ANALISIS SENTIMEN BERITA COVID-19 PADA PORTAL BERITA DETIK.COM MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBOR*

SKRIPSI

Disusun Sebagai Syarat Memperoleh Gelar Sarjana Teknik
Prodi Teknik Informatika Fakultas Teknik
Universitas Malikussaleh

DISUSUN OLEH:

NAMA : RINI KHOLISYAH
NIM : 180170027
PRODI : TEKNIK INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MALIKUSSALEH
2024

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah puji syukur kehadirat Allah SWT. yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini. Shalawat dan salam kepada Nabi Muhammad SAW. yang menjadi panutan kita sepanjang masa.

Penghargaan dan terimakasih yang tak ternilai kepada orang tua tercinta atas segala pengorbanan, jerih payah, do'a, serta nasehat yang telah diberikan, yang sangat penulis sayangi yaitu Ayahanda tercinta Sakino, S.Pd.,SD dan Ibunda tersayang Salamah semoga Allah senantiasa melindungi serta memberikan keselamatan.

Alhamdulillah penulis dapat menyelesaikan tugas akhir ini dengan judul **"Analisis Sentimen Berita Covid-19 pada Portal Berita Detik.com Menggunakan Algoritma K-Nearest Neighbor"**. Banyak ilmu serta pengalaman baru dan berharga penulis peroleh dari kegiatan penelitian ini. Oleh karena itu, penulis ucapkan terimakasih banyak atas segala bantuan dan dukungan sehingga kegiatan penelitian ini berjalan dengan lancer. Maka dari itu pada kesempatan ini penulis ingin menyampaikan terima kasih banyak kepada:

1. Bapak Prof. Dr. Ir. Herman Fitrah, M.T., ASEAN Eng., selaku Rektor Universitas Malikussaleh.
2. Bapak Dr. Muhammad Daud, S.T., M.T selaku Dekan Fakultas Teknik.
3. Bapak Munirul Ula, S.T., M.Eng., Ph.D selaku Ketua Jurusan Informatika.
4. Ibu Zara Yunizar, S.Kom., M.Kom selaku Ketua Prodi Teknik Informatika.
5. Bapak Rizal, S.Si., M.IT selaku Dosen Pembimbing Utama dan Ibu Lidya Rosnita, S.T., M.Kom selaku Dosen Pembimbing Pendamping, yang selama ini telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan memberikan masukan kepada penulis dalam mengerjakan tugas akhir ini hingga selesai.
6. Bapak Wahyu Fuadi, S.T., M.IT selaku Dosen Penguji I dan Bapak Mukti Qamal, S.T., M.IT selaku Dosen Penguji II, yang telah memberikan

masukan yang sangat bermanfaat dalam proses menyelesaikan penelitian tugas akhir ini.

7. Bapak dan ibu dosen serta staf akademik yang telah membantu penulis selama menjalankan perkuliahan di Program Studi Teknik Informatika Universitas Malikussaleh.
8. Teman-teman seperjuangan Angkatan 2018 Teknik Informatika Universitas Malikussaleh yang telah banyak memberikan dukungan dan semangat.
9. Semua pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari sepenuhnya bahwa dalam penulisan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan pada masa yang akan dating. Semoga Tugas Akhir ini memberikan informasi dan bermanfaat untuk pengembangan wawasan dan peningkatan ilmu pengetahuan bagi kita semua.

Akhir kata semoga Tugas Akhir ini dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya. Mohon maaf atas segala kekhilafan. Semoga rahmat dan hidayah serla linfungan Allah SWT. senantiasa dilimpahkan kepada kita semua. Aamiin ya Robbal'alamin.

Wassalamu'alaikum Wr. Wb.

Lhokseumawe, 01 Februari 2024

Penulis,

Rini Kholisyah

NIM. 180170027

ABSTRAK

Laju perkembangan jumlah kasus Covid-19 di negara Indonesia dalam beberapa tahun belakangan mengalami peningkatan, dengan adanya peningkatan kasus ini membuat keresahan di tengah-tengah masyarakat. Peningkatan kasus ini juga menyebabkan banyak sekali berita-berita yang memuat tentang Covid-19, salah satu portal berita yang banyak memuat pemberitaan terkait Covid-19 adalah detik.com. Oleh karena sebab tersebut, perlu dikembangkan suatu sistem yang melakukan permodelan terhadap analisis sentimen guna mengklasifikasikan berita tekait Covid-19 pada portal berita detik.com menjadi tiga kelas yaitu kelas berita positif, kelas berita netral, dan kelas berita negatif. Sistem yang dibangun menggunakan algoritma TF-IDF yang dapat digunakan untuk menghitung nilai bobot dari setiap katayang ada pada berita, serta algoritma K-NN untuk mengklasifikasikan berita Covid-19 terhadap tiga kategori kelas yakni kelas positif, netral, dan negatif dengan jumlah berita *testing* sebanyak 50 data berita Covid-19. Penelitian menggunakan jumlah data latih sebanyak 450 data berita ini menghasilkan nilai akurasi 74%, presisi 68,38%, *recall* 69,07%, serta f1-*score* sebesar 67,58% dengan prediksi sistem berita pada portal berita detik.com lebih cenderung ke berita bersifat negatif sebanyak 26 berita, sedangkan berita bersifat netral sebanyak 12 berita, dan berita bersifat positif sebanyak 12 berita.

Kata kunci: Sentimen, Berita, Covid-19, K-NN

ABSTRACT

The rate of COVID-19 cases in Indonesia has experienced a significant increase in recent years, causing concern among the population. This surge in cases has also led to a proliferation of news articles related to COVID-19, with one prominent news portal being detik.com. Due to these circumstances, there is a need to develop a system for sentiment analysis that can classify COVID-19 news articles on detik.com into three categories: positive, neutral, and negative. The system employs the TF-IDF algorithm to calculate the weight of each word in the news articles and the K-NN algorithm to classify them into the three aforementioned categories. The testing dataset consists of 50 COVID-19 news articles. This research achieved the highest accuracy when using a training dataset of 450 news articles, resulting in an accuracy rate of 74%. The precision value was 72.22%, recall was 73.43%, and the F1-score was 71.31%. The system's predictions for news articles on detik.com leaned more towards negative sentiment, with 24 articles classified as negative, 14 as neutral, and 12 as positive.

Keywords: Sentiment; News; Covid-19; K-NN

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan penyebaran informasi di internet selalu meningkat dari waktu ke waktu. Teknologi informasi adalah salah satu hal yang tidak akan lepas dari kehidupan manusia. Manusia akan kesusahan dalam berkomunikasi dan menyampaikan informasi tanpa adanya teknologi. Berita merupakan media informasi yang juga turut mengalami peningkatan. Pada awalnya banyak lembaga penyaluran informasi menyampaikan berita melalui media cetak koran beralih ke media elektronik seperti radio dan televisi. Seiring berjalannya waktu, berita memiliki media penyebaran baru yang akan mendukung mobilitas manusia yang semakin tinggi, yaitu media digital menggunakan sistem berbasis *web* secara *update*. Berdasarkan laporan DataReportal, per Januari 2022, pengguna internet di Indonesia mencapai 204,7 juta pengguna. Jumlah ini mencakup 73,7% dari total populasi Indonesia. Jumlah pengguna ini meningkat tipis 1,03% dibandingkan dengan tahun sebelumnya pada Januari 2021 yaitu tercatat sebanyak 202,6 juta pengguna (DataReportal, 2022).

Salah satu media digital yang menggunakan sistem berbasis *web* adalah portal berita detik.com. Detik.com adalah sebuah portal *web* yang berisikan artikel dan berita daring di Indonesia yang dapat diakses pada alamat URL www.detik.com. Berita dan artikel yang disampaikan terdiri dari beberapa kategori seperti kesehatan, politik, olahraga, teknogi, dan masih banyak lagi kategori berita lainnya. Beberapa tahun belakangan ini, banyak berita yang dicari oleh masyarakat dengan kategori kesehatan terutama berita mengenai pandemi Covid-19 yang sempat menggemparkan Indonesia hingga dunia belakangan ini.

Berita mengenai Covid-19 (*Corona Virus Desaese 19*) beberapa tahun belakangan ini sangat ramai diperbincangkan dari awal kehadirannya di Wuhan, China hingga menyebar ke seluruh dunia termasuk Indonesia dan menelan banyak korban jiwa. Berdasarkan data dari laman covid19.go.id, dari awal ditemukan

hingga pada 01 April 2022 Covid-19 telah memakan korban jiwa di Indonesia hingga mencapai 155.164 orang meninggal (Covid19.go.id, 2022). Berita yang mengangkat tema Covid-19 pun banyak tersebar lewat berbagai *platform* media, dari media massa hingga media sosial. Hal tersebut memudahkan masyarakat untuk mendapatkan informasi tentang virus tersebut. Namun, dengan banyaknya berita yang beredar, banyak oknum-oknum tidak bertanggung jawab yang membuat dan menyebarkan berita-berita *hoax* terutama di media sosial, sehingga membuat kebanyakan orang semakin merasa was-was dan khawatir hingga menimbulkan perasaan cemas dan panik akan virus ini, yang mana hal tersebut tentu dapat mempengaruhi imun tubuh dari orang tersebut. Oleh karena itu, dibutuhkan model *sentiment analysis* untuk mengklasifikasikan berita Covid-19 pada portal berita detik.com menjadi data sentimen positif, netral, dan negatif. Berita dianggap positif apabila terdapat kalimat-kalimat bersifat positif, kalimat dianggap positif apabila terdapat kata-kata yang mengandung makna positif. Berita dianggap netral apabila terdapat kalimat-kalimat bersifat netral, kalimat dianggap netral apabila terdapat kata-kata yang mengandung makna netral. Berita dianggap negatif apabila terdapat kalimat-kalimat bersifat negatif, kalimat dianggap negatif apabila terdapat kata-kata yang mengandung makna negatif.

Sentiment analysis atau *opinion mining* merupakan proses mengemukakan informasi dengan mengklasifikasi dokumen teks ke dalam beberapa kelompok yang sesuai dengan keseluruhan sentimen yang diterangkan di dalam setiap dokumen tersebut (P. Setiawan, 2018). Pengimplementasian sistem *sentiment analysis* dapat digunakan untuk menganalisis sentimen dari data teks salah satunya yaitu teks berita. Isi berita yang dipublikasi dapat memunculkan opini berita positif, negatif, maupun netral terhadap suatu hal yang sedang dibahas oleh masyarakat. Terutama berita tentang Covid-19 yang sangat berpengaruh pada kesehatan mental dan fisik seseorang. Beberapa algoritma yang dapat digunakan untuk mengklasifikasi berita Covid-19 pada portal berita detik.com, salah satunya adalah algoritma *K-Nearest Neighbor*.

Algoritma *K-Nearest Neighbor* (K-NN) adalah salah satu algoritma yang paling sering kali digunakan untuk klasifikasi. Algoritma K-NN adalah salah satu

metode yang menerapkan algoritma *supervised*. Akurasi algoritma K-NN ditentukan oleh ada dan tidaknya data yang tidak relevan atau jika bobot fitur tersebut sebanding dengan relevansinya terhadap klasifikasi. Kelebihan dari algoritma K-NN ini yaitu efektif saat dipakai untuk data dengan jumlah yang besar dan sanggup menghasilkan data yang cukup kuat dan jelas (Fairuz, 2020). Oleh karena itu, algorita K-NN ini sangat cocok digunakan untuk mengklasifikasi berita, karena berita lebih banyak memuat kata yang lebih banyak dibandingkan dengan satu postingan di *Facebook* dan sebuah *Twitter*.

Penelitian lain yang juga menggunakan metode yang sama juga pernah dilakukan oleh Ar Razi dengan judul "Klasifikasi Penerimaan Beasiswa Aceh Carong (Aceh Pintar) di Universitas Malikussaleh Menggunakan Algoritma KNN (*K-Nearest Neighbors*)", dengan hasil algoritma K-Nearest Neighbor cukup efektif dan efisien dalam mengklasifikasikan penerima beasiswa Aceh Carong dengan hasil 82,00% (Razi, 2022).

Penelitian serupa juga pernah dikakukan oleh Faisal Briliansyah dengan judul penelitian "Sistem Klasifikasi Kategori Berita Menggunakan Metode *K-Nearest Neighbor*" dengan hasil presentase nilai *accuracy* sebesar 76%, *precision* 35%, *recall* 35%, f-*measure* 35%, *specificity* 84%, dan UAC 60% (Briliansyah, 2020).

Berdasarkan latar belakang di atas, maka penulis mengangkat judul "**Analisis sentimen berita Covid-19 pada portal berita detik.com menggunakan *algoritma K-Nearest Neighbor***". Adapun penelitian ini bertujuan untuk mengklasifikasikan berita Covid-19 pada portal berita detik.com menjadi kelompok positif, negatif, dan netral menggunakan metode algoritma *K-Nearest Neighbor*. Hasil dari penelitian ini akan memberikan gambaran kepada masyarakat umum apakah berita Covid-19 pada portal berita detik.com cenderung ke berita positif, negatif, atau netral.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan teknik *sentiment analysis* terhadap berita Covid-19 pada portal berita detik.com menggunakan metode algoritma *K-Nearest Neighbor* ?
2. Bagaimana tingkat akurasi sistem dengan metode *K-Nearest Neighbor* dalam penelitian *sentiment analysis* terhadap berita Covid-19 ?

1.3 Batasan Masalah

Agar tujuan dari penelitian ini tercapai, maka penelitian ini perlu dibatasi. Adapun batasan penelitian yang dibuat penulis yakni sebagai berikut:

1. Metode yang digunakan pada penelitian *sentiment analysis* ini adalah metode algoritma *K-Nearest Neighbor*.
2. Data sumber penelitian yang digunakan didapat dari portal berita detik.com.
3. Data yang diambil hanya berita mengenai Covid-19.
4. Data yang diambil sebanyak 450 data untuk data latih dan 50 data untuk data uji dalam kurun waktu dari tahun 2020-2021 dengan menggunakan *crawling* berdasarkan *term* Covid-19.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini yaitu sebagai berikut:

1. Menerapkan teknik *sentiment analysis* untuk mendapatkan informasi berupa klasifikasi yang dihasilkan dari setiap berita Covid-19 menggunakan metode *K-Nearest Neighbor*.
2. Melihat sejauh mana algoritma *K-Nearest Neighbor* dalam mengenali pola pada sebuah berita untuk mengetahui klasifikasi berita Covid-19.
3. Menganalisis sejauh mana tingkat klasifikasi positif, negatif, dan netral yang dihasilkan dari sebuah berita Covid-19 pada portal berita detik.com.
4. Mengetahui tingkat akurasi sistem dengan metode *K-Nearest Neighbor* dalam penelitian *sentiment analysis* terhadap berita Covid-19.

1.5 Manfaat Penelitian

Penelitian diharapkan dapat membawa manfaat. Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Untuk mengetahui sejauh mana keakuratan metode algoritma *K-Nearest Neighbor* untuk diterapkan pada penelitian sentiment analysis berita Covid-19 pada portal berita detik.com.
2. Sebagai studi pustaka pada kegiatan-kegiatan penelitian selanjutnya.
3. Diharapkan dapat menjadi suatu referensi yang berguna bagi dunia akademik untuk mengetahui sejauh mana tingkat kemampuan algoritma *K-Nearest Neighbor* dalam melakukan analisis teks pada sebuah berita.

BAB II
TINJAUAN PUSTAKA

2.1 Data Mining

Data mining adalah proses yang mempekerjakan satu atau lebih Teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis.

Data mining merupakan proses mencari pola atau informasi menarik dalam data yang terpilih dengan menggunakan metode atau teknik tertentu. Metode-metode, teknik-teknik, atau algoritma di dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat amat bergantung pada tujuan dan proses *Knowledge Discovery in Database* (KDD) secara keseluruhan (Mardi, 2017).

Knowledge discovery ataupun *pattern recognition* merupakan beberapa istilah sepadan yang dimiliki data mining. Kedua istilah tersebut sebetulnya memiliki ketepatannya masing-masing. Istilah *knowladge discovery* (penemuan pengetahuan) tepat digunakan sebab tujuan utama dari data mining memang guna mendapatkan pengetahuan yang masih tersembunyi di dalam bongkahan data. Istilah *pattern recognition* (pengenalan pola) juga cocok digunakan sebab pengetahuan yang ingin digali memang berupa pola-pola yang kemungkinan juga masih perlu digali dari dalam bongkahan yang sedang dihadapi (Susanto & Suryadi, 2010).

Ada banyak sekali fungsi yang dimiliki data mining, namun untuk fungsi utamanya sendiri adalah sebagai berikut (Susanto & Suryadi, 2010):

1. *Descriptive*, merupakan suatu fungsi guna mengetahui lebih jauh mengenai data yang tengah diamati. Dengan melakukan suatu proses ini diharapkan dapat memahami perilaku dari suatu data itu sendiri. Data itulah yang setelahnya bisa manfaatkan untuk memahami karakteristik data yang dimaksud. Dengan memanfaatkan fungsi deskripsi ini, setelahnya dapat menemukan pola-pola tertentu yang bersembunyi dalam suatu data.

2. *Predictive*, adalah suatu fungsi bagaimana suatu proses yang nantinya akan menjumpai pola-pola tertentu dari sebuah data. Pola-pola ini bisa diketahui dari bermacam variabel yang terdapat pada data. Setelah mejumpai pola pola tersebut, maka pola tersebut dapat dimanfaatkan guna memprediksi variable lain yang masih belum diketahui jenis ataupun nilainya.

Selain fungsi utama di atas, data mining juga memiliki fungsi-fungsi lainnya. Beberapa fungsi lain dari data mining diantaranya adalah, sebagai berikut:

1. *Classification and prediction*, membangun model fungsi yang membedakan dan mendeskripsikan konsep atau kelas guna memprediksi masa depan.
2. *Cluster analysis*, membentuk grup data untuk membuat kelas baru.
3. *Multidimensional concept description*, karakteristik dan diskriminasi berfungsi guna meringkas, menggenderalisasikan, membedakan karakteristik data, dll.
4. *Outlier analysis*, merupakan objek data yang mana tidak sesuai dengan sifat umum dari data, berguna untuk analisis peristiwa langka dan deteksi penipuan.

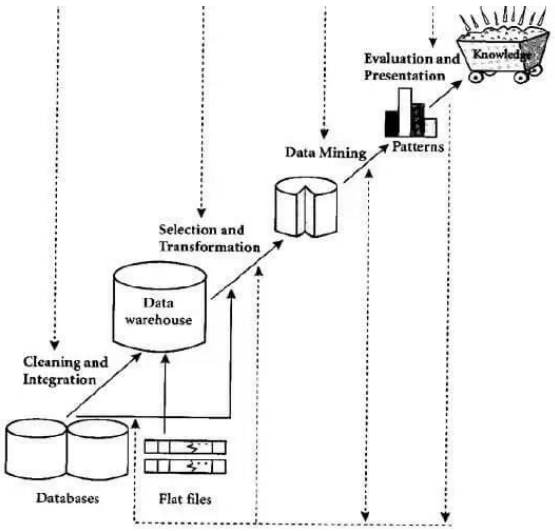
Data mining merupakan peranan yang diantaranya adalah pengumpulan, pemakaian, dan histori yang menemukan pola, hubungan, dan keteraturan dalam set data berukuran besar. Maksudnya, proses pencarian informasi yang belum diketahui sebelumnya dari sekelompok data besar. Adapun karkteristik data mining yakni, sebagai berikut (Ginting et al., 2014):

1. Data mining berkenaan dengan penemuan sesuatu yang tersembunyi serta pola data tertentu yang belum diketahui sebelumnya.
2. Data mining biasa memakai data yang sangat besar. Biasanya, data yang besar tersebut dimanfaatkan guna membuat hasil lebih dipercaya.
3. Data mining bermanfaat untuk menciptakan keputusan yang kritis terutama dalam hal strategi.

Dalam data mining, terdapat pula metode-metode untuk melakukan pengumpulan informasi. Yang mana metode itu akan membantu pada proses menemukan data. Berikut ini metode-metode yang terdapat dalam data mining:

2.1.1 Proses Pengambilan Data

Proses pengambilan data dapat dilakukan dengan *Knowledge Discovery in Database* (KDD).



Gambar 2.1 Proses *Knowledge Discovery in Database* (KDD)

Sumber: *Journal of Informatics and Technology*

Tahapan atau proses-proses tersebut diawali dengan data mentah hingga berakhir dengan informasi atau pengetahuan yang sudah diproses. Proses-proses tersebut yaitu, sebagai berikut (Anggraeni et al., 2013):

1. *Data cleansing*, merupakan tahapan dimana data-data yang mengandung *error*, tidak konsisten, dan tidak lengkap akan dibuang dari *data collection*.
2. *Data integration*, merupakan proses integrasi data dimana data yang berulang akan dikombinasikan.
3. *Data selection*, merupakan tahap pemilihan atau seleksi data yang relevan pada analisis agar diterima dari *data collection* yang ada.
4. *Data Transformation*, merupakan tahapan transformasi data yang telah dipilih ke bentuk prosedur mining dengan cara agresi data.
5. Data mining, merupakan proses terpenting yang mana akan dilakukan berbagai macam teknik yang digunakan untuk mengekstrak pola-pola potensial agar mendapatkan data yang dibutuhkan.

6. *Pattern evolution*, merupakan sebuah tahapan untuk memproses pola-pola menarik yang sebelumnya telah ditemukan dengan menggunakan identifikasi berdasarkan *measure* yang sudah diberikan.
7. *Knowledge presentation*, adalah proses dimana memanfaatkan teknik visualisasi yang bermaksud membantu user untuk mengerti serta menginterpretasikan hasil dari data mining.

2.1.2 Teknik dalam Proses Data Mining

Dalam proses data mining, terdapat berbagai bermacam-macam teknik yang dapat digunakan. Teknik-teknik yang dapat digunakan pada proses penambangan data tersebut adalah, sebagai berikut (Khomarudin, 2016):

1. *Nearest neighbor*, merupakan teknik tertua yang dipakai dalam data mining. Teknik ini merupakan Teknik yang memprediksi pengklasifikasian atau pengelompokkan.
2. *Decision tree*, adalah suatu model prediktif yang bisa dideskripsikan sebagai pohon. Setiap *node* yang ada pada struktur pohon tersebut mewakili suatau pertanyaan yang dipakai dalam penggolongan data.
3. *Clustering*, adalah teknik yang digunakan untuk mengelompokkan data berdasarkan standar tiap-tiap data.
4. *Predictive modeling*, merupakan model yang memiliki dua teknik, yakni *value prediction* dan *classification*.
5. *Database segmentation*, merupakan teknik yang melakukan partisi *database* menjadi beberapa segmen, klaster, atau *record*.
6. *Link analysis*, merupakan suatu teknik yang digunakan untuk membuat hubungan antar *record* yang individu ataupun sekumpulan *record* pada *database*.
7. *Deviation detection*, merupakan suatu teknik yang dipakai untuk mengidentifikasi *outlier* yang mengungkapkan suatu penyimpangan dari ekspektasi yang telah diketahui sebelumnya.

