *Modified K-Nearest Neighbor (MK-NN)*

Pada pengklasifikasian dengan *MK-NN*, tahapan pertama yang harus dilakukan pembagian data latih dan data uji. Data latih adalah data yang digunakan untuk melatih model dan data uji adalah data yang digunakan untuk menguji model. Proses klasifikasi dengan *MK-NN* menggunakan 4.191 komentar. Pembagian data dilakukan secara 80% data latih dan 20% data uji yang disajikan pada Tabel 4.11.

Tabel 4. 11 Pembagian data latih dan data uji

Persentase	Jumlah data	
	Data latih	Data uji
80% : 20%	3.352	839

Perhitungan proses *Modified K-Nearest Neighbor (MK-NN)* akan dicontohkan dengan 10 komentar pertama dari data penelitian yang disajikan pada tabel 4.12.

Tabel 4. 12. 10 komentar pertama

D	Komentar	Label
D1	keren mudah praktis no ribet	Positif
D2	mudah lebih cepat	Positif
D3	keren banget	Positif
D4	bagus	Positif
D5	kasih 3 kirim otp kode email habis	Negatif
D6	tampil warna cantik menu aplikasi lengkap mudah akses	Positif
D7	good	Positif
D8	transaksi cepat aman menu mudah paham	Positif
D9	tolong fitur tarik tunai setor cashless bni mobile	Negatif
D10	best	Positif

a. Cosine similarity

Perhitungan *cosine similarity* pada data latih yang digunakan untuk menghitung kemiripan antar komentar yang mana perhitungannya menggunakan rumus pada persamaan (2.4). Dari perhitungan kemiripan yang dihasilkan, maka akan mempermudah penentuan kelas untuk data uji. Perhitungan hasil *cosine similarity* menggunakan 8 komentar pertama yang dipilih untuk menjadi data latih, sama pada contoh perhitungan TF-IDF yang disajikan dalam bentuk matriks pada Tabel 4.13.

Tabel 4. 13 Hasil cosine similarity

	D1	D2	D3	D4	D5	D6	D7	D8
D1	1	0.13	0.26	0	0	0.07	0	0.09
D2	0.13	1	0	0	0	0.10	0	0.35
D3	0.26	0	1	0	0	0	0	0
D4	0	0	0	1	0	0	0	0
D5	0	0	0	0	1	0	0	0
D6	0.07	0.10	0	0	0	1	0	0.19
D7	0	0	0	0	0	0	1	0
D8	0.09	0.35	0	0	0	0.19	0	1

Hasil *cosine similarity* berkisar dari 0 hingga 1 yang mana jika nilai *cosine similarity* mendekati 1, maka dapat disimpulkan bahwa komentar memiliki kesamaan. Sebaliknya, jika nilai *cosine similarity* mendekati 0 maka komentar tidak memiliki kesamaan. Pada 8 komentar yang sudah dihitung *cosine similarity* menunjukkan D2 dan D8 memiliki nilai *cosine similarity* terbesar yaitu 0,35 yang berarti D2 dan D8 memiliki kemiripan komentar.

Hasil *cosine similarity* keseluruhan data latih pada penelitian ini disajikan dalam Lampiran 10.

b. Validitas

Validitas berfungsi sebagai informasi tambahan yang digunakan untuk menghitung jumlah titik dengan label yang sama pada data latih. Validitas

setiap data, tergantung pada tetangga terdekatnya. Jika ada suatu data x memiliki label positif, maka tetangga yang memiliki label positif akan dihitung dengan nilai 1. Sebaliknya, jika tetangga x memiliki nilai negatif, maka tetangganya akan dihitung dengan nilai 0.

Untuk menghitung nilai validitas dari setiap data latih dengan dimisalkan nilai $K=3$. Maka akan dilihat terlebih dahulu 3 tetangga terdekat dari masing-masing data latih menggunakan rumus pada persamaan (2.5), diperoleh operasi hitung sebagai berikut.

$$validity(D1) = \frac{1}{3}(1 + 1 + 1) = 1$$

$$validity(D2) = \frac{1}{3}(1 + 1 + 1) = 1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$validity(D7) = \frac{1}{3}(1 + 1 + 1) = 1$$

$$validity(D8) = \frac{1}{3}(1 + 1 + 1) = 1$$

Kemudian hasil validitas pada 8 komentar data latih dimuat dalam bentuk vektor $validity = [1, 1, 1, 1, 0, 1, 0,66, 1]$ yang mana terdapat perbedaan nilai validitas pada data latih. Apabila ditemukan nilai salah satu atau lebih tetangga data latih memiliki label yang berbeda seperti $D5$, maka operasi hitungnya dilakukan sebagai berikut.

$$validity(D_5) = \frac{1}{3}(0 + 0 + 0) = 0$$

Berikut disajikan perhitungan validitas dari masing-masing nilai K 10 data pertama untuk seluruh data latih dalam penelitian ini yang disajikan dalam tabel 4.14.

