DETEKSI PLAGIARISME BERBASIS PARAFRASE PADA TEKS BAHASA INDONESIA

THESIS

Oleh :
FAUZIYAH AMINI
NIM. 200605210002

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

DETEKSI PLAGIARISME BERBASIS PARAFRASE PADA TEKS BAHASA INDONESIA

THESIS

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang Untuk Memenuhi Salah Satu Persyaratan dalam Memperoleh Gelar Magister Komputer (M.Kom)

Oleh:
FAUZIYAH AMINI
NIM. 200605210002

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2022

DETEKSI PLAGIARISME BERBASIS PARAFRASE PADA TEKS BAHASA INDONESIA

THESIS

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang Untuk memenuhi Salah Satu Persyaratan dalam Memperoleh Gelar Magister Komputer (M.Kom)

Oleh:
FAUZIYAH AMINI
NIM. 200605210002

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2022

DETEKSI PLAGIARISME BERBASIS PARAFRASE PADA TEKS BAHASA INDONESIA

THESIS

Oleh :
FAUZIYAH AMINI
NIM. 200605210002

Telah diperiksa dan disetujui untuk diuji:
Tanggal : 7 Desember 2022

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Cahyo Crysdian	Dr. M. Amin Hariyadi
NIP. 19740424 200901 1 008	NIP. 19670118 200501 1 001

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

DETEKSI PLAGIARISME BERBASIS PARAFRASE PADA TEKS BAHASA INDONESIA

THESIS

Oleh:

FAUZIYAH AMINI
NIM. 200605210002

Telah Dipertahankan di Depan Dewan Penguji Thesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal : 7 Desember 2022

Susunan Dewan Penguji

1. Penguji Utama : Dr. Fachrul Kurniawan ST., M.MT ., IPM ( )
   NIP. 19771020 200912 1 001
2. Ketua Penguji : Dr. M. Imamudin Lc, MA
   NIP. 19740602 200901 1 010
3. Sekretaris Penguji : Dr. Cahyo Crysdian ( )
   NIP. 19740424 200901 1 008
4. Anggota Penguji : Dr. M. Amin Hariyadi ( )
   NIP. 19670118 200501 1 001

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Fauziyah Amini

NIM : 200605210002

Program Studi : Magister Informatika

Fakultas : Sains dan Teknologi

Judul Thesis: Deteksi Plagiarisme Berbasis Parafrase Pada Teks Bahasa Indonesia

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan Thesis ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Desember 2022

Yang membuat pernyataan,

Fauziyah Amini

NIM. 200605210002

HALAMAN MOTTO

“Penuhilah hidupmu dengan berbagi manfaat kepada orang lain, dimanapun kamu berada”

HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Puji syukur kehadirat Allah, shalawat dan salam bagi Rasul-Nya

Penulis persembahkan sebuah karya ini kepada:

Kedua orang tua tercinta penulis, Bapak Nurcahyo dan Ibu Kismayanti yang selalu memberikan motivasi agar terus semangat dalam rangka menuntut ilmu. Serta berkat doa merekalah penulis bisa berada sampai di titik pencapaian sekarang ini.

Dosen pembimbing penulis Bapak Dr. Cahyo Crysdian dan Bapak Dr. M. Amin Hariyadi yang selalu memberikan masukan dan saran dalam berjalannya penelitian ini, sehingga penulis dapat menyelesaikan thesis dengan baik.

Seluruh dosen Magister Informatika UIN Maulana Malik Ibrahim Malang, yang telah memberikan ilmu dan pengalaman yang akan berguna pada masa depan penulis kelak.

Orang-orang yang penulis penulis sayangi, yang tak bisa penulis sebutkan satu per satu yang selalu memberikan support dan mendoakan penulis untuk menyelesaikan thesis ini.

Penulis ucapkan terimakasih banyak. Semoga tali persaudaraan kita tetap terjaga dan selalu diridhoi Allah SWT. Aamiin Allahumma Aamiin.

KATA PENGANTAR

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Puji dan syukur kehadirat Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan karuniaNya kepada kita, sehingga penulis bisa menyelesaikan Thesis yang berjudul "Deteksi Plagiarisme berdasarkan paraphrase pada teks Bahasa Indonesia". Tujuan dari penyusunan thesis ini guna memenuhi salah satu syarat untuk bisa menempuh ujian magister komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Magister Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Dalam pengerjaan thesis ini telah melibatkan banyak pihak yang sangat membantu dan memotivasi dalam proses pengerjaan thesis ini. Oleh sebab itu, penulis sampaikan rasa terima kasih kepada:

1. Prof Dr HM. Zainuddin MA, selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini,M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian, Selaku Ketua Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang sekaligus selaku Dosen Pembimbing I.
4. Dr. M. Amin Hariyadi, selaku Dosen Pembimbing II yang telah memberi arahan mulai dari perencanaan penelitian thesis hingga tahapan penyusunan thesis selesai.

5. Para staff laboran dan Admin Program Studi Magister Informatika UIN Malang yang telah membantu dalam urusan administrasi penulis dalam menyelesaikan penyusunan thesis.
6. Orang tua dan keluarga penulis yang telah banyak memberikan doa dan dukungan baik secara moral ataupun materi.
7. Segenap sivitas akademika Program Studi Magister Informatika, terutama seluruh Bapak/Ibu dosen yang telah memberikan segenap ilmu dan bimbingan,
8. Semua pihak yang telah membantu dalam penyusunan thesis ini yang tidak bisa penulis sebutkan semuanya.

Penulis menyadari bahwa dalam thesis ini masih terdapat kekurangan, namun penulis berharap semoga skripsi ini bisa memberikan manfaat untuk UIN Maulana Malik Ibrahim Malang khususnya bagi penulis secara pribadi.

Malang,13 Desember 2022

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Amini, Fauziyah. 2022. **Deteksi Plagiarisme Berdasarkan Parafrase Pada Teks Bahasa Indonesia.** Thesis Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) Dr. Cahyo Crysdian (II) Dr. M. Amin Hariyadi, M.T

Kata Kunci : Klasifikasi , *K-Nearest Neighbor* , Parafrase , *Support Vector Machine*.

Plagiarisme dipandang sebagai pelanggaran ilmiah yang serius, pencurian terhadap ide-ide intelektual. Salah satu bentuk plagiarisme yang sering ditemukan saat ini adalah menjiplak ide orang lain dan menuliskannya dengan kata-kata yang berbeda atau yang biasa kita sebut dengan parafrase kalimat. Maka diperlukan sistem yang mampu melakukan klasifikasi terhadap dokumen parafrase dan non parafrase. Dalam penelitian ini klasifikasi dilakukan menggunakan metode *K-Nearest Neighbor* dan *Support Vector Machine*. Pada pengujian algoritma KNN dilakukan percobaan nilai K terhadap 1 sampai dengan 15 menggunakan *elbow method*. Hasil dari *elbow method* menunjukan bahwa nilai K yang paling optimal pada dataset penelitian ini adalah K=4. Hal ini terbukti dengan hasil akurasi dari nilai K=4 menghasilkan akurasi yang paling tertinggi yaitu sebesar 88%. Pada uji coba algoritma KNN juga dilakukan pengukuran kedekatan jarak antar obyek dengan 4 metode, yaitu *Manhattan Distance*, *Euclidean Distance*, *Minkowsky, Distance*, dan *Chebyshev Distance*. Hasil menunjukan bahwa hasil perhitungan dari ke empat metode yang di uji coba menghasilkan jarak yang sama dalam setiap perhitungan jarak setiap pasangan kalimat. Maka dapat disimpulkan bahwa pada penelitian ini pemilihan pengukuran jarak tidak memberikan pengaruh yang signifikan terhadap performa algoritma knn dan menghasilkan performa yang sama pada setiap metode pengukuran jarak. Berdasarkan hasil uji coba pada penelitian ini algoritma KNN menghasilkan nilai akurasi 88%, *precision* 100% , *recall* 70%, dan *f-measure* 82%. Selanjutnya, pada uji coba yang telah dilakukan pada algoritma *support vector machine* dengan kernel linear menghasilkan performa klasifikasi yang lebih rendah dari hasil performa klasifikasi algoritma KNN dengan nilai akurasi SVM sebesar 83%, *precision* 82% , *recall* 82%, dan *f-measure* 82%.

ABSTRACT

Amini, Fauziyah. 2022. **Plagiarism Detection Based on Paraphrasing in Indonesian Text.** Thesis Informatics Masters Study Program Faculty of Science and Technology. Islamic State of Maulana Malik IbrahimMalang.

Supervisor : (I) Dr. Cahyo Crysdian (II) Dr. M. Amin Hariyadi, M.T

Keyword : Classification , *K-Nearest Neighbor* , Paraphrase , *Support Vector Machine*.

Plagiarism is seen as a serious scientific violation, theft of intellectual ideas. One form of plagiarism that is often found today is plagiarizing other people's ideas and writing them in different words or what we usually call paraphrasing sentences. So we need a system that is able to classify paraphrased and non-paraphrased documents. In this study the classification was carried out using the K-Nearest Neighbor method and the Support Vector Machine. In testing the KNN algorithm, an experiment was carried out on the K value of 1 to 15 using the elbow method. The results of the elbow method show that the most optimal K value in this research dataset is K=4. This is proven by the results of the accuracy of the value of K = 4 resulting in the highest accuracy of 88%. In testing the KNN algorithm, the closeness of the distance between objects was also measured using 4 methods, namely Manhattan Distance, Euclidean Distance, Minkowsky Distance, and Chebyshev Distance. The results show that the calculation results of the four methods tested produce the same distance in each calculation of the distance for each pair of sentences. So it can be concluded that in this study the choice of distance measurement does not have a significant effect on the performance of the KNN algorithm and produces the same performance for each distance measurement method. Based on the test results in this study, the KNN algorithm produces an accuracy of 88%, precision of 100%, recall of 70%, and f-measure of 82%. Furthermore, in trials that have been carried out on the support vector machine algorithm with a linear kernel, the classification performance is lower than the results of the classification performance of the KNN algorithm with SVM accuracy values of 83%, 82% precision, 82% recall, and 82% f-measure.

الملخص

أميني ، فوزية. ٢٠٢٢. كشف الانتحال بناء على إعادة الصياغة على النص الإندونيسي. رسالة الماجستير، قسم المعلومات، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج.

المشرف الأول: د. جاهيو كريسديان. المشرف الثاني: د. أمين هاريادي.

الكلمات الرئيسية: التصنيف، *K-Nearest Neighbor*، إعادة صياغة، آلة المتجهات الداعمة.

ينظر إلى الانتحال على أنه جريمة علمية خطيرة، وسرقة الأفكار الفكرية. أحد أشكال الانتحال التي غالبا ما توجد اليوم هو سرقة أفكار الآخرين وكتابتها بكلمات مختلفة أو ما نسميه عادة إعادة صياغة الجملة. لذلك هناك حاجة إلى نظام قادر على تصنيف المستندات المعاد صياغتها وغير المعاد صياغتها. في هذه الدراسة، تم إجراء التصنيف باستخدام خوارزمية K-Nearest Neighbor و Support Vector Machine. في اختبار خوارزمية KNN، تم إجراء تجربة لقيمة ك مقابل ١ إلى ١٥ باستخدام طريقة الكوع. أشارت نتائج طريقة الكوع إلى قيمة K المثلى في مجموعة بيانات البحث هذه هي ك = ٤. تم إثبات ذلك من خلال دقة قيمة ك = ٤ مما أدى إلى أعلى دقة بنسبة ٨٨%. في تجربة خوارزمية KNN، تم قياس قرب المسافات بين الكائنات أيضا بواسطة ٤ طرق؛ وهي مسافة مانهاتن، وايوكليديان، ومينكوفسكي، و جيبيشيف. أظهرت النتائج أن نتائج حساب الطرق الأربع التي تم اختبارها أنتجت نفس المسافة في كل حساب مسافة لكل من الجمل المقابلة. لذلك، يمكن الاستنتاج منها أن اختيار قياسات المسافة لا يؤثر تأثيرا كبيرا على أداء خوارزمية KNN وأنتجت نفس الأداء في كل طريقة قياس المسافة. بناء على نتائج التجارب في هذه الدراسة، أنتجت خوارزمية KNN قيمة ثبات تبلغ ٨٨%، ودقة ١٠٠%، واسترجاع ٧٠%، وف-قياس ٨٢%. علاوة على ذلك، أسفرت التجارب التي أجريت على خوارزمية آلة المتجهات الداعمة باستخدام نواة خطية عن أداء تصنيف أقل من نتائج أداء تصنيف خوارزمية KNN مع قيمة ثبات SVM تبلغ ٨٣%، ودقة ٨٢%، والاسترجاع ٨٢%، وف-قياس ٨٢%.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Ketersediaaan informasi digital yang semakin luas dan semakin mudah memberikan dampak peningkatan terhadap plagiarisme. Berdasarkan hasil dari beberapa survei penelitian menunjukkan peningkatan kasus plagiarisme baik dalam karya akademis maupun literatur ilmiah (Fatima *et al.* 2019; S. Hanbidge *et al*. 2020). Plagiarisme dipandang sebagai pelanggaran ilmiah yang serius, pencurian terhadap ide-ide intelektual. Salah satu bentuk plagiarisme yang sering ditemukan saat ini adalah menjiplak karya orang lain dan menuliskannya dengan susunan kata yang berbeda atau yang biasa kita sebut dengan parafrase kalimat.