2.2 *Sentiment Analysis*

Sentiment analysis atau *opinion mining* merupakan sebuah tugas mencari pendapat mengenai suatu entitas tertentu untuk menentukan apakah sebuah teks termasuk kalimat opini atau bukan. *Sentiment analysis* bertujuan guna mengetahui perilaku penulis atau pembicara berhubungan dengan beberapa polaritas kontekstual atau topik keseluruhan dokumen. Sikap mungkinnya evaluasi atau penilaian, keadaan efektif (yakni, keadaan penulis saat menulis), atau komunikasi emosional (yakni, efek emosional si penulis yang hendak ditanamkan kepada pembaca) (K. Y. Setiawan et al., 2014). Cara kerja analisis sentimen dalam mendapatkan data bisa dibagi menjadi 3 langkah yakni sebagai berikut:

1. Klasifikasi, mesin harus mengelompokkan data yang dianggap sebagai opini dari suatu teks. Terdapat tiga klasifikasi dalam metode *sentiment analysis* yang dapat dikerjakan, yaitu *machine learning*, *lexicon-based*, dan campuran. *Mechine learning* memiliki fitur-fitur yang dapat mengenali sentiment dalam suatu teks. *Lexicon-based* memakai berbagai suku kata yang dinilai menggunakan skor polaritas guna mengetahui pendapat masyarakat tentang sebuah topik. Klasifikasi campuran menggabungkan *machine learning* dan *lexicon*, walaupun jarang dipakai namun metode ini biasanya mengeluarkan hasil yang lebih menjanjikan.
2. Evaluasi, proses ini mengikut sertakan pengukuran rata-rata makro, mikro, dan skor F_1 tertimbang guna mengatur data yang masuk ke dua klasifikasi atau lebih. Metrik yang dipakai berdasarkan pada keseimbangan pengelompokkan data set. Skema yang digunakan, yakni tinjauan data set, *pre-processing*, *tokenizer*, penghapusan *stopwords*, transformasi, klasifikasi, dan yang terakhir evaluasi.
3. Visualisasi data, dikerjakan memanfaatkan bagan sesuai kebutuhan perusahaan atau siapa yang menggunakan data-data ini. Pada umumnya orang biasanya memakai teknik yang telah dikenal, seperti histogram, matriks, atau grafik.

Sentiment analysis adalah salah satu bidang *Natural Languange Processing* (NLP) untuk membangun sistem guna mengekstraksi dan mengenali opini dengan

bentuk teks. Informasi yang berbentuk teks sekarang ini ada banyak terdapat di internet dengan format blog, forum, situs berita *review*, serta media sosial. *Sentiment analysis* membantu informasi yang awalnya belum terstruktur dapat diubah jadi data yang lebih terstruktur. Data ini dapat mendeskripsikan opini masyarakat tentang politik, layanan, merek, produk, atau topik lainnya. Pemerintah, perusahaan, ataupun bidang lainnya yang selanjutnya mengunakan data-data tersebut guna membuat layanan masyarakat, *review* produk, analisis *marketing*, dan umpan balik produk.

Terdapat beberapa jenis *sentiment analysis* yang bisa dimanfaatkan guna mengidentifikasi respon pengguna. Mulai dari melihat polaritas pendapat hingga untuk mengidentifikasi niat pengguna. Di bawah ini merupakan beberapa tipe *sentiment analysis* tersebut antara lain adalah sebagai berikut (K. Y. Setiawan et al., 2014):

2.2.1 *Fine-Grained Sentiment Analysis*

Fine-Grained adalah salah satu tipe *sentiment analysis* yang paling umum, yang mana tefokus kepada tingkat polaritas pendapat. Jenis *sentiment analysis* ini akan mengklasifikasi pendapat atau respon ke beberapa kategori seperti sangat *positive*, netral, agak *negative*, dan *negative*.

2.2.2 Intent Sentiment Analysis

Tipe ini bermaksud untuk mengidentifikasi serta menggali lebih dalam alasan di balik komentar pengguna untuk mengetahui apakah itu termasuk pendapat, saran, pertanyaan, keluhan, atau malah penghargaan kepada layanan atau produk.

2.2.3 *Emotion Detection*

Sentiment analysis ini bertujuan guna mendeteksi emosi, seperti kemarahan, frustasi, kebahagiaan, dan kesedihan. Akan tetapi, salah satu kekurangan dari *emotion detection* yakni cara orang mengekspresikan emosinya berbeda-beda. Contohnya seperti kata ‘gila’ sebetulnya bermakna *negative*, tetapi

bila seseorang mengatakan, “gila sih, ini bangus banget,” menjadi bermakna *positive*.

2.2.4 *Multilingual Sentiment Analysis*

Tipe analisis yang satu ini digunakan guna menganalisis kata-kata dengan bermacam bahasa. Akan tetapi, tipe *sentiment analysis* ini termasuk cukup sulit dikarenakan harus mempunyai daftar kata dari berbagai bahasa, selain itu juga harus selalu melakukan *update* daftar ini sesuai dengan perkembangan bahasa tersebut.

2.2.5 *Aspect-Based Sentiment Analysis*

Analysis sentiment tipe ini terfokus terhadap elemen-elemen yang lebih spesifik pada layanan atau produk. Tipe ini juga memungkinkan menghubungkan antara s*entiment* spesifik dengan berbagai aspek layanan atau produk.

2.3 *K-Nearest Neighbor* (K-NN)

K-Nearest Neighbor (K-NN) adalah suatu algoritma yang dipergunakan untuk melakukan klasifikasi pada objek berdasarkan data pembelajaran yang mempunyai jarak paling dekat dengan objek tersebut. Nilai *k* yang paling baik tergantung dengan data. Secara umum, nilai *k* yang tinggi akan mengurangi noise pada klasifikasi, akan tetapi hal tersebut menyebabkan batasan antar setiap klasifikasi menjadi kabur (Romadloni et al., 2019).

K-Nearest Neighbor menjalankan klasifikasi dengan menggunakan proyeksi data pembelajaran terhadap ruang berdimensi banyak. Ruang tersebut terbagi menjadi bagian-bagian yang menerangkan kriteria data pembelajaran. Masing-masing data pembelajaran dilambangkan menjadi titik-titik c pada ruang dimensi banyak.

Mencari jarak terdekat antara data yang akan dievaluasi dengan *K-Nearest Neighbor* terdekatnya dalam data pelatihan merupakan prinsip kerja *K-Nearest Neighbor* (K-NN). Berikut ini merupakan rumus pencarian jarak menggunakan rumus *Euclidean* (Agusta, 2007).

$$d_i = \sqrt{\sum_{i=1}^{p}(x_{2i} - x_{1i})^2} \quad \text{.....(2.1)}$$

Keterangan: d_i : Jarak ke i
i : Variabel data
p : Dimensi data
x_1 : Data uji
x_2 : Data latih

Metode K-NN bekerja berdasarkan langkah-langkah seperti di bawah ini (Dinata, Fajriana, et al., 2020):

1. Langkah 1 : Tentukan jumlah tetangga terdekat (k) yang ingin dipertimbangkan sebagai dasar klasifikasi.
2. Langkah 2 : Hitung jarak antara data baru terhadap seluruh titik data pada dataset.
3. Langkah 3 : Urutkan jarak pada langkah 2 dari kecil ke besar, lalu ambil titik data dengan jarak yang paling kecil sejumlah *k* titik.
4. Langkah 4 : Hitung jumlah titik data *k* dalam setiap kelas atau kategori.
5. Langkah 5 : Masukkan data baru ke dalam kelas yang paling banyak jumlah *k*.

2.3.1 Banyaknya k Tetangga Terdekat

Untuk menerapkan algoritma *K-Nearest Neighbor*, harus ditentukan banyaknya k tetangga terdekat yang dipakai untuk mengerjakan pengelompokan data baru (Lestari, 2014). Banyaknya k sebaiknya adalah jika kelasnya genap maka k sebanyak angka ganjil seperti 3, 5, 7, dan seterusnya. Sebaliknya, apabila kelasnya ganjil maka k sebanyak angka genap seperti 2, 4, 6, dan seterusnya. Penentuan nilai k berdasarkan seberapa banyak data yang ada serta ukuran dimensi yang terbentuk oleh data. Semakin banyak data yang ada, maka sebaiknya angka k yang dipilih semakin rendah. Akan tetapi, semakin besar ukuran dimensi data, sebaiknya angka k yang dipilih semakin tinggi. Dalam menentukan nilai k sebenarnya tidak ada cara khusus namun dapat menggunakan *confusion matrix* untuk mengetahui nilai k mana

yang memiliki tingkat akurasi yang lebih tinggi, sehingga nilai k tersebutlah yang akan digunakan untuk implementasi algoritma *K-Nearest Neighbor*.

2.3.2 Contoh Perhitungan Manual Metode K-NN

Diberikan data training berupa dua atribut yaitu perempuan (Pr) dan laki-laki (Lk) untuk menghasilkan sebuah data apakah tergolong Pr atau Lk, berikut ini adalah contoh datanya:

Tabel 2.1 Contoh Data *Training* dan Data *Testing*

Tinggi Badan (cm)	Berat Badan (kg)	Jenis Kelamin
155	50	Pr
175	63	Lk
160	55	Pr
177	68	Lk
163	52	Pr
176	78	Lk
172	58	?

Keterangan:

- *Independent variables*, merupakan variabel yang nilainya tidak dipengaruhi oleh variabel lain. Pada contoh di atas, yang termasuk *independent variable* yaitu tinggi badan dan berat badan.
- *Dependent variable*, merupakan variabel yang nilainya dipengaruhi oleh variabel lain. Pada contoh di atas, yang termasuk *dependent variable* yaitu jenis kelamin.

Diberikan data baru yang akan diklasifikasikan, yakni tinggi badan = 172 dan berat badan = 58. Jadi termasuk klasifikasi apa data baru tersebut? Pr atau Lk? Langkah penyelesaian:

1. Menentukan parameter k, misalnya saja kita buat juamlah tetangga terdekat k = 3.
2. Hitung jarak antara data baru dengan semua data *training*. Kita dapat menggunakan rumus *euclidean distance*, dengan perhitungan sebagai berikut:

Tabel 2.2 Perhitungan Jarak dengan *Euclidean Distance*

X	Y	***Euclidean Distance* (172, 58)**
155	50	$\sqrt{(155-172)^2+(50-58)^2} = \sqrt{(-7)^2+(-8)^2} = \sqrt{353} =$ 18,78829423
175	63	$\sqrt{(175-172)^2+(63-58)^2} = \sqrt{(3)^2+(5)^2} = \sqrt{34} =$ 5,830951895
160	55	$\sqrt{(160-172)^2+(55-58)^2} = \sqrt{(-12)^2+(-3)^2} = \sqrt{153} =$ 12,36931688
177	68	$\sqrt{(177-172)^2+(68-58)^2} = \sqrt{(5)^2+(28)^2} = \sqrt{809} =$ 11,18033989
163	52	$\sqrt{(163-172)^2+(52-58)^2} = \sqrt{(-9)^2+(-6)^2} = \sqrt{353} =$ 10,81665383
176	78	$\sqrt{(176-172)^2+(78-58)^2} = \sqrt{(4)^2+(20)^2} = \sqrt{416} =$ 20,39607805

3. Mengurutkan jarak dari data baru dengan data *training* dan menentukan tetangga terdekat berdasarkan jarak minimum k.

Tabel 2.3 Pengurutan Jarak Terdekat Data Baru dengan Data *Training*

Rangking	***Euclidean Distance***	**Jenis Kelamin**
5	18,78829423	Pr
1	5,830951895	Lk
4	12,36931688	Pr
3	11,18033989	Lk
2	10,81665383	Pr
6	20,39607805	Lk

4. Parameter k yang telah ditentukan k = 3. Maka diambil data nilai terdekat yang telah di rangking sejumlah 3 data *training*.

Tabel 2.4 Penentuan Kategori yang Termasuk k = 3

Rangking	***Euclidean Distance***	**Jenis Kelamin**
1	5,830951895	Lk
2	10,81665383	Pr

Tabel 2.4 Penentuan Kategori yang Termasuk k = 3 (Lanjutan)

Rangking	***Euclidean Distance***	**Jenis Kelamin**
3	11,18033989	Lk

5. Gunakan kategori mayoritas yang sederhana dari tetangga terdekat tersebut sebagai nilai prediksi data baru.

Tabel 2.5 Hasil Klasifikasi Berdasarkan Kategori Mayoritas

Tinggi Badan (cm)	**Berat Badan (kg)**	**Jenis Kelamin**
155	50	Pr
175	63	Lk
160	55	Pr
177	68	Lk
163	52	Pr
176	78	Lk
172	58	Lk

Data yang kita miliki pada rangking 1, 2, dan 3 mempunyai 2 jenis kelamin Lk dan 1 jenis kelamin Pr. Dari jumlah mayoritas (Lk > Pr) tersebut dapat disimpulkan bahwa data baru termasuk dalam jenis kelamin Lk.

2.3.3 Kelebihan dan Kekurangan Metode K-NN

Berikut ini kelebihan dan kekurang dari metode *K-Nearest Neighbor* (Budiman & Firmansyah, 2015):

Tabel 2.6 Kelebihan dan Kekurangan Metode K-NN

Kelebihan	Kekurangan
• Sangat sederhana dan mudah untuk diimplementasikan. • Beberapa parameter untuk acuan yakni jarak matrik dan k. • Efektif dalam menghitung data berskala besar maupun kecil. • Kuat untuk melatih data *noisy*.	• Perlu untuk menentukan nilai k untuk menyatakan tetangga terdekatnya. • Perhitungan jarak harus dilakukan pada setiap *querry instance* sehingga membuat biaya komputasi yang tinggi

Tabel 2.6 Kelebihan dan Kekurangan Metode K-NN (Lanjutan)

	• Rentan terhadap variable non-informatif.

2.4 *Web Scraping*

Web scraping merupakan metode guna mengestraksi informasi yang berasal dari situs *web*, sehingga menjadi data yang bisa dianalisis serta digunakan untuk berbagai tujuan (Djufri, 2020). Proses *web scraping* data dari internet terbagi menjadi dua langkah berurutan, yakni menemukan *web* yang hendak diekstrak datanya kemudian mengekstrak informasi/data yang diperlukan dari *web* tersebut (Zhao, 2017). Sistem *web scraping* terdiri dari dua bagian, yaitu *web crawler* dan *web scraper*. *Web crawler* sebagai 'laba-laba' yang memiliki kecerdasan buatan (AI) bergerak menjajaki internet guna mencari informasi pada sebuah URL. Sedangkan *web scraper* merupakan alat yang dipakai guna mengekstrak data dari URL yang sebelumnya telah ditelusuri oleh *crawler*.

2.4.1 Teknik *Web Scraping*

Ada beberapa teknik *web scraping* yang dapat digunakan untuk mendapatkan data dari internet. Teknik-teknik tersebut yaitu sebagai berikut (Josi et al., 2014):

1. *Copy paste* data secara manual, merupakan cara *web scraping* yang paling sederhana dikarenakan harus mengambil dan menyimpan informasi yang dibutuhkan satu per satu, sehingga teknik ini memakan waktu yang lama.
2. *Regular expression*, merupakan baris kode yang dimanfaatkan dalam algoritma pencarian guna menemukan tipe data tertentu dari suatu file. Konsisten *syntax* dalam berbagai bahasa pemograman sehingga teknik ini sangat fleksibel merupakan salah satu keuntungan utama dari penggunaan *regular expression*.
3. *Parsing* HTML, merupakan metode yang digunakan dengan mengirimkan HTTP permintaan ke server penyimpanan data *website* yang datanya ingin diekstrak. Teknik ini juga dapat digunakan untuk melakukan *web scraping*

bukan hanya di halaman *website* yang bersifat statis saja, namun juga dapat digunakan di halaman *website* yang bersifat dinamis.

4. Menganalisa DOM (Dokumen Objek Model), merupakan representasi struktur suatu halaman *website* yang ditulis menggunakan HTML. Saat melakukan parsing HTML, DOM yang berasal dari halaman yang hendak diekstrak datanya akan dimuat terlebih dahulu. DOM juga membawa data yang terdapat pada file HTML, sehingga analisis DOM dapat menjadi alternatif guna melakukan *web scraping* pada halaman situs dinamis bila *parsing* HTML tidak mengeluarkan hasil.
5. XPath, merupakan bahasa *query* yang dipakai guna memilih node dari struktur file HTML dan XML. Implementasinya tidak berbeda jauh dengan analisa DOM, yakni digunakan guna mencari data pada elemen teks di file penunjang halaman.
6. *Google sheet*, merupakan aplikaksi *web* kepunyaan *google* yang biasanya dipakai guna membuat *spreadsheet*. Penggunaan *google sheet* guna melakukan web scraping membutuhkan adanya browser yang terdapat fitur *inspect element*. Setelahnya, hanya cukup mengopi *expression* XPath pada elemen halaman *website* yang datanya akan disalin ke dalam *command* IMPORTXML yang berada di *google sheet*.

2.4.2 Langkah-Langkah *Web Scraping*

Terdapat langkah-langkah yang perlu dilakukan untuk melakukan *web scraping*, yakni sebagai berikut (Josi et al., 2014):

1. *Create scraping template*, yaitu pembuat program mempelajari dokumen HTML dari *website* yang akan diambil informasinya dari tag HTML yang mengapit informasi yang akan diambil.
2. *Explore site navigation*, yaitu pembuat program mempelajari Teknik navigasi pada website yang akan diambil informasinya untuk ditirukan pada aplikasi *web scraping* yang akan dibuat.

3. *Automate navigaton and extraction*, berdasarkan informasi yang didapatkan dari langkah 1 dan 2 di atas, aplikasi *web scraper* dibuat agar pengambilan informasi dapat dilakukan secara otomatis dari *website* yang ditentukan.
4. *Extracted data and package history*, informasi yang didapat dari langkah 3 disimpan dalam tabel atau tabel-tabel *database*.

2.4.3 *Web Scraping Tools*

Selain menggunakan Teknik-teknik *web scraping*, pengguna juga dapat memakai beberapa *tools* atau *software*. *Tools* tersebut adalah sebagai berikut (Zhao, 2017):

1. *Scrapy*, mempunyai beberapa fitur diantaranya memproses, mengelola dan menyaring data yang diterima berasal dari berbagai *website*. *Software* ini juga diketahui paling efisien dalam melakukan *web scraping* dengan data yang besar. Format JSON, CSV hingga XML digunakan untuk mengekspor data pada *scrapy*.
2. Data *scraper*, *software* ini dapat dipakai tanpa mengeluarkan biaya, serta dapat melakukan *web scraping* sampai 500 halaman *website*. Ekspor data dapat dilakukan menggunakan format file CSV atau XSL.

Parsehub, dapat diterapkan ke semua sistem operasi dari OS, seperti *Linux*, *Mac*, serta *Windows* oleh karena itu *software* ini cukup fleksibel. Namun *software* ini tidak gratis sehingga perlu mengeluarkan biaya untuk menggunakannya, digunakan 20 proyek *web scraping* untuk *subscription plan software* ini.

2.5 TF-IDF (*Term Frequency Inverse Document Frequency*)

Metode TF-IDF adalah metode guna menghitung bobot setiap kata paling umum dipakai pada *information retrieval*. Metode ini juga terkenal mudah, efisien, dan memiliki hasil yang akurat. Metode TF-IDF merupakan cara pemberian bobot hubungan sebuah kata (*term*) terhadap dokumen. TF-IDF ini merupakan sebuah ukuran *statistic* yang dipakai guna mengevaluasi seberapa penting suatau kata di dalam suatu dokumen atau dalam sekelompok kata. Untuk dokumen tunggal, setiap kalimat dianggap sebagai dokumen. Frekuensi kemunculan kata pada dokumen

yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen itu. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam sebuah dokumen dan semakin kecil jika muncul dalam banyak dokumen (Melita, 2018).

Pada metode TF-IDF digunakan rumus guna menghitung bobot (W) masing-masing dokumen terhadap kata kunci dengan rumus (Asiyah, 2016):

$W_{dx} = TF_{dx} \times IDF_x$ ……(2.2)

Keterangan: d : Dokumen ke-d

x : Kata ke-x dari kata kunci

W : Bobot doumen ke-d terhadap kata ke-t

TF : Banyaknya kata yang dicari pada sebuah dokumen

IDF : *Inversed document frequency*

Sedangkan untuk mencari IDF pada sebuah dokumen dapat menggunakan rumus berikut:

$\text{IDF} = \log(\frac{N}{df})$ ……(2.3)

Dimana:

IDF : *Inversed document frequency*

N : Jumlah dokumen

df : Jumlah dokumen yang mempunyai kata yang dicari

Term weighting atau pembobotan *term* sangat dipengaruhi oleh hal-hal berikut ini:

1. *Term frequency* (tf) *factor*, yakni faktor yang menentukan bobot *term* pada sebuah dokumen berdasarkan jumlah kemunculannya dalam dokumen tersebut. Nilai jumlah kemunculan sebuah kata diperhitungkan ketika pemberian bobot terhadap suatu kata. Semakin besar jumlah kemunculan *term* (tf tinggi) pada dokumen, maka semakin besar juga bobotnya dalam dokumen atau juga akan memberikan nilai kesesuaian yang semakin besar pula.
2. *Inverse document frequency* (idf) *factor*, yakni pengurangan dominansi *term* yang sering muncul di berbagai dokumen. Hal ini diperlukan dikarenakan *term* yang banyak muncul di berbagai dokumen, dapat

dianggap sebagai *term* umum (*common term*) sehingga tidak penting nilainya. Sebaiknya, faktor kejarang munculan kata (*term scarcity*) dalam koleksi dokumen harus diperhatikan dalam pemberian bobot.

2.5.1 Contoh Perhitungan Manual TF-IDF

Berikut ini merupakan contoh perhitungan manual TF-IDF, sebagai berikut:

Tabel 2.7 Dokumen 1

Istilah	Jumlah
Ini	1
Adalah	1
Sebuah	2
Sampel	1

Tabel 2.8 Dokumen 2

Istilah	Jumlah
Ini	1
Adalah	1
Contoh	3
Lainnya	2

Untuk menghitung TF-IDF kata ‘ini’, dapat dilakukan langkah-langkah berikut. Dalam tiap dokumen, kata ‘ini’ sama-sama muncul sekali. Nilai IDF bersifat tetap per korpus dan bergantung pada jumlah dokumen yang memiliki kata ‘ini’. Dalam kasus ini, kita memiliki korpus yang semua dokumennya memiliki kata ‘ini’.

$$\text{IDF}\ (\text{"ini"}, d) = \log(\tfrac{2}{2}) = 0$$

TF-IDF(“ini”, d_1) = 1 × 0 = 0

TF-IDF(“ini”, d_2) = 1 × 0 = 0

Jadi, nilai TF-IDF kata ‘ini’ adalah nol yang berarti bahwa kata-kata ini tidak terlalu bermakna karena muncul dalam seluruh dokumen. Contoh lainnya, kata ‘contoh’ muncul tiga kali, tapi hanya pada dokumen 2.

$$\text{IDF}(\text{"contoh"}, \text{d}) = \log(\tfrac{2}{1}) = 0{,}301$$

TF-IDF(“contoh”, d_1, D) = 0 × 0,301 = 0

TF-IDF(“contoh”, d_2, D) = 3 × 0,301 = 0,903

2.6 *Confusion Matrix*

Tahap pengujian untuk menentukan apakah suatu objek benar atau salah adalah matriks kebingungan. Pengelompokan uji *confusion matrix* di mana kelas yang diantisipasi akan ditampilkan pada titik tertinggi kisi dan kelas yang diperhatikan di sebelah kiri. Terdapat angka di setiap sel yang menunjukkan prediksi jumlah kasus aktual di kelas yang diamati (Dinata, Akbar, et al., 2020).

Tabel 2.9 *Confusion Matrix*

	Nilai Prediksi	
Nilai Aktual	TP	FN
	FP	TN

Keterangan:

TP : Data aktual positif yang diprediksi positif.

TN : Data aktual negatif yang diprediksi negatif.

FP : Data aktual negatif namun diprediksi positif.

FN : Data aktual positif namun diprediksi negatif.