Tabel 4. 14. Hasil validitas antar data latih

K	Validity
1	[1, 1, 1, 1, 1, 1, 1, 1, 1, 0, ...]
2	[1, 1, 1, 1, 1, 1, 1, 1, 1, 0.5, ...]
3	[1, 1, 1, 1, 0.6, 1, 1, 1, 1, 0.6, ...]

4	[1, 1, 1, 1, 0.75, 1, 1, 1, 1, 0.75, ...]
5	[1, 1, 1, 1, 0.8, 0.8, 1, 1, 1 0.8, ...]
6	[1, 1, 1, 1, 0.83, 0.66, 1, 1, 1, 0.66, ...]
7	[1, 1, 1, 1, 0.857, 0.571, 1, 1, 1, 0.714, ...]
8	[1, 1, 1, 1, 0.875, 0.625, 1, 1, 1, 0.75, ...]
9	[0.88, 1, 1, 1, 0.88, 0.55, 1, 1, 1, 0.66, ...]
10	[0.9, 1, 1, 1, 0.9, 0.5, 1, 1, 1, 0.7, ...]

c. *Cosine Distance*

Cosine distance digunakan untuk menghitung jarak data latih dan data uji dengan perhitungan seperti persamaan (2.6). Yang menjadi data uji adalah D9 dan D10, dan D lainnya merupakan data latih. Maka diperoleh hasil *cosine distance* matriks yang disajikan dalam tabel 4.15.

Tabel 4. 15 Hasil cosine distance data latih dan data uji

	D9	D10
D1	1	0,888
D2	1	0,854
D3	1	1
D4	1	1
D5	1	1
D6	1	1
D7	1	1
D8	1	1

Pada *cosine similarity* ketika hasil perhitungan mendekati 0, maka dapat disimpulkan kedua dokumen memiliki kemiripan dan ketika perhitungan mendekati 1, maka dapat disimpulkan kedua dokumen memiliki perbedaan. Hasil *cosine distance* data uji dan data latih pada *Pyhton* disajikan pada Lampiran 15.

d. *Weight Voting*

Setelah diperoleh nilai validitas antar data latih dan nilai *cosine similarity* antara data latih dan data uji, maka dilanjutkan dengan

perhitungan weight voting. Untuk penentuan label data uji adalah dengan melihat nilai weight voting tertinggi. Jika nilai weight voting tertinggi memiliki label positif, maka data uji akan berlabel positif. Sebaliknya, jika nilai weight voting tertinggi memiliki label negatif, maka data uji akan berlabel negatif. Perhitungan weight voting untuk D9 dan D10 terhadap D1 dilakukan seperti pada persamaan (2.7).

$$W(D9, D1) = 1 \times \frac{1}{1 + 0,5} = 0,67$$

$$W(D10, D1) = 1 \times \frac{1}{0,888 + 0,5} = 0,72$$

Maka diperoleh hasil weight voting pada Tabel 4.16.

Tabel 4. 16 Hasil *weight voting*

	D9	D10
D1	0.67	0.72
D2	0.67	0.74
D3	0.67	0.88
D4	0.67	0.67
D5	0.67	0.67
D6	0.67	0.67
D7	0.67	0.67
D8	0.67	0.67

Berdasarkan hasil weight voting untuk data uji D9 adalah 0,67 pada seluruh data latih, sehingga kelas D9 sama dengan D1. Untuk data uji D10, nilai weight voting tertinggi adalah D3, sehingga kelas D10 juga sama dengan D3.

Hasil *weight voting* untuk pada penelitian ini disajikan pada Lampiran 17. Setelah diperoleh hasil *weight voting* terbesar, maka label pada data uji akan mengikut kepada label dari nilai *weight voting* terbesar.

e. Evaluasi model

Setelah menghitung *weight voting* dari seluruh nilai K, maka diperoleh nilai akurasi, *precision*, dan *recall* yang disajikan dalam Tabel 4.17.

Tabel 4. 17 Hasil akurasi, *precision*, dan *recall*

Nilai K	Akurasi	<i>Precision</i>	<i>Recall</i>
1	86,29%	87,63%	94,18%
2	87,24%	87,55%	95,84%
3	86,05%	85,92%	96,34%
4	86,05%	85,60%	96,84%
5	85,45%	84,68%	97,34%
6	85,33%	84,31%	97,67%
7	84,98%	83,80%	98,00%
8	84,38%	83,30%	97,84%
9	84,38%	83,30%	97,84%
10	84,26%	83,00%	98,17%

Dari tabel di atas, nilai akurasi tertinggi diperoleh pada K=2 yang mana memiliki nilai akurasi sebesar 87,24%. Nilai *precision* terbaik diperoleh pada K=1 dengan nilai sebesar 87,63%. Kemudian, nilai *recall* tertinggi diperoleh pada K=10 yang bernilai 98,17%. Berdasarkan rumusan masalah untuk menemukan nilai akurasi terbaik, maka nilai K=2 merupakan jawaban untuk rumusan masalah. Lalu diperoleh hasil *confusion matrix* yang disajikan pada Tabel 4.18.