Plagiarisme termasuk salah satu praktik curang karena mengambil hak orang lain dalam hal ini ide, gagasan, atau karya ilmiah dan diakui sebagai miliknya sendiri. Perbuatan curang tentu dilarang dalam agama islam. Pada surat Al-mutaffifin ayat 1 Allah S.W.T berfirman :

وَيْلٌ لِّلْمُطَفِّفِيْنَۙ

Artinya : "Celakalah bagi orang-orang yang curang(dalam menakar dan menimbang)"

Berdasarkan kitab tafsir Al-Madinah Al-Munawwarah ayat ini menjelaskan bahwa Allah memperingatkan manusia dari berbuat curang dalam menunaikan hak orang lain dalam timbangan dan takaran, dan mengancam mereka dengan siksaan dan kebinasaan bagi orang-orang yang mengurangi hak orang lain, jika mereka mengambil barang yang ditimbang atau ditakar dari orang lain maka mereka akan

meminta agar mendapat timbangan dan takaran yang sempurna, namun ketika mereka menimbang atau menakar untuk orang lain maka mereka akan mengurangi hak orang tersebut, atau bahkan mereka minta untuk mendapat lebih dari yang seharusnya mereka dapatkan dari orang lain.

Menurut Tafsir Li Yaddabbaru Ayatih / Markaz Tadabbur di bawah pengawasan Syaikh Prof. Dr. Umar bin Abdullah al-Muqbil, professor fakultas syari'ah Universitas Qashim - Saudi Arabia, surat al-muthafifin diturunkan atas perlakuan orang-orang yang berbuat curang pada timbangan dan takaran dalam jual beli, akan tetapi makna yang terkandung didalamnya juga berlaku bagi siapa saja yang berbuat curang diluar perkara tersebut. Salah satunya yaitu praktik plagiarisme juga bisa dikatakan bentuk kecurangan karena mengakui karya orang lain menjadi karyanya sendiri. Orang yang melakukan plagiarisme terhadap karya orang lain tentu akan mengurangi keberkahan dari ilmu yang dia dapat.

Larangan berbuat curang juga dikuatkan dengan dalil hadist yang diriwayatkan oleh Imam Muslim :

مَنْ غَشَّنَا فَلَيْسَ مِنَّا

"Barangsiapa yang berbuat curang/menipu kepada kami (kaum Muslimin), maka ia bukan termasuk golongan kami."(HR. Muslim)

Hadits tersebut mengandung pelajaran yang amat penting. Perbuatan penipuan, kecurangan yang merugikan orang adalah perbuatan yang tercela. Hal ini bukan saja terbatas masalah jual beli, tapi segala bentuk dan macam kekurangan. Dalam bahasa Arab, kata "man" (siapa saja) menunjukkan pada keumuman.

Artinya, siapa saja yang berbuat curang, menipu orang dalam berbagai bidang yang menyalahi koridor syariat, maka masuk dalam kategori bukan golongan nabi.

Syaikh Al-Albani rahimahullah berkata: “Para ulama mengatakan, “Di antara keberkahan ilmu ialah menisbatkan setiap perkataan kepada orang yang mengatakannya,” karena yang demikian itu lebih selamat dari pemalsuan. Praktik plagiarisme ini juga termasuk perbuatan tidak jujur karena mengakui ide orang lain sebagai idenya sendiri. Salah satu akhlak terpuji dari Rasulullah SAW adalah shidiq (jujur). Shidiq berarti benar atau jujur. Sifat jujur juga sangat dijunjung tinggi dalam Islam. Maka setiap muslim juga harus memiliki sikap jujur di mana dan kapanpun berada sebagaimana telah dicontohkan oleh rasulullah. Nabi SAW menjadikan kejujuran sebagai asas dari setiap kebaikan, sebagaimana sabdanya :

عَنْ عَبْدِ اللَّهِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ عَلَيْكُمْ بِالصِّدْقِ فَإِنَّ الصِّدْقَ يَهْدِي إِلَى الْبِرِّ وَإِنَّ الْبِرَّ يَهْدِي إِلَى الْجَنَّةِ وَمَا يَزَالُ الرَّجُلُ يَصْدُقُ وَيَتَحَرَّى الصِّدْقَ حَتَّى يُكْتَبَ عِنْدَ اللَّهِ صِدِّيقًا وَإِيَّاكُمْ وَالْكَذِبَ فَإِنَّ الْكَذِبَ يَهْدِي إِلَى الْفُجُورِ وَإِنَّ الْفُجُورَ يَهْدِي إِلَى النَّارِ وَمَا يَزَالُ الرَّجُلُ يَكْذِبُ وَيَتَحَرَّى الْكَذِبَ حَتَّى يُكْتَبَ عِنْدَ اللَّهِ كَذَّابًا

Dari Abdullah dia berkata; Rasulullah shallallahu ‘alaihi wasallam bersabda: ‘Kalian harus berlaku jujur, karena kejujuran itu akan membimbing kepada kebaikan. Dan kebaikan itu akan membimbing ke surga. Seseorang yang senantiasa berlaku jujur dan memelihara kejujuran, maka ia akan dicatat sebagai orang yang jujur di sisi Allah. Dan hindarilah dusta, karena kedustaan itu akan menggiring kepada kejahatan dan kejahatan itu akan menjerumuskan ke neraka. Seseorang yang senantiasa berdusta dan memelihara kedustaan, maka ia akan dicatat sebagai pendusta di sisi Allah.” (HR. Bukari, Muslim, Tirmidzi dan Ahmad ibn Hanbal).

Penelitian yang membahas tentang plagiarisme sudah dilakukan sejak tahun 1990-an. Namun deteksi plagiarisme hanya didasarkan pada kata-kata yang sama

kemudian dihitung presentase kemiripan antar dokumen. Metode seperti ini masih belum maksimal untuk menangani plagiarisme yang semakin marak saat ini. Karena saat ini bentuk tindakan plagiarisme tidak hanya menjiplak karya orang lain, namun pelaku plagiarisme saat ini mengambil karya ilmiah atau ide orang lain kemudian dituliskan kembali dengan makna yang sama tetapi dengan susunan kata yang berbeda. Tindakan plagiarisme seperti ini biasa kita sebut dengan parafrase kalimat.

Identifikasi Parafrase atau *Natural Language Sentence Machine* (NLSM) adalah salah satu hal yang menantang dalam pemrosesan teks. Dimana peneliti harus mengidentifikasi apakah sebuah kalimat adalah parafrase dari kalimat lain di pasangan kalimat yang diberikan. Parafrase kalimat menyampaikan arti yang sama tetapi struktur dan urutan kata-katanya bervariasi. Ini adalah suatu hal menantang karena sulit untuk menyimpulkan konteks yang tepat dalam sebuah kalimat. Parafrase terjadi ketika teks dimodifikasi secara leksikal atau sintaksis (Clough dan Stevenson, 2011) agar terlihat berbeda dari sumbernya, tetapi tetap memiliki makna yang sama. Parafrase itu sendiri legal bila dilakukan dengan benar seperti dalam penggunaan kembali teks Jurnalistik (Piao *et. al* 2002), tetapi ketika teks dimodifikasi dan digunakan tanpa menyebutkan sumbernya dengan benar, itu adalah termasuk tindakan plagiarisme.

Berdasarkan permasalahan yang telah diuraikan sebelumnya, sistem pendeteksi plagiarisme konvensional perlu modifikasi agar mendapatkan hasil yang maksimal. Penelitian ini mengeksplorasi berbagai algoritma *machine learning* untuk memodelkan teks yang di parafrase dengan teks yang lain. Dengan adanya

penelitian ini diharapkan menjadi solusi untuk mencegah tindakan plagiarisme yang semakin marak terjadi pada saat ini.

1.2 Pernyataan Masalah

1. Mengapa plagiarisme berdasarkan parafrase pada teks bahasa Indonesia sulit untuk diidentifikasi?
2. Metode apakah yang dapat secara optimal mendeteksi plagiarisme berdasarkan parafrase pada teks bahasa Indonesia?

1.3 Tujuan Penelitian

1. Menganalisa kompleksitas masalah plagiarisme berdasarkan parafrase pada teks bahasa Indonesia.
2. Membandingkan keunggulan metode-metode klasifikasi dalam mendeteksi plagiarisme berdasarkan parafrase pada teks bahasa Indonesia.

1.4 Batasan Masalah

Data yang digunakan dalam penelitian ini adalah dokumen berbentuk teks dalam bahasa Indonesia

1.5 Manfaat Penelitian

1. Dapat membantu instansi pendidikan dalam mencegah praktik plagiarisme dalam lingkungan instansi pendidikan
2. Deteksi plagiarisme memberikan dukungan kepada pelaku akademik untuk membangun nilai-nilai kejujuran, keadilan, dan kepercayaan dalam mengejar integritas akademik.

3. Dapat membantu instansi pendidikan dalam mengembangkan konsep integritas pendidikan yang salah satunya mencakup deteksi plagiarisme.

1.6 Sistematika Penulisan

Sistematika penulisan Thesis ini sebagai berikut:

BAB I. PENDAHULUAN (INTRODUCTION)

Pada bab pendahuluan menguraikan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi dan sistematika penyusunan laporan. Uraian di bab ini memberikan gambaran kepada pembaca terkait maksud dan tujuan dalam penelitian yaitu tentang deteksi plagiarisme berdasarkan paraphrase pada teks Bahasa Indonesia.

BAB II STUDI PUSTAKA (LITERATURE REVIEW)

Dalam tinjauan pustaka akan menguraikan penelitian teoritis dan ilmiah yang berkaitan dengan proses dan metode yang digunakan dalam penelitian yang diambil dari berbagai sumber referensi seperti buku, jurnal, dan sumber yang valid.

BAB III s/d N. DESAIN PENELITIAN

Pada bab desain penelitian ini menjelaskan tentang pola dan desain penelitian, bahan atau materi penelitian, alat, jalannya penelitian, dan analisis hasil penelitian.

BAB N+1. PEMBAHASAN (DISCUSSION)

Bab ini menjelaskan hasil yang diperoleh dalam bentuk penjelasan teoritis, kualitatif, kuantitatif atau statistik. Dalam hal ini hasil yang diperoleh dari ujicoba setiap metode untuk dikupas dan dibandingkan dengan hasil ujicoba metode lainnya serta melalui perspektif Informatika dan inspirasi Al-Qur'an dan Hadist.

BAB N+2. KESIMPULAN (CONCLUSION)

Berisi kesimpulan dari hasil penelitian dan saran terkait penelitian yang dilakukan untuk memperbaiki model yang dibangun dalam mencapai hasil yang lebih baik di masa yang akan datang.

BAB II
STUDI PUSTAKA

2.1 Sistem Deteksi Plagiarisme Berdasarkan Parafrase

Berbagai metode untuk menangani plagiarisme berdasarkan parafrase telah diusulkan dalam beberapa penelitian. Diantaranya, yaitu:

Alvi *et al.* (2021) mengusulkan metode untuk mengidentifikasi dua jenis parafrase, yaitu : substitusi sinonim dan penataan ulang kata (*reordering words*) dalam pasangan kalimat yang diparafrasekan dan dijiplak. Penelitian ini dilakukan melalui tiga tahap pendekatan, menggunakan pencocokan konteks penyisipan kata yang telah dilatih sebelumnya untuk mengidentifikasi substitusi sinonim dan penataan ulang kata. Metode yang digunakan untuk mengidentifikasi pengurutan ulang kata (*reordering words*) menggunakan permutasi dan pola parafrase. Untuk mendeteksi substitusi sinonim, menggunakan Algoritma *Smith Waterman* dan penyematan kata (*word embeddings*) yang telah dilatih Concept Net Numberbatch mengungguli kombinasi metode penyelarasan dan penyematan kata lainnya. Penelitian ini memberikan hasil deteksi pengurutan ulang kata dalam hal *precision*, *recall*, dan *f1score*. Dapat diamati bahwa *f1score* keseluruhan adalah 0,905. Hasil deteksi substitusi sinonim dalam pasangan kalimat parafrase menunjukan bahwa Algoritma *Smith Waterman* dengan *Concept-Net Numberbatch* menghasilkan *f1score* 0,80184. Dimana algoritma *Smith Waterman* merupakan metode yang terbaik dalam menangani substitusi sinonim pada teks.

Ehan *et al.* (2019) melakukan identifikasi parafrase menggunakan berbagai model *machine learning* dan membuat perbandingan kinerja di antara model

machine learning tersebut. Model yang digunakan diantaranya adalah Regresi Logistik, Support Vector Machines, dan arsitektur yang berbeda dari *Neural Networks*. Di antara model yang dibandingkan, seperti yang diharapkan, *Recurrent Neural Network* (RNN) paling cocok untuk tugas identifikasi parafrase. Hasil penelitian ini menunjukan akurasi terendah yang diperoleh adalah regresi logistik menggunakan *one-hot encoding*. Akurasinya sekitar 0,43 sementara SVM tampil sedikit lebih baik di 0,45. Dalam penelitian ini, menggunakan *N-gram* sebagai fitur, ukuran ruang fitur adalah 1300 dan menggunakan kata-kata sebagai fitur, ukuran ruang fitur sekitar 80000. Selanjutnya, input direpresentasikan sebagai penyisipan *word2vec* dan hasil dari dua model yang sama ditingkatkan secara signifikan. *Word2vec* menghasilkan representasi input yang padat dengan mempertimbangkan semua konteks teks yang diberikan. Akurasi untuk dua model adalah sekitar 0,6 dan 0,62. Hasil dari berbagai model dengan menggunakan Neural Network menunjukan akurasi tertinggi diperoleh model LSTM. Seperti yang diharapkan, model LSTM memberikan performa lebih baik daripada varian jaringan saraf lainnya karena lebih cocok untuk data urutan.