Rumus untuk perhitungan *confusion matrix* seperti dibawah ini:

1. *Precision*, berguna untuk menentukan tingkat presisi antara data yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem.

   $$Precision = \frac{TP}{(TP+FP)} \quad \text{.....(2.4)}$$

2. *Recall*, berguna untuk mengevaluasi tingkat keberhasilan sistem menemukan kembali sebuah informasi.

   $$Recall = \frac{TP}{(TP+FN)} \quad \text{.....(2.5)}$$

3. *Accuracy*, untuk mengukur kinerja sebuah metode.

   $$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \quad \text{.....(2.6)}$$

4. F1-*Score*, untuk membandingkan rata-rata presisi dan *recall*.

   $$\text{F1-}Score = 2 \times \frac{presision \times recall}{presision + recall} \quad \text{.....(2.7)}$$

2.7 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian yang terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian ini. Di bawah ini merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis, sebagai berikut:

Tabel 2.10 Penelitian Terdahulu

No.	Penulis	Judul	Kesimpulan
1.	• Muhammad Iqbal Ahmadi • Dudih Gustian • Falentino Sembiring	Analisis *Sentiment* Masyarakat Terhadap Kasus Covid-19 pada Media Sosial Youtube dengan Metode *Naïve Bayes*	Penelitian ini dilakukan pada tahun 2021. Pada penelitian ini peneliti menggunakan media sosial *Youtube* dari *channel* kompas.tv dengan menggunakan metode *Naïve Bayes* guna mengetahui tingkat persentase respon dan komentar masyarakat pada beberapa video yang berisikan berita mengenai perkembangan kasus Covid-19 di Indonesia dengan hasil tanggapan masyarakat lebih dominan komentar negatif dengan jumlah sebanyak 800 komentar dan sedangkan jumlah komentar positif diberikan masyarakat sebanyak 361 komentar, dengan metode algoritma *Naïve Bayes* menghasilkan tingkat akurasi sebesar 74% (Ahmadi et al., 2021).

Tabel 2.10 Penelitian Terdahulu (Lanjutan)

No.	Penulis	Judul	Kesimpulan
2.	• Anni Karimatul Fauziyyah • Deden Hardan Gautama	Analisis Sentimen Pandemi Covid-19 pada *Streaming Twitter* dengan *Text Mining Python*	Penelitian ini dilakukan pada tahun 2020. Peneliti melakukan *streaming* data *twit* dengan pencarian data collection yang menghasilkan nilai paling tinggi kategori netral yakni 58,94% untuk sentimen Covid-19 dan *variable coronavirus* dengan nilai 55,10% dibandingkan polaritas negatif atau positif (Fauziyyah, 2020).
3.	Lilyana Asri Utami	Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Algoritma *Support Vector Machine* dan *K-Nearest Neighbor* Berbasis *Particle Swarm Optimization*	Penelitian ini di lakukan pada tahun 2017. Metode yang digunakan oleh peneliti pada penelitian ini yakni *Support Vector Machine* (SVM), *Support Vector Machine* berbasis Particle Swarm Optimization (SVM+PSO), K-Nearest Neighbor (K-NN), dan *K-Nearest Neighbor* berbasis Particle Swarm Optimization (K-NN+PSO) untuk meninjau opini publik mengenai berita kebakaran hutan dengan review opini publik yang dikumpulkan sebanyak 360 data yang di dalamnya terdapat 180 opini positif dan 180 opini negatif. Dengan hasil penelitian perhitungan metode SVM memiliki *accuracy* sebesar 80,83% dan AUC sebesar 0,947. Metode SVM+PSO menghasilkan

Tabel 2.10 Penelitian Terdahulu (Lanjutan)

No.	Penulis	Judul	Kesimpulan
			accuracy sebesar 86,11% dan AUC 0,922. Metode K-NN memiliki *accuracy* sebesar 85,00% dan AUC sebesar 0,918. Serta metode K-NN+PSO memiliki nilai *accuracy* sebesar 73,06% dan AUC sebesar 0,500 (Utami, 2017).
4.	• Wahyu Hidayat • Ema Utami • Ahmad Fikri Iskandar • Anggit Dwi Hartanto • Agus Budi Prasetyo	Perbandingan Performansi Model pada Algoritma K-NN Terhadap Klasifikasi Berita Fakta Hoaks Tentang Covid-19	Penelitian ini dilakukan pada tahun 2021. Pada penelitian ini peneliti menggunakan algoritma K-NN untuk mengklasifikasikan berita fakta *hoax* mengenai Covid-19 dengan hasil nilai pengujian tertinggi pada model *Jaccard Distance* dengan nilai k = 4 mendapatkan nilai hasil pengujian 0,696 *accuracy*, 0,710 *precision*, 0,599 F1-*score* dan 0,572 untuk *recall* (Hidayat et al., 2021).
5	• Dwi Selvy Wisdayani • Indah Manfaati Nur • Rochdi Wasoni	Penerapan Algoritma *K-Nearest Neighbor* dalam Klasifikasi Tingkat Keparahan Korban Kecelakaan Lalu Lintas di	Peneltian ini dilakukan pada tahun 2019. Metode yang digunakan peneliti pada penelitian ini yakni algoritma K-NN untuk mengklasifikasikan tingkat keparahan korban kecelakaan lalu lintas di Kabupaten Pati, Jawa Tengah dengan hasil penelitian tingkat keparahan kecelakaan lalu lintas di Kabupaten Pati, Jawa Tengah memiliki karakteristik secara umum yakni korban kecelakaan terluka lebih

Tabel 2.10 Penelitian Terdahulu (Lanjutan)

No.	Penulis	Judul	Kesimpulan
		Kabupaten Jawa Tengah	banyak berjenis kelamin laki-laki sebagai pengendara. Kecelakaan terjadi pada kisaran pukul 06:00-12:00 di waktu kejadian harian, dan korban sering tidak menggunakan alat keselamatan dan lengah mengakibatkan kecelakaan depan-samping. Berdasarkan data sampel yang digunakan menghasilkan 64,40% tingkat accuracy, 11,18% nilai error, 60,43% recall, dan 62,33% f-measure (Wisdayani et al., 2019).

BAB III
METODE PENELITIAN

3.1 Tempat dan Waktu Pelaksanaan Penelitian

Penelitian ini dilakukan dengan mengambil data berupa berita-berita Covid-19 pada portal berita detik.com guna memisahkan data yang menghasilkan opini positif, opini negatif, dan opini netral dari berita Covid-19 pada tahun 2020-2021. Waktu pelaksanaan penelitian ini dimulai setelah selesai dilakukannya seminar proposal.

3.2 Studi Literatur

Pada tahap pengumpulan data studi literatur diambil dari berbagai sumber seperti jurnal, buku, paper yang memiliki hubungan dengan penelitian mengenai *text mining* baik yang berkenaan dengan pencarian Covid-19 atau data bidang/topik lain.

3.3 Teknik Pengumpulan Data

Dalam penelitian ini, data yang digunakan diambil dari portal berita detik.com. Data yang diperoleh dengan proses *crawling text* berita dari detik.com menggunakan *helper* dari php yang bernama '*simple_html_dom*' milik Jose Solorzano.

3.4 Teknik Pengolahan Data

Data teks dari url berita di portal berita detik.com akan dipisahkan menjadi dua bagian, yaitu data latih dan data uji.

1. Data latih merupakan kumpulan data yang akan menjadi bahan latihan untuk melatih algoritma K-NN untuk mengenali pola-pola opini positif, negatif, atau netral. Data latih akan ditentukan dengan kategori positif, negatif, atau netral, untuk mendapatkan frekuensi term (TF-IDF).

2. Data uji merupakan kumpulan data yang akan menjadi bahan uji pada penerapan algoritma K-NN pada program nantinya, untuk melihat kemampuan algoritma K-NN untuk menentukan opini yang dihasilkan dari data-data tersebut berupa opini positif, opini negatif, dan opini netral.

3.5 Analisis Kebutuhan Sistem

Dalam mendukung penelitian analisis sentimen berita Covid-19 pada portal berita detik.com mengggunakan metode *K-Nearest Neighbor* membutuhkan beberapa alat, adapun alat yang dimaksud adalah:

3.5.1 Perangkat Keras (*Hardware*)

Spesifikasi komputer/PC yang digunakan yaitu:

1. *RAM 4 GB*
2. *HDD 1 TB*
3. *CPU Intel Core i3-6006U, 2.0GHz*

3.5.2 Perangkat Lunak (*Software*)

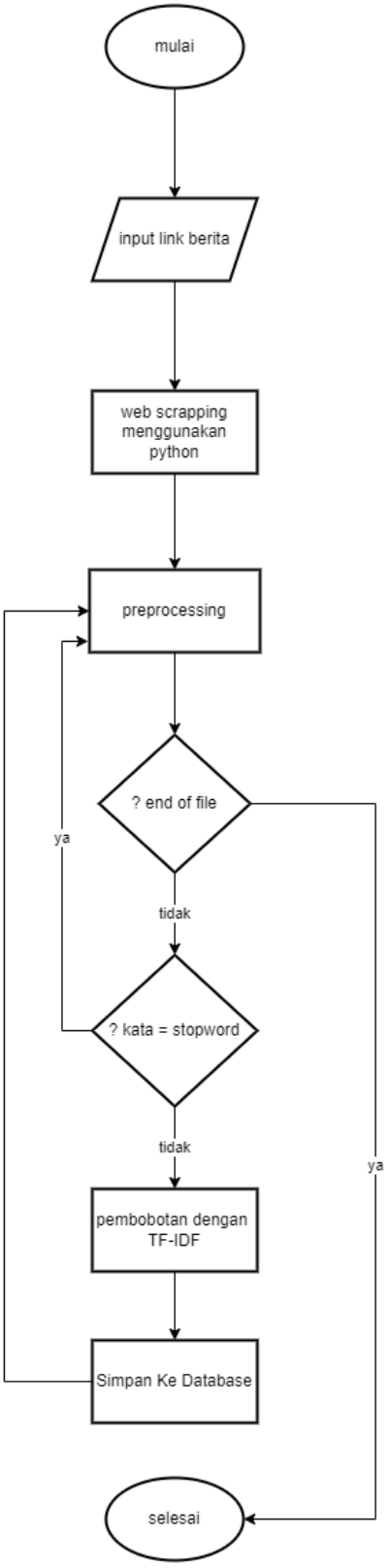
Spesifikasi *software* yang digunakan yaitu :

1. Sistem Operasi *Windows* 10
2. *Web Browser*
3. *Web Scraping*
4. *Database* MySQL
5. XAMPP

3.6 Skema Sistem

Terdapat dua diagram pada skema sistem tentang *sentiment analysis* pada penelitian ini, yakni sebagai berikut:

3.6.1 Diagram Data Latih



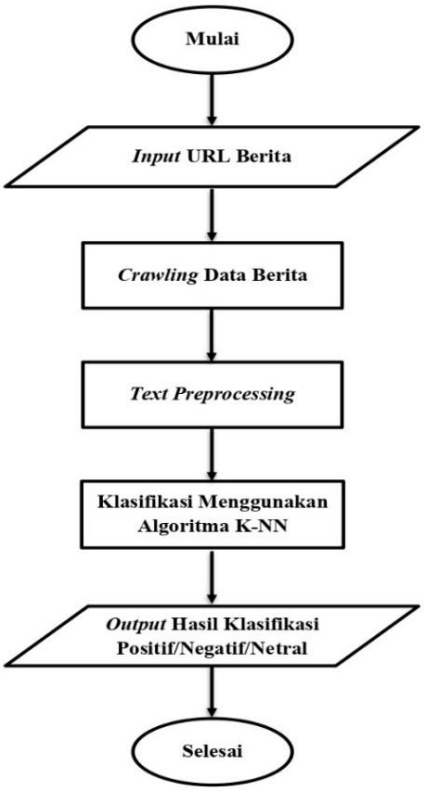
Gambar 3.1 Diagram Data Latih

Keterangan:

1. Mulai
2. *Input link* berita, pada tahapan ini user akan meng-*input link* berita yang diambil dari *website* www.detik.com.
3. *Web scraping* menggunakan *python*, pada tahap ini *code* program akan melakukan *crawling* data berita berdasarkan URL yang telah di-*input* oleh *user*. Data berita yang diambil adalah judul dan isi dari berita tersebut.
4. *Preprocessing*, merupakan suatu proses untuk menyeleksi data *text* agar menjadi lebih terstruktur lagi dengan melalui serangkaian tahapan yang meliputi:
    - *Case folding*, peran dari tahap ini yaitu untuk menyamaratakan penggunaan huruf kapital, sedangkan karakter lain yang bukan termasuk huruf dan angka seperti tanda baca dan spasi dianggap sebagai delimiter.
    - *Tokenizing*, untuk memudahkan proses analisis data maka harus memecahkan kalimat-kalimat menjadi kata atau disebut dengan token.
    - *Filtering*, tahap ini dilakukan guna mendapatkan kata-kata yang penting dari hasil token tadi. Kata umum yang sering muncul dan tidak mempunyai makna seperti kata penghubung (dan, yang, serta, setelah, dan lainnya) akan dihilangkan.
    - *Stemming*, tahap ini juga diperlukan untuk memperkecil jumlah *index* yang berbeda dari suatu data sehingga sebuah kata yang memiliki *suffix* maupun *prefix* akan kembali ke bentuk dasarnya. Selain itu, juga untuk melakukan pengelompokan kata-kata lain yang mempunyai kata dasar dan kata arti yang serupa tetapi memiliki bentuk yang berbeda karena mendapatkan imbuhan yang berbeda pula. Tahap ini akan dilakukan menggunakan bantuan *library* php ‘sastrawi’ yang dikembangkan oleh Andy Librian pada situs *repository* github.com.

5. *End* of file, apakah kata tersebut merupakan kata terakhir dari data berita, jika ‘ya’ maka proses sistem selesai, namun jika ‘tidak’ maka akan lanjut ke tahap berikutnya.
6. Kata = *stopword*, proses ini dilakukan untuk mengecek apakah kata yang telah melalui *preprocessing* tersebut merupakan *stopword* atau bukan, jika ‘ya’ maka kata tersebut akan dikembalikan ke proses *preprocessing* untuk dihapus, jika ‘tidak’ maka akan masuk ke proses pembobotan kata.
7. Pembobotan dengan TF-IDF, jika kata yang dicek tersebut bukan *stopword* maka kata tersebut masuk ke proses pembobotan dengan TF-IDF yang mana pada proses ini akan menghitung bobot dari setiap kata.
8. Simpan ke *database*, selanjutnya data akan disimpan ke *database*, setelah kata tersebut disimpan sistem akan kembali ke tahap *preprocessing* untuk memproses kata berikutnya.
9. Selesai, proses selesai setelah kata terakhir diproses.

3.6.2 Diagram Data Uji



Gambar 3.2 Diagram Data Uji

Keterangan:

1. Mulai
2. *Input* URL berita, pada tahapan ini user akan meng-*input* URL berita yang diambil dari *website* www.detik.com.
3. *Crawling* data berita, pada tahapan ini *code* program akan melakukan *crawling* data berita berdasarkan URL yang telah di-*input* oleh *user*. Data berita yang diambil adalah judul dan isi dari berita tersebut.
4. *Text preprocessing*, merupakan suatu proses untuk menyeleksi data *text* agar menjadi lebih terstruktur lagi dengan melalui serangkaian tahapan yang meliputi:
    - *Case folding*, peran dari tahap ini yaitu untuk menyamaratakan penggunaan huruf kapital, sedangkan karakter lain yang bukan termasuk huruf dan angka seperti tanda baca dan spasi dianggap sebagai delimiter.
    - *Tokenizing*, untuk memudahkan proses analisis data maka harus memecahkan kalimat-kalimat menjadi kata atau disebut dengan token.
    - *Filtering*, tahap ini dilakukan guna mendapatkan kata-kata yang penting dari hasil token tadi. Kata umum yang sering muncul dan tidak mempunyai makna seperti kata penghubung (dan, yang, serta, setelah, dan lainnya) akan dihilangkan.
    - *Stemming*, tahap ini juga diperlukan untuk memperkecil jumlah *index* yang berbeda dari suatu data sehingga sebuah kata yang memiliki *suffix* maupun *prefix* akan kembali ke bentuk dasarnya. Selain itu, juga untuk melakukan pengelompokan kata-kata lain yang mempunyai kata dasar dan kata arti yang serupa tetapi memiliki bentuk yang berbeda karena mendapatkan imbuhan yang berbeda pula. Tahap ini akan dilakukan menggunakan bantuan *library* php ‘sastrawi’ yang dikembangkan oleh Andy Librian pada situs *repository* github.com.

5. Klasifikasi menggunakan algoritma K-NN, selanjutnya data berita akan diuji atau diklasifikasikan menggunakan algoritma K-NN berdasarkan data latih sebelumnya dengan rumus *euclidean distance*.
6. *Output* hasil klasifikasi positif/negatif/netral, selanjutnya menampilkan hasil klasifikasi bernilai positif, negatif, atau netral.
7. Selesai.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Sistem

Pada penelitian ini, penulis membangun sebuah sistem untuk membantu menganalisis berita-berita pada portal berita **detik.com** terkait dengan Covid-19. Sistem ini melakukan beberapa tahapan yaitu pengumpulan data yang diambil dari portal berita **detik.com** dengan kata kunci "Covid-19" sebanyak 50 berita untuk data uji dan 450 berita untuk data latih yang diambil secara acak, kemudian berita latih dengan kata kunci "Covid-19" tersebut diberi label yaitu positif, netral, atau negatif. Data latih (*training*) dan data uji (*testing*) akan dimasukkan ke dalam *text preprocessing* yang bertujuan untuk mengolah data agar dianalisis ke dalam algoritma *K-Nearest Neighbor*. Namun sebelum itu data-data tersebut akan dicek apakah data tersebut merupakan berita mengenai Covid-19 atau bukan dengan cara sistem akan mengecek apakah di dalam teks berita tersebut mengandung kata-kata yang sebelumnya telah penulis tentukan sebagai kata kunci pada tabel 4.1, jika di dalam berita tidak memiliki kata kunci tersebut maka berita tersebut akan digolongkan ke dalam berita non-Covid dan tidak akan dianalisis ke dalam algoritma *K-Nearest Neighbor*. Namun apabila di dalam berita memiliki kata kunci yang dicari maka akan digolongkan ke dalam berita Covid-19 dan selanjutnya akan dilakukan pembobotan kata dengan algoritma TF-IDF dan hasil perhitungannya akan dimasukkan ke dalam klasifikasi dengan algoritma K-NN. Kemudian akan menghasilkan data kelas tersebut termasuk kelas positif, netral, atau negatif.

Tabel 4.1 Tabel kata Kunci (*Keyword*)

No	Kata Kunci (*Keyword*)
1	Covid
2	Pandemi
3	Omicron
4	Corona

4.2 Analisis Data

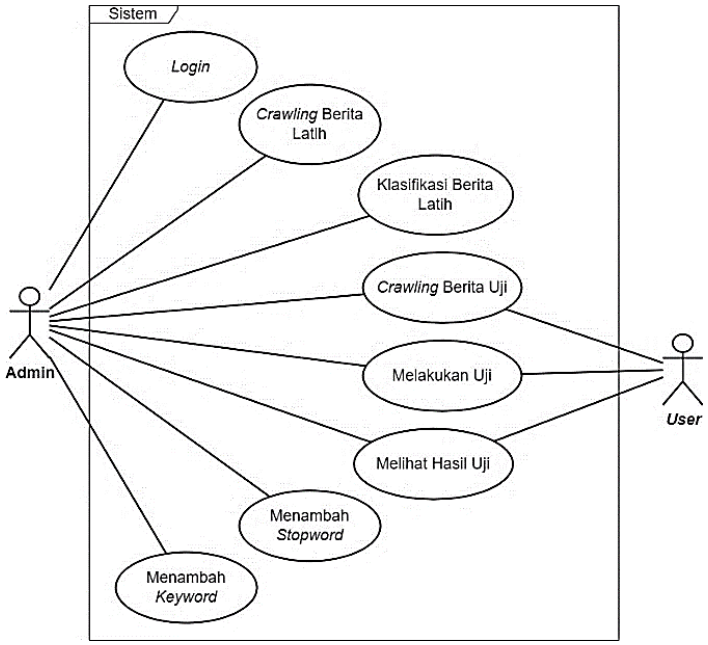
Berdasarkan kebutuhan yang digunakan pada sistem ini, data diperoleh dalam bentuk berita pada portal berita **detik.com** tentang Covid-19 menggunakan *scraping* dengan memasukkan judul berita dan *link* URL berita detik.com mengenai Covid-19 dengan total data sebanyak 450 data berita latih yang kemudian diberikan pelabelan berupa positif, netral, dan negatif masing-masing label sebanyak 150 data, serta data uji sebanyak 50 data berita Covid-19 yang dipilih secara acak.

4.3 Perancangan Sistem

Perancangan sistem merupakan tahap pendefinisian terhadap kebutuhan dalam membangun perangkat lunak. Pada proses pembuatannya sistem ini menggunakan UML *use case diagram*, *sequence diagram*, dan *activity diagram* untuk menjelaskan alur dari proses yang ada pada sistem.

4.3.1 *Use Case Diagram*

Pada penelitian ini, *Use case diagram* yang digunakan pada sistem yakni sebagai berikut.



Gambar 4.1 *Use Case Diagram*

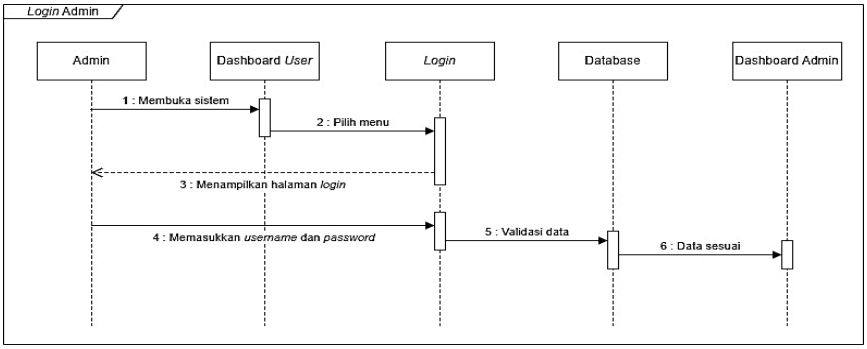
Dalam *use case diagram* di atas terdapat dua orang aktor yaitu admin dan *user*. Pada sistem ini, seorang admin dapat melakukan *login*, *crawling* berita latih,

klasifikasi berita latih, *crawling* berita uji, melakukan uji berita, melihat hasil uji, menambah *stopword*, dan menambah *keyword*. Sedangkan *user* hanya dapat melakukan *crawling* berita uji, melakukan uji berita, dan melihat hasil uji saja.

4.3.2 *Sequence Diagram*

Sequence diagram yang digunakan pada sistem ini adalah sebagai berikut.

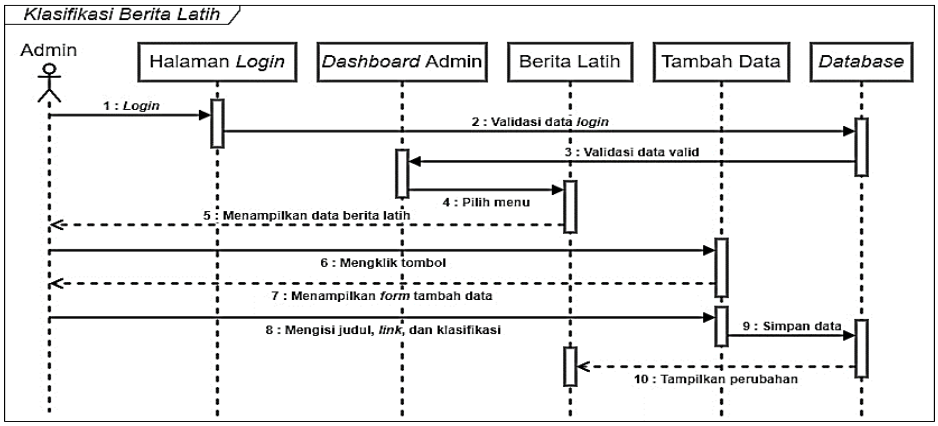
1. *Sequence Diagram Login*



Gambar 4.2 *Sequence Diagram Login*

Sequence diagram di atas menggambarkan bagaimana proses *login* yang dilakukan oleh admin, yang dimana halaman pertama yang akan muncul ketika memasuki sistem adalah halaman dashboard user, kemudian admin harus memilih menu login sehingga akan dibawa ke halaman login, kemudian admin akan memasukkan *username* dan *password*, kemudian sistem akan melakukan pencocokan ke *database*, jika *username* dan *password* ditemukan dalam *database* maka admin akan dialihkan ke halaman *dashboard* admin.