Tabel 4. 18 Hasil *confusion matrix*

Hasil Prediksi	Hasil Aktual	
	Positif	Negatif
Positif	577	82
Negatif	25	155

Berdasarkan hasil *confusion matrix* pada 839 data uji menggunakan metode *Modified K-Nearest Neighbor (MK-NN)* menghasilkan hasil klasifikasi yang baik, dimana 732 data diklasifikasikan dengan benar yang terbagi menjadi 577 data memiliki nilai aktual positif dan hasil prediksi juga

menunjukkan nilai yang positif, dan 155 data memiliki nilai aktual negatif dan hasil prediksi juga menunjukkan nilai yang negatif. Sisanya 107 data belum diklasifikasikan dengan tepat, yaitu terdapat 25 data yang memiliki nilai aktual positif tetapi hasil prediksi menunjukkan nilai yang negatif, dan 82 data yang memiliki nilai aktual negatif tetapi hasil prediksi menunjukkan nilai yang positif.

Dari hasil *confusion matrix*, dapat dihitung nilai akurasi, *precision*, dan *recall*. Nilai akurasi dihitung dengan rumus seperti pada persamaan (2.10) dan diperoleh nilai akurasi sebagai berikut.

$$\text{akurasi} = \frac{577 + 155}{577 + 25 + 155 + 82} = \frac{732}{839} = 0,8724$$

Dari hasil akurasi yang didapatkan, maka model memprediksi secara benar data aktual sebesar 87,24%. Maka, dengan hasil tersebut, diketahui bahwa model mampu memprediksi label positif dan negatif secara baik. Semakin mendekati 100%, maka dapat disimpulkan ketepatan model dalam memprediksi label semakin baik, sebaliknya ketika nilai akurasi semakin rendah, maka ketepatan model dalam memprediksi label positif dan negatif juga menurun.

Nilai *precision* dihitung dengan rumus pada persamaan (2.8) dan diperoleh hasil perhitungan *precision* sebagai berikut.

$$\text{Precision} = \frac{577}{82 + 577} = \frac{577}{659} = 0,8755$$

Dari hasil *precision* yang didapatkan, maka dari hasil prediksi yang memiliki label kelas positif, sebesar 87,55% sesuai dengan data aktual yang memiliki label kelas positif. Ini menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam mengidentifikasi data yang benar-benar termasuk dalam kelas positif dari seluruh prediksi kelas positif yang dibuat. Semakin tinggi nilai *precision*, semakin kecil kemungkinan model menghasilkan prediksi *false positive*.

Nilai *recall* dihitung dengan rumus pada persamaan (2.9) dan diperoleh hasil perhitungan *precision* sebagai berikut.

$$Recall = \frac{577}{25 + 577} = \frac{577}{602} = 0,9584$$

Dari hasil *recall* yang didapatkan, maka model sangat baik dalam menangkap sebesar 95,84% data aktual dengan label kelas positif. Hal ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi data positif yang sebenarnya, dengan tingkat *false negative* yang relatif kecil.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian ini diperoleh kesimpulan sebagai berikut:

1. *User* pada *wondr by BNI* dari 4.191 data, diperoleh 3.056 data berlabel positif dan 1.135 data berlabel negatif. Kata-kata yang sering dibahas *user* diperoleh pada *wordcloud* yang memiliki ukuran besar pada kata yang ditampilkan. Pada sentimen positif kata yang sering dibahas adalah “aplikasi”, “wondr”, “bni”, “ mudah”, “bagus”, “transaksi”, “mantap”, dan “keren”. *User* yang memberikan penilaian positif banyak berpendapat bahwa *wondr by BNI* mudah digunakan dan dari segi tampilan aplikasi juga bagus. Kata-kata yang sering dibahas *user* pada sentimen negatif adalah “aplikasi”, “bni”, “masuk”, “gak”, “verifikasi wajah”, dan “transaksi”. *User* yang memberikan penilaian negatif banyak berpendapat bahwa *wondr by BNI* terkendala dalam fitur verifikasi wajah saat pendaftaran aplikasi dan juga terdapat transaksi yang gagal
2. Hasil akurasi dari analisis *Modified K-Nearest Neighbor (MK-NN)* tertinggi diperoleh pada nilai $K=2$ sebesar 87,24% dan nilai *precision* 87,55% dan nilai *recall* sebesar 95,84%. Hasil akurasi yang diperoleh menunjukkan tingkat akurasi yang baik dengan hasil prediksi data uji sebanyak 839 data yang terbagi atas: menjadi 577 data memiliki nilai aktual positif dan hasil prediksi juga menunjukkan nilai yang positif, dan 154 data memiliki nilai aktual negatif dan hasil prediksi juga menunjukkan nilai yang negatif. Sisanya 107 data belum diklasifikasikan dengan tepat, yaitu terdapat 25 data yang memiliki nilai aktual positif tetapi hasil prediksi menunjukkan nilai yang negatif, dan 82 data yang memiliki nilai aktual negatif tetapi hasil prediksi menunjukkan nilai yang positif.