Rajkumar *et al.* (2014) telah mengembangkan teknik deteksi plagiarisme *monolingual* untuk mengatasi kasus plagiarisme parafrase. Deteksi parafrase berbasis *machine learning*, yang beroperasi dengan mengekstraksi fitur leksikal, sintaksis, dan semantik, telah digunakan untuk mendeteksi plagiarisme dalam bagian teks. Support Vector Machine berbasis pengenal parafrase digunakan untuk mengklasifikasikan pasangan kalimat masukan sebagai parafrase dengan mengekstraksi berbagai fitur leksikal, sintaksis, dan semantik dari pasangan

kalimat. Sistem diuji pada tiga korpora yang berbeda: *Webis CPC*, *subset METER*, dan *Wikipedia Rewrite Corpus*, baik di tingkat bagian maupun di tingkat kalimat, dengan pendekatan tingkat bagian menunjukkan kinerja yang lebih baik. Sistem ini juga menunjukkan kinerja yang sebanding atau lebih baik jika dibandingkan dengan sistem dengan kinerja terbaik pada ketiga *corpora*. Selanjutnya, sistem juga diuji pada berbagai subkategori korpus P4P dengan hasil yang baik. Ini menunjukkan bahwa menggunakan teknik pengenalan parafrase adalah arah yang menjanjikan untuk dieksplorasi dalam pengembangan sistem deteksi plagiarisme.

Foltýnek *et al.* (2019) merangkum penelitian tentang metode komputasi untuk mendeteksi plagiarisme akademik dengan meninjau secara sistematis 239 makalah penelitian yang diterbitkan antara tahun 2013 dan 2018. Untuk menyusun presentasi kontribusi penelitian, Foltýnek mengusulkan topologi baru yang berorientasi teknis untuk upaya pencegahan dan deteksi plagiarisme, bentuk-bentuk akademik plagiarisme, dan metode deteksi plagiarisme komputasi. Penelitian ini meningkatkan metode analisis teks semantik yang lebih baik, penyelidikan fitur konten non-tekstual, dan penerapan *machine learning*. Kesimpulan dari analisis penelitian ini, integrasi metode analisis heterogen untuk fitur konten tekstual dan non-tekstual menggunakan *machine learning* sebagai area yang paling menjanjikan untuk kontribusi penelitian masa depan untuk meningkatkan deteksi plagiarisme akademik lebih lanjut.

Alzahrani *et al.* (2012) Menyajikan taksonomi baru plagiarisme yang menyoroti perbedaan antara plagiarisme literal dan plagiarisme cerdas, dari sudut pandang perilaku plagiarisme. Taksonomi mendukung pemahaman yang mendalam

tentang pola kebahasaan yang berbeda dalam melakukan plagiarisme, misalnya, mengubah teks menjadi setara secara makna tetapi dengan kata dan organisasi yang berbeda, memperpendek teks dengan generalisasi dan spesifikasi konsep, dan mengadopsi ide dan kontribusi penting dari orang lain. Fitur teks yang berbeda yang mencirikan jenis plagiarisme yang berbeda. Kerangka sistematis dan metode deteksi plagiarisme monolingual, ekstrinsik, intrinsik, dan lintas bahasa disurvei dan dikorelasikan dengan jenis plagiarisme, yang tercantum dalam taksonomi plagiarisme. Penelitian ini melakukan studi tentang teknik canggih untuk deteksi plagiarisme, termasuk karakter berbasis *N-gram* (CNG), berbasis vektor (VEC), berbasis sintaks (SYN), berbasis semantik (SEM), berbasis fuzzy (FUZZY), berbasis struktural (STRUC), berbasis stilometrik (STYLE), dan teknik lintas bahasa (CROSS). Studi ini menguatkan bahwa sistem yang ada untuk deteksi plagiarisme fokus pada penyalinan teks tetapi gagal mendeteksi plagiarisme cerdas ketika ide disajikan dalam kata-kata yang berbeda. Hasil dari analisis penelitian ini meunjukan bahwa metode berbasis semantik (SEM) dan FUZZY tepat untuk mendeteksi plagiarisme ide makna berbasis semantik di tingkat paragraf, misalnya, ketika ide diringkas dan disajikan dalam kata-kata yang berbeda. Penelitian ini juga mengusulkan penggunaan fitur struktural dan informasi kontekstual dengan metode berbasis STRUC yang efisien untuk mendeteksi kepentingan berdasarkan bagian dan plagiarisme ide adaptasi berbasis konteks.

Alfikri *et al.* (2014) Menggunakan Naive Bayes dan Support Vector Machine (SVM) sebagai algoritma pembelajaran untuk mendeteksi plagiarisme secara lebih rinci. Fitur pembelajaran yang digunakan dalam metode ini adalah

kesamaan kata, kesamaan *fingerprint*, kesamaan analisis semantik laten (LSA), dan pasangan kata. Fitur-fitur tersebut diadaptasi dari beberapa metode mutakhir dalam analisis rinci sistem pendeteksi plagiarisme. Tujuan pemilihan fitur tersebut adalah untuk mengambil informasi dari metode analisis detail yang mutakhir (kesamaan kata, *fingerprint*, dan LSA) untuk mengintegrasikan kekuatan masing-masing metode dalam mendeteksi plagiarisme. Beberapa percobaan dilakukan untuk menguji kinerja metode yang diusulkan dalam mendeteksi banyak kasus plagiarisme. Percobaan menggunakan 70 data uji. Pengujian data berisi kasus plagiarisme literal, plagiarisme literal parsial, plagiarisme parafrase, plagiarisme dengan struktur kalimat yang diubah, dan plagiarisme terjemahan. Pengujian data juga berisi kasus non-plagiarisme topik yang berbeda dan non-plagiarisme topik yang sama. Hasil yang diperoleh pada percobaan menggunakan SVM menunjukkan akurasi rata-rata sebesar 92,86% (mencapai 95,71% tanpa menggunakan fitur kesamaan kata). Sedangkan hasil yang diperoleh dengan menggunakan *Naive Bayes* menunjukkan akurasi rata-rata sebesar 54,29% (mencapai 84,29% tanpa menggunakan fitur *word pair*). Penggunaan algoritma SVM menunjukkan hasil yang lebih baik karena secara alami cocok dalam mengklasifikasikan masalah yang memiliki dua kelas dan lebih baik daripada *Naive Bayes* dalam menyelesaikan masalah berdimensi tinggi (yang memiliki banyak fitur). Metode yang diusulkan (SVM) memiliki rata-rata akurasi tinggi untuk setiap kasus plagiarisme yang diuji. Ini membuktikan bahwa metode yang diusulkan (SVM) mampu mengintegrasikan informasi dalam mendeteksi plagiarisme dari metode analisis detail yang canggih

(kesamaan kata, *fingerprint*, dan LSA) untuk mendapatkan hasil deteksi yang lebih akurat.

2.2 TF IDF

Dalam penelitian Albitar et al (2014), algoritma TF-IDF digunakan dalam menilai kesamaan semantik antara teks berdasarkan TF/IDF dengan fungsi baru yang menggabungkan kesamaan semantic antara konsep yang merepresentasikan dokumen teks yang dibandingkan secara berpasangan. Hasil eksperimen pada penelitian ini menunjukkan peningkatan akurasi yang signifikan sebesar 18,13%.

Penggunaan TF-IDF untuk pembobotan kata pada klasifikasi teks pada dokumen Bangladesh yang dikombinasikan dengan metode *Support Vector Machine* telah menghasilkan tingkat akurasi sebesar 92,57% (Islam et al, 2017). Pembobotan TF-IDF akan menghasilkan skor pada setiap kata. Dari skor tersebut dapat diurutkan secara *ascending* ataupun *descending*. Skor kata yang terbesar menunjukkan bahwa kata tersebut sering muncul pada dokumen tersebut. Dengan menggunakan TF-IDF maka akan diketahui kesesuaian kata terhadap dokumen tersebut (Qaiser dan Ali, 2018).

TF-IDF diterapkan dalam sistem klasifikasi dokumen skripsi sebagai vektorisasi teks. Dengan menggunakan algoritma *Cosine Similarity*, TF-IDF berpengaruh dalam proses klasifikasi dokumen ke dalam kategori yang sesuai. Pada penelitian ini menghasilkan akurasi sebesar 98% (Wahyuni, dkk., 2017).

Algoritma TF-IDF. Algoritma TF-IDF adalah metode statistik yang digunakan untuk menilai pentingnya kata untuk dokumen atau kategori dalam kumpulan file

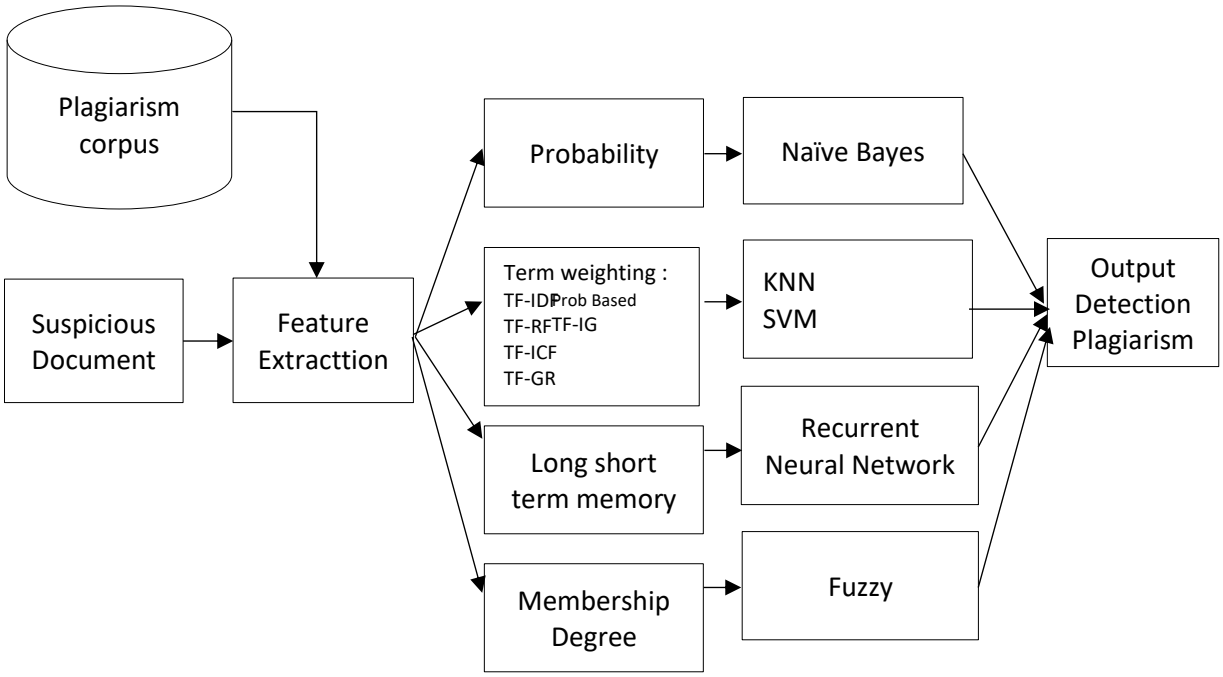
atau corpora. Gagasan utamanya adalah jika suatu kata atau frase sering muncul dalam sebuah artikel dan jarang ditemukan pada artikel lain, maka kata atau frase tersebut dianggap memiliki kemampuan pembedaan kelas yang baik dan layak untuk klasifikasi (Liu *et al*., 2018).

Kombinasi algoritma *K-Nearest Neighbor* dengan TF-IDF juga digunakan dalam penelitian yang dilakukan oleh Irianda *et. al* (2018). Dalam penelitian ini mengusulkan pendekatan yang dapat mengukur dan mengidentifikasi kemiripan dokumen laporan yang dilakukan secara terkomputerisasi yang dapat mengidentifikasi kemiripan antara *query* dengan Document. Skema pembobotan kata pada penelitian ini berbasis *Class-Based Indexing*, kemudian dikomparasi dengan skema pembobotan yaitu TFIDF. Nilai bobot TFIDF dan *Cosine Similarity* untuk menganalisa kemiripan dokumen. Nilai Cosine Similarity pembobotan tersebut CoSimTFIDF kemudian ditetapkan sebagai set fitur untuk proses klasifikasi. Berikutnya dilakukan proses klasifikasi teks menggunakan metode K-Nearest Neighbor (K-NN). Penilitian ini menunjukan hasil akurasi optimal dengan *preprocessing Stemming* dan hasil terbaik dari semua fitur adalah rasio data latih 75% dan data uji 25% pada fitur *Cosine Similarity* berbasis *term weighting* TFIDF yaitu 84%.

Algoritma TF-IDF menghitung nilai pada setiap term untuk mengekstrak istilah. Pada penelitian Zhu *et al* (2019) penggunaan TF-IDF pada penentuan topik berita terpopuler telah menghasilkan akurasi sebesar 80%. Sedangkan tingkat akurasi tanpa menggunakan TF-IDF hanya sebesar 72%.