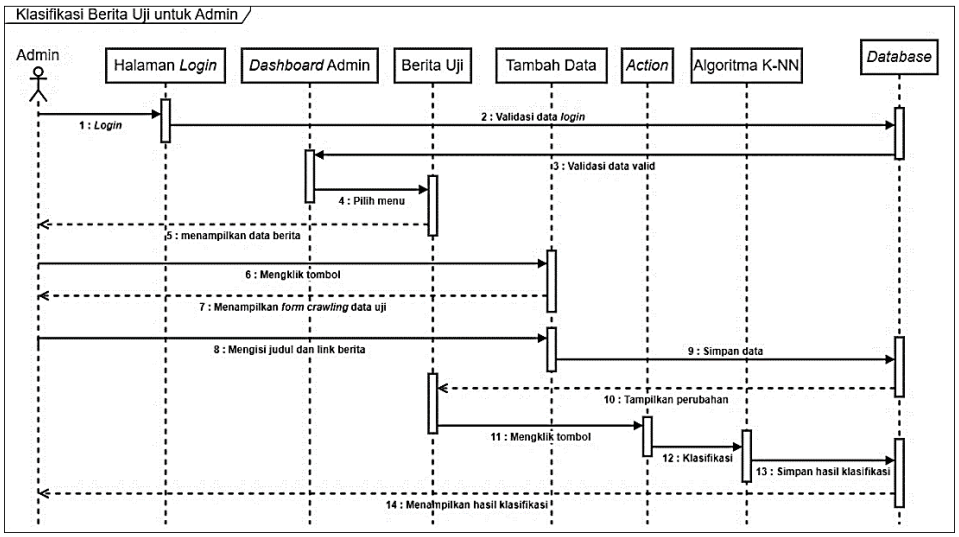
2. *Sequence Diagram* Klasifikasi Berita Latih



Gambar 4.3 *Sequence Diagram* Klasifikasi Berita Latih

Sequence diagram di atas menggambarkan bagaimana proses klasifikasi data latih dilakukan. Admin harus melakukan *login* di halaman *login* admin kemudian setelah proses validasi selesai dengan hasil valid maka sistem akan menampilkan *dashboar* admin, pada *dashboard* admin ini admin memilih menu berita latih yang kemudian akan menampilkan data berita latih. Jika admin ingin menambahkan data berita latih maka admin harus mengklik tombol tambah data lalu akan diarahkan ke halaman *form* tambah data latih yang merupakan halaman *crawling* data latih dan admin diharuskan mengisi judul berita, URL berita, dan memilih klasifikasi berita positif, netral, atau negatif kemudian setelah selesai melakukan tambah data latih sistem akan kembali menampilkan halaman data latih dan data latih pun telah ditambahkan.

3. *Sequence Diagram* Klasifikasi Berita Uji Admin

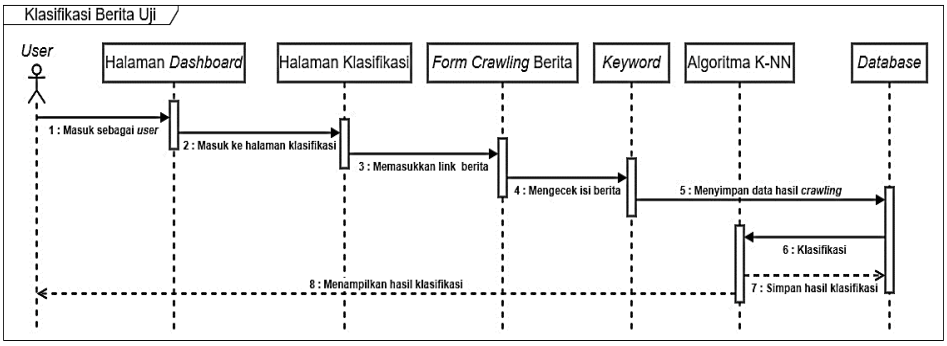


Gambar 4.4 *Sequence Diagram* Klasifikasi Berita Uji Admin

Sequence diagram di atas menggambarkan bagaimana proses klasifikasi data uji yang dilakukan oleh admin. Admin harus melakukan *login* di halaman *login* admin kemudian setelah proses validasi selesai dengan hasil valid maka sistem akan menampilkan *dashboar* admin, pada *dashboard* admin ini admin memilih menu berita uji yang kemudian akan menampilkan data berita uji. Jika admin ingin menguji data berita maka admin harus mengklik tombol tambah data lalu akan diarahkan ke halaman *form*

crawling data uji, pada halaman ini admin diharuskan mengisi judul berita dan *link* berita kemudian setelah selesai melakukan tambah data uji sistem akan kembali menampilkan halaman data uji dan data uji pun telah ditambahkan kemudian admin harus menekan tombol *action* yang ada pada sebelah data berita uji yang baru ditambahkan setelah itu halaman pun akan pindah ke halaman hasil klasifikasi berita yang akan menunjukkan berita tersebut termasuk berita positif, netral, atau negatif dan menyimpan data ke database.

4. *Sequence Diagram* Klasifikasi Berita Uji *User*



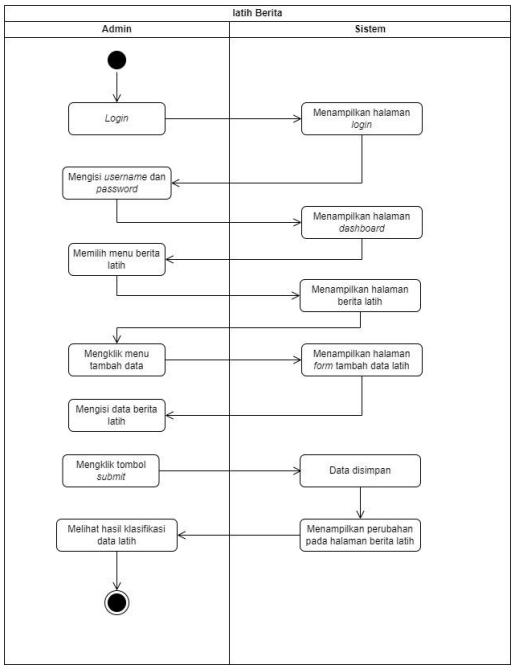
Gambar 4.5 *Sequence Diagram* Klasifikasi Berita Uji *User*

Sequence diagram di atas menggambarkan bagaimana proses klasifikasi data uji yang dilakukan oleh *user*. Ketika *user* memasuki sistem maka akan langsung diditampilkan halaman *dashboard user*, kemudian *user* memilih menu klasifikasi untuk masuk ke halaman klasifikasi yang berisi *form crawling* berita kemudian *user* diharuskan memasukkan *link* URL berita, setelah itu sistem akan melakukan pengecekkan apakah berita tersebut merupakan berita Covid-19 atau bukan dengan mengecek apakah didalam berita tersebut terdapat *keyword* yang dicari atau tidak, jika berita tersebut benar berita Covid-19 maka sistem akan menyimpan data hasil *crawling* ke dalam *database* lalu melakukan klasifikasi menggunakan algoritma K-NN, lalu sistem akan menampilkan hasil klasifikasi kepada *user* dan sistem menyimpan hasil klasifikasi ke dalam *database*.

4.3.3 *Activity Diagram*

Berikut ini merupakan *activity diagram* dari aplikasi yang dibangun.

1. *Activity Diagram* Latih



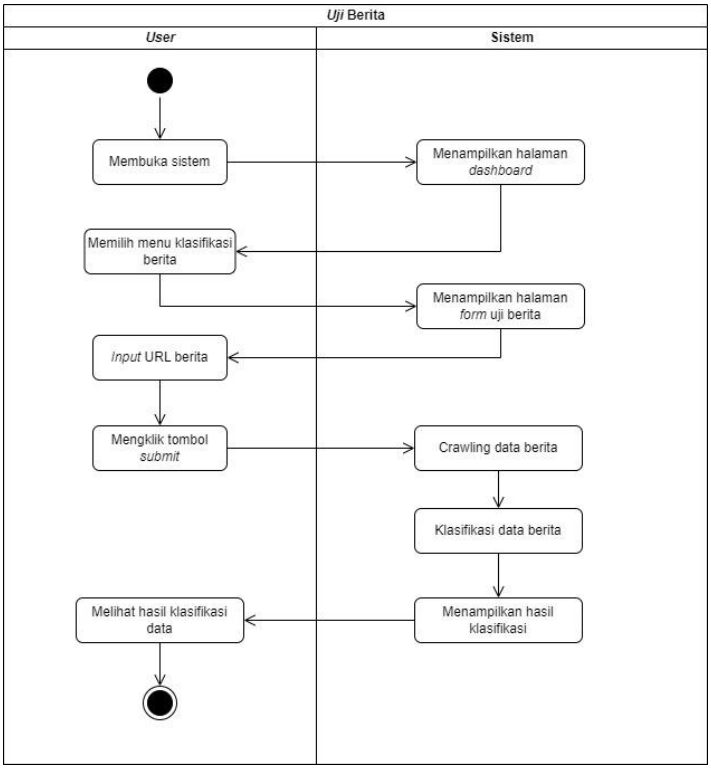
Gambar 4.6 *Activity Diagram* Latih

Activity diagram di atas merupakan diagram aktivitas yang dilakukan oleh sistem dalam melakukan *training*/latih data berita. Untuk melakukan *training* berita memulainya dengan *login* kemudian sistem akan menampilkan halaman *login*, mengisi *username* dan *passwoard* setelah selesai sistem akan menampilkan halaman *dashboard*, memilih menu berita latih lalu sistem menampilkan halaman berita latih, mengklik tombol tambah data kemudian sistem akan menampilkan halaman *form* tambah data latih, mengisi data berita latih dan mengklik tombol *submit* kemudian sistem akan menyimpan data dan menampilkan perubahan pada halaman berita latih, lalu terakhir melihat hasil klasifikasi data latih dan selesai.

2. *Activity Diagram* Uji

Activity diagram yang dilakukan oleh sistem untuk melakukan *testing* data berita ini dimulai dengan membuka sistem lalu sistem akan menampilkan halaman *dashboard*, memilih menu klasifikasi berita lalu sistem akan menampilkan halaman *form* uji berita, *input* URL berita dan mengklik tombol *submit* kemudian sistem akan melakukan *crawling* data berita dan melakukan klasifikasi data berita kemudian sistem menampilkan hasil

klasifikasi, melihat hasil klasifikasi dan selesai. *Activity diagram* yang dilakukan oleh sistem untuk melakukan *testing* data berita dapat dilihat pada gambar di bawah ini.



Gambar 4.7 *Activity Diagram* Uji

4.4 Perancangan *Database*

Perancangan *database* ini meliputi penggunaan tabel-tabel yang akan diaplikasikan pada sistem analisis sentimen terhadap berita Covid-19 pada portal berita detik.com. berikut ini dalah perancangan *database* yang digunakan yaitu sebagai berikut.

1. Tabel *User*

   Tabel *user* merupakan tabel *database* untuk menyimpan data *user*, berikut perancangan data *user*.

Tabel 4.2 Tabel *User*

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	*username*	*Varchar*	Admin

Tabel 4.2 Tabel *User* (Lanjutan)

No	Nama *Field*	*Type*	Keterangan
3	*password*	*Varchar*	Kata sandi
4	nama	*Varchar*	Nama admin
5	email	*Varchar*	Email admin

2. Tabel Berita Latih

   Pada tabel berita latih ini merupakan tabel *database* untuk menyimpan data berita latih. Berikut perancangan tabel berita latih.

Tabel 4.3 Tabel Berita Latih

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	judul	*Text*	Judul berita
3	*link*	*Text*	URL
4	klasifikasi	*Int*	Divisualisasi dengan angka: posistif (1), netral (2), negatif (3)
5	isi	*Text*	Teks berita

3. Tabel Berita Uji

   Berikut tabel *database* berita uji untuk menyimpan data berita uji.

Tabel 4.4 Tabel Berita Uji

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	judul	*Text*	Judul berita
3	isi	*Text*	Teks berita
4	*link*	*Text*	URL
5	klasifikasi	*Int*	Divisualisasi dengan angka: posistif (1), netral (2), negatif (3)

4. Tabel Bobot Kata Latih

   Pada tabel bobot kata latih ini berisi tabel *database* untuk menyimpan bobot kata latih. Berikut rancangan tabel bobot kata latih.

Tabel 4.5 Tabel Bobot Kata Latih

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	kata_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dan seterusnya
3	berita_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dan seterusnya
4	bobot	*double*	Nilai bobot per kata

5. Tabel Bobot Kata Uji

   Tabel bobot kata uji merupakan tabel *database* untuk menyimpan bobot kata uji. Berikut rancanga tabel bobot data uji.

Tabel 4.6 Tabel Bobot Kata Uji

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	kata_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dan seterusnya.
3	berita_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dan seterusnya.
4	bobot	*Float*	Nilai bobot per kata

6. Tabel Hasil K-NN

   Pada tabel hasil K-NN ini merupakan tabel *database* untuk menyimpan hasil *K-Nearest Neighbor*. Berikut rancanga tabel hasil K-NN.

Tabel 4.7 Tabel Hasil K-NN

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*

Tabel 4.7 Tabel Hasil K-NN (Lanjutan)

No	Nama *Field*	*Type*	Keterangan
2	uji_id	*Int*	Data uji divisualisasikan dengan angka 1, 2, 3, dst.
3	latih_id	*Int*	Data latih divisualisasikan dengan angka 1, 2, 3, dst.
4	bobot	*Double*	Nilai bobot
5	klasifikasi	*Int*	Divisualisasi dengan angka: posistif (1), netral (2), negatif (3)

7. Tabel Kata

   Tabel kata ini merupakan tabel *database* untuk menyimpan kata. Berikut rancangan tabel kata.

Tabel 4.8 Tabel Kata

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	kata	*Varchar*	Kata dalam berita

8. Tabel Kata Latih

   Berikut rancangan tabel *database* kata latih untuk menyimpan kata latih.

Tabel 4.9 Tabel Kata Latih

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	kata_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dan seterusnya
3	berita_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dan seterusnya

Tabel 4.9 Tabel Kata Latih (Lanjutan)

No	Nama *Field*	*Type*	Keterangan
4	frekuensi	*Int*	Frekuensi kata pada berita

9. Tabel Kata Uji

Pada tabel kata uji ini merupakan tabel *database* untuk menyimpan kata uji. Berikut rancangan tabel *database* kata uji yang digunakan pada sistem ini.

Tabel 4.10 Tabel Kata Uji

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary key*
2	kata	*Varchar*	Kata pada berita uji
3	berita_id	*Int*	Divisualisasikan dengan angka 1, 2, 3, dst.
4	frekuensi	*Int*	Frekuensi kata pada berita

10. Tabel *Keyword*

Pada tabel *keyword* ini merupakan tabel *database* untuk menyimpan *keyword* atau kata kunci yang digunakan pada sistem ini. Berikut rancangan tabel *keyword*.

Tabel 4.11 Tabel *Keyword*

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary Key*
2	kata	*Varchar*	Kata kunci

11. Tabel *Stopword*

Tabel *stopword* ini adalah tabel *database* untuk menyimpan *stopword*. Berikut ini merupakan rancangan tabel *stopwod*.

Tabel 4.12 Tabel Stopword

No	Nama *Field*	*Type*	Keterangan
1	id	*Int*	*Primary Key*
2	kata	*Varchar*	Kata pada berita yang akan dihilangkan

4.5 Pembahasan

Pembahasan ini bertujuan untuk menganalisis proses-proses yang menunjang kerja sistem dalam menentukan hasil analisis klasifikasi sentimen berita Covid-19 pada portal berita detk.com dengan menggunakan algoritma *K-Nearest Neighbor* secara lebih rinci dari sumber data yang telah dikumpulkan sebelumnya. Proses-proses tersebut yakni sebeagai berikut.

4.5.1 Implementasi Perhitungan *K-Nearest Neighbor*

Metode *K-Nearest Neighbor* digunakan untuk mengklasifikasikan data dalam penelitian ini. Dalam implementasinya, masing-masing data *training* harus diberi label positif, netral, dan negatif terlebih dahulu, serta masing-masing data harus memiliki nilai bobot kata pada masing-masing data yang dapat dihitung menggunakan TF-IDF.

1. Penentuan Data *Training* Positif, Netral, dan Negatif

Penentuan data positif, netral, dan negatif pada data *training* dilakukan berdasarkan dari makna yang terkandung dalam kata pada kalimat. Menurut penulis, suatu kalimat dikatakan positif jika di dalam kalimat terdapat kata-kata pada tabel 4.13 berikut.

Tabel 4.13 Tabel Kata Positif

No	Kata	No	Kata	No	Kata	No	Kata
1	hubung	23	hati	45	kerja	67	informasi
2	sambut	24	khusus	56	cepat	68	potensi
3	giat	25	bersih	47	rawat	69	muncul
4	sehat	26	seru	48	wenang	70	berangkat
5	wajib	27	nasihat	49	sumber	71	teliti
6	pakai	28	capai	50	daya	72	buka
7	hubung	29	bijak	51	ampuh	73	lepas
8	resmi	30	tegas	52	lapang	74	sedia
9	terang	31	sains	53	berkat	75	lindung
10	sumbang	32	nyata	54	tenaga	76	apresiasi
11	konfirmasi	33	terap	55	tahan	77	untung

Tabel 4.13 Tabel Kata Positif (Lanjutan)

No	Kata	No	Kata	No	Kata	No	Kata
12	pimpin	34	transparan	56	layak	78	bantu
13	mudah	35	kasih	57	proses	79	efisien
14	ahli	36	terima	58	saran	80	kebal
15	peluang	37	masuk	59	kembang	81	fakta
16	sambung	38	dekat	60	lengkap	82	luas
17	khas	39	studi	61	percaya	83	produktivitas
18	alam	40	efektif	62	akurat	84	antibodi
19	timbul	41	hidup	63	rutin	85	vaksin
20	tengah	42	cipta	64	paham	86	alamiah
21	unggul	43	gratis	65	mampu	87	bisa
22	karantina	44	tahu	66	mendapat	88	utama

Menurut penulis, suatu kalimat dikatakan negatif jika di dalam kaliamt terdapat kata-kata pada tabel 4.14 berikut.

Tabel 4.14 Tabel Kata Negatif

No	Kata	No	Kata	No	Kata	No	Kata
1	duka	18	sakit	35	wafat	52	tular
2	ganggu	19	kasus	36	tempur	53	omicron
3	bahaya	20	infeksi	37	lawan	54	xbb
4	pandemi	21	mati	38	bludak	55	corona
5	desak	22	pilek	39	rusak	56	picu
6	wabah	23	sumbat	40	parah	57	gejala
7	covid	24	bersin	41	dampak	58	fatal
8	beban	25	batuk	42	rendah	59	rajalela
9	virus	26	dahak	43	darurat	60	putus
10	kalah	27	serak	44	lambat	61	henti
11	dorong	28	nyeri	45	kena	62	salah
12	negatif	29	lonjak	46	limbah	63	hilang
13	tolak	30	jatuh	47	sempit	64	potong

Tabel 4.14 Tabel Kata Negatif (Lanjutan)

No	Kata	No	Kata	No	Kata	No	Kata
14	tuduh	31	beda	48	akibat	65	khawatir
15	ledak	32	landa	49	waspada	66	terobos
16	tutup	33	cabut	50	kritis	67	pasien
17	diskriminatif	34	phk	51	gempur	68	luka

Menurut penulis, suatu kalimat dikatakan netral jika di dalam kalimat terdapat kata-kata positif maupun negatif dengan jumlah kata yang seimbang atau di dalam kalimat tersebut terdapat kata-kata pada tabel 4.15 berikut.

Tabel 4.15 Tabel Kata Netral

No	Kata	No	Kata	No	Kata	No	Kata
1	jakarta	47	undang	90	ketat	137	rebak
2	kementrian	48	kawasan	91	dokter	138	jiwa
3	cina	49	milik	92	pemerintah	139	tawar
4	jepang	50	bawa	93	beijing	140	kabar
5	jumat	51	spike	94	daerah	141	negara
6	imbau	52	transportasi	95	migran	142	tes
7	obat	53	kombinasi	96	pulang	143	pcr
8	mudik	54	masker	97	kampung	144	terbang
9	salur	55	surat	98	sebar	145	aju
10	imlek	56	pribadi	99	kota	146	argumen
11	januari	57	kelompok	100	penduduk	147	protein
12	jalan	58	wakil	101	padat	148	spesifik
13	monoklonal	59	menteri	102	sistem	149	varian
14	temu	60	wartawan	103	saham	150	uni
15	libat	61	buah	104	ekonomi	151	eropa
16	orang	62	arah	105	desa	152	rabu
17	lanjut	63	layan	106	minim	153	tulang
18	usia	64	warga	107	upaya	154	anggota
19	badan	65	tinggal	108	icu	155	laku

Tabel 4.15 Tabel Kata Netral (Lanjutan)

No	Kata	No	Kata	No	Kata	No	Kata
20	hamil	66	rumah	109	syarat	156	uji
21	anak	67	penuh	110	pelancong	157	amerika
22	hasil	68	prospek	111	evolusi	158	serikat
23	otopsi	69	batas	112	ribu	159	harap
24	jam	70	paru	113	booster	160	butuh
25	kritik	71	hong	114	mesti	161	sektor
26	hitung	72	kong	115	pos	162	wisata
27	tindak	73	dana	116	periksa	163	pada
28	balas	74	seberang	117	laut	164	tahap
29	juru	75	darat	118	unit	165	analisis
30	bicara	76	minggu	119	intensif	166	kunjung
31	situasi	77	izin	120	tren	167	pejabat
32	kendali	78	puluh	121	biar	168	asing
33	jamu	79	evaluasi	122	teknis	169	tumbuh
34	presiden	80	mantan	123	lipat	170	yang
35	filipina	81	duta	124	ganda	171	telah
36	bulan	82	surabaya	125	identifikasi	172	ikat
37	hari	83	subvarian	126	inggris	173	erat
38	pekan	84	kuat	127	mutasi	174	redar
39	negeri	85	organisasi	128	peneliti	175	ubah
40	jadwal	86	dunia	129	tempel	176	beber
41	korea	87	who	130	sel	177	deteksi
42	jantung	88	berat	131	ganti	178	ungkap
43	singapura	89	banding	132	imbuh	179	kutip
44	prinsip	90	acu	133	sorot	180	alasan
45	oktober	91	cegah	134	lantas	181	liput
46	gang	92	wilayah	135	setara	182	kira
47	lantik	93	asia	136	hindar	183	gelombang

Untuk melihat perbedaan antara kalimat positif, kalimat netral, dan kalimat negatif dapat dilihat pada contoh seperti pada tabel di bawah ini.

Tabel 4.16 Contoh Kalimat Positif, Netral, dan Negatif

No	Dokumen	Nilai
1	Pemprov Jabar akan memanfaatkan drone disinfektan untuk melawan serta mencegah penyebaran virus corona	Positif
2	Di tengah gempuran virus corona yang membuat industri perhotelan babak belur, ia memprediksi akan terjadinya PHK lagi di bulan April mendatang	Negatif
3	Dua pasien baru tersebut berusia 44 tahun asal Kota Mataram dan usia 46 tahun asal Bali yang bertamu ke wilayah NTB	Netral

Pada kalimat pertama terdapat beberapa kata yang bermakna positif, yaitu: memanfaatkan, disinfektan, mencegah, dan penyebaran. Kata-kata tersebut merupakan kata-kata yang bermakna positif, meskipun terdapat kata melawan, virus, dan corona yang bermakna negatif, namun jumlahnya lebih sedikit dari jumlah kata yang bermakna positif. Sehingga kalimat pertama memiliki nilai positif yang lebih besar dibandingkan dengan nilai negatifnya.