5.2 Saran

Dari penelitian yang dilakukan untuk mengklasifikasikan penilaian aplikasi *wondr by BNI* masih terdapat kelemahan yang berdampak terhadap hasil penelitian ini. Sehingga, saran yang diharapkan penulis pada penelitian ini sebagai berikut:

1. Bagi pihak BNI, hasil informasi dari komentar-komentar *user* yang berbentuk negatif dapat dijadikan bahan evaluasi dalam peningkatan kepuasan pengguna dan memberikan pelayanan semaksimal mungkin, serta untuk pengembangan pembaruan aplikasi *wondr by BNI*.
2. Bagi peneliti selanjutnya, dapat menggunakan pendekatan *machine learning* lain sebagai pembanding performa algoritma *Modified K-Nearest Neighbor (MK-NN)* untuk mengklasifikasikan penilaian *wondr by BNI*.

DAFTAR PUSTAKA

- A. Prasetyo, Rino, Indriati, P. Adikara, P. (2018). Klasifikasi Hoax Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Modified K-Nearest Neighbor. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 7466–7473.
- Amalia, P., & Hastriana, A. Z. (2022). Pengaruh Kemanfaatan, Kemudahan Keamanan, dan Fitur M-Banking terhadap Kepuasan Nasabah dalam Bertransaksi pada Bank Syariah Indonesia (Studi Kasus BSI KCP Sumenep). *Islamic Sciences, Sumenep*, 1, 70–89.
- Amelia, S. (2022). Penerapan Metode Modified K-Nearest Neighbor pada Pengklasifikasian Status Pembayaran Kredit Barang Elektronik dan Furniture. *STATISTIKA Journal of Theoretical Statistics and Its Applications*, 22(1), 95–104. <https://doi.org/10.29313/statistika.v22i1.345>
- APJII Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang. (2024). <https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>
- Arafat, M. D. (2021). *Klasifikasi Penyakit dengan Gejala Demam pada Anak Menggunakan Algoritma Genetic Modified K-Nearest Neighbor* [UPN “Veteran” Yogyakarta]. <http://eprints.upnyk.ac.id/25653/>
- Arizal, I. H. (2023). *Implementasi Metode Multinomial Naive Bayes Untuk Pengklasifikasian Ulasan Aplikasi Traveloka* [Universitas Muhammadiyah Semarang]. <http://repository.unimus.ac.id/7116/>
- Brahimi Belgacem, M. T. A. T. (2021). *Improving sentiment analysis in Arabic: A combined approach*. 33(10). <https://www.sciencedirect.com/science/article/pii/S1319157819303283>
- Dewi Ayu Kusuma, S. (2022). Analisis Sentimen Ekspedisi Sicepat Dari Ulasan Google Play Menggunakan Algoritma Naïve Bayes. *Jurnal Teknik Informatika Dan Sistem Informasi*, 9(2), 796–805. <https://eprints.unisbank.ac.id/id/eprint/9255>
- Erdiansyah, U., Irmansyah Lubis, A., & Erwansyah, K. (2022). Komparasi Metode

- K-Nearest Neighbor dan Random Forest Dalam Prediksi Akurasi Klasifikasi Pengobatan Penyakit Kutil. *Jurnal Media Informatika Budidarma*, 6(1), 208. <https://doi.org/10.30865/mib.v6i1.3373>
- Feldman, R., & Sanger, J. (2012). *The Text Mining Handbook: Advanced Approaches In Analyzing Unstructured Data*. Cambridge University Press.
- Fitria, A., Munawar, A., & Pratama, P. P. (2021). Pengaruh Penggunaan Internet Banking, Mobile Banking Dan SMS Banking Terhadap Kepuasan Nasabah Bank BNI. *Jurnal Informatika Kesatuan*, 1(1), 43–52. <https://doi.org/10.37641/jikes.v1i1.406>
- Halim, A., Yusra, Y., Fikry, M., Irsyad, M., & Budianita, E. (2023). Klasifikasi Sentimen Masyarakat Di Twitter Terhadap Prabowo Subianto Sebagai Bakal Calon Presiden 2024 Menggunakan M-KNN. *Journal of Information System Research (JOSH)*, 5(1), 202–212. <https://doi.org/10.47065/josh.v5i1.4054>
- Han Jiawei, Jian Pei, M. K. (2006). *Data Mining : Concepts and Techniques Second Edition*. Morgan Kauffman.
- Hidayat T.F.T, Garno, G, R. A. . (2021). Analisis Sentimen Opini Pemindahan Ibu Kota Pada Twitter Dengan Metode Support Vector Machine. *Jurnal Ilmu Komputer*.
- Ikhsan, R. M. N., & Adytia, P. (2024). *Penerapan Algoritma K-Nearest Neighbor Pada Analisis Sentimen Terhadap Ulasan Aplikasi MYBCA di Google Play Store*. <https://repository.wicida.ac.id/5591/>
- Isnain, A. R., Supriyanto, J., & Kharisma, M. P. (2021). Implementation of K-Nearest Neighbor (K-NN) Algorithm For Public Sentiment Analysis of Online Learning. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 15(2), 121. <https://doi.org/10.22146/ijccs.65176>
- Luqyana, W. A. (2018). *Analisis Sentimen Cyberbullying Pada Komentar Instagram Dengan Metode Klasifikasi Support Vector Machine*. Universitas Brawijaya.
- Maharani, D., Wijoyo, S. H., & Wicaksono, S. A. (2023). *Klasifikasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Modified K-Nearest Neighbor (Studi Pada S1 Pendidikan Teknologi Informasi Fakultas Ilmu Komputer)*.