2.3 Kerangka Teori

Kerangka teori pada penelitian ini mengacu pada jurnal-jurnal sebelumnya yang telah melakukan penelitian pada deteksi plagiarisme. Kerangka teori pada penelitian ini dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Kerangka Teori

Berdasarkan kerangka teori pada Gambar 2.1 dokumen dataset yang akan diuji coba akan dilakukan *feature extraction* terlebih dahulu untuk mengambil *insight* baru dan informasi penting dalam data. Kemudian langkah selanjutnya adalah mencoba beberapa metode yang telah dilakukan pada penelitian sebelumnya dalam menangani kasus plagiarisme berdasarkan parafrase. Adapun beberapa penelitian atau jurnal yang menjadi acuan dalam penelitian ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Rangkuman Penelitian Deteksi Parafrase

No	Penulis & Tahun	Algortima	Judul	Hasil
1	E. Hunt et al. (2019)	• *Neural Network* • *Logistic Regression* • *Support Vector Machine*	*Machine Learning Models For Parafrase Identification And Its Applications On Plagiarism Detection.*	• Akurasi Neural Network 80% • akurasi Logistic Regression 59% • akurasi SVM 62,5%
2	Alfikri, Zakiy & Purwarianti, Ayu. (2014).	• *Support Vector Machine* • *Naive Bayes*	*Detailed Analysis Of Extrinsic Plagiarism Detection System Using Machine Learning Approach (Naive Bayes And SVM)*	• akurasi SVM 92,86% • akurasi Naive Bayes 54,29%
3	Julianto, B. I., Mubarok, M. S., Batu, T. B., & Barat, J. (2017)	• *Naïve Bayes*	Identifikasi Parafrasa Bahasa Indonesia Menggunakan Naïve Bayes	• akurasi naive bayes 71%
4	Alvi, F., Stevenson, M., & Clough, P. (2021)	• *Smith Waterman Algotihm* • *Meteor Algorithm*	*Parafrase Type Identifcation For Plagiarism Detection Using Contexts And Word Embeddings*	• Smith Waterman Algotihm f1 score 0,80184 • Meteor Algorithm f1 score 0.76911
5	Thompson, V. (2017).	• *K Nearest Neighbor* • *Multi Layer Perceptron* • *Naïve Bayes*	*Method For Detecting Paraphpase Plagiarism(2017)*	• KNN f1 score 0.865 • MLP f1 score 0.861 • Naive Bayes F1score 0.855
6	A. Chitra & Anupriya Rajkumar (2013)	• *Support Vector Mechine*	*Genetic Algorithm Based Feature Selection For Parafrase Recognition*	• Akurasi SVM 76.97%

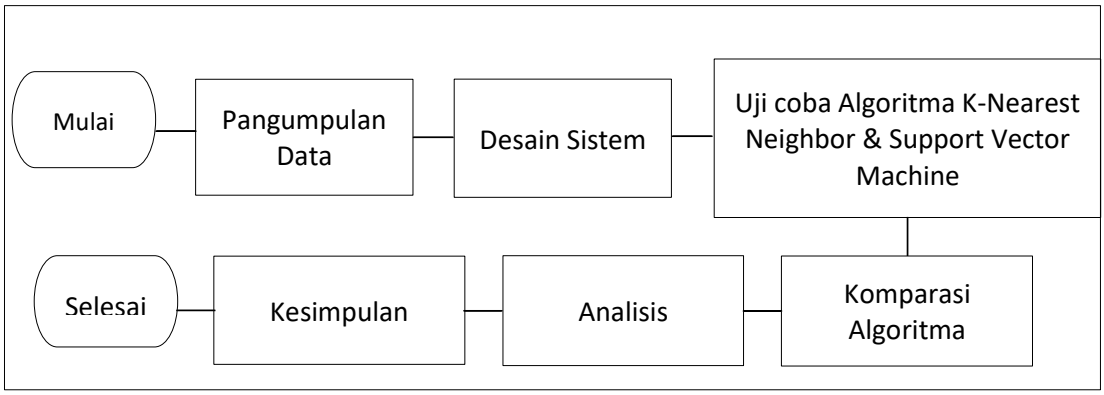
7	Subroto, Imam & Selamat, Ali (2014)	• *Support Vector Machine*	*Plagiarism Detection through Internet using Support Vectors Machine*	• Akurasi SVM 97.6%

Berdasarkan Tabel 2.1 pada penelitian sebelumnya menunjukan bahwa algoritma *K-Nearest Neighbor* dan *Support Vector Machine* memiliki performa yang terbaik diantara metode lain. Hal ini dibuktikan dengan nilai akurasi *Support Vector Machine* sebesar 97,6% dan nilai *f1 score K Nearest Neighbor* sebesar 0.865. Berdasarkan hal tersebut peneliti memilih untuk menggunakan algoritma *Support Vector Machine* karena memiliki akurasi yang paling tinggi dan algoritma *K Nearest Neighbor* karena memiliki nilai *f1 score* tertinggi.

BAB III
DESAIN PENELITIAN

3.1 Prosedur Penelitian

Pada penelitian ini ada beberapa tahapan, yaitu : pengumpulan data, kemudian desain sistem, pada tahap desain sistem ini meliputi *data extraction*, *text preprocessing, dan term weighting*. Kemudian dilanjutkan dengan uji coba terhadap data menggunakan algoritma *K-Nearest Neighbor* & *Support Vector Machine*. Selanjutnya, kedua algoritma dikomparasi untuk melakukan analisis dan terakhir diambil kesimpulan dari penelitian ini. Alur proses desain penelitian dapat dilihat pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Prosedur Penelitian

Penjelasan dari setiap tahap pada prosedur penelitian pada Gambar 3.1 akan diuraikan pada bagian berikutnya.

3.2 Pengumpulan Data

Tahap pengumpulan data ini dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan pada penelitian ini. Dalam penelitian ini data berbentuk teks, yang didapatkan dari abstrak tugas akhir mahasiswa. Data primer ini didapat melalui *e-theses* uin malang yang diakses melalui (http://etheses.uin-malang.ac.id). Sample dataset yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 *Sample Dataset*

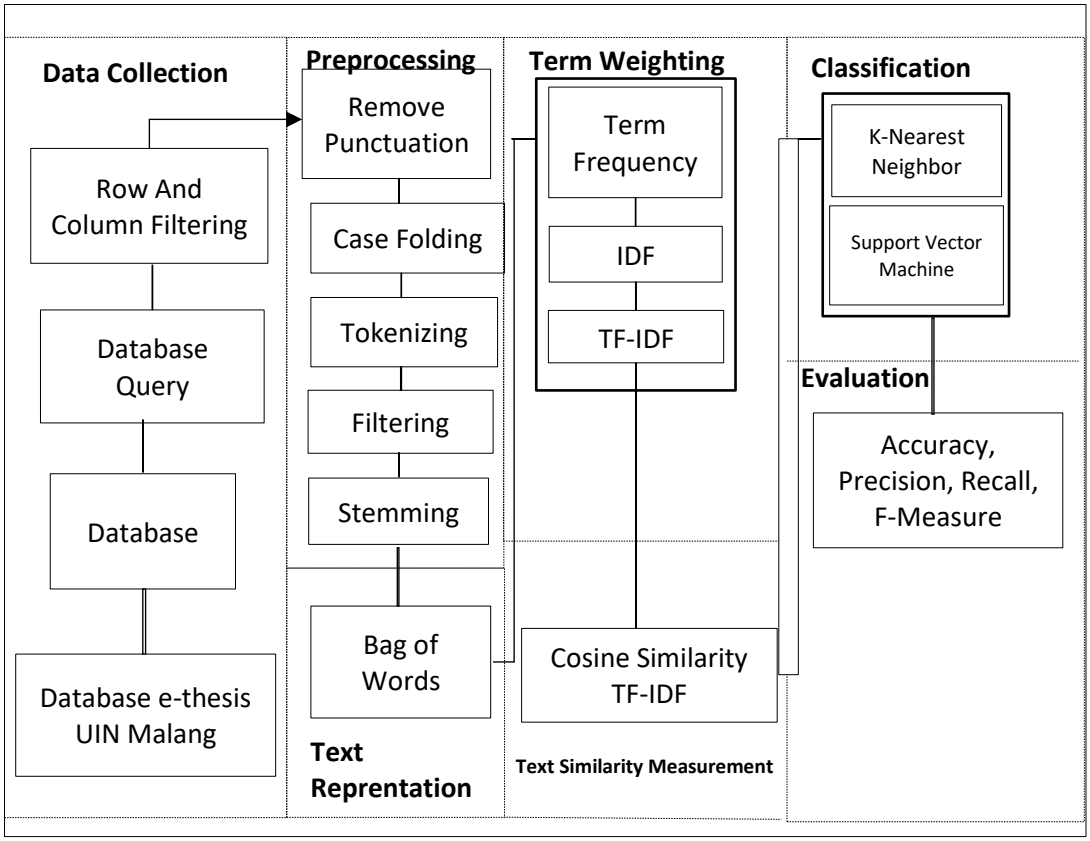
Id	qid1	qid2	Dok1	Dok2	Parafrase
0	1	2	Metode fuzzy Sugeno untuk membantu Arduino Uno mengolah data menjadi suatu acuan suhu dan kelembaban yang ideal untuk perkembangan optimal jamur tiram. NodeMCU untuk membantu Arduino Uno mengirimkan data hasil pengolahan di website monitoring.	Tata cara fuzzy Sugeno buat menolong Arduino Uno mencerna informasi jadi sesuatu acuan temperatur serta kelembaban yang sempurna buat pertumbuhan maksimal jamur tiram. NodeMCU buat menolong Arduino Uno mengirimkan informasi hasil pengolahan di web monitoring.	1
1	3	4	Berkumpulnya orang di suatu tempat sehingga membentuk sebuah kerumunan merupakan hal yang lumrah saat ini. Memperkirakan jumlah orang dalam kerumunan merupakan masalah penting untuk berbagai tujuan mulai dari keselamatan umum hingga strategi industri.	Berkumpulnya orang di sesuatu tempat sehingga membentuk suatu kerumunan ialah perihal yang lumrah dikala ini. Memperkirakan jumlah orang dalam kerumunan ialah permasalahan berarti buat bermacam tujuan mulai dari keselamatan universal sampai strategi industri.	1

2	5	6	Sebagai sarana ujian saringan masuk pada perguruan tinggi islam, Pelaksanaan SSE-UMPTKIN tentu perlu mempersiapkan infrastruktur yang berhubungan dengan proses pelaksanannya. Sehingga faktor yang mempengaruhi terjadinya kendala pada pelaksanaan SSE-UMPTKIN dapat dicegah dan diatasi.	Selaku fasilitas tes saringan masuk pada akademi besar islam, Penerapan SSE- UMPTKIN pasti butuh mempersiapkan infrastruktur yang berhubungan dengan proses pelaksanannya. Sehingga aspek yang pengaruhi terbentuknya hambatan pada penerapan SSE- UMPTKIN bisa dicegah serta diatasi.	1
3	7	8	Proses penyusunan aksi rekontruksi rehabilitasi pasca terjadinya bencana merupakan hal penting, karena kegiatan ini dilakukan untuk mengetahui tingkat kerusakan dan tindakan yang perlu dilakukan setelah terjadinya bencana alam sesuai dengan data dilapangan langsung, maka perlunya dilakukan penelitian dengan metode Fuzzy-VIšekriterijumsko KOmpromisno Rangiranje (Fuzzy-VIKOR).	Proses penataan aksi rekontruksi rehabilitasi pasca terbentuknya musibah ialah perihal berarti, sebab aktivitas ini dicoba buat mengenali tingkatan kehancuran serta aksi yang butuh dicoba sehabis terbentuknya musibah alam cocok dengan informasi dilapangan langsung, hingga perlunya dicoba riset dengan tata cara Fuzzy- VIšekriterijumsko KOmpromisno Rangiranje(Fuzzy-VIKOR).	1
…	…	…	…	…	…
30	61	62	Di dalam kecerdasan buatan, agen cerdas (AI) adalah sebuah entitas otonom yang mengamati dan bertindak atas suatu lingkungan dan mengarahkan aktivitasnya tersebut untuk mencapai tujuan.	Chatbot merupakan sebuah program komputer yang dibangun untuk menampilkan percakapan atau komunikasi interaktif dengan pengguna (manusia) melalui teks, ucapan, dan atau Gambar.	0
31	63	64	Proses pemilihan aplikasi Point of Sale harus didasarkan pada kemampuan dan kebutuhan pembeli.	Ketika pembeli dihadapkan pada banyak pilihan merk POS dan berbagai spesifikasinya kebanyakan pembeli jadi kebingungan memilih aplikasi yang sesuai untuk usahanya.	0

32	65	66	Toko Orisha Beauty merupakan toko yang menjual produk kecantikan yang berlokasi di Batu, Malang.	Toko ini khusus menjual produk bermerek Nu-Skin selain toko offline juga online, sehingga membutuhkan sebuah sistem yang mampu memudahkan dalam merekapitulasi pembelian dari konsumen.	0
33	67	68	Indonesia merupakan negara tropis dan rentan dengan bencana alam. Bencana alam di Indonesia sangat sering terjadi seperti gempa bumi, banjir, kebakaran, Angin Topan dan lain-lain.	Bencana alam menimbulkan dampak bagi warga terdampak, infrastruktur ataupun sektor yang terdapat di Indonesia. Pemerintah pusat, daerah, maupun kota berupaya untuk memberikan rehabilitasi pasca terjadinya bencana alam.	0

3.3 Desain Sistem

Desain sistem akan menjelaskan bagaimana sistem mengklasifikasikan teks yang terdeteksi plagiarisme berdasarkan parafrase. Dalam melakukan klasifikasi teks, penelitian ini menggunakan Algoritma *K-Nearest Neighbor* dan *Support Vector Machine*. Alur proses desain sistem dalam penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Desain Sistem

3.3.1 Data *Preprocessing*

Tahap *preprocessing* dalam *text mining* bertujuan untuk memperkecil ukuran teks. Penelitian sebelumnya telah banyak menggunakan teknik preprocessing pada beberapa aplikasi seperti *clustering*, *classification*, *document indexing*, *summarization*, dan *automatic essay grading.*

Hasanah et. al (2019) mengukur keefektifan teknik *preprocessing* pada *Automatic Short Answer Grading* menggunakan tanya jawab dalam bahasa Indonesia. Teknik *preprocessing* yang digunakan dalam penelitian ini adalah *Case Folding, Tokenization, Punctuation Removal, Stopword Removal*, dan *Stemming*. Selanjutnya dilakukan perhitungan nilai korelasi dan *Mean Absolute Error* untuk

mengukur keefektifan teknik *preprocessing* yang telah digunakan. Berdasarkan uji coba yg dilakukan menunjukkan bahwa tidak ada perbedaan yang signifikan pada teknik *preprocessing* yang digunakan.