Pada kalimat kedua terdapat beberapa kata yang bermakna negatif, yaitu: gempuran, virus, corona, phk, serta kata babak belur yang memiliki makna negatif. Kata-kata tersebut merupakan kata-kata yang bermakna negatif, serta tidak terdapat kata-kata yang bermakna positif pada kalimat kedua.

Pada kalimat ketiga dapat dilihat bahwa tidak terdapat kata-kata yang bermakna positif maupun negatif. Sehingga dapat disimpulkan bahwa kalimat ketiga memiliki nilai netral.

2. Perhitungan Manual Algoritma TF-IDF

Agoritma TF-IDF digunakan untuk melakukan pembobotan terhadap data *training* berupa teks, hasil pembobotan tersebut akan digunakan untuk menghitung nilai batas yang dibutuhkan dalam algoritma *K-Nearest Neighbor* nantinya. Di bawah ini merupakan contoh perhitungan manual algoritma TF-IDF, diberikan 10

buah data berita yang dibagi menjadi 1 data *testing* dan 9 data *training* yang diberi label sebagai berikut.

Tabel 4.17 Contoh Data Berita

No	Dokumen	Nilai
1	Berita uji (D-L)	?
2	Berita 1 (D-1)	Positif
3	Berita 2 (D-2)	Positif
4	Berita 3 (D-3)	Negatif
5	Berita 4 (D-4)	Negatif
6	Berita 5 (D-5)	Netral
7	Berita 6 (D-6)	Netral
8	Berita 7 (D-7)	Netral
9	Berita 8 (D-8)	Positif
10	Berita 9 (D-9)	Negatif

Algoritma TF-IDF menghitung bobot dari setiap kata, oleh karena itu dokumen yang telah melalui tahap *preprocessing* di atas harus dipecah menjadi kata, sehingga menjadi seperti pada tabel berikut.

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning*

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
1	subvarian	0	0	0	8	3	4	0	0	1	4
2	xbb	0	0	0	8	3	12	0	0	1	4
3	omicron	0	0	0	7	1	7	0	0	1	4
4	gejala	0	0	0	6	0	1	1	0	3	4
5	covid	2	3	10	5	17	4	9	7	3	9
6	infeksi	0	0	0	4	3	1	7	10	3	6
7	varian	1	0	0	3	0	4	2	0	1	5
8	sakit	0	1	0	3	3	1	8	0	1	6
9	van	0	0	0	3	2	0	0	0	0	2
10	amerika	1	1	0	2	3	1	1	0	2	7
11	serikat	1	1	0	2	3	1	1	0	2	7
12	corona	0	8	0	2	1	0	2	0	0	4
13	tular	0	0	0	2	2	0	1	0	0	3
14	dunia	0	0	0	2	1	1	2	0	0	4
15	picu	0	0	0	2	0	0	0	1	0	2

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning* (Lanjutan)

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
16	banding	0	0	0	2	1	1	0	1	0	4
17	cegah	0	0	0	2	1	0	0	0	0	2
18	kerkhove	0	0	0	2	2	0	0	0	0	2
19	virus	1	2	0	2	2	0	6	0	6	6
20	sel	0	0	0	2	0	0	0	0	0	1
21	ubah	0	0	1	2	0	0	0	0	0	2
22	gelombang	0	0	0	2	1	0	3	0	0	3
23	pasien	0	1	0	2	0	0	2	1	4	5
24	daftar	0	0	0	1	0	0	0	0	0	1
25	isi	0	0	1	1	0	0	0	0	0	2
26	ledak	0	0	0	1	0	0	0	0	0	1
27	terang	0	0	0	1	0	0	0	0	0	1
28	organisasi	0	0	0	1	0	1	1	0	0	3
29	sehat	0	0	0	1	0	0	0	0	0	1
30	berat	0	0	1	1	0	0	0	0	0	2
31	rebak	1	0	0	1	1	0	1	0	0	4

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning* (Lanjutan)

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
32	acu	0	0	0	1	0	0	0	0	2	2
33	pusat	0	0	0	1	0	1	0	1	0	3
34	kendali	1	0	0	1	0	1	1	0	0	4
35	cdc	0	0	0	1	0	2	0	0	0	2
36	sebar	2	0	0	1	1	1	2	0	1	6
37	cepat	1	0	0	1	2	0	0	0	0	3
38	wilayah	0	5	0	1	0	0	0	0	0	2
39	sumbang	0	0	0	1	0	1	0	0	0	2
40	konfirmasi	0	0	0	1	0	0	0	0	0	1
41	advertisement	1	1	1	1	1	1	1	1	1	9
42	scroll	1	1	1	1	1	1	1	1	1	9
43	resume	1	1	1	1	1	1	1	1	1	9
44	content	1	1	1	1	1	1	1	1	1	9
45	pimpin	0	0	0	1	1	0	0	0	0	2
46	teknis	0	0	0	1	1	0	1	0	0	3
47	maria	0	0	0	1	1	0	0	0	0	2

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning* (Lanjutan)

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
48	kerkhov	0	0	0	1	0	0	0	0	0	2
49	orang	3	0	4	1	1	0	1	4	2	8
50	lipat	0	0	0	1	0	0	0	0	0	2
51	Ganda	0	0	0	1	0	0	0	0	0	2
52	minggu	1	0	0	1	1	1	4	0	1	7
53	deteksi	0	0	0	1	2	0	0	0	0	3
54	jenewa	0	0	0	1	0	0	0	0	0	2
55	kutip	0	0	1	1	0	1	1	0	1	6
56	detikhealth	0	0	0	1	0	0	0	0	0	2
57	new	0	0	0	1	0	0	0	0	0	2
58	york	0	0	0	1	0	0	0	0	0	2
59	post	0	0	0	1	0	0	0	0	0	2
60	jumat	1	0	0	1	0	1	0	0	0	4
61	alas	0	0	1	1	1	0	0	0	0	4
62	mutasi	0	0	0	1	0	0	2	0	0	3
63	tempel	0	0	0	1	0	0	0	0	0	2

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning* (Lanjutan)

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
64	ganti	0	0	0	1	0	0	0	0	0	1
65	mudah	0	0	0	1	0	0	0	0	0	1
66	imbuh	0	0	0	1	0	0	0	0	0	1
67	ahli	0	0	0	1	2	0	0	0	0	2
68	sorot	0	0	0	1	0	0	1	0	0	2
69	milik	1	1	2	1	1	0	0	3	1	7
70	tara	0	0	0	1	0	0	0	0	0	1
71	hindar	0	0	0	1	0	0	0	0	0	1
72	antibodi	0	0	0	1	0	0	0	4	0	2
73	vaksin	0	0	0	1	0	0	1	6	0	3
74	alamiah	0	0	0	1	0	0	0	5	0	2
75	ikat	0	0	0	1	0	0	0	0	0	1
76	erat	0	0	0	1	0	0	0	0	0	1
77	unggul	0	0	0	1	0	0	0	0	0	1
78	tumbuh	0	0	0	1	0	0	2	0	0	2
79	edar	0	0	0	1	0	0	0	0	0	1

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning* (Lanjutan)

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
80	peluang	0	0	0	1	0	0	0	0	0	1
81	beber	0	0	0	1	0	0	0	0	0	2
82	arti	0	0	0	1	1	0	0	0	0	2
83	mati	0	0	0	1	6	0	6	0	1	4
84	tindak	1	0	0	1	1	0	0	0	0	3
85	hasil	1	1	2	1	0	0	1	1	3	7
86	sambung	0	0	0	1	0	0	0	0	0	1
87	lapor	0	1	0	1	1	1	2	0	0	5
88	perihal	0	0	0	1	0	0	0	0	0	1
89	khas	0	0	0	1	0	0	0	0	0	1
90	alami	0	0	0	1	0	0	0	0	2	2
91	keluarga	0	0	0	1	0	0	0	0	0	1
92	timbul	0	0	0	1	0	0	0	1	2	3
93	tenggorok	0	0	0	1	0	0	0	0	0	1
94	pilek	0	0	0	1	0	0	0	0	0	1
95	hidung	0	0	0	1	0	0	0	0	0	1

Tabel 4.18 Tabel Kata dan Frekuensi Data *Traning* (Lanjutan)

No	Kata	Frekuensi									DF
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	
96	sumbat	0	0	0	1	0	0	0	0	0	1
97	bersin	0	0	0	1	0	0	0	0	0	1
98	batuk	0	0	0	1	0	0	0	0	0	1
99	dahak	0	0	0	1	0	0	0	0	0	1
100	kepala	0	0	0	1	1	0	2	0	0	3
101	suara	0	0	0	1	1	0	0	0	0	2
102	serak	0	0	0	1	0	0	0	0	0	1
103	nyeri	0	0	0	1	0	0	0	0	0	1
104	otot	0	0	0	1	0	0	0	0	0	1
105	indra	0	0	0	1	0	0	0	0	0	1
106	cium	0	0	0	1	0	0	0	0	0	1
107	kraken	0	0	0	1	0	0	0	0	0	1
108	ri	0	0	0	1	0	0	0	0	0	1

Setelah mendapatkan nilai frekuensi untuk setiap kata pada data *training*, selanjutnya adalah memasukkan semua variable ke dalam rumus 2.2. Perhitungan manual untuk mencari bobot kata latih sebagai berikut:

$$W = \text{Frekuensi} \times \text{Log}\left(\frac{Jumlah\ data\ latih}{Jumlah\ data\ latih\ yang\ mengandung\ kata}\right)$$

W-D1($_{subvarian}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D1($_{xbb}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D2($_{subvarian}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D2($_{xbb}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D3($_{subvarian}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D3($_{xbb}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D4($_{subvarian}$) = 8 × log $\left(\frac{9}{4}\right)$ = 2,8175

W-D4($_{xbb}$) = 8 × log $\left(\frac{9}{4}\right)$ = 2,8175

W-D5($_{subvarian}$) = 3 × log $\left(\frac{9}{4}\right)$ = 1,0565

W-D5($_{xbb}$) = 3 × log $\left(\frac{9}{4}\right)$ = 1,0565

W-D6($_{subvarian}$) = 4 × log $\left(\frac{9}{4}\right)$ = 1,4087

W-D6($_{xbb}$) = 12 × log $\left(\frac{9}{4}\right)$ = 4,2262

W-D7($_{subvarian}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D7($_{xbb}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D8($_{subvarian}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D8($_{xbb}$) = 0 × log $\left(\frac{9}{4}\right)$ = 0

W-D9($_{subvarian}$) = 1 × log $\left(\frac{9}{4}\right)$ = 0,3522

W-D9($_{xbb}$) = 1 × log $\left(\frac{9}{4}\right)$ = 0,3522

Sehingga didapat hasil pembobotan semua kata data *training* sebagai berikut.

Tabel 4.19 Tabel Bobot Per Kata Data *Training*

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
1	subvarian	0	0	0	2,8175	1,0565	1,4087	0	0	0,3522
2	xbb	0	0	0	2,8175	1,0565	4,2262	0	0	0,3522
3	omicron	0	0	0	2,4653	0,3522	2,4653	0	0	0,3522
4	gejala	0	0	0	2,1131	0	0,3522	0,3522	0	1,0565
5	covid	0	0	0	0	0	0	0	0	0
6	infeksi	0	0	0	0,7044	0,5283	0,1761	1,2326	1,7609	0,5283
7	varian	0,2553	0	0	0,7658	0	1,0211	0,5105	0	0,2553
8	sakit	0	0,1761	0	0,5283	0,5283	0,1761	1,4087	0	0,1761
9	van	0	0	0	1,9596	1,3064	0	0	0	0
10	amerika	0,1091	0,1091	0	0,2183	0,3274	0,1091	0,1091	0	0,2183
11	serikat	0,1091	0,1091	0	0,2183	0,3274	0,1091	0,1091	0	0,2183
12	corona	0	2,8175	0	0,7044	0,3522	0	0,7044	0	0
13	tular	0	0	0	0,9542	0,9542	0	0,4771	0	0
14	dunia	0	0	0	0,7044	0,3522	0,3522	0,7044	0	0
15	picu	0	0	0	1,3064	0	0	0	0,6532	0
16	banding	0	0	0	0,7044	0,3522	0,3522	0	0,3522	0

Tabel 4.19 Tabel Bobot Per Kata Data *Training* (Lanjutan)

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
17	cegah	0	0	0	1,3064	0,6532	0	0	0	0
18	kerkhove	0	0	0	1,3064	1,3064	0	0	0	0
19	virus	0,1761	0,3522	0	0,3522	0,3522	0	1,0565	0	1,0565
20	sel	0	0	0	1,9085	0	0	0	0	0
21	ubah	0	0	0,6532	1,3064	0	0	0	0	0
22	gelombang	0	0	0	0,9542	0,4771	0	1,4314	0	0
23	pasien	0	0,2553	0	0,5105	0	0	0,5105	0,2553	1,0211
24	daftar	0	0	0	0,9542	0	0	0	0	0
25	isi	0	0	0,6532	0,6532	0	0	0	0	0
26	ledak	0	0	0	0,9542	0	0	0	0	0
27	terang	0	0	0	0,9542	0	0	0	0	0
28	organisasi	0	0	0	0,4771	0	0,4771	0,4771	0	0
29	sehat	0	0	0	0,9542	0	0	0	0	0
30	berat	0	0	0,6532	0,6532	0	0	0	0	0
31	rebak	0,3522	0	0	0,3522	0,3522	0	0,3522	0	0
32	acu	0	0	0	0,6532	0	0	0	0	1,3064

Tabel 4.19 Tabel Bobot Per Kata Data *Training* (Lanjutan)

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
33	pusat	0	0	0	0,4771	0	0,4771	0	0,4771	0
34	kendali	0,3522	0	0	0,3522	0	0,3522	0,3522	0	0
35	cdc	0	0	0	0,6532	0	1,3064	0	0	0
36	sebar	0,3522	0	0	0,1761	0,1761	0,1761	0,3522	0	0,1761
37	cepat	0,4771	0	0	0,4771	0,9542	0	0	0	0
38	wilayah	0	3,2661	0	0,6532	0	0	0	0	0
39	sumbang	0	0	0	0,6532	0	0,6532	0	0	0
40	konfirmasi	0	0	0	0,9542	0	0	0	0	0
41	advertisement	0	0	0	0	0	0	0	0	0
42	scroll	0	0	0	0	0	0	0	0	0
43	resume	0	0	0	0	0	0	0	0	0
44	content	0	0	0	0	0	0	0	0	0
45	pimpin	0	0	0	0,6532	0,6532	0	0	0	0
46	teknis	0	0	0	0,4771	0,4771	0	0,4771	0	0
47	maria	0	0	0	0,6532	0,6532	0	0	0	0
48	kerkhov	0	0	0	0,9542	0	0	0	0	0

Tabel 4.19 Tabel Bobot Per Kata Data *Training* (Lanjutan)

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
49	orang	0,3274	0	0,4366	0,1091	0,1091	0	0,1091	0,4366	0,2183
50	lipat	0	0	0	0,9542	0	0	0	0	0
51	ganda	0	0	0	0,9542	0	0	0	0	0
52	minggu	0,1761	0	0	0,1761	0,1761	0,1761	0,7044	0	0,1761
53	deteksi	0	0	0	0,6532	1,3064	0	0	0	0
54	jenewa	0	0	0	0,9542	0	0	0	0	0
55	kutip	0	0	0,2553	0,2553	0	0,2553	0,2553	0	0,2553
56	detikhealth	0	0	0	0,9542	0	0	0	0	0
57	new	0	0	0	0,9542	0	0	0	0	0
58	york	0	0	0	0,9542	0	0	0	0	0
59	post	0	0	0	0,9542	0	0	0	0	0
60	jumat	0,4771	0	0	0,4771	0	0,4771	0	0	0
61	alas	0	0	0,4771	0,4771	0,4771	0	0	0	0
62	mutasi	0	0	0	0,6532	0	0	1,3064	0	0
63	tempel	0	0	0	0,9542	0	0	0	0	0
64	ganti	0	0	0	0,9542	0	0	0	0	0

Tabel 4.19 Tabel Bobot Per Kata Data *Training* (Lanjutan)

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
65	mudah	0	0	0	0,9542	0	0	0	0	0
66	imbuh	0	0	0	0,9542	0	0	0	0	0
67	ahli	0	0	0	0,6532	1,3064	0	0	0	0
68	sorot	0	0	0	0,6532	0	0	0,6532	0	0
69	milik	0,1091	0,1091	0,2183	0,1091	0,1091	0	0	0,3274	0,1091
70	tara	0	0	0	0,9542	0	0	0	0	0
71	hindar	0	0	0	0,9542	0	0	0	0	0
72	antibodi	0	0	0	0,6532	0	0	0	2,6128	0
73	vaksin	0	0	0	0,4771	0	0	0,4771	2,8627	0
74	alamiah	0	0	0	0,6532	0	0	0	3,2661	0
75	ikat	0	0	0	0,9542	0	0	0	0	0
76	erat	0	0	0	0,9542	0	0	0	0	0
77	unggul	0	0	0	0,9542	0	0	0	0	0
78	tumbuh	0	0	0	0,6532	0	0	1,3064	0	0
79	edar	0	0	0	0,9542	0	0	0	0	0
80	peluang	0	0	0	0,9542	0	0	0	0	0

Tabel 4.19 Tabel Bobot Per Kata Data *Training* (Lanjutan)

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
81	beber	0	0	0	0,6532	0	0	0	0	0
82	arti	0	0	0	0,6532	0,6532	0	0	0	0
83	mati	0	0	0	0,3522	2,1131	0	2,1131	0	0,3522
84	tindak	0,4771	0	0	0,4771	0,4771	0	0	0	0
85	hasil	0,1091	0,1091	0,2183	0,1091	0	0	0,1091	0,1091	0,3274
86	sambung	0	0	0	0,9542	0	0	0	0	0
87	lapor	0	0,2553	0	0,2553	0,2553	0,2553	0,5105	0	0
88	perihal	0	0	0	0,9542	0	0	0	0	0
89	khas	0	0	0	0,9542	0	0	0	0	0
90	alami	0	0	0	0,6532	0	0	0	0	1,3064
91	keluarga	0	0	0	0,9542	0	0	0	0	0
92	timbul	0	0	0	0,4771	0	0	0	0,4771	0,9542
93	tenggorok	0	0	0	0,9542	0	0	0	0	0
94	pilek	0	0	0	0,9542	0	0	0	0	0
95	hidung	0	0	0	0,9542	0	0	0	0	0
96	sumbat	0	0	0	0,9542	0	0	0	0	0

Tabel 4.19 Tabel Bobot Per Kata *Data* Training (Lanjutan)

No	Kata	Bobot								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
97	bersin	0	0	0	0,9542	0	0	0	0	0
98	batuk	0	0	0	0,9542	0	0	0	0	0
99	dahak	0	0	0	0,9542	0	0	0	0	0
100	kepala	0	0	0	0,4771	0,4771	0	0,9542	0	0
101	suara	0	0	0	0,6532	0,6532	0	0	0	0
102	serak	0	0	0	0,9542	0	0	0	0	0
103	nyeri	0	0	0	0,9542	0	0	0	0	0
104	otot	0	0	0	0,9542	0	0	0	0	0
105	indra	0	0	0	0,9542	0	0	0	0	0
106	cium	0	0	0	0,9542	0	0	0	0	0
107	kraken	0	0	0	0,9542	0	0	0	0	0
108	ri	0	0	0	0,9542	0	0	0	0	0
Total		3,8594	7,5589	3,5652	86,134	21,662	15,354	19,116	13,590	10,768

Tabel 4.20 Tabel Kata dan Frekuensi Data *Testing*

No	Kata	Frekuensi	No	Kata	Frekuensi
1	subvarian	8	29	kutip	1
2	xbb	8	30	detikhealth	1
3	omicron	7	31	new	1
4	gejala	6	32	york	1
5	covid	5	33	post	1
6	infeksi	4	34	jumat	1
7	varian	3	35	alas	1
8	sakit	3	36	mutasi	1
9	van	3	37	tempel	1
10	amerika	2	38	ganti	1
11	serikat	2	39	mudah	1
12	corona	2	40	imbuh	1
13	tular	2	41	ahli	1
14	dunia	2	42	sorot	1
15	picu	2	43	milik	1
16	banding	2	44	tara	1
17	cegah	2	45	hindar	1
18	kerkhove	2	46	antibodi	1
19	virus	2	47	vaksin	1
20	sel	2	48	alamiah	1
21	ubah	2	49	ikat	1
22	gelombang	2	50	erat	1
23	pasien	2	51	unggul	1
24	daftar	1	52	tumbuh	1
25	isi	1	53	edar	1
26	ledak	1	54	peluang	1
27	terang	1	55	beber	1
28	organisasi	1	56	arti	1

Tabel 4.20 Tabel Kata dan Frekuensi Data *Testing* (Lanjutan)

No	Kata	Frekuensi	No	Kata	Frekuensi
57	sehat	1	83	mati	1
58	berat	1	84	tindak	1
59	rebak	1	85	hasil	1
60	acu	1	86	sambung	1
61	pusat	1	87	lapor	1
62	kendali	1	88	perihal	1
63	cdc	1	89	khas	1
64	sebar	1	90	alami	1
65	cepat	1	91	keluarga	1
66	wilayah	1	92	timbul	1
67	sumbang	1	93	tenggorok	1
68	konfirmasi	1	94	pilek	1
69	advertisement	1	95	hidung	1
70	scroll	1	96	sumbat	1
71	resume	1	97	bersin	1
72	content	1	98	batuk	1
73	pimpin	1	99	dahak	1
74	teknis	1	100	kepala	1
75	maria	1	101	suara	1
76	kerkhov	1	102	serak	1
77	orang	1	103	nyeri	1
78	lipat	1	104	otot	1
79	Ganda	1	105	indra	1
80	minggu	1	106	cium	1
81	deteksi	1	107	kraken	1
82	jenewa	1	108	ri	1

Setelah mendapatkan nilai frekuensi kata pada data *testing*, selanjutnya adalah memasukkan semua variable ke dalam rumus 2.2 berikut.

$$W = \text{Frekuensi} \times \text{Log}\left(\frac{Jumlah\ data\ latih}{Jumlah\ data\ latih\ yang\ mengandung\ kata + 1}\right)$$

W-DL(subvarian) = $8 \times \log\left(\frac{9}{5}\right) = 2{,}0422$

W-DL(xbb) = $8 \times \log\left(\frac{9}{5}\right) = 2{,}0422$

W-DL(omicron) = $7 \times \log\left(\frac{9}{5}\right) = 1{,}7869$

W-DL(gejala) = $6 \times \log\left(\frac{9}{5}\right) = 1{,}5316$

W-DL(covid) = $5 \times \log\left(\frac{9}{10}\right) = -0{,}2288$

W-DL(infeksi) = $4 \times \log\left(\frac{9}{7}\right) = 0{,}4366$

W-DL(varian) = $3 \times \log\left(\frac{9}{6}\right) = 0{,}5283$

W-DL(sakit) = $3 \times \log\left(\frac{9}{7}\right) = 0{,}3274$

W-DL(van) = $3 \times \log\left(\frac{9}{3}\right) = 1{,}4314$

W-DL(amerika) = $8 \times \log\left(\frac{9}{8}\right) = 0{,}1091$

W-DL(serikat) = $8 \times \log\left(\frac{9}{8}\right) = 0{,}1091$

W-DL(corona) = $2 \times \log\left(\frac{9}{5}\right) = 0{,}5105$

W-DL(tular) = $2 \times \log\left(\frac{9}{4}\right) = 0{,}7044$

W-DL(dunia) = $2 \times \log\left(\frac{9}{5}\right) = 0{,}5105$

W-DL(picu) = $2 \times \log\left(\frac{9}{3}\right) = 0{,}9542$

W-DL(banding) = $2 \times \log\left(\frac{9}{5}\right) = 0{,}5105$

W-DL(cegah) = $2 \times \log\left(\frac{9}{3}\right) = 0{,}9542$

W-DL(kerkhove) = $2 \times \log\left(\frac{9}{3}\right) = 0{,}9542$

W-DL(virus) = $2 \times \log\left(\frac{9}{7}\right) = 0{,}2183$

W-DL(sel) = $2 \times \log\left(\frac{9}{2}\right) = 1{,}3064$

W-DL(ubah) = $2 \times \log\left(\frac{9}{3}\right) = 0{,}9542$

Sehingga didapat hasil pembobotan semua kata data *testing* sebagai berikut.