$I(1)$.

- Maulana, R., Voutama, A., & Ridwan, T. (2023). Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC. *Jurnal Teknologi Terpadu*, 9(1), 42–48. <https://doi.org/10.54914/jtt.v9i1.609>
- Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). Data Mining In Agriculture. In *Springer* (Vol. 1). Springer. https://doi.org/10.1007/978-0-387-39940-9_561
- Nguyen, H., Veluchamy, A., Diop, M., Al Khaimah Academy, R., Diop, M. L., Iqbal, R., Towers, J., Hisn Rd, A., & Al Khaimah, R. (2018). Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches. *SMU Data Science Review*, 1(4), 1–22. <https://scholar.smu.edu/datasciencereview> Available at: <https://scholar.smu.edu/datasciencereview/vol1/iss4/7http://digitalrepository.smu.edu>
- Nugroho, G. A. P. (2016). *Analisis Sentimen Data Twitter Menggunakan K-Means Clustering*. Universitas Sanata Dharma.
- Paramitha, A. A., Indriati, & Sari, Y. A. (2020). Analisis Sentimen Terhadap Ulasan Pengguna MRT Jakarta Menggunakan Information Gain dan Modified K-Nearest Neighbor. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(4), 1125–1132.
- Praptiwi, D. Y. (2018). *Analisis Sentimen Online Review Pengguna E-commerce Menggunakan Metode Support Vector Machine dan Maximum Entropy*. Universitas Islam indonesia.
- Purnaramadhan, R. (2021). Recommendation System Model Untuk Merekomendasikan Produk Pada Website Menggunakan Metode Content-Based. In *Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia*. Univeristas Islam Indonesia.
- Rizki, M. A. (2022). *Klasifikasi Hoax Terhadap Berita Cryptocurrency Pada Sosial Media Twitter Dengan Metode Modified K- Nearest Neighbor (MK-NN)*. Institut Teknologi Sumatera.
- Romadhon, M. R., & Kurniawan, F. (2021). A Comparison of Naive Bayes

- Methods, Logistic Regression and KNN for Predicting Healing of Covid-19 Patients in Indonesia. *3rd 2021 East Indonesia Conference on Computer and Information Technology*, 41–44.
- Safitri, R., & Miftah Andriansyah. (2020). Analisis Penerimaan Teknologi Keuangan (Fintech) Terhadap Penggunaan Aplikasi Fintech Ovo. *Jurnal Mitra Manajemen*, 4(4), 538–549. <https://doi.org/10.52160/ejmm.v4i4.369>
- Saputro, I. ., & Sari, B. W. (2020). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa. *Creative Information Technology Journal*, 6(1), 1. <https://doi.org/https://doi.org/10.24076/citec.2019v6i1.178>
- Sofiah, Yusra, Muhammad Fikry, & Lola Oktavia. (2023). Klasifikasi Sentimen Masyarakat di Twitter terhadap Kenaikan Harga Bahan Bakar Minyak dengan Metode Modified K-Nearest Neighbor. *SATIN - Sains Dan Teknologi Informasi*, 9(1), 137–148. <https://doi.org/10.33372/stn.v9i1.988>
- Turban Efraim, Jay E. Aronson, T.-P. L. (2005). *Decision Support Systems and Intelligent Systems*. Prentice Hall.
- Zafikri, A. (2008). *Implementasi Metode Term Frequency Inverse Document Frequency (TF-IDF) Pada Sistem Temu Kembali Informasi*. Univeristas Sumatera Utara.