Hardaya et al., (2017) mengimplementasikan text mining untuk klasifikasi pengaduan dan usulan masyarakat. Model klasifikasi dibangun menggunakan algoritma Support Vector Machine (SVM). Hasil pengembangan model klasifikasi dokumen menunjukkan bahwa model klasifikasi dengan stemming dan pengenalan sinonim paling akurat diantara yang lain dengan tingkat akurasi 91,37%.

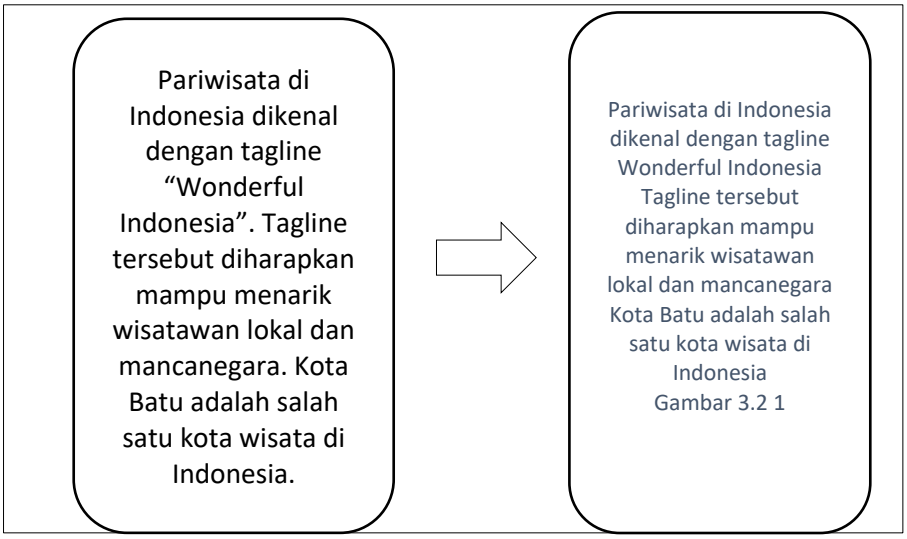
Ayedh et al., (2016) melakukan klasifikasi teks menggunakan algoritma naïve bayes, SVM, dan K-Nearest Neighbor. Teknik preprocessing yang digunakan dalam penelitian ini adalah *Stop word removal* dan *Stemming*. Hasil menunjukan bahwa pada algoritma SVM dan K-Neirest Neighbor menunjukkan hasil yang optimal ketika menggunakan Teknik stemming. Namun, pada algoritma Naïve bayes menunjukkan hasil yang optimal ketika menggunakan Teknik *Stop word removal* .

Uysal dan Gunal (2014) melakukan penelitian untuk mengetahui pengaruh teknik *prepreprocessing* yang banyak digunakan pada klasifikasi teks dalam dua bahasa yang berbeda. Berdasarkan penelitian ini menunjukan bahwa konversi huruf kecil/*lowercase* meningkatkan akurasi klasifikasi *machine learning* untuk mengklasifikasikan email sebagai spam dan mengidentifikasi kategori berita dalam bahasa Inggris dan Turki.

Maka merujuk pada penelitian sebelumnya, perlu dilakukan tahapan *preprocessing* untuk medapatkan hasil yang lebih optimal. pada penelitian ini teknik *preprocessing* meliputi beberapa tahap yaitu: *remove punctuation, case folding, tokenizing, filtering,* dan *stemming.*

1) Remove Punctuation

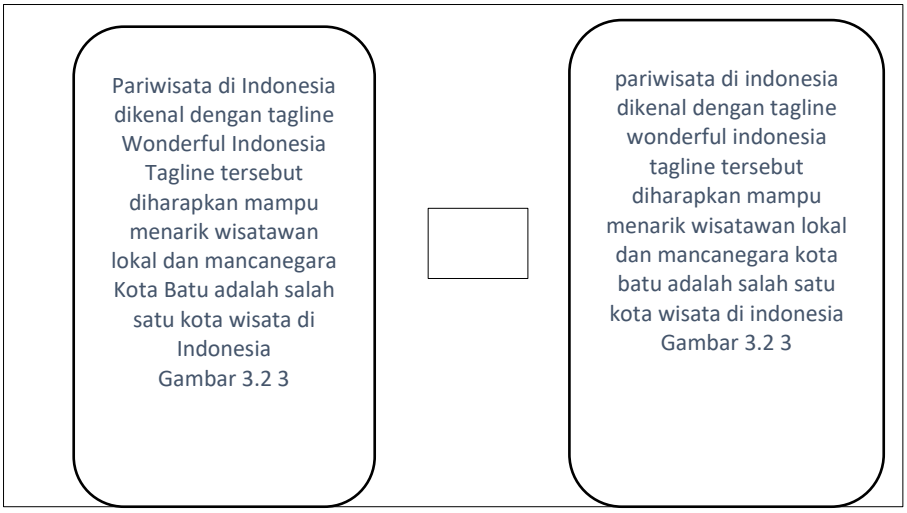
Remove punctuation adalah proses untuk manghapus karakter atau tanda baca yang terdapat dalam sebuah dokumen. Adapun karakter yang dihapus meliputi : !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~. Hasil dari proses *remove punctuation* dicontohkan pada Gambar 3.3.



Gambar 3.3 Proses *Remove Punctuation*

2) Case Folding

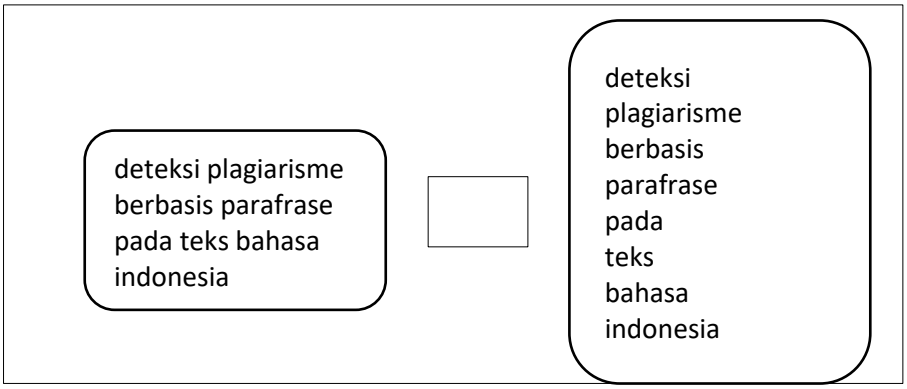
Case folding adalah proses untuk menyeragamkan semua penggunaan huruf kapital menjadi huruf kecil *(lowercase).* Hasil dari proses *case folding* dicontohkan pada Gambar 3.4.



Gambar 3.4 Proses *Case Folding*

3) Tokenizing

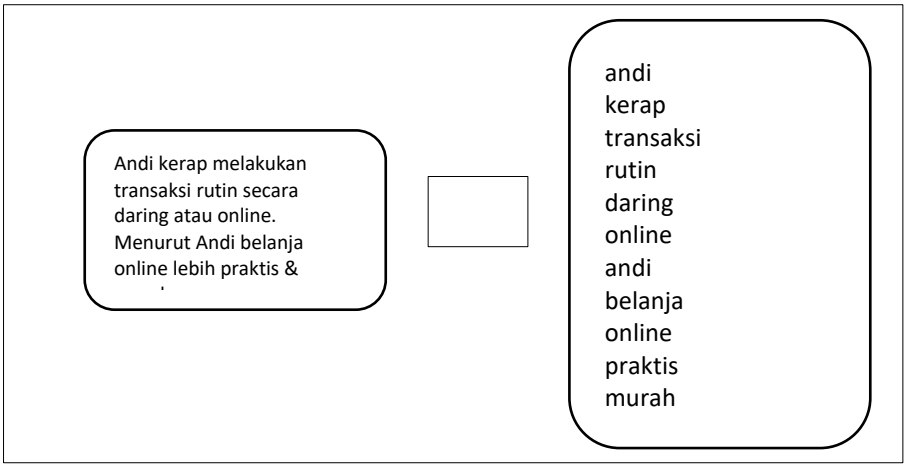
Tokenizing adalah proses pemisahan teks menjadi potongan-potongan yang disebut sebagai token untuk kemudian di analisa. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token. Didalam NLP, token diartikan sebagai "kata" meskipun *tokenize* juga dapat dilakukan pada paragraf maupun kalimat. Hasil dari proses *tokenizing* dicontohkan pada Gambar 3.5.



Gambar 3.5 Proses *Tokenizing*

4) Filtering

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). *Stopword* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh stopword dalam bahasa Indonesia adalah "yang", "dan", "di", "dari", dll. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya. Penghilangan stopword ini dapat mengurangi ukuran *index* dan waktu pemrosesan. Selain itu, juga dapat mengurangi level noise. Hasil dari proses *filtering* dicontohkan pada Gambar 3.6.

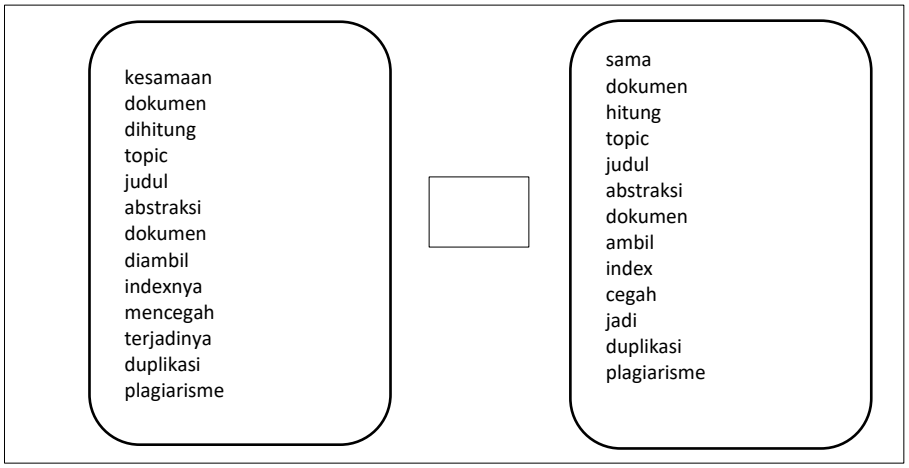


Gambar 3.6 Proses *Filtering*

5) Stemming

Stemming adalah proses untuk menghapus imbuhan didalam suatu kata. Dengan kata lain proses *stemming* adalah mengubah suatu kata kembali ke kata dasarnya. Misalnya kata "mendengarkan", "dengarkan", "didengarkan" akan

ditransformasi menjadi kata “dengar”. Hasil dari proses *stemming* dicontohkan pada Gambar 3.7.



Gambar 3.7 Proses *Stemming*

3.3.2 *Split Data*

Split data adalah teknik untuk memisahkan antara data *training* dan data *testing*. Pemisahan data ini berfungsi untuk mengevaluasi performa model *machine learning* yang akan dibangun. *Data training* adalah sekumpulan data yang akan dilatih untuk membuat model dari algoritma *machine learning*. Sedangkan *data testing* adalah sekumpulan data untuk mengevaluasi performa dari model yang telah dihasilkan apakah memiliki hasil yang akurat atau sebaliknya.

3.3.4 *Term Weighting*

Sebelum melakukan klasifikasi suatu teks, harus dilakukan pembobotan term (*term weighting*) terlebih dahulu. *Term weighting* adalah melakukan transformasi data teks menjadi sekumpulan angka (vektor) sebelum melakukan modelling. Pembobotan teks pada penelititan ini menggunakan teknik TF-IDF dan

TF-ICF. *Term Frequency* (TF) menghitung frekuensi jumlah kemunculan kata pada sebuah dokumen. *Inverse Document Frequency* (IDF) merupakan nilai untuk mengukur seberapa penting sebuah kata. IDF akan menilai kata yang sering muncul sebagai kata yang kurang penting berdasarkan kemunculan kata tersebut pada seluruh dokumen. Semakin kecil nilai IDF maka akan dianggap semakin tidak penting kata tersebut, begitu pula sebaliknya. Untuk melakukan pembobotan TF-IDF menggunakan Rumus (3.1) dengan rincian sebagai berikut:

$$TF - IDF = tf * log\left(\frac{d}{df}\right) \quad (3.1)$$

tf = frekuensi kemunculan term dalam dokumen
d = jumlah dokumen
df = jumlah document yang mengandung term

Adapun proses pembobotan kata menggunakan metode TF-IDF dimulai dengan mengambil hasil tahap *preprocessing* pada tahap sebelumnya. Berikut ini dicontohkan beberapa pasangan kalimat pada Tabel 3.2.