Tabel 4.21 Tabel Bobot Per Kata Data *Testing*

No	Kata	Bobot	No	Kata	Bobot
1	subvarian	2,0422	29	kutip	0,6532
2	xbb	2,0422	30	detikhealth	0,4771
3	omicron	1,7869	31	new	0,2553
4	gejala	1,5316	32	york	0,4771
5	covid	-0,2288	33	post	0,3522
6	infeksi	0,4366	34	jumat	0,2553
7	varian	0,5283	35	alas	0,4771
8	sakit	0,3274	36	mutasi	0,1091
9	van	1,4314	37	tempel	0,3522
10	amerika	0,1023	38	ganti	0,4771
11	serikat	0,1023	39	mudah	0,4771
12	corona	0,5105	40	imbuh	0,6532
13	tular	0,7044	41	ahli	-0,0458
14	dunia	0,5105	42	sorot	-0,0458
15	picu	0,9542	43	milik	-0,0458
16	banding	0,5105	44	tara	-0,0458
17	cegah	0,9542	45	hindar	0,4771
18	kerkhove	0,9542	46	antibodi	0,3522
19	virus	0,2183	47	vaksin	0,4771
20	sel	1,3064	48	alamiah	0,6532
21	ubah	0,9542	49	ikat	0,0511
22	gelombang	0,7044	50	erat	0,6532
23	pasien	0,3522	51	unggul	0,6532
24	daftar	0,6532	52	tumbuh	0,1091
25	isi	0,4771	53	edar	0,4771
26	ledak	0,6532	54	peluang	0,6532
27	terang	0,6532	55	beber	0,1761
28	organisasi	0,3522	56	arti	0,6532

Tabel 4.21 Tabel Bobot Per Kata Data *Testing* (lanjutan)

No	Kata	Bobot	No	Kata	Bobot
57	sehat	0,6532	83	mati	0,2553
58	berat	0,6532	84	tindak	0,3522
59	rebak	0,6532	85	hasil	0,0511
60	acu	0,3522	86	sambung	0,6532
61	pusat	0,3522	87	lapor	0,1761
62	kendali	0,4771	88	perihal	0,6532
63	cdc	0,6532	89	khas	0,6532
64	sebar	0,6532	90	alami	0,4771
65	cepat	0,6532	91	keluarga	0,6532
66	wilayah	0,6532	92	timbul	0,3522
67	sumbang	0,4771	93	tenggorok	0,6532
68	konfirmasi	0,4771	94	pilek	0,6532
69	advertisement	0,0511	95	hidung	0,6532
70	scroll	0,6532	96	sumbat	0,6532
71	resume	0,6532	97	bersin	0,6532
72	content	0,4771	98	batuk	0,6532
73	pimpin	0,3522	99	dahak	0,6532
74	teknis	0,4771	100	kepala	0,3522
75	maria	0,6532	101	suara	0,4771
76	kerkhov	0,6532	102	serak	0,6532
77	orang	0,6532	103	nyeri	0,6532
78	lipat	0,4771	104	otot	0,6532
79	Ganda	0,6532	105	indra	0,6532
80	minggu	0,6532	106	cium	0,6532
81	deteksi	0,4771	107	kraken	0,6532
82	jenewa	0,4771	108	ri	0,6532
Total			60,065		

3. Perhitungan Manual Algoritma *K-Nearest Neighbor*

Perhitungan algoritma *K-Nearest Neighbor* dapat lakukan setelah semua kata pada dokumen latih dan uji telah memiliki bobot. Perhitungan manual algoritma *K-Nearest Neighbor* dapat dilakukan menggunakan rumus *euclidean distance* 2.1 berikut.

$d = \sqrt{(Data\ latih - Data\ uji)^2}$

$$D\text{-}1 = \sqrt{\begin{array}{c}(0-2{,}0422)^2+(0-2{,}0422)^2+(0-1{,}7869)^2+(0-1{,}5316)^2\\ +\big(0-(-0{,}2288)\big)^2+(0-0{,}4366)^2+(0{,}2553-0{,}5283)^2+\\ (0{,}3522-0{,}1091)^2+(0{,}4771-0{,}3522)^2+(0{,}1023-0{,}1023)^2+\\ (0{,}1023-0{,}1023)^2+(0-0{,}5105)^2+(0-0{,}7044)^2+\\ (0-0{,}5105)^2+(0-0{,}9542)^2+(0-0{,}5105)^2+(0-0{,}9542)^2+\\ (0-0{,}9542)^2+(0{,}1761-0{,}2183)^2+(0-1{,}3064)^2+\\ (0-0{,}9542)^2+(0-0{,}7044)^2+(0-0{,}3522)^2+(0-0{,}6532)^2+\\ (0-0{,}4771)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}3522)^2+\\ (0-0{,}6532)^2+(0{,}2553-0{,}3522)^2+(0{,}3522-0{,}2553)^2+\\ (0-0{,}4771)^2+(0-0{,}3522)^2+(0-0{,}4771)^2+(0-0{,}4771)^2+\\ (0-0{,}3274)^2+(0-1{,}4314)^2+(0-0{,}4771)^2+(0-0{,}4771)^2+\\ (0-0{,}6532)^2+\big(0-(-0{,}0458)\big)^2+(0-(-0{,}0458))^2+\\ \big(0-(-0{,}0458)\big)^2+\big(0-(-0{,}0458)\big)^2+(0-0{,}4771)^2+\\ (0-0{,}4771)^2+(0-0{,}3522)^2+(0-0{,}4771)^2+(0-0{,}6532)^2+\\ (0{,}3274-0{,}0511)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+\\ (0{,}1761-0{,}1091)^2+(0-0{,}4771)^2+(0-0{,}6532)^2+(0-0{,}1761)^2\\ +(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+\\ (0{,}4771-0{,}3522)^2+(0-0{,}3522)^2+(0-0{,}4771)^2+(0-0{,}6532)^2\\ +(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}4771)^2+\\ (0-0{,}4771)^2+(0{,}1091-0{,}0511)^2+(0-0{,}6532)^2+(0-0{,}6532)^2\\ +(0-0{,}4771)^2+(0-0{,}3522)^2+(0-0{,}4771)^2+(0-0{,}6532)^2+\\ (0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}4771)^2+(0-0{,}6532)^2+\\ (0-0{,}6532)^2+(0-0{,}4771)^2+(0-0{,}4771)^2+(0-0{,}2553)^2+\\ (0{,}4771-0{,}3522)^2+(0{,}1091-0{,}0511)^2+(0-0{,}6532)^2+\\ (0-0{,}1761)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}4771)^2+\\ (0-0{,}6532)^2+(0-0{,}3522)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+\\ (0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+\\ (0-0{,}6532)^2+(0-0{,}3522)^2+(0-0{,}4771)^2+(0-0{,}6532)^2+\\ (0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+(0-0{,}6532)^2+\\ (0-0{,}6532)^2+(0-0{,}6532)^2\end{array}}$$

$= \sqrt{0{,}3892}$ $= 0{,}6239$

Hasil kuadrat serta akar dari bobot yang telah dihitung dapat dilihat pada tabel bawah ini.

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor*

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
1	subvarian	0	0	0	0,6011	0,9715	0,4013	0	0	2,8561
2	xbb	0	0	0	0,6011	0,9715	4,7699	0	0	2,8561
3	omicron	0	0	0	0,4602	2,0584	0,4601	0	0	2,0584
4	gejala	0	0	0	0,3381	0	1,3911	1,3911	0	0,2257
5	covid	0	0	0	0	0	0	0	0	0
6	infeksi	0	0	0	0,0717	0,0084	0,0678	0,6337	1,7539	0,0084
7	varian	0,0745	0	0	0,0564	0	0,2429	0,0003	0	0,0745
8	sakit	0	0,0229	0	0,0403	0,0403	0,0229	1,1692	0	0,0229
9	van	0	0	0	0,2791	0,0156	0	0	0	0
10	amerika	4,6777	4,6777	0	0,0134	0,0507	4,6777	4,6777	0	0,0134
11	serikat	4,6777	4,6777	0	0,0134	0,0507	4,6777	4,6777	0	0,0134
12	corona	0	5,3219	0	0,0376	0,0251	0	0,0376	0	0
13	tular	0	0	0	0,0624	0,0624	0	0,05164	0	0
14	dunia	0	0	0	0,0376	0,0251	0,0251	0,0376	0	0
15	picu	0	0	0	0,1240	0	0	0	0,0906	0
16	banding	0	0	0	0,0376	0,0251	0,0251	0	0,0251	0

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor* (Lanjutan)

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
17	cegah	0	0	0	0,1240	0,0906	0	0	0	0
18	kerkhove	0	0	0	0,1240	0,1240	0	0	0	0
19	virus	0,0018	0,0179	0	0,0179	0,0179	0	0,7027	0	0,7027
20	sel	0	0	0	0,3625	0	0	0	0	0
21	ubah	0	0	0,0906	0,1240	0	0	0	0	0
22	gelombang	0	0	0	0,0624	0,0516	0	0,5285	0	0
23	pasien	0	0,0094	0	0,0251	0	0	0,0251	0,0094	0,4474
24	daftar	0	0	0	0,0906	0	0	0	0	0
25	isi	0	0	0,0310	0,0310	0	0	0	0	0
26	ledak	0	0	0	0,0906	0	0	0	0	0
27	terang	0	0	0	0,0906	0	0	0	0	0
28	organisasi	0	0	0	0,0156	0	0,0156	0,0156	0	0
29	sehat	0	0	0	0,0906	0	0	0	0	0
30	berat	0	0	0,0310	0,0310	0	0	0	0	0
31	rebak	0,0094	0	0	0,0094	0,0094	0	0,0094	0	0
32	acu	0	0	0	0,0310	0	0	0	0	0,6877

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor* (Lanjutan)

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
33	pusat	0	0	0	0,0156	0	0,0156	0	0,0156	0
34	kendali	0,00941	0	0	0,0094	0	0,0094	0,0094	0	0
35	cdc	0	0	0	0,0310	0	0,6877	0	0	0
36	sebar	0,0591	0	0	0,0045	0,0045	0,0045	0,0591	0	0,0045
37	cepat	0,0156	0	0	0,0156	0,3625	0	0	0	0
38	wilayah	0	7,7782	0	0,0310	0	0	0	0	0
39	sumbang	0	0	0	0,0310	0	0,0310	0	0	0
40	konfirmasi	0	0	0	0,0906	0	0	0	0	0
41	advertisement	0	0	0	0	0	0	0	0	0
42	scroll	0	0	0	0	0	0	0	0	0
43	resume	0	0	0	0	0	0	0	0	0
44	content	0	0	0	0	0	0	0	0	0
45	pimpin	0	0	0	0,0310	0,0310	0	0	0	0
46	teknis	0	0	0	0,0156	0,0156	0	0,0156	0	0
47	maria	0	0	0	0,0310	0,0310	0	0	0	0
48	kerkhov	0	0	0	0,0906	0	0	0	0	0

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor* (Lanjutan)

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
49	orang	0,0763	0	0,1485	0,0034	0,0034	0	0,0034	0,1485	0,0279
50	lipat	0	0	0	0,0906	0	0	0	0	0
51	ganda	0	0	0	0,0906	0	0	0	0	0
52	minggu	0,0045	0	0	0,0045	0,0045	0,0045	0,3543	0	0,0045
53	deteksi	0	0	0	0,0310	0,6877	0	0	0	0
54	jenewa	0	0	0	0,0906	0	0	0	0	0
55	kutip	0	0	0,0063	0,0063	0	0,0063	0,0063	0	0,0063
56	detikhealth	0	0	0	0,0906	0	0	0	0	0
57	new	0	0	0	0,0906	0	0	0	0	0
58	york	0	0	0	0,0906	0	0	0	0	0
59	post	0	0	0	0,0906	0	0	0	0	0
60	jumat	0,0156	0	0	0,0156	0	0,0156	0	0	0
61	alas	0	0	0,0156	0,0156	0,0156	0	0	0	0
62	mutasi	0	0	0	0,0310	0	0	0,6877	0	0
63	tempel	0	0	0	0,0906	0	0	0	0	0
64	ganti	0	0	0	0,0906	0	0	0	0	0

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor* (Lanjutan)

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
65	mudah	0	0	0	0,0906	0	0	0	0	0
66	imbuh	0	0	0	0,0906	0	0	0	0	0
67	ahli	0	0	0	0,0310	0,6877	0	0	0	0
68	sorot	0	0	0	0,0310	0	0	0,0310	0	0
69	milik	0,0034	0,0034	0,0279	0,0034	0,0034	0	0	0,0763	0,0034
70	tara	0	0	0	0,0906	0	0	0	0	0
71	hindar	0	0	0	0,0906	0	0	0	0	0
72	antibodi	0	0	0	0,0310	0	0	0	4,5613	0
73	vaksin	0	0	0	0,0156	0	0	0,0156	6,3028	0
74	alamiah	0	0	0	0,0310	0	0	0	7,7782	0
75	ikat	0	0	0	0,0906	0	0	0	0	0
76	erat	0	0	0	0,0906	0	0	0	0	0
77	unggul	0	0	0	0,0906	0	0	0	0	0
78	tumbuh	0	0	0	0,0310	0	0	0,6877	0	0
79	edar	0	0	0	0,0906	0	0	0	0	0
80	peluang	0	0	0	0,0906	0	0	0	0	0

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor* (Lanjutan)

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
81	beber	0	0	0	0,0310	0	0	0	0	0
82	arti	0	0	0	0,0310	0,0310	0	0	0	0
83	mati	0	0	0	0,0094	3,4515	0	3,4515	0	0,0094
84	tindak	0,0156	0	0	0,0156	0,0156	0	0	0	0
85	hasil	0,0034	0,0034	0,0279	0,0034	0	0	0,0034	0,0034	0,0763
86	sambung	0	0	0	0,0906	0	0	0	0	0
87	lapor	0	0,0063	0	0,0063	0,0063	0,0063	0,1119	0	0
88	perihal	0	0	0	0,0906	0	0	0	0	0
89	khas	0	0	0	0,0906	0	0	0	0	0
90	alami	0	0	0	0,0310	0	0	0	0	0,6877
91	keluarga	0	0	0	0,0906	0	0	0	0	0
92	timbul	0	0	0	0,0156	0	0	0	0,0156	0,3625
93	tenggorok	0	0	0	0,0906	0	0	0	0	0
94	pilek	0	0	0	0,0906	0	0	0	0	0
95	hidung	0	0	0	0,0906	0	0	0	0	0
96	sumbat	0	0	0	0,0906	0	0	0	0	0

Tabel 4.22 Hasil Perhitungan Manual *K-Nearest Neighbor* (Lanjutan)

No	Kata	K-NN								
		D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9
97	bersin	0	0	0	0,0906	0	0	0	0	0
98	batuk	0	0	0	0,0906	0	0	0	0	0
99	dahak	0	0	0	0,0906	0	0	0	0	0
100	kepala	0	0	0	0,0156	0,0156	0	0,3625	0	0
101	suara	0	0	0	0,0310	0,0310	0	0	0	0
102	serak	0	0	0	0,0906	0	0	0	0	0
103	nyeri	0	0	0	0,0906	0	0	0	0	0
104	otot	0	0	0	0,0906	0	0	0	0	0
105	indra	0	0	0	0,0906	0	0	0	0	0
106	cium	0	0	0	0,0906	0	0	0	0	0
107	kraken	0	0	0	0,0906	0	0	0	0	0
108	ri	0	0	0	0,0906	0	0	0	0	0
Total		0,3892	0,2886	13,1634	0,3789	8,2251	9,9963	8,2028	10,4018	20,7808
Hasil Diakarkan		0,6239	0,5372	3,6281	0,6156	2,8679	3,1617	2,8640	3,2252	4,5586

Kemudian mengurutkan jarak atau nilai kemiripan dari yang terbesar hingga yang terkecil, seperti pada tabel berikut ini.

Tabel 4.23 Mengurutkan Hasil K-NN

Ranking	Dokumen	*Euclidean Distance*	Nilai
1	D-9	4,5586	Positif
2	D-3	3,6281	Negatif
3	D-8	3,2252	Negatif
4	D-6	3,1617	Netral
5	D-5	2,8679	Netral
6	D-7	2,8640	Netral
7	D-1	0,6239	Positif
8	D-4	0,6156	Negatif
9	D-2	0,5372	Positif

Tahap selanjutnya menentukan nilai k. Nilai k yang diambil untuk contoh di atas adalah k=4. Kemudian menentukan kelas yang dominan sebanyak k=4 seperti pada tabel di bawah ini.

Tabel 4.24 Mengurutkan Data Sebanyak k=4

Ranking	Dokumen	*Euclidean Distance*	Nilai
1	D-9	4,5586	Positif
2	D-3	3,6281	Negatif
3	D-8	3,2252	Negatif
4	D-6	3,1617	Netral

Pada tabel di atas, dapat dilihat bahwa kelas negatif terdapat sebanyak 2 kelas, sedangkan kelas positif dan netral masing-masing sebanyak 1 kelas. Maka dapat di simpulkan bahwa D-L diklasifikasikan dalam kelas sentimen negatif.

Dalam penyelesaian penelitian ini diambil beberapa nilai k untuk menentukan parameter k yang akan digunakan dalam penelitian. Berikut ini tabel pemilihan nilai k.

Tabel 4.25 Menentukan Nilai k

Keterangan	K = 4	K = 6	K = 8
Berhasil	37	30	30

Tabel 4.25 Menentukan Nilai k (Lanjutan)

Keterangan	K = 4	K = 6	K = 8
Gagal	13	20	20

Berdasarkan tabel di atas, penulis memilih nilai k=4 dikarenakan lebih optimal dan tingkat kegagalannya lebih rendah.

4.5.2 Pengujian Menggunakan *Confusion Matrix*

Penelitian ini menggunakan 450 data berita latih dan 50 data uji berita Covid-19. Berikut ini merupakan *confusion matrix* untuk mencari nilai *accuracy*, *precission*, *recall*, dan f1-*score* data positif, netral, dan negatif dari 50 data berita Covid-19 yang telah diuji.

Tabel 4.26 Tabel *Confusion Matrix* Data Uji Berita Covid-19

Aktual	Prediksi		
	Positif	Netral	Negatif
Positif	9	4	3
Netral	3	5	0
Negatif	0	3	23

$$Precission\ x = \frac{TP\ x}{Total\ nilai\ kelas\ x\ yang\ diprediksi}$$

$$Precission(positif) = \frac{9}{9+3+0} = 0.75$$

$$Precission(netral) = \frac{5}{4+5+3} = 0.4167$$

$$Precission(negatif) = \frac{23}{3+0+23} = 0.8846$$

$$Macro\ precission = \frac{0.75 + 0.4167 + 0.8846}{3} \times 100 = 68{,}38\%$$

$$Recall\ x = \frac{TP\ x}{Total\ nilai\ kelas\ x\ yang\ harusnya\ aktual}$$

$$Recall(positif) = \frac{9}{9+4+3} = 0.5625$$

$$Recall(netral) = \frac{5}{3+5+0} = 0.625$$

$$Recall(negatif) = \frac{23}{0+3+23} = 0.8846$$

$$Macro\ recall = \frac{0.5625 + 0.625 + 00.8846}{3} \times 100 = 69{,}07\%$$

$$F1 - score\ x = 2 \times \frac{Precission\ x\ \times Recall\ x}{Precission\ x + Recall\ x}$$

$$F1 - score(positif) = 2 \times \frac{0.75 \times 0.5625}{0.75 + 0.5625} = 0.6429$$

$$F1 - score(netral) = 2 \times \frac{0.4167\ \times 0.625}{0.4167 + 0.625} = 0.5$$

$$F1 - score(negatif) = 2 \times \frac{0.8846\ \times 0.8846}{0.8846 + 0.8846} = 0.8846$$

$$Macro\ f1 - score = \frac{0.6429 + 0.5 + 0.8846}{3} \times 100 = 67{,}58\%$$

$$Accuracy = \frac{total\ semua\ TP}{total\ semua\ data\ yang\ diuji} = \frac{9 + 5 + 23}{50} \times 100$$

$$= 74\%$$

Dari hasil di atas untuk algoritma *K-Nearest Neighbor* dengan menggunakan *euclidean distance* dengan data latih sebanyak 450 data berita mendapatkan persentase *precission* sebesar 68,38%, *recall* sebesar 69,07%, f1-*score* sebesar 67,58%, dan *accuracy* sebesar 74% serta sistem memprediksi berita bersifat positif sebanyak 12 berita, berita bersifat netral sebanyak 12 berita, dan berita bersifat negatif sebanyak 26 berita dari 50 data uji berita Covid-19.

4.5.3 Pengujian dengan Menambah Data

Berikut ini merupakan pengujian dengan menggunakan jumlah data latih sebanyak 450, 500, 1000, dan 1500 data berita Covid-19 dengan 50 data *testing* yang sama. Pengujian ini dilakukan untuk mencari tahu apakah banyaknya jumlah data latih mempengaruhi nilai akurasi yang didapat.

Tabel 4.27 Tabel Data Latih

Data Latih	Data		
	Positif	Netral	Negatif
450	150	150	150
500	167	166	167
1000	333	333	334
1500	500	500	500

Berikut tabel *confusion matrix* untuk mencari nilai akurasi dari masing-masing data latih.

1. Data latih 450 data

Tabel 4.28 Tabel *Confusion Matrix* dan Akurasi 450 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	9	4	3	$\frac{9+5+23}{50} \times 100 = 74\%$
Netral	3	5	0	
Negatif	0	3	23	

2. Data latih 500 data

Tabel 4.29 Tabel *Confusion Matrix* dan Akurasi 500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	10	3	3	$\frac{10+7+24}{50} \times 100 = 82\%$
Netral	0	7	1	
Negatif	1	1	24	

3. Data latih 1000 data

Tabel 4.30 Tabel *Confusion Matrix* dan Akurasi 1000 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	15	0	1	$\frac{15+5+21}{50} \times 100 = 82\%$
Netral	2	5	1	
Negatif	1	4	21	

4. Data latih 1500 data

Tabel 4.31 Tabel *Confusion Matrix* dan Akurasi 1500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	15	0	1	$\frac{15+6+21}{50} \times 100 = 84\%$
Netral	1	6	1	
Negatif	1	4	21	

Hasil perhitungan nilai akurasi dari tabel-tabel *confusion matrix* di atas didapatkan hasil sebagai berikut.

Tabel 4.32 Tabel Akurasi Tambah Data Latih

	Data Latih			
	450	500	1000	1500
Akurasi	74%	82%	82%	84%

Pengujian sistem dengan menggunakan jumlah data latih sebanyak 450, 500, 1000, dan 1500 data latih dapat dilihat pada tabel 4.32 di atas bahwa nilai akurasi meningkat dengan bertambahnya jumlah data latih.

Pengujian juga dilakukan dengan menambah jumlah data uji yaitu sebanyak 30 data, 40 data, 50 data, dan 60 data uji. Berikut tabel *confusion matrix* untuk mencari nilai akurasi dari masing-masing data uji.