LAMPIRAN

Lampiran 1. Data penelitian

No.	Komentar	Label
1	Mantap	Positif
2	Woow..suka bangetðŸ˜• wondr by BNI mantap & kerennðŸ• sangat simpel dan transaksi apapun makin praktisðŸ—wondrfull#jadiinMaumu	Positif
3	Bayar QRIS di wondr cepat banget,cuman 3 step selesai	Positif
4	Transaksiku lebih gampang pakai aplikasi dari BNI wondr	Positif
5	Sangat membantu sekali	Positif
6	sangat mudah untuk di pahami ..	Positif
7	Muda di gunakan dan pliturnya bagus	Positif
8	ðŸ‘ðŸ‘ðŸ‘	Positif
9	Pembayaran melalui QRIS sangat cepat dan simpelðŸ•	Positif
10	Wahh,, dengan adanya aplikasi wondr by Bni semua makin mudahhhh... terimakasih BNI	Positif
11	Wahh,, dengan adanya aplikasi wondr by Bni semua makin mudahhhh... terimakasih BNI	Positif
12	tolong di perbaiki, tidak bisa tf ke dana	Negatif
13	Tampilan lebih fresh, menarik, satset scan qris dimanapun dan banyakk promo menarik ðŸ¥°	Positif
14	Tampilannya keren,promonya bejibun, kalo lupa pin kartu tdk perlu ke bank lagi...mantap	Positif
15	Sangat membantu transaksi dengan qris 3 step slesai.. cepet dn mudah	Positif
⋮	⋮	⋮
4264	The best	Positif

Lampiran 2. Syntax input data

```
#import data
from google.colab import files
uploaded = files.upload()

#menampilkan data yang telah di import
df = pd.read_csv('scrapped_data.csv')
df.head()

#menghitung berapa banyak data
print(len(df))
```

Lampiran 3. *Syntax Pre-processing data*

#1. cleaning data untuk menghapus emoji, url, tagar, tanda baca, dan spasi berulang

```
import re
def clean_text(text):
    #Remove emojis
    emoji_pattern = re.compile("[
        u"\U0001F600-\U0001F64F"
        u"\U0001F300-\U0001F5FF"
        u"\U0001F680-\U0001F6FF"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        "]" + "", flags=re.UNICODE)
    text = emoji_pattern.sub(r'', text)

    #Remove hashtags (words starting with #)
    text = re.sub(r'#\w+', '', text)

    #Remove URLs (http/https links)
    text = re.sub(r'http\S+|www\S+', '', text)

    #remove punctuation (. , ,, ?, !)
    text = text.translate(str.maketrans('', '',
string.punctuation))

    #remove extra space
    text = text.strip()

    return text

df['cleaned_text'] = df['content'].apply(clean_text)
df.head()
```

#2. mengubah cleaned_text menjadi lower case/ case folding

```
df['cleaned_text'] = df['cleaned_text'].str.lower()
df.head()
```

#3. melakukan tokenizing

```
df['tokenized_text'] = df['cleaned_text'].apply(word_tokenize)
df.head()
```

```

#4. stopwords removal bahasa indonesia

indonesian_stopwords = stopwords.words('indonesian')

def remove_stopwords(tokens):
    filtered_tokens = [token for token in tokens if token not
in indonesian_stopwords]
    return filtered_tokens

df['no_stopword_text'] =
df['tokenized_text'].apply(remove_stopwords)
df.head()

#5. stemming

!pip install Sastrawi
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory, StopWordRemover, ArrayDictionary
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

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

def stem_text(tokens):
    stemmed_tokens = [stemmer.stem(token) for token in tokens]
    return stemmed_tokens

df['stemmed_text'] = df['no_stopword_text'].apply(stem_text)
df['stemmed_text'] = df['stemmed_text'].apply(' '.join)
df['tokenized_text'] = df['stemmed_text'].apply(word_tokenize)
df['tokenized_text'].to_csv('preprocessingdata.csv',
index=False)
df.head()

# menghapus data kosong

df = df[df['tokenized_text'].apply(lambda x: isinstance(x,
list) and len(x) > 0)]
df.head()

# menghitung berapa banyak data yang tersisa

remaining_data_count = len(df)
print(f"Jumlah data yang tersisa: {remaining_data_count}")

```


Lampiran 4. *Syntax* Pelabelan data

```
#labelling
def label_sentimen(score):
    if score <= 3:
        return 'Negatif'
    else:
        return 'Positif'

df.loc[:, 'sentimen'] = df['score'].apply(label_sentimen)
df

# Menghitung jumlah sentimen negatif dan positif
sentimen_counts = df['sentimen'].value_counts()

# Menampilkan hasil
print(sentimen_counts)

# menyajikan hasil pelabelan dengan bar chart

import matplotlib.pyplot as plt

# Menghitung jumlah sentimen positif dan negatif
sentimen_counts = df['sentimen'].value_counts()

# Membuat bar chart
plt.figure(figsize=(8, 6))
plt.bar(sentimen_counts.index, sentimen_counts.values,
        color=['blue', 'red'])
plt.title('Jumlah Sentimen Positif dan Negatif')
plt.xlabel('Sentimen')
plt.ylabel('Jumlah')
plt.show()

# coding sentimen positif = 1 dan negatif = 0

def encode_sentimen(sentimen):
    if sentimen == 'Positif':
        return 1
    else:
        return 0

df['sentimen_encoded'] = df['sentimen'].apply(encode_sentimen)
df
```