Tabel 3.2 Contoh pasangan kalimat parafrase

NO	Kalimat1	Kalimat2	Parfrase
1	tingkat kasus positif harian covid-19 di indonesia mengalami kenaikan dan penurunan yang sangat bervariasi.	angka harian kasus positif covid19 di indonesia sangat bervariasi.	Ya
2	pariwisata adalah salah satu sektor pendapatan yang memiliki dampak besar bagi pemerintah.	wisata adalah bepergian secara bersama dengan tujuan untuk bersenang-senang, menambah pengetahuan, dan lain-lain.	Tidak
3	jual beli merupakan transaksi tukar menukar barang yang dimana barang tersebut memiliki nilai atau harga	jual beli adalah pertukaran barang dimana barang tersebut memiliki nilai atau harga.	Ya

4	proses pertukaran barang yang memiliki nilai atau harga disebut jual beli	?

D1 = tingkat kasus positif harian covid-19 di indonesia mengalami kenaikan dan penurunan yang sangat bervariasi

D2 = angka harian kasus positif covid19 di indonesia sangat bervariasi

D3 = pariwisata adalah salah satu sektor pendapatan yang memiliki dampak besar bagi pemerintah

D4 = wisata adalah bepergian secara bersama dengan tujuan untuk bersenang-senang, menambah pengetahuan, dan lain-lain.

D5 = jual beli merupakan transaksi tukar menukar barang yang dimana barang tersebut memiliki nilai atau harga

D6= jual beli adalah pertukaran barang dimana barang tersebut memiliki nilai atau harga

Q = proses pertukaran barang yang memiliki nilai atau harga disebut jual beli

Corpus atau kumpulan kata yang dibentuk dari beberapa kalimat pada Tabel 3.1 selanjutnya dihitung bobot setiap kata menggunakan rumus IDF seperti dicontohkan pada Tabel 3.3.

Tabel 3. 3Perhitungan Bobot IDF

Term	**D1**	**D 2**	**D3**	**D4**	**D5**	**D6**	**Q**	**DF**	**D/DF**	**IDF**
Tingkat	1	0	0	0	0	0	0	1	6	0,778151
Positif	1	1	0	0	0	0	0	2	3	0,477121
Indonesia	1	1	0	0	0	0	0	2	3	0,477121
Angka	0	1	0	0	0	0	0	1	6	0,778151
covid19	1	1	0	0	0	0	0	2	3	0,477121
Harga	0	0	0	0	1	1	1	3	2	0,30103

Jual	0	0	0	0	1	1	1	3	2	0,30103
Transaksi	0	0	0	0	1	0	0	1	6	0,778151
Beli	0	0	0	0	1	1	1	3	2	0,30103
Nilai	0	0	0	0	1	1	1	3	2	0,30103
Dimana	0	0	0	0	1	1	0	2	3	0,477121
Tukar	0	0	0	0	1	0	0	1	6	0,778151
Tersebut	0	0	0	0	1	1	0	2	3	0,477121
Merupakan	0	0	0	0	1	0	0	1	6	0,778151
…	…	…	…	…	…	…	…	…	…	…
Proses	0	0	0	0	0	0	1	1	6	0,778151

Setelah mendapatkan bobot IDF dari masing-masing kalimat pada Tabel 3.3. Selanjutnya dilakukan perhitungan bobot menggunakan rumus TF IDF seperti yang dicontohkan pada Tabel 3.4.

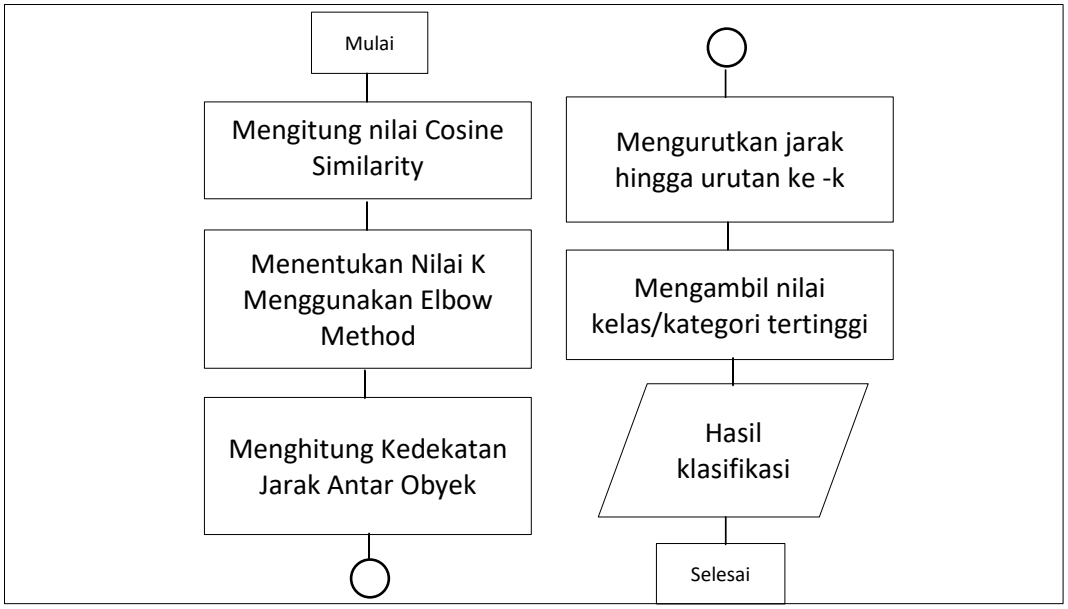
Tabel 3.4 Perhitungan Bobot TF IDF

Term	**TF IDF**						
	D1	**D2**	**D3**	**D4**	**D5**	**D6**	**Q**
Tingkat	0,778151	0	0	0	0	0	0
Positif	0,477121	0,477121	0	0	0	0	0
Indonesia	0,477121	0,477121	0	0	0	0	0
Angka	0	0,778151	0	0	0	0	0
covid19	0,477121	0,477121	0	0	0	0	0
Harga	0	0	0	0	0,30103	0,30103	0,30103
Jual	0	0	0	0	0,30103	0,30103	0,30103
Transaksi	0	0	0	0	0,778151	0	0
Beli	0	0	0	0	0,30103	0,30103	0,30103
Nilai	0	0	0	0	0,30103	0,30103	0,30103
Dimana	0	0	0	0	0,477121	0,477121	0
Tukar	0	0	0	0	0,778151	0	0
Tersebut	0	0	0	0	0,477121	0,477121	0
Merupakan	0	0	0	0	0,778151	0	0
…	…	…	…	…	…	…	…
Proses	0	0	0	0	0	0	0,778151

BAB IV
METODE K-NEAREST NEIGHBOR

4.1 Desain & Implementasi

K-Nearest Neighbor merupakan salah satu metode klasifikasi digunakan dalam *data mining* dan *machine learning*. Kinerja klasifikasi dari KNN sangat bergantung pada metrik yang digunakan untuk menghitung jarak berpasangan antara titik data. Untuk menghitung k titik data tetangga terdekat yang diinginkan, pada penellitian ini menggunakan *cosine similarity* sebagai metrik kesamaan. Aturan klasifikasi KNN dibuat oleh sampel pelatihan saja, tanpa data tambahan lainnya. Klasifikasi KNN, menemukan sekelompok k objek dalam set pelatihan yang paling dekat dengan objek uji, dan mendasarkan penetapan label pada dominasi kelas tertentu di lingkungan ini. Algoritma *K-Nearest Neighbor* (k-NN) adalah metode untuk mengklasifikasikan objek berdasarkan contoh pelatihan terdekat dalam ruang fitur. Seperti yang telah disebutkan sebelumnya, pada penelitian ini menggunakan algoritma dalam melakukan klasifikasi teks. Berikut ini adalah *Flowchart* proses klasifikasi KNN ditunjukan pada Gambar 4.1.



Gambar 4.1 *Flowchart* Algoritma KNN

4.1.1 Mengitung Nilai *Cosine Similarity*

Tahap pertama dalam mengklasifikasikan teks adalah menghitung nilai *cosine similarity*, untuk menghitung kedekatan jarak antar dokumen. Rumus *cosine similarity* dituliskan pada Rumus 4.1.

$$cosSim(Q, d_j) = \frac{\sum_{i=1}^{m} x_i . d_{ji}}{\sqrt{\sum_{i=1}^{m} x^2 i} \cdot \sqrt{\sum_{i=1}^{m} d^2 ji}} \quad (4.1)$$

Keterangan :
Q : dokumen query
dj : data sampel
xi : nilai bobot term i pada dokumen
dji : nilai bobot term $i$pada dokumen sampel

Dalam menghitung *cosine similarity* menggunakan pembobotan TF-IDF yang telah dilakukan sebelumnya pada Tabel 3.3. Pertama, hitung hasil perkalian skalar antara bobot Q (dokumen *query*) dengan bobot dokumen yang telah diklasifikasikan. Kemudian, hasil perkalian dari setiap dokumen dengan Q

dijumlahkan sesuai pembilang pada rumus diatas *cosine similarity*. Lalu, hitung panjang vektor setiap dokumen,dengan mengkuadratkan bobot setiap term dalam setiap dokumen, jumlahkan nilai kuadrat tersebut dan kemudian akarkan. Adapun contoh perhitungan nilai *cosine similarity* pada dokumen query (Q) dicontohkan seperti pada Tabel 4.1.

Tabel 4.1 Nilai Bobot TF IDF Dokumen Uji & Dokumen Query

d1	d2	d3	…	d57	d58	d59	q1	q2	q3	…	q57	q58	q59
0	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0
4,393	0	0	…	0	0	0	4,393	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	5,086	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…
9,361	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0
0	0	0	…	0	0	0	0	0	0	…	0	0	0

Tabel 4.1 merupakan nilai TF IDF atau bobot seriap kata yang terkandung dalam dokumen uji dan dokumen *query*. Nilai TF IDF ini digunakan sebagai parameter untuk menghitung nilai *cosine similarity* antara dokumen uji dan dokumen *query*. Dalam menghitung *cosine sismilarity*, pertama lakukan perkalian antar *vector* dokumen uji dan dokumen *query*. Lalu, hitung panjang *vector* setiap dokumen dengan mengkuadratkan bobot setiap *term* dalam setiap dokumen, jumlahkan nilai kuadrat tersebut dan kemudian akarkan. Adapun contoh perhitungan nilai *cosine similarity* pada dokumen *query* (Q) adalah sebagai berikut:

$$(d1, q1)$$

$$= \frac{(0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (4{,}39x4{,}39) + (0x0) + \cdots + (9{,}3}{\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 4{,}39^2 + 0^2 + \cdots + 9{,}36^2 + 0^2 + 0^2} x \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 +}$$

$$= 0{,}57$$

$$(d2, q2)$$

$$= \frac{(0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \cdots + (0x0) + (0x0)}{\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2} x \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0}$$

$$= 0{,}87$$

$$(d3, q3)$$

$$= \frac{(0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \cdots + (0x0) + (0x0)}{\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2} x \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0}$$

$$= 0{,}34$$

$$(d57, q57)$$

$$= \frac{(0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (5{,}08x0) + \cdots + (0x0) + (0}{\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2} x \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 5}$$

$$= 0{,}25$$

$$(d58, q58)$$

$$= \frac{(0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \cdots + (0x0) + (0x0)}{\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2} x \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0}$$

$$= 0{,}22$$

$$(d59, q59)$$

$$= \frac{(0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \cdots + (0x0) + (0x0)}{\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2} x \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0}$$

$$= 0$$

4.1.2 Menentukan Nilai K Menggunakan *Elbow Method*

Setelah mendapatkan nilai *cosine similarity*, langkah selanjutnya adalah menentukan parameter nilai K (jumlah tetangga paling dekat). Penentuan nilai K dalam algoritma sangatlah krusial karena nilai K sangat mempengaruhi hasil

klasifikasi algoritma KNN, jika nilai K yang digunakan kurang baik maka akan mengurangi performa algoritma KNN. Maka dari itu pada penelitian ini menggunakan metode *elbow method* untuk mengetahui nilai K yang terbaik.

Elbow method merupakan suatu metode yang digunakan untuk menghasilkan informasi dalam menentukan jumlah nilai K terbaik dengan cara melihat persentase hasil perbandingan antara jumlah K yang akan membentuk siku pada suatu titik. Metode ini memberikan ide/gagasan dengan cara memilih nilai K dan kemudian menambah nilai K tersebut untuk dijadikan model data dalam penentuan nilai K terbaik. Hasil persentase yang berbeda dari setiap nilai K dapat ditunjukan dengan menggunakan grafik sebagai sumber informasinya. Jika nilai K pertama dengan nilai K kedua memberikan sudut dalam grafik atau nilainya mengalami penurunan paling besar maka nilai K tersebut yang terbaik.