1. Data uji 30 data

Tabel 4.33 Data Uji 30 Data dengan 450 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	7	2	3	$\frac{7+5+10}{30} \times 100$ $= 73{,}33\%$
Netral	3	5	0	
Negatif	0	0	10	

Tabel 4.34 Data Uji 30 Data dengan 500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	7	2	3	$\frac{7+7+9}{30} \times 100$ $= 76{,}67\%$
Netral	0	7	1	
Negatif	1	0	9	

Tabel 4.35 Data Uji 30 Data dengan 1000 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	11	0	1	$\frac{11+5+8}{30} \times 100$ $= 80\%$
Netral	2	5	1	
Negatif	1	1	8	

Tabel 4.36 Data Uji 30 Data dengan 1500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	11	0	1	$\frac{11+6+9}{30} \times 100 = 86{,}67\%$
Netral	1	6	1	
Negatif	1	0	9	

2. Data uji 40 data

Tabel 4.37 Data Uji 40 Data dengan 450 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	9	4	3	$\frac{9+5+15}{40} \times 100 = 72{,}5\%$
Netral	3	5	0	
Negatif	0	1	15	

Tabel 4.38 Data Uji 40 Data dengan 500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	10	3	3	$\frac{10+7+14}{40} \times 100 = 77{,}5\%$
Netral	0	7	1	
Negatif	1	1	14	

Tabel 4.39 Data Uji 40 Data dengan 1000 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	15	0	1	$\frac{15+5+12}{40} \times 100 = 76{,}91\%$
Netral	2	5	1	
Negatif	1	3	12	

Tabel 4.40 Data Uji 40 Data dengan 1500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	15	0	1	$\frac{15+6+13}{40} \times 100 = 85\%$
Netral	1	6	1	

Tabel 4.40 Data Uji 40 Data dengan 1500 Data Latih (Lanjutan)

Aktual	Prediksi			Accuracy
	Positif	Netral	Negatif	
Negatif	1	2	13	

3. Data uji 50 data

Tabel 4.41 Data Uji 50 Data dengan 450 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	9	4	3	$\frac{9+5+23}{50} \times 100 = 74\%$
Netral	3	5	0	
Negatif	0	3	23	

Tabel 4.42 Data Uji 50 Data dengan 500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	10	3	3	$\frac{10+7+24}{50} \times 100 = 82\%$
Netral	0	7	1	
Negatif	1	1	24	

Tabel 4.43 Data Uji 50 Data dengan 1000 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	15	0	1	$\frac{15+5+21}{50} \times 100 = 82\%$
Netral	2	5	1	
Negatif	1	4	21	

Tabel 4.44 Data Uji 50 Data dengan 1500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	15	0	1	$\frac{15+6+21}{50} \times 100 = 84\%$
Netral	1	6	1	
Negatif	1	4	21	

4. Data uji 60 data

Tabel 4.45 Data Uji 60 Data dengan 450 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	10	6	4	$\frac{10+7+24}{60} \times 100 = 68{,}33\%$
Netral	4	7	2	
Negatif	1	3	24	

Tabel 4.46 Data Uji 60 Data dengan 500 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	10	6	4	$\frac{10+10+224}{60} \times 100 = 773{,}33\%$
Netral	2	10	1	
Negatif	2	1	24	

Tabel 4.47 Data Uji 60 Data dengan 1000 Data Latih

Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	17	2	1	$\frac{17+7+22}{60} \times 100 = 76{,}67\%$
Netral	4	7	2	
Negatif	1	4	22	

Tabel 4.48 Data Uji 60 Data dengan 1500 Data Latih

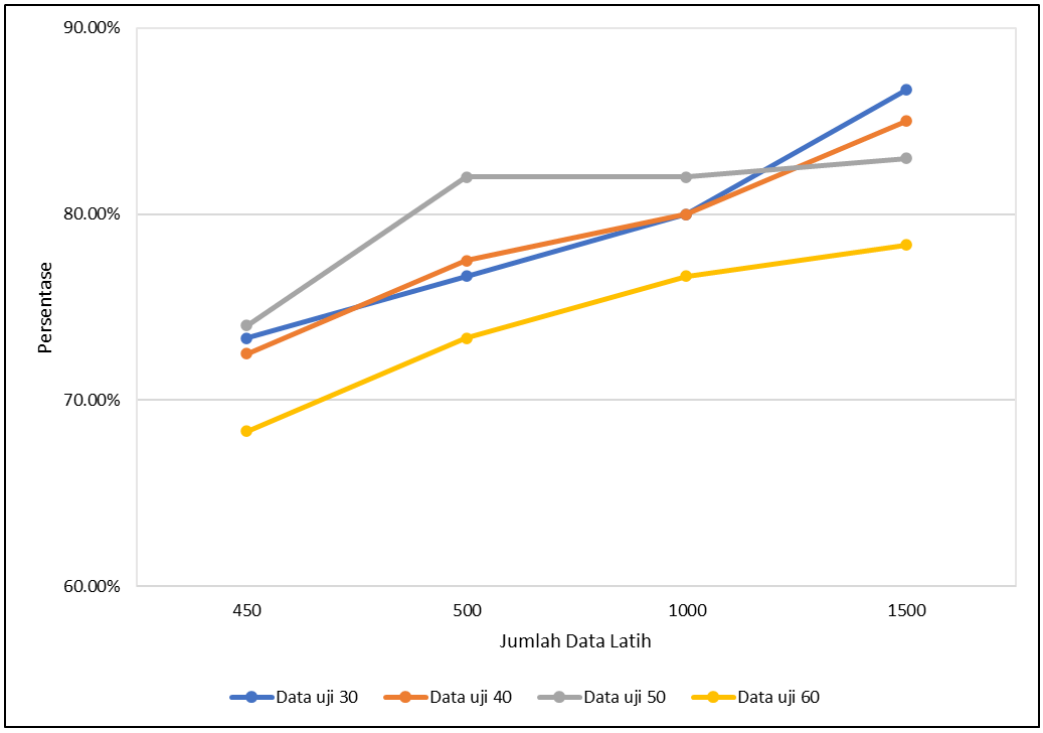
Aktual	Prediksi			*Accuracy*
	Positif	Netral	Negatif	
Positif	17	1	2	$\frac{17+8+22}{60} \times 100 = 78{,}33\%$
Netral	3	8	2	
Negatif	1	4	22	

Hasil perhitungan nilai akurasi dari tabel-tabel *confusion matrix* di atas dapat dilihat pada table dan gambar grafik sebagai berikut.

Tabel 4.49 Tabel Akurasi Tambah Data Uji

Data Uji	Data Latih			
	450	500	1000	1500
30	73.33%	76.67%	80%	86.67%
40	72.50%	77.50%	80%	85%
50	74%	82%	82%	84%
60	68.33%	73.33%	76.67%	78.33%

Tabel 4.49 di atas menunjukkan bahwa terjadi perubahan yang naik turun dalam pengujian menggunakan data uji terhadap data set. Tabel 4.49 dapat dilihat bahwa nilai akurasi cenderung menurun pada setiap penambahan data uji, penurunan nilai akurasi tersebut dapat disebabkan oleh penambahan data-data uji yang belum masuk ke dalam data set, sehingga saat proses uji dilakukan terdapat data-data uji yang tidak dapat dihitung nilai bobotnya sehingga mempengaruhi hasil pengujian. Nilai akurasi mencapai nilai tertinggi saat data latih berjumlah 1500 data dan data uji berjumlah 30 data, sedangkan nilai akurasi terendah terdapat pada data latih berjumlah 450 dan data uji berjumlah 60 data.



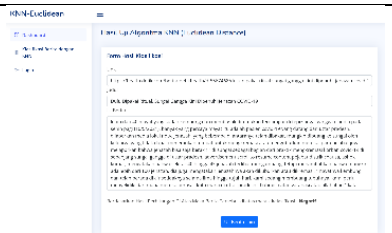
Gambar 4.8 Grafik Persentase Akurasi

Gambar 4.8 di atas memvisualisasikan data nilai akurasi kedalam diagram garis (*line chart*), dimana diagram garis tersebut menunjukkan perbedaan dari hasil pengujian untuk setiap populasi data, dapat dilihat untuk setiap satu kali tahapan pengujian diwakili oleh satu garis. Garis berwarna biru untuk data uji berjumlah 30 data, garis berwarna merah untuk data uji berjumlah 40 data, garis berwarna abu-abu untuk data berjumlah 50 data, dan garis berwarna kuning untuk data uji berjumlah 60 data.

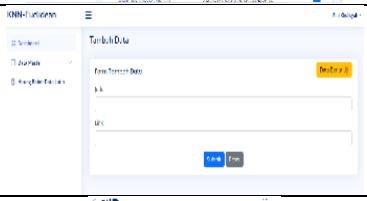
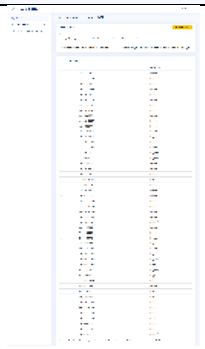
4.5.4 Pengujian Sistem

Proses pengujian sistem pada aplikasi klasifikasi berita Covid-19 dari portal berita detik.com menggunakan *black box testing*. Berikut ini merupakan hasil pengujian sistem dengan *black box*.

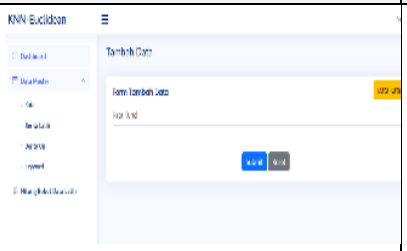
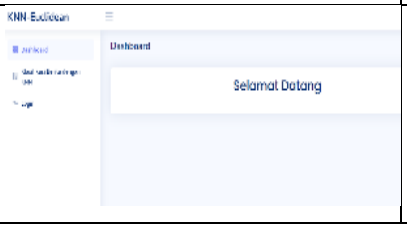
Tabel 4.50 Pengujian Sistem

No	Aksi	Hasil yang Diharapkan	Hasil Pengujian Sistem	Hasil
1	Membuka sistem klasifikasi	Menampilkan halaman *dashboard user*		Valid
2	Mengklik menu klasifikasi berita dengan K-NN	Menampilkan halaman uji berita		Valid
3	Memasukkan URL berita dan mengklik tombol *submit*	Menampilkan halaman hasil klasifikasi		Valid
4	Mengklik menu *login*	Menampilkan halaman *login* admin		Valid

Tabel 4.50 Pengujian Sistem (Lanjutan)

No	Aksi	Hasil yang Diharapkan	Hasil Pengujian Sistem	Hasil
5	Memasukkan data *username* dan *password*	Menampilkan halaman utama admin		Valid
6	Mengklik menu kata	Menampilkan halaman data kata		Valid
7	Mengklik menu berita latih	Menampilkan halaman data berita latih		Valid
8	Mengklik tombol tambah data	Menampilkan halaman *form* tambah data		Valid
9	Mengklik menu berita uji	Menampilkan halaman data berita uji		Valid
10	Mengklik tombol tambah data	Menampilkan halaman *form* tambah data uji		Valid
11	Memasukkan URL, judul berita, dan mengklik tombol *submit*	Menampilkan halaman hasil pengujian dengan K-NN		Valid
12	Mengklik menu *keyword*	Menampilkan halaman *keyword* (kata kunci)		Valid

Tabel 4.50 Pengujian Sistem (Lanjutan)

No	Aksi	Hasil yang Diharapkan	Hasil Pengujian Sistem	Hasil
13	Mengklik tombol tambah data	Menampilkan halaman *form* tambah data *keyword*		Valid
14	Mengklik tombol *sign out*	Kembali ke halaman dashboard user		Valid

4.5.5 Implementasi Sistem

Adapun hasil implementasi sistem yang telah dirancang terdapat beberapa tampilan sebagai berikut.

1. Halaman *User*

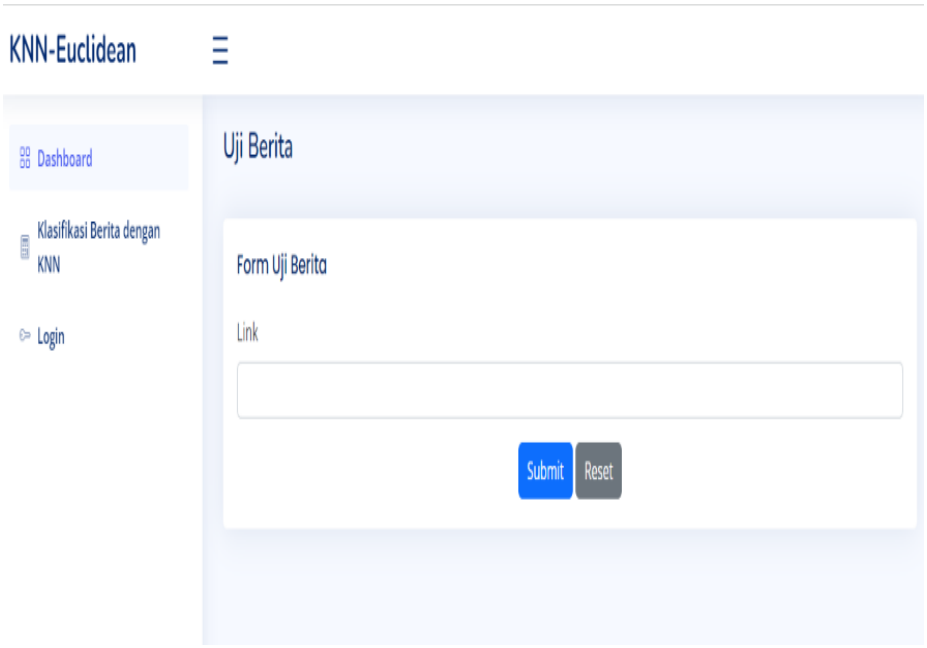
   a. Halaman *Dashboard User*

Gambar 4.9 Halaman *Dashboard User*

Halaman ini adalah halaman *dashboar* untuk *user* sekaligus halaman pertama yang akan muncul ketika masuk ke sistem. Pada halaman ini terdapat beberapa menu seperti menu *dashboard*, klasifikasi berita dengan K-NN, serta menu *login* untuk admin.

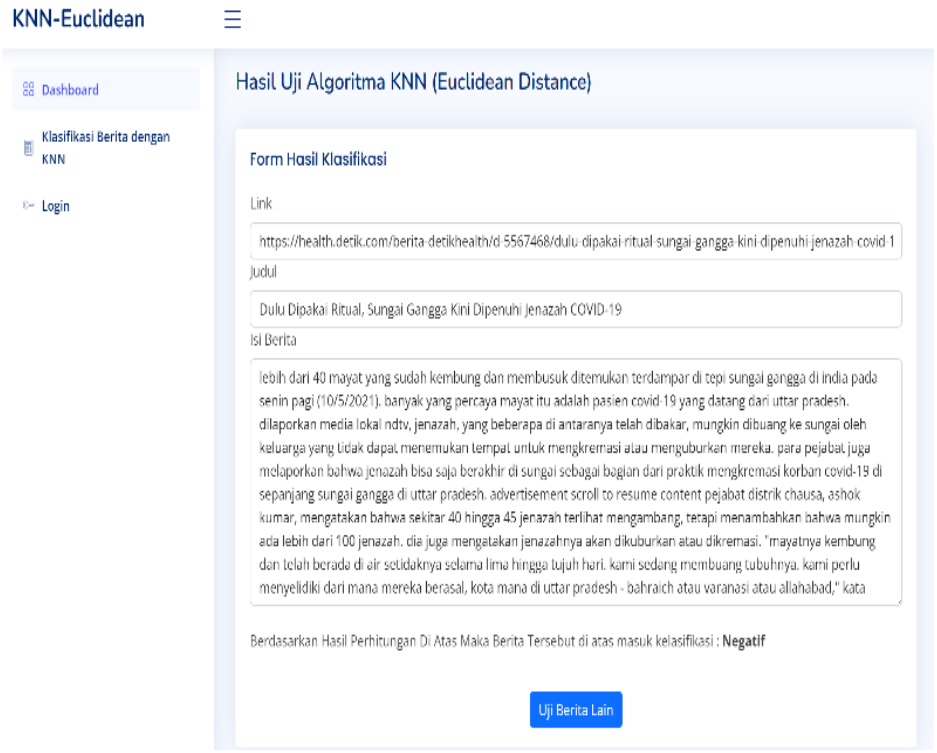
b. Halaman Klasifikasi Berita Uji

Halaman ini merupakan halaman *crawling* data uji yang mana *user* diharuskan memasukkan URL berita dan menekan tombol *submit*.



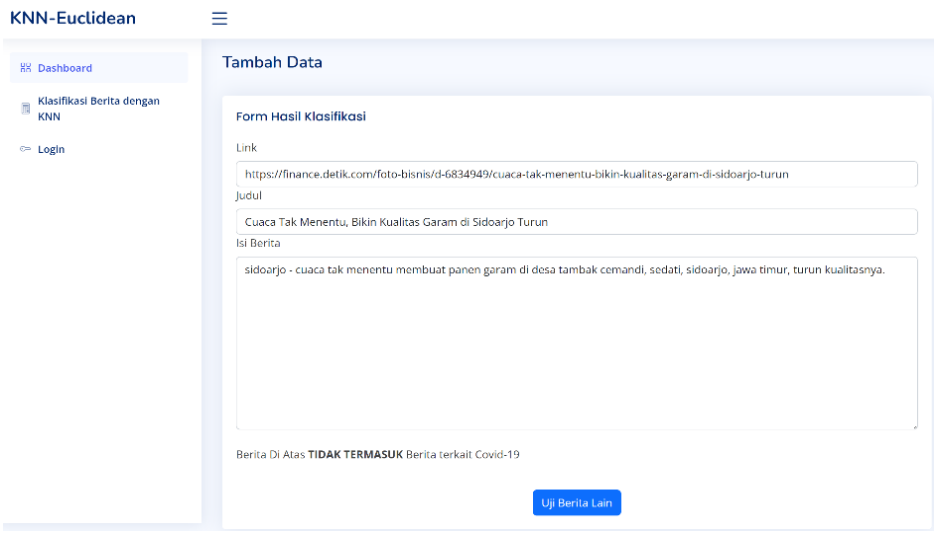
Gambar 4.10 Halaman Klasifikasi Berita Uji

c. Halaman Hasil Klasifikasi



Gambar 4.11 Halaman Hasil Klasifikasi

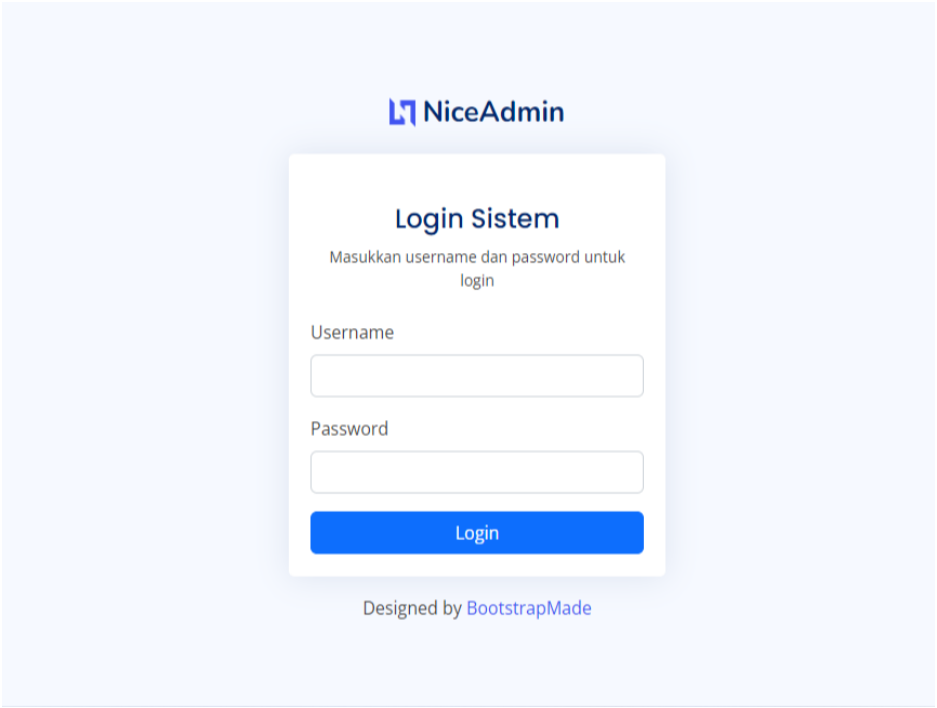
Halaman ini merupakan halaman hasil dari *crawling* dan klasifikasi berita uji yang terdiri dari *link*, judul, isi berita, dan hasil klasifikasi positif, netral, atau negatif. Namun apabila yang dimasukkan bukan termasuk berita Covid-19, maka akan muncul halaman hasil klasifikasi sebagai berikut.



Gambar 4.12 Halaman Klasifikasi Data Uji Bukan Berita Covid-19

2. Halaman Admin

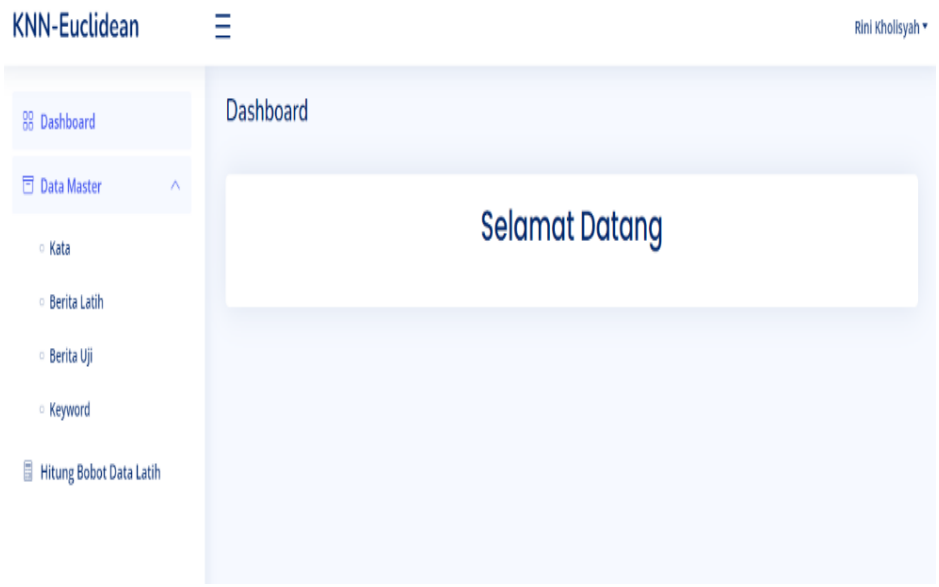
a. Halaman *Login* Admin



Gambar 4.13 Halaman *Login* Admin

Halaman *login* ini diperuntukkan hanya untuk admin. Pada halaman ini admin diharuskan memasukkan *username* dan *password* lalu setelah itu menekan tombol *login*.

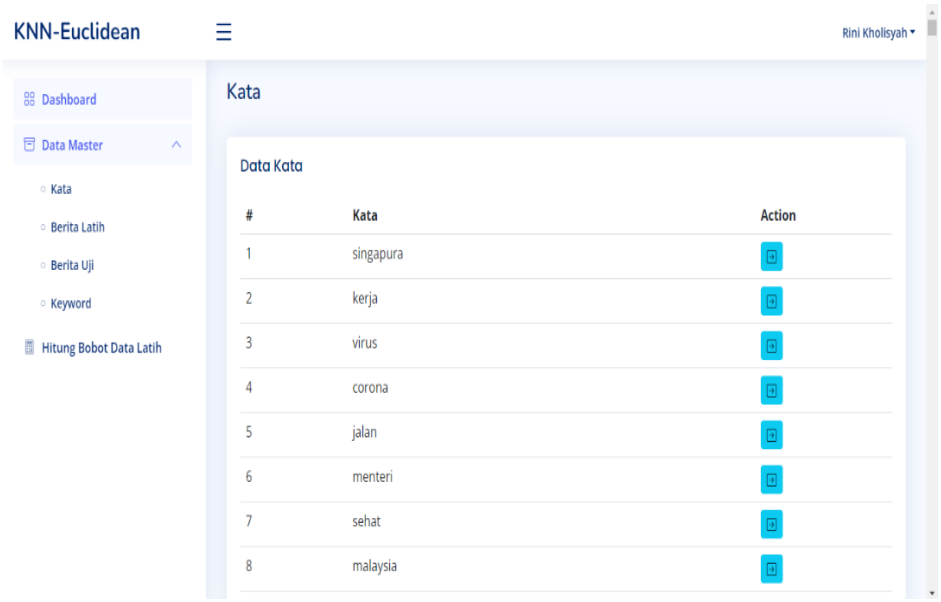
b. Halaman *Dashboard* Admin



Gambar 4.14 Halaman *Dashboard* Admin

Halaman ini merupakan halaman utama admin yang berisi beberapa menu seperti menu *dashboard*, data master (kata, berita latih, berita uji, *keyword*), dan hitung bobot data latih.