Lampiran 5. Syntax wordcloud

```
#1. Wordcloud untuk sentimen positif

from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Filtering data untuk mengambil sentimen positif
positive_df = df[df['sentimen'] == 'Positif']

# Gabungkan semua stemmed_text untuk sentimen positif ke dalam
satu string
text_positive = ' '.join(positive_df['stemmed_text'].tolist())

# Membentuk wordcloud
wordcloud_positive = WordCloud(width=800, height=400,
background_color='white').generate(text_positive)

# Menyajikan word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_positive, interpolation='bilinear')
plt.axis('off')
plt.title('Wordcloud sentimen positif')
plt.show()
```

```
#2. Wordcloud untuk sentimen negatif

import matplotlib.pyplot as plt
# Filtering data untuk mengambil sentimen negatif
negative_df = df[df['sentimen'] == 'Negatif']

# Gabungkan semua stemmed_text untuk sentimen positif ke dalam
satu string
text_negative = ' '.join(negative_df['stemmed_text'].tolist())

# Membentuk wordcloud
wordcloud_negative = WordCloud(width=800, height=400,
background_color='white').generate(text_negative)

# Menyajikan wordcloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_negative, interpolation='bilinear')
plt.axis('off')
plt.title('Wordcloud sentimen negatif')
plt.show()
```

Lampiran 6. *Syntax* pembobotan TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(df['stemmed_text'])

print(tfidf_matrix)
```

Lampiran 7. *Output* Pembobotan TF-IDF

```
(3, 614)      0.3069263330717489
(3, 3618)     0.34388417247918895
(3, 3854)     0.34757701852352746
(3, 1248)     0.5839675626732005
(3, 2656)     0.49733736356374353
(3, 256)      0.27994204887568286
:
(4186, 2280)  0.2950356305730202
(4186, 1973)  0.2688605437723009
(4186, 3905)  0.241124033516752
(4186, 150)   0.35584989737236844
(4186, 3819)  0.34549886420170267
(4186, 927)   0.432644338088754
(4186, 742)   0.48734225067900927
(4187, 1324)  1.0
```

Lampiran 8. Membagi data latih dan data uji

```
#membagi data training 80% dan data testing 20%

from sklearn.model_selection import train_test_split

# Memisahkan fitur (X) dan label (y)
X = tfidf_matrix
y = df['sentimen_encoded']

# Membagi data menjadi data training dan data testing dengan
rasio 80:20
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Menampilkan ukuran data training dan data testing
print("Ukuran data training:", X_train.shape)
print("Ukuran data testing:", X_test.shape)
```

Lampiran 9. Syntax menghitung nilai *Cosine Similarity*

```
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity

# Menghitung cosine similarity antara semua dokumen dalam data
latih (TF-IDF matrix atau data X_train)
cosine_sim_matrix = cosine_similarity(X_train)

# Menampilkan matriks cosine similarity
print("Matriks Cosine Similarity:")
print(cosine_sim_matrix)

# Konversi matriks cosine similarity ke DataFrame untuk
menyimpan ke file CSV
cosine_sim_df = pd.DataFrame(cosine_sim_matrix)
```

Lampiran 10. Output *Cosine Similarity*

```
Matriks Cosine Similarity:
[[1.         0.         0.04196907 ... 0.1924415  0.         0.06489819]
 [0.         1.         0.         ... 0.         0.         0.         ]
 [0.04196907 0.         1.         ... 0.08167635 0.         0.0626415 ]
 ...
 [0.1924415  0.         0.08167635 ... 1.         0.         0.1262989 ]
 [0.         0.         0.         ... 0.         1.         0.         ]
 [0.06489819 0.         0.0626415  ... 0.1262989  0.         1.         ]]
```

Lampiran 11. Syntax mengurutkan nilai *Cosine Similarity*

```
import numpy as np
import pandas as pd

# Mengurutkan nilai cosine similarity dari terkecil hingga
terbesar
sorted_cosine_sim = np.sort(cosine_sim_matrix, axis=None)

# Menampilkan nilai cosine similarity yang telah diurutkan
print("Nilai Cosine Similarity yang Diurutkan:")
print(sorted_cosine_sim)
```

Lampiran 12. Syntax Menghitung nilai validitas antar data latih (K=2)

```
# menghitung nilai validitas

import numpy as np
```

```
def calculate_validity(cosine_sim_matrix, y_train):