Langkah pertama dalam *elbow method* adalah inisialisasi nilai K. Pada penelitian ini akan melakukan percobaan terhadap nilai K=1 sampai dengan K=15 maka inisialisasi nilai K pada penelitian ini adalah K=1-15. Setelah inisialisasi, tahap selanjutnya adalah menghitung nilai rata-rata error dari setiap nilai K. Rata-rata nilai error dihitung menggunakan Rumus 4.2.

$$Error = \frac{\sum_i^n (nilaiaktual - nilaiprediksi)}{n} \quad (4.2)$$

Keterangan :

n : Jumlah data

i : Nilai K (1-15)

4.1.3 Menghitung Kedekatan Jarak Antar Obyek

Dalam menentukan klasifikasi suatu obyek, algoritma knn mengklasifikasikan berdasarkan kedekatan jarak antar obyek. Untuk menghitung kedekakan jarak ada beberapa metode pengukuran jarak yang dapat digunakan antara lain yaitu: *Manhattan Distance, Euclidean Distance, Chebyshev Distance, Minkowsky Distance, Chi Square, Cosine Distance* dll.

Mulak dan Talhar (2013) melakukan implementasi K-nearest neighbor menggunakan jarak *Euclidian*, jarak *Manhattan* dan jarak *Chebychev* pada dataset KDD. Proses klasifikasi KNN dievaluasi dalam hal akurasi, spesifikasi, dan sensitivitas. Berdasarkan uji coba diketahui bahwa jarak *Manhattan* memberikan hasil yang lebih baik daripada ukuran jarak lainnya. Performa jarak *Euclidian* lebih rendah dibandingkan dengan jarak *Chebychev*.

Chomboon *et.al* (2015) dalam penelitiannya mempelajari kinerja dari *k-nearest neighbor* dalam menerapkan pengukuran jarak yang berbeda. Dalam penelitian ini, membandingkan 11 metrik jarak termasuk *Euclidean, Mahalanobis, Manhattan, Minkowski, Chebychev, Cosinus, Correlation, Hamming, Jaccard*, dan *Spearman*. Serangkaian percobaan telah dilakukan pada delapan dataset sintetik dengan berbagai macam distribusi. Perhitungan jarak yang memberikan prediksi yang sangat akurat pada penelitian ini adalah metode *Manhattan, Chebychev, Euclidean, Mahalanobis, Minkowski.*

Perbandingan penghitungan jarak pada *k-nearest neighbour* dalam klasifikasi data tekstual dilakukan dalam penelitian Wahyono dkk 2019 Hasil penelitian ini menunjukkan bahwa jarak *Euclidean* dan *Minkowski* pada algoritma KNN pada data dengan representasi vektor dari kalimat Sebagian besar

menghasilkan akurasi terbaik dibandingkan *Chebyshev* maupun *Manhattan*. Hasil terbaik pada KNN diperoleh ketika K bernilai 3.

Penelitian tersebut menunjukkan perbedaan kinerja metode pengukuran jarak terhadap performansi algoritma KNN. Maka, pada penelitian ini akan melakukan pengukuran jarak menggunakan metode *Manhattan Distance, Euclidean Distance, Chebyshev Distance,* dan *Minkowsky Distance* untuk mengetahui metode pengukuran jarak mana yang paling ideal dalam penelitian ini.

1) Pengukuran Jarak Menggunakan *Manhattan Distance*

Manhattan distance atau jarak *manhattan* sering juga disebut *city block distance. Manhattan distance* adalah metrik ukur yang umumnya digunakan untuk menghitung jarak antara dua titik data dalam jalur seperti *grid*. Jarak *manhattan* dihitung sebagai jumlah dari perbedaan mutlak antara dua vektor. Adapun persamaan dari *manhattan distance* dapat dihitung menggunakan persamaan 4.3.

$$d_i = \sum_{i=1}^{n} X_{2i} - X_{i1} \vee \quad (4.3)$$

Keterangan:

X1 = Sampel Data

X2 = data uji/testing

i = variabel data

d = jarak

n = jumlah data

Adapun hasil pengukuran kedekatan jarak antara obyek menggunakan Manhattan distance dapat dilihat pada Tabel 4.2.

Tabel 4.2 Pengukuran Jarak Menggunakan *Manhattan Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.10383739	0.26197487	…	0.3400778
2	0.	0.	…	0.
3	0.041431	0.07348965	…	0.00461328
4	0.08440503	0.07810293	…	0.07810293

2) Pengukuran Jarak Menggunakan *Euclidean Distance*

Euclidean distance atau jarak *Euclidean* adalah metrik jarak ukur antara dua vektor dengan menghitung akar kuadrat dari jumlah selisih kuadrat antara keduanya. Perhitungan ini mirip seperti yang digunakan pada teorema *Pythagoras*. *Euclidean distance* dapat dihitung menggunakan persamaan 4.4.

$$d_i = \sqrt{\sum_{i=1}^{n}(X_{2i} - X_{i1})^2} \quad (4.4)$$

Keterangan:

X1 = Sampel Data

X2 = data uji/testing

i = variabel data

d = jarak

n = jumlah data

Adapun hasil pengukuran kedekatan jarak antara obyek menggunakan *Euclidean distance* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengukuran Jarak Menggunakan *Euclidean Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.10383739	0.26197487	…	0.3400778
2	0.	0.	…	0.
3	0.041431	0.07348965	…	0.00461328

4	0.08440503	0.07810293	…	0.07810293

3) Pengukuran Jarak Menggunakan *Minkowsky Distance*

Minkowsky Distance adalah pengukuran jarak/kesamaan antara dua titik dalam ruang vektor bernorma (ruang nyata dimensi N) dan merupakan generalisasi dari jarak Euclidean dan jarak Manhattan. Jika parameter p dari metrik Jarak *Minkowski* adalah 1, itu akan mewakili Jarak *Manhattan* dan ketika Parameter p adalah 2, itu akan mewakili Jarak *Euclidean*. Parameter p dalam Jarak *Minkowski* adalah bilangan bulat antara dua titik riil, X = (x1, x2,..., xn) dan Y = (y1, y2,..., yn). Adapun nilai p yang digunakan dalam penelitian ini adalah p=3. *Minkowsky Distance* dapat dihitung menggunakan persamaan 4.5.

$$d_i = \qquad (4.5)$$

Keterangan:

X1 = sampel data

X2 = data uji/testing

i = variabel data

d = jarak

n = jumlah data

Adapun hasil pengukuran kedekatan jarak antara obyek menggunakan *Minkowsky Distance* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Pengukuran Jarak Menggunakan *Minkowsky Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.10383739	0.26197487	…	0.3400778

2	0.	0.	…	0.
3	0.041431	0.07348965	…	0.00461328
4	0.08440503	0.07810293	…	0.07810293

4) Pengukuran Jarak Menggunakan *Chebyshev Distance*

Chebyshev distance atau jarak *Chebyshev* merupakan metode yang mengukur jarak berdasarkan nilai mutlak atau absolut dari selisih pasangan koordinat sebuah titik. Artinya apabila kita memiliki 2 buah vektor yang berbeda nilai masing-masing elemen maka jarak yang diukur menggunakan chebyshev adalah berdasarkan nilai mutlak dari selisih elemen-elemen pada vektor-vektor tersebut secara otomatis jumlah datanya harus sama. Ketika nilai mutlak dari selisih tersebut sudah didapat yang tentu saja berbentuk vektor dengan panjang data yang sama dengan vektor awal maka kita mencari nilai mutlak tertinggi atau maksimal. Jarak *Chebysev* ini merupakan kasus khusus dari *minkowski* apabila p → ∞. *Chebyshev distance* dapat dihitung menggunakan persamaan 4.6.

$$d_i = \lim_{p \infty} \tag{4.6}$$

Keterangan:

X1 = sampel data

X2 = data uji/testing

i = variabel data

d = jarak

n = jumlah data

Adapun hasil pengukuran kedekatan jarak antara obyek menggunakan *Chebyshev Distance* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Pengukuran Jarak Menggunakan *Chebyshev Distance*

K=11	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.10383739	0.26197487	**…**	0.3400778
2	0.	0.	**…**	0.
3	0.041431	0.07348965	**…**	0.00461328
4	0.08440503	0.07810293	**…**	0.07810293

4.1.4 Mengurutkan Jarak Pada Setiap Pengukuran

Setelah mendapatkan jarak antar 2 obyek langkah selanjutnya yaitu megurutkan berdasarkan jarak terkecil sampai jarak terbesar. Adapun pengurutan jarak akan diuraikan pada bagian selanjutnya.

1) Pengurutan Jarak *Manhattan*

Pengurutan kedekatan jarak antara obyek menggunakan *manhattan distance* dapat dilihat pada Tabel 4.6.

Tabel 4.6 Pengurutan *Manhattan Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.	0.	**…**	0.
2	0.041431	0.07348965	**…**	0.00461328
3	0.08440503	0.07810293	**…**	0.07810293
4	0.10383739	0.26197487	**…**	0.3400778

2) Pengurutan Jarak *Euclidean*

Pengurutan kedekatan jarak antara obyek menggunakan *euclidean distance* dapat dilihat pada Tabel 4.7.

Tabel 4.7 Pengurutan *Euclidean Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.	0.	**…**	0.

2	0.041431	0.07348965	…	0.00461328
3	0.08440503	0.07810293	…	0.07810293
4	0.10383739	0.26197487	…	0.3400778

3) Pengurutan Jarak *Minkowsky*

Pengurutan kedekatan jarak antara obyek menggunakan *minkowsky distance* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Pengurutan *Minkowsky Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.	0.	…	0.
2	0.041431	0.07348965	…	0.00461328
3	0.08440503	0.07810293	…	0.07810293
4	0.10383739	0.26197487	…	0.3400778

4) Pengurutan Jarak *Chebyshev*

Pengurutan kedekatan jarak antara obyek menggunakan *chebyshev distance* dapat dilihat pada Tabel 4.9.

Tabel 4.9 Pengurutan *Chebyshev Distance*

K=4	**Jarak (Q_1,D_1)**	**Jarak (Q_1,D_1)**	**…**	**Jarak (Q_n,D_n)**
1	0.	0.	…	0.
2	0.041431	0.07348965	…	0.00461328
3	0.08440503	0.07810293	…	0.07810293
4	0.10383739	0.26197487	…	0.3400778

Setelah mengurutkan jarak Manhattan, Euclidean, *Minkowsky,* dan *C*heby*shev* terlihat bahwa hasil perhitungan dari ke empat metode yang di uji coba menghasilkan jarak yang sama dalam setiap perhitungan jarak setiap pasangan kalimat. Hal ini menunjukan pemilihan pengukuran jarak dalam dataset pada

penelitian ini menghasilkan performa yang sama pada setiap metode pengukuran jarak.

4.1.5 Mengambil Nilai Kategori Tertinggi

Pada langkah ini akan dilakukan penentuan hasil kelas/kategori yang dihasilkan dari prediksi model algoritma knn yang bersesuaian pada langkah sebelumnya. Pengambilan nilai kategori tertinggi dicontohkan pada Tabel 4.10.

Tabel 4.10 Menentukan Kategori Tertinggi

K=4	**Jarak (Q_1,D_1)**	**Kelas/ Kategori**	**Jarak (Q_2,D_2)**	**Kelas/ Kategori**	**…**	**Jarak (Q_n,D_n)**	**Kelas/ Kategori**
1	0.	1	0.	1	…	0.	1
2	0.04	1	0.07	1	…	0.004	1
3	0.08	1	0.078	1	…	0.07	0
4	0.1	0	0.26	1	…	0.34	1

Pada Tabel 4.10 menunjukan pasangan setiap jarak dan kategori dari pengurutan jarak K terdekat. Pada bagian sebelumnya telah dijelaskan bahwa kategori 1 adalah pasangan kalimat yang di parafrase dan kategori 0 adalah pasangna kalimat yang tidak di parafrase. Pada pasangan jarak (Q_1,D_1) kategori K terdekat adalah (1, 1, 1,0). Maka untuk menentukan kategori/kelas pada pasangan kalimat (Q_1,D_1) adalah dengan menghitung masing-masing jumlah kategori. Kategori 1 berjumlah 3 dan kategori 0 berjumlah 1, maka prediksi kategori pasangan kalimat (Q_1,D_1) adalah kategori 1 karena jumlahnya lebih dominan dari kategori 0.

Pada pasangan jarak (Q_2,D_2) kategori K terdekat adalah (1, 1, 1, 1). Maka untuk menentukan kategori/kelas pada pasangan kalimat (Q_2,D_2) adalah dengan menghitung masing-masing jumlah kategori. Kategori 1 berjumlah 4 dan kategori

0 berjumlah 0, maka prediksi kategori pasangan kalimat (Q_2,D_2) adalah kategori 1 karena jumlahnya lebih dominan dari kategori 0.

Pada pasangan jarak (Q_n,D_n) kategori K terdekat adalah (1, 1, 0, 1). Maka untuk menentukan kategori/kelas pada pasangan kalimat (Q_n,D_n) adalah dengan menghitung masing-masing jumlah kategori. Kategori 1 berjumlah 3 dan kategori 0 berjumlah 1, maka prediksi kategori pasangan kalimat (Q_n,D_n) adalah kategori 1 karena jumlahnya lebih dominan dari kategori 0.