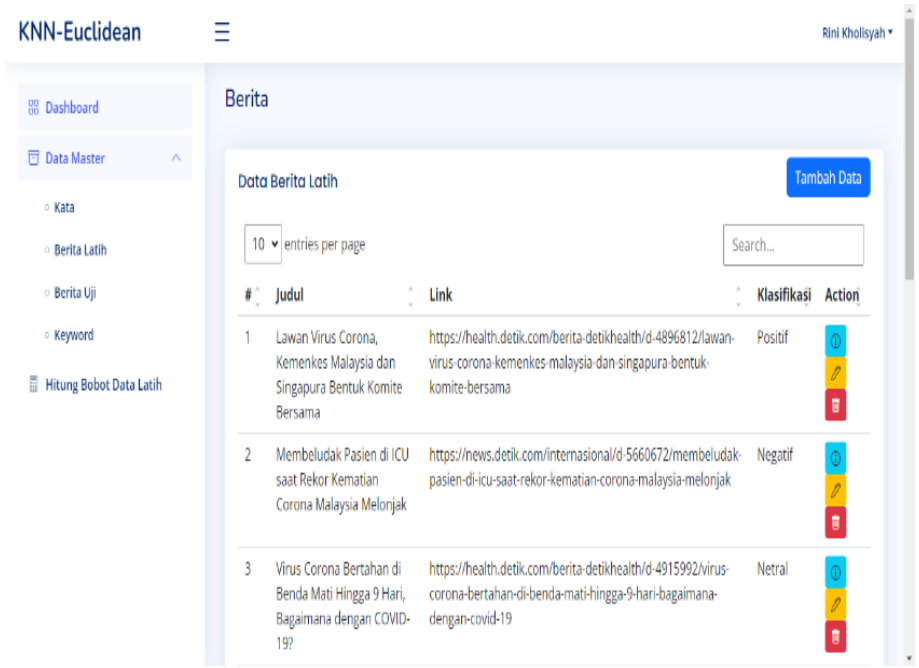
c. Halaman Daftar Kata



Gambar 4.15 Halaman Daftar Kata

Halaman ini berisi daftar kata dari data latih dan data uji serta terdapat tombol untuk menambahkan kata ke dalam *stopword*.

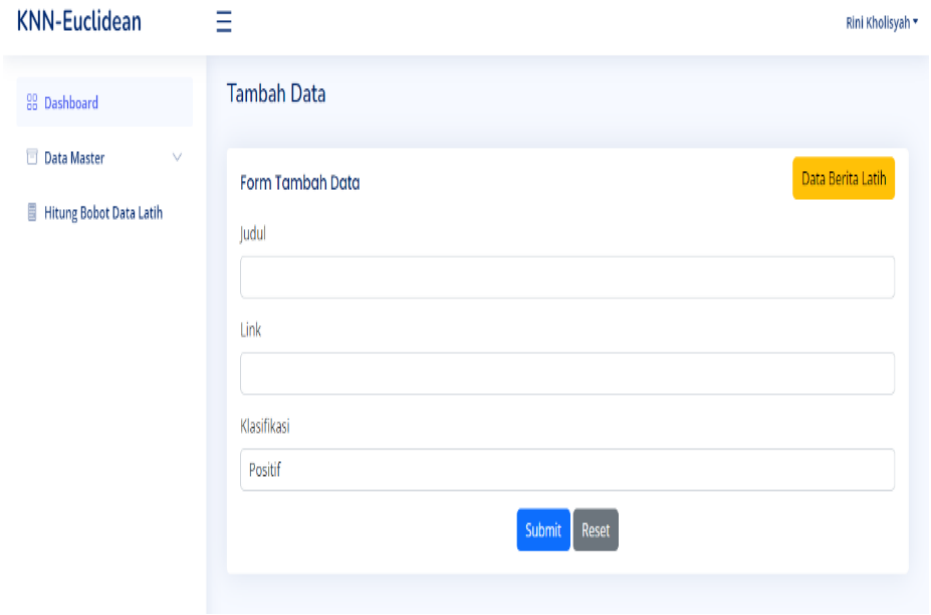
d. Halaman Data Berita Latih



Gambar 4.16 Halaman Data Berita Latih

Halaman ini berisi data berita latih serta admin dapat pula menambah data berita latih baru.

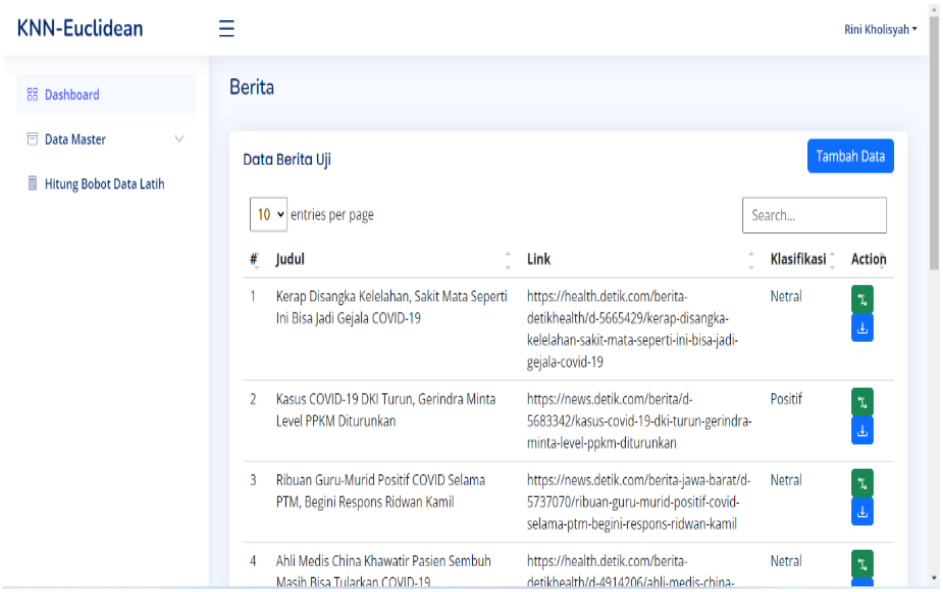
e. Halaman *Form* Tambah Data Latih



Gambar 4.17 Halaman *Form* Tambah Data Latih

Halaman ini berupa *form* tambah data latih yang mana admin diharuskan mengisi judul dan URL berita serta memilih klasifikasi berita berupa positif, netral, atau negatif. Setelah itu admin diharuskan untuk menekan tombol submit untuk menyimpan data baru tersebut.

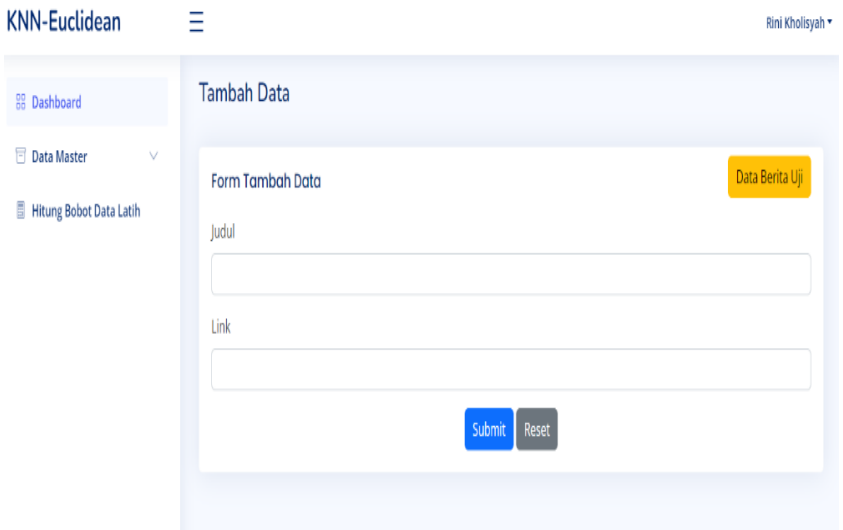
f. Halaman Data Berita Uji



Gambar 4.18 Halaman Data Berita Uji

Halaman ini berisi data-data berita yang pernah diuji serta terdapat beberapa tombol seperti tombol *action* untuk melakukan klasifikasi berita tersebut, serta tombol tambah data latih untuk menambah data berita uji.

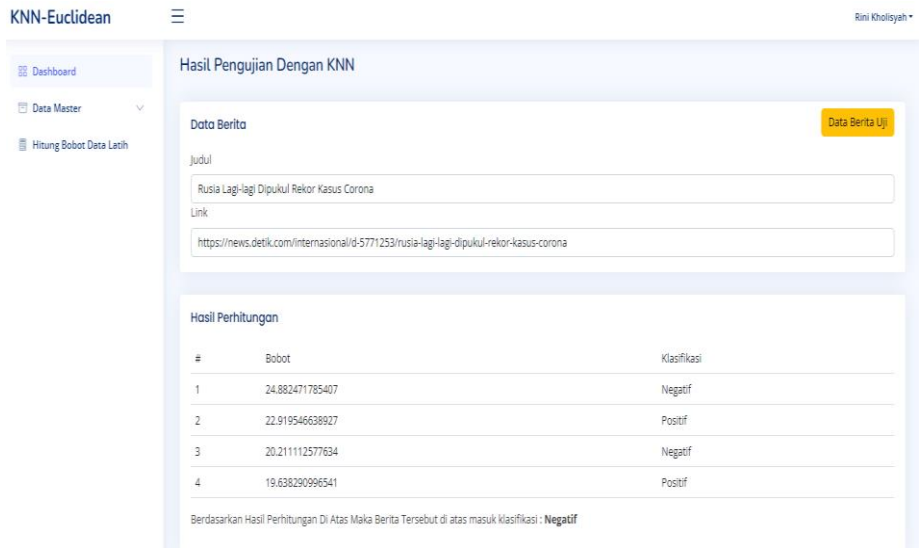
g. Halaman Klasifikasi Data Uji



Gambar 4.19 Halaman Klasifikasi Data Uji

Halaman ini merupakan halaman yang akan muncul ketika admin menekan tombol tambah data. Pada halaman ini terdapat *form* untuk melakukan *crawling* data berita, admin diharuskan mengisi judul dan *link* berita serta menekan tombol *submit*.

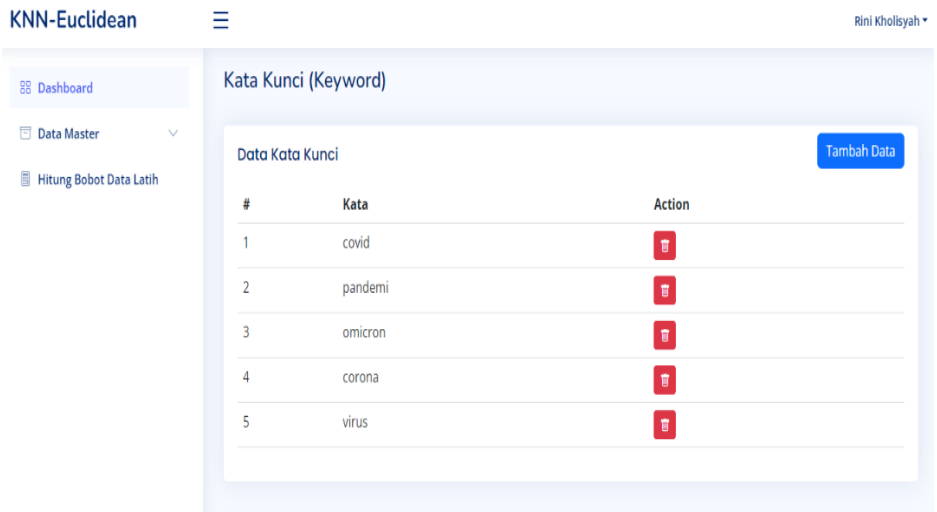
h. Halaman Hasil Klasifikasi Data Uji Admin



Gambar 4.20 Halaman Hasil Klasifikasi Data Uji

Halaman ini merupakan halaman yang akan muncul Ketika admin sudah menambahkan data berita uji serta telah menekan tombol *action*. Pada halaman ini berisi judul dan *link* yang telah diinput sebelumnya serta nilai bobot dan hasil klasifikasi berupa positif, netral, dan negatif.

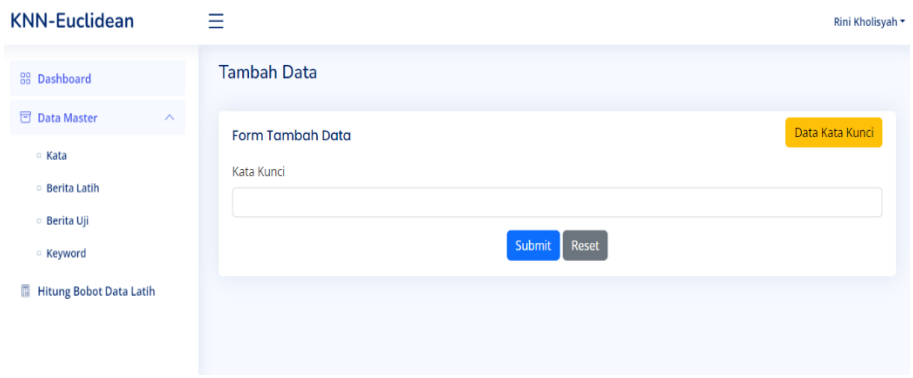
i. Halaman *Keyword*



Gambar 4.21 Halaman *Keyword*

Halaman ini merupakan halaman dari daftar *keyword* atau kata kunci yang digunakan pada sistem ini serta terdapat tombol untuk menambah data dan menghapus data *keyword*.

j. Halaman Tambah Kata Kunci



Gambar 4.22 Halaman Tambah Kata Kunci

Halaman ini merupakan halaman yang akan muncul ketika admin menekan tombol tambah data pada halaman *keyword*. Halaman ini berisi sebuah *form* kata kunci baru yang harus diisi oleh admin, serta tombol *submit* untuk menyimpan kata data baru tersebut.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis sentimen berita Covid-19 pada portal berita detik.com menggunakan algoritma *K-Nearest Neighbor* yang telah dilakukan, maka diperoleh beberapa kesimpulan sebegai berikut.

1. Berdasarkan hasil implementasi, algoritma *K-Nearest Neighbor* dapat digunakan untuk mengklasifikasikan sentimen berita Covid-19 pada portal berita detik.com menjadi tiga kelas yaitu positif, netral, dan negatif.
2. Dalam penelitian ini penulis memilih nilai k = 4 karena memiliki tingkat akurasi yang tinggi dan tingkat *error* yang rendah.
3. Parameter yang digunakan dalam penelitian ini terdiri dari tiga kelas yaitu kelas positif, netral, dan negatif, dimana setiap kelas digunakan 150 data berita sehingga terdapat 450 data berita untuk data latih.
4. Hasil penelitian dari klasifikasi analisis sentimen berita Covid-19 pada portal berita detik.com dari data uji sebanyak 50 data berita Covid-19 menghasilkan 12 data berita yang berifat positif, 12 data berita bersifat netral, dan 26 data berita bersifat negatif.
5. Berdasarkan hasil evaluasi klasifikasi dengan algoitma *K-Nearest Neighbor* dari data uji sebanyak 50 data berita Covid-19 menghasilkan nilai *accuracy* sebesar 74%, *precission* sebesar 68,38%, *recall* sebesar 69,07%, dan f1-*score* sebesar 67,58%.
6. Semakin besar data latih yang digunakan, maka semakin tinggi tingkat keakuratan yang didapat.

5.2 Saran

Setelah mengevaluasi hasil akhir dari penelitian yang telah dilakukan, masih terdapat beberapa kekurangan yang perlu dikembangkan untuk penelitian

selanjutnya. Adapun saran dari penulis yang dapat digunakan untuk penelitian selanjutnya sebagai berikut.

1. *Keyword* yang digunakan masih terbatas sehingga diperlukan *keyword* yang lebih banyak supaya mendapatkan *dataset* yang lebih banyak.
2. Pada penelitian ini, data diambil dari berita pada portal berita detik.com mengenai Covid-19, maka peneliti selanjutnya diharapkan dapat mengambil data dari portal berita lainnya seperti portal berita kompas.com, tribunnews.com, dan portal berita lannya.
3. Penulis menyarankan agar melakukan penambahan *dataset* dan mengkombinasikan metode lain dalam melakukan penelitian analisis sentimen berita Covid-19 ini untuk memperoleh hasil performa yang lebih akurat.

DAFTAR PUSTAKA

Agusta, Y. (2007). *K-Means-Penerapan, Permasalahan dan Metode Terkait*. Jurnal Sistem Dan Informatika, 3(1), 47–60.

Ahmadi, M. I., Gustian, D., & Sembiring, F. (2021). *Analisis Sentiment Masyarakat terhadap Kasus Covid-19 pada Media Sosial Youtube dengan Metode Naive Bayes*. Jurnal Sains Komputer & Informatika (J-SAKTI), 5(2), 807–814. https://doi.org/10.30645/j-sakti.v5i2.378

Anggraeni, H. D., Saputra, R., & Noranita, B. (2013). *Aplikasi Data Mining Analisis Data Transaksi Penjualan Obat Menggunakan Algoritma Apriori (Studi Kasus di Apotek Setya Sehat Semarang*). Journal of Informatics and Technology, 2(2), 22–28.

Asiyah, S. N. (2016). *Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K-Nearest Neighbor*. Skripsi. Institut Teknologi Sepuluh Nopember. Surabaya.

Briliansyah, F. (2020). *Sistem Klasifikasi Kategori Berita Menggunakan Metode K-Nearest Neighbor*. Skripsi. Universitas Islam Negeri Maulana Malik Ibrahim. Malang.

Budiman, S., & Firmansyah, Y. (2015). *Makalah Pembelajaran Mesin KNN (K-Nearest Neighbor)*. Makalah. Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Amikom Purwokerto. Purwokerto.

Covid19.go.id. (2022). *Data Sebaran Perkembangan Covid-19*. https://covid19.go.id. [Diakses, 03 April 2022, Pukul 10:15 WIB].

DataReportal. (2022). *Digital 2022: Indonesia*. https://datareportal.com/reports/digital-2022-indonesia. [Diakses, 05 April 2022, Pukul 07:08 WIB].

Dinata, R. K., Akbar, H., & Hasdyna, N. (2020). *Algoritma K-Nearest Neighbor dengan Euclidean Distance dan Manhattan Distance untuk Klasifikasi Transportasi Bus*. ILKOM Jurnal Ilmiah, 12(2), 104–111. https://doi.org/10.33096/ilkom.v12i2.539.104-111

Dinata, R. K., Fajriana, Zulfa, & Hasdyna, N. (2020). *Klasifikasi Sekolah Menengah Pertama/Sederajat Wilayah Bireuen Menggunakan Algoritma K-Nearest Neighbors Berbasis Web*. Journal of Computer Engineering System and Science, 5(1), 33–37. https://doi.org/10.24114/cess.v5i1.14962

Djufri, M. (2020). *Penerapan Teknik Web Scraping Untuk Penggalian Potensi Pajak (Studi Kasus pada Online Market Place Tokopedia, Shopee dan Bukalapak)*. Jurnal BPPK, 13(2), 65–75. https://doi.org/10.48108/jurnalbppk.v13i2.636

Fairuz, A. L. (2020). *Analisis Sentimen Masyarakat Terhadap Covid-19 pada Media Sosial Twitter Menggunakan Metode K-Nearest Neighbor (K-NN) dan Naive Bayes*. Skripsi. Institut Teknologi Telkom Purwakerto. Purwokerto.

Fauziyyah, A. K. (2020). *Analisis Sentimen Pandemi Covid19 pada Streaming Twitter dengan Text Mining Python*. Jurnal Ilmiah SINUS, 18(2), 31. https://doi.org/10.30646/sinus.v18i2.491

Ginting, S. L., Wendi, Z., & Hamidah, I. (2014). *Dalam Data Mining untuk Memprediksi Masa Studi Mahasiswa Berdasarkan Data Nilai Akademik*. Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST).

Hidayat, W., Utami, E., Iskandar, A. F., Hartanto, A. D., & Prasetio, A. B. (2021). *Perbandingan Performansi Model pada Algoritma K-NN terhadap Klasifikasi Berita Fakta Hoaks Tentang Covid-19*. Edumatic: Jurnal Pendidikan Informatika, 5(2), 167–176. https://doi.org/10.29408/edumatic.v5i2.3664

Josi, A., Abdillah, L. A., & Suryayusra. (2014*). Penerapan Teknik Web Scraping pada Mesin Pencari Artikel Ilmiah*. Jurnal Sistem Informasi (SISFO), 5, 159–164. https://doi.org/10.48550/arXiv.1410.5777

Khomarudin, A. N. (2016). *Teknik Data Mining : Algoritma K-Means Clustering*. Jurnal Ilmu Komputer.

Lestari, M. (2014). *Penerapan Algoritma Klasifikasi Nearest Neighbor (K-NN) untuk Mendeteksi Penyakit Jantung*. Faktor Exacta, 7(4), 366–371. https://doi.org/10.30998/faktorexacta.v7i4.290

Mardi, Y. (2017). *Jurnal Edik Informatika Data Mining : Klasifikasi Menggunakan Algoritma C4.5*. Jurnal Edik Informatika Penelitian Bidang Komputer Sains Dan Pendidikan Informatika, 2(2), 213–219. https://doi.org/10.22202/ei.2016.v2i2.1465

Melita, R. (2018). *Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)*. Skripsi. Universitas Islam Negeri Syarif Hidayatullah. Jakarta.

Razi, A. (2022). *Klasifikasi Penerimaan Beasiswa Aceh Carong (Aceh Pintar) di Universitas Malikussaleh Menggunakan Algoritma KNN (K-Nearest Neighbors)*. Jurnal TIKA, 7(1), 79–84. https://doi.org/10.51179/tika.v7i1.1116

Romadloni, N. T., Santoso, I., & Budilaksono, S. (2019). *Perbandingan Metode Naive Bayes, KNN dan Decision Tree Terhadap Analisis Sentimen Transportasi KRL Commuter Line*. IKRA-ITH INFORMATIKA: Jurnal Komputer Dan Informatika, 3(2), 1–9.

Setiawan, K. Y., Hidayati, H., & Gozali, A. A. (2014). *Analisis User Opinion Twitter pada Level Fine-grained Sentiment Analysis Terhadap Tokoh Publik*. Eprocedings of Engineering 1, 1(1), 639–646.

Setiawan, P. (2018). *Sistem Sentiment Analysis Berita dengan Metode Naive Bayes Classifier*. Skripsi. STIKOM Bali. Bali.

Susanto, S., & Suryadi, D. (2010). *Pengantar Data Mining*. C.V. ANDI OFFSET. Yogyakarta.

Utami, L. A. (2017). *Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Algoritma Support Vector Machine dan K-Nearest Neighbor Berbasis Particle Swarm Optimization*. Jurnal Pilar Nusa Mandiri, 13(1), 103–112. https://doi.org/10.33480/pilar.v13i1.153

Wisdayani, D. S., Nur, I. M., & Wasono, R. (2019). *Penerapan Algoritma K-Nearest Neighbor dalam Klasifikasi Tingkat Keparahan Korban Kecelakaan Lalu Lintas di Kabupaten Jawa Tengah*. Prosiding Mahasiswa Seminar Nasional Unimus, 2.

Zhao, B. (2017). *Web Scraping*. Springer International Publishing AG (Outside the USA), 1–3. https://doi.org/10.1007/978-3-319-32001-4_483-1