    K=2
    validity_values = []
    for i in range(cosine_sim_matrix.shape[0]):
        neighbor_indices = np.argsort(cosine_sim_matrix[i])[:-1][1:3]

        validity = 0
        for neighbor_index in neighbor_indices:
            if y_train.iloc[neighbor_index] == y_train.iloc[i]:
                validity += 1

        validity_values.append((1/K) * validity)

    return validity_values

# Menghitung nilai Validitas untuk data latih
validity_values = calculate_validity(cosine_sim_matrix,
y_train)

# Menyajikan nilai Validitas untul 10 data pertama
print("Validity Values (first 10 data points):",
validity_values[:10])
```

Lampiran 13. *Output* nilai validitas antar data latih (K=2)

Validity Values (first 10 data points): [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.5]

Lampiran 14. *Syntax* menghitung jarak *cosine distance* antara data latih dengan data uji

```
#Menghitung jarak dokumen data latih dan data uji menggunakan
cosine distance

import pandas as pd
from sklearn.metrics.pairwise import cosine_distances

# Menghitung cosine distance antara data latih (X_train) dan
data uji (X_test)
cosine_dist_matrix = cosine_distances(X_train, X_test)

# Menampilkan matriks cosine distance
print("Matriks Cosine Distance:")
print(cosine_dist_matrix)
```

```
# Konversi matriks cosine distance ke DataFrame untuk
menyimpan ke file CSV
cosine_dist_df = pd.DataFrame(cosine_dist_matrix)
```

Lampiran 15. Output jarak cosine distance antara data latih dan data uji

```
Matriks Cosine Distance:
[[0.94343675 1.          1.          ... 1.          0.944364    0.96397327]
 [1.          1.          1.          ... 1.          1.          1.          ]
 [1.          1.          1.          ... 1.          0.98502719 0.9871361 ]
 ...
 [1.          1.          1.          ... 1.          0.9623464 1.          ]
 [1.          1.          1.          ... 1.          1.          1.          ]
 [1.          1.          0.88103006 ... 1.          1.          1.          ]]
```

Lampiran 16. Syntax menghitung nilai weight voting

```
# menghitung nilai weight voting antara data latih dan data
uji

import pandas as pd
import numpy as np
weight_voting_matrix = np.zeros((X_train.shape[0],
X_test.shape[0]))
validitas_df = pd.DataFrame({'validitas': validity_values}) #
Create validitas_df from validity_values

for j in range(X_test.shape[0]):
    for i in range(X_train.shape[0]):
        weight_voting_matrix[i, j] =
validitas_df['validitas'].iloc[i] * (1 /
(cosine_dist_matrix[i, j] + 0.5))

print("Matriks Weight Voting:")
print(weight_voting_matrix)
```

Lampiran 17. Output perhitungan nilai weight voting

```
Matriks Weight Voting:
[[0.69279101 0.66666667 0.66666667 ... 0.66666667 0.69234625 0.68307258]
 [0.66666667 0.66666667 0.66666667 ... 0.66666667 0.66666667 0.66666667]
 [0.66666667 0.66666667 0.66666667 ... 0.66666667 0.67338834 0.67243341]
 ...
 [0.66666667 0.66666667 0.66666667 ... 0.66666667 0.6838325 0.66666667]
 [0.66666667 0.66666667 0.66666667 ... 0.66666667 0.66666667 0.66666667]
 [0.66666667 0.66666667 0.7240972 ... 0.66666667 0.66666667 0.66666667]]
```

Lampiran 18. *Syntax* untuk menentukan label dari data uji

```
# menentukan sentimen dari setiap data uji menggunakan nilai
weight voting terbesar

import pandas as pd
import numpy as np

predicted_sentiments = []

for j in range(X_test.shape[0]):
    current_test_weights = weight_voting_matrix[:, j]
    max_weight_index = np.argmax(current_test_weights)
    predicted_sentiment = y_train.iloc[max_weight_index]
    predicted_sentiments.append(predicted_sentiment)

# Konversi list hasil prediksi sentimen ke dalam DataFrame
predicted_sentiments_df = pd.DataFrame({'predicted_sentiment':
predicted_sentiments})
```

Lampiran 19. *Syntax* untuk *confusion matrix*

```
# confusion matrix

from sklearn.metrics import confusion_matrix

# Assuming y_test contains the true labels and
predicted_sentiments contains the predicted labels
cm = confusion_matrix(y_test, predicted_sentiments)

print("Confusion Matrix:")
print(cm)
```

Lampiran 20. *Output confusion matrix*

```
Confusion Matrix:
[[155  82]
 [ 25 577]]
```

Lampiran 21. *Syntax* untuk akurasi, *precision*, dan *recall*

```
# precession, recall, and accuracy

from sklearn.metrics import precision_score, recall_score,
accuracy_score

# Calculate precision
```

```
precision = precision_score(y_test, predicted_sentiments)

# Calculate recall
recall = recall_score(y_test, predicted_sentiments)

# Calculate accuracy
accuracy = accuracy_score(y_test, predicted_sentiments)

# Print the results
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"Accuracy: {accuracy}")
```

Lampiran 21. *Output* perhitungan akurasi, *precision*, dan *recall*

```
Precision: 0.8755690440060698
Recall: 0.9584717607973422
Accuracy: 0.8724672228843862
```