4.2 Uji Coba

Pada penelitian ini untuk melakukan uji coba algoritma K-Nearest Neighbor digunakan skenario eksperimen sebagai berikut :

4.2.1 Skenario Uji Coba K-Nearest Neighbor

1. Rasio pembagian *dataset* dibagi kedalam data training dan data testing dengan rasio pembagian 80% data training dan 20% data testing
2. Nilai k dalam Pengujian *K-Nearest Neighbor* dilakukan dengan menggunakan beberapa nilai k yaitu K = 1 - 15. Dalam mencari nilai K yang paling optimal pada penelitian ini menggunakan *Elbow Method.*
3. Menggunakan 4 metode pengukuran jarak, yaitu:
   a) Klasifikasi teks berdasarkan pengukuran *Manhattan Distance*
   b) Klasifikasi teks berdasarkan pengukuran *Euclidean Distance*
   c) Klasifikasi teks berdasarkan pengukuran *Minkowsky Distance*
   d) Klasifikasi teks berdasarkan pengukuran *Chebyshev Distance*
4. Evaluasi performa dari masing- masing *algoritma K-Nearest Neighbor* akan dihitung Menggunakan *Confusion Matrix*. Kemudian dari Tabel *Confusion*

Matrix akan dihitung nilai akurasi, *precision*, *recall* dan *f1score.* Adapun rumus Confusion Matrix dituliskan pada Tabel 4.11.

Tabel 4.11 Tabel Confusion Matrix

		Actual Values	
		Positif	Negatif
Predicted Value	Positive	TP	FP
	Negative	FN	TN

Keterangan :

TP : *True Positive* merupakan data positif yang diprediksi sebagai data positif.

TN : *True Negative* merupakan data negatif yang diprediksi sebagai data negative

FP : *False Positive* merupakan data positif yang diprediksi sebagai data negatif.

FN : *False Negative* merupakan data positif yang diprediksi sebagai data negatif.

$$RumusAkurasi = \frac{TP + TN}{TP + TN + FP + FN} * 100\% (4.7)$$

$$RumusPrecision = \frac{TP}{TP + FP} * 100\% (4.8)$$

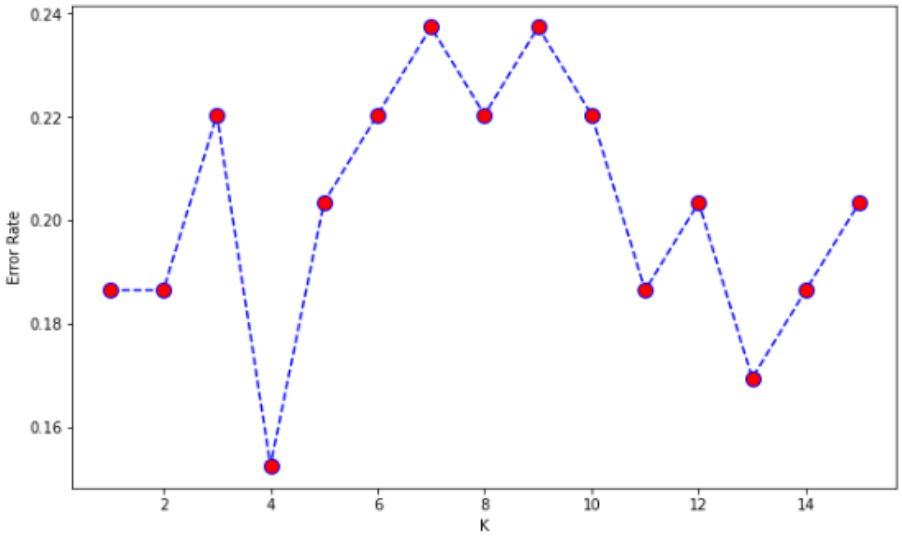
$$RumusRecall = \frac{TP}{TP + FN} * 100\% (4.9)$$

$$RumusF1Score = 2\frac{PrecisionxRecall}{Precision + Recall} * 100\% (4.9)$$

4.2.2 Hasil Uji Coba

1) Hasil Uji Coba *Elbow Method*

Berdasarkan hasil uji coba yang telah dilakukan pada nilai K 1-15 menggunakan elbow method nilai K yang paling optimal adalah ketika K=4. Adapun visualisasi nilai error pada percobaan setiap nilai K menggunakan *elbow method* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Grafik Nilai Error K

Berdasarkan Gambar 4.2 dapat dilihat bahwa ketika nilai K=4 mengalami penurunan nilai error yang sangat drastis dari nilai K

sebelumnya. Maka nilai K yang ideal adalah k=4. Hal ini selaras dengan nilai error yang dihasilkan ketika K=4 bernilai paling kecil. Rangkuman hasil nilai error pada setiap pengujian nilai K dapat dilihat pada Tabel 4.12.

Tabel 4.12 Nilai Error K

Nilai K	Nilai Error
1	0.18
2	0.18
3	0.22
4	**0.15**
5	0.20
6	0.22
7	0.23
8	0.22
9	0.23
10	0.22
11	0.18
12	0.20
13	0.16
14	0.18
15	0.20

Dapat dilihat pada Tabel 4.12 nilai error ketika nilai K= 4 bernilai sama yaitu 0.15, nilai ini merupakan nilai error paling kecil dibandingkan dengan nilai K yang lain. Maka dapat disimpulkan nilai K yang paling ideal dalam penelitian ini adalah K=4.

2) Hasil Uji Coba Algoritma *K-Nearest Neighbor* berdasarkan pengukuran *Manhattan distance*

Hasil klasifikasi algoritma *K-Nearest Neighbor* dengan pengukuran jarak terdekat menggunakan metode *Manhattan distance* dalam penelitian ini dapat dilihat pada Tabel 4.13.

Tabel 4.13 Hasil Klasifikasi Algoritma KNN Dengan *Manhattan Distance*

Data Ke-	Y Aktual	Y Pred	TP	TN	FP	FN
1	1	0			1	
2	1	1	1			
3	1	1	1			
4	1	1	1			
5	1	1	1			
6	1	1	1			
7	1	1	1			
8	1	1	1			
9	1	1	1			
10	1	1	1			
11	1	0			1	
12	1	1	1			
13	1	1	1			
14	1	1	1			
15	1	1	1			
16	1	1	1			
17	1	0			1	
18	1	1	1			
19	1	1	1			
20	1	0			1	
21	1	1	1			
22	1	1	1			
23	1	1	1			
24	1	0			1	
25	1	0			1	
26	1	0			1	

27	1	1	1			
28	1	1	1			
29	1	1	1			
30	0	0		1		
31	0	0		1		
32	0	0		1		
33	0	0		1		
34	0	0		1		
35	0	0		1		
36	0	0		1		
37	0	0		1		
38	0	0		1		
39	0	0		1		
40	0	0		1		
41	0	0		1		
42	0	0		1		
43	0	0		1		
44	0	0		1		
45	0	0		1		
46	0	0		1		
47	0	0		1		
48	0	0		1		
49	0	0		1		
50	0	0		1		
51	0	0		1		
52	0	0		1		
53	0	0		1		
54	0	0		1		
55	0	0		1		
56	0	0		1		
57	0	0		1		
58	0	0		1		
59	0	0		1		
60	0	0		1		
Jumlah			**22**	**31**	**7**	**0**

Dari hasil klasifikasi algoritma KNN dengan pengukuran jarak terdekat menggunakan metode *Manhattan distance* pada Tabel 4.13 dapat dilihat bahwa dari

60 pasangan data ada 22 data yang termasuk dalam *true positif*, 31 data termasuk true negative, 7 data termasuk false positif, dan tidak ada data yang termasuk *false negative*. Maka berdasarkan Tabel 4.13 dapat dihitung nilai akurasi, presisi, *recall* dan *f1 measure.*

$$Akurasi = \frac{31 + 22}{31 + 22 + 7 + 0} * 100\% = 88\%$$

$$Presisi = \frac{22}{22 + 0} * 100\% = 100\%$$

$$Recall = \frac{22}{7 + 22} * 100\% = 70\%$$

$$f - measure = 2\frac{100x70}{100 + 70} = 82\%$$

3) Hasil Uji Coba Algoritma *K-Nearest Neighbor* Berdasarkan Pengukuran *Euclidean Distance*

Hasil klasifikasi algoritma *K-Nearest Neighbor* dengan pengukuran jarak terdekat menggunakan metode *Euclidean distance* dalam penelitian ini dapat dilihat pada Tabel 4.14.

Tabel 4.14 Hasil Klasifikasi Algoritma KNN Dengan *Euclidean Distance*

Data Ke-	Y Aktual	Y Pred	TP	TN	FP	FN
1	1	0			1	
2	1	1	1			
3	1	1	1			
4	1	1	1			
5	1	1	1			
6	1	1	1			
7	1	1	1			
8	1	1	1			
9	1	1	1			

10	1	1	1			
11	1	0			1	
12	1	1	1			
13	1	1	1			
14	1	1	1			
15	1	1	1			
16	1	1	1			
17	1	0			1	
18	1	1	1			
19	1	1	1			
20	1	0			1	
21	1	1	1			
22	1	1	1			
23	1	1	1			
24	1	0			1	
25	1	0			1	
26	1	0			1	
27	1	1	1			
28	1	1	1			
29	1	1	1			
30	0	0		1		
31	0	0		1		
32	0	0		1		
33	0	0		1		
34	0	0		1		
35	0	0		1		
36	0	0		1		
37	0	0		1		
38	0	0		1		
39	0	0		1		
40	0	0		1		
41	0	0		1		
42	0	0		1		
43	0	0		1		
44	0	0		1		
45	0	0		1		
46	0	0		1		
47	0	0		1		
48	0	0		1		

49	0	0		1		
50	0	0		1		
51	0	0		1		
52	0	0		1		
53	0	0		1		
54	0	0		1		
55	0	0		1		
56	0	0		1		
57	0	0		1		
58	0	0		1		
59	0	0		1		
60	0	0		1		
Jumlah			**22**	**31**	**7**	**0**

Dari hasil klasifikasi algoritma KNN dengan pengukuran jarak terdekat menggunakan metode *Euclidean distance* pada Tabel 4.14 dapat dilihat bahwa dari 60 pasangan data ada 22 data yang termasuk dalam *true positif*, 31 data termasuk *true negative*, 7 data termasuk *false positif*, dan tidak ada data yang termasuk *false negative*. Maka berdasarkan Tabel 4.14 dapat dihitung nilai akurasi, presisi, *recall* dan *f1 measure*.

$$Akurasi = \frac{31 + 22}{31 + 22 + 7 + 0} * 100\% = 88\%$$

$$Presisi = \frac{22}{22 + 0} * 100\% = 100\%$$

$$Recall = \frac{22}{7 + 22} * 100\% = 70\%$$

$$f - measure = 2\frac{100x70}{100 + 70} = 82\%$$

4) Hasil Uji Coba Algoritma *K-Nearest Neighbor* Berdasarkan Pengukuran *Minkowsky Distance*

Hasil klasifikasi algoritma *K-Nearest Neighbor* dengan pengukuran jarak terdekat menggunakan metode *Minkowsky distance* dalam penelitian ini dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil Klasifikasi Algoritma KNN Dengan *Minkowsky Distance*

Data Ke-	Y Aktual	Y Pred	TP	TN	FP	FN
1	1	0			1	
2	1	1	1			
3	1	1	1			
4	1	1	1			
5	1	1	1			
6	1	1	1			
7	1	1	1			
8	1	1	1			
9	1	1	1			
10	1	1	1			
11	1	0			1	
12	1	1	1			
13	1	1	1			
14	1	1	1			
15	1	1	1			
16	1	1	1			
17	1	0			1	
18	1	1	1			
19	1	1	1			
20	1	0			1	
21	1	1	1			
22	1	1	1			
23	1	1	1			
24	1	0			1	
25	1	0			1	
26	1	0			1	

27	1	1	1			
28	1	1	1			
29	1	1	1			
30	0	0		1		
31	0	0		1		
32	0	0		1		
33	0	0		1		
34	0	0		1		
35	0	0		1		
36	0	0		1		
37	0	0		1		
38	0	0		1		
39	0	0		1		
40	0	0		1		
41	0	0		1		
42	0	0		1		
43	0	0		1		
44	0	0		1		
45	0	0		1		
46	0	0		1		
47	0	0		1		
48	0	0		1		
49	0	0		1		
50	0	0		1		
51	0	0		1		
52	0	0		1		
53	0	0		1		
54	0	0		1		
55	0	0		1		
56	0	0		1		
57	0	0		1		
58	0	0		1		
59	0	0		1		
60	0	0		1		
Jumlah			**22**	**31**	**7**	**0**

Dari hasil klasifikasi algoritma KNN dengan pengukuran jarak terdekat menggunakan metode *Minkowsky distance* pada Tabel 4.15 dapat dilihat bahwa dari

60 pasangan data ada 22 data yang termasuk dalam *true positif*, 31 data termasuk *true negative*, 7 data termasuk *false positif*, dan tidak ada data yang termasuk *false negative*. Maka berdasarkan Tabel 4.15 dapat dihitung nilai akurasi, presisi, *recall*, dan *f1 measure*.

$$Akurasi = \frac{31 + 22}{31 + 22 + 7 + 0} * 100\% = 88\%$$

$$Presisi = \frac{22}{22 + 0} * 100\% = 100\%$$

$$Recall = \frac{22}{7 + 22} * 100\% = 70\%$$

$$f - measure = 2\frac{100x70}{100 + 70} = 82\%$$

5) Hasil Uji Coba Algoritma *K-Nearest Neighbor* Berdasarkan Pengukuran *Cebyshev Distance*

Hasil klasifikasi algoritma *K-Nearest Neighbor* dengan pengukuran jarak terdekat menggunakan metode *Cebyshev distance* dalam penelitian ini dapat dilihat pada Tabel 4.16.

Tabel 4.16 Hasil Klasifikasi Algoritma KNN Dengan *Cebyshev Distance*

Data Ke-	Y Aktual	Y Pred	TP	TN	FP	FN
1	1	0			1	
2	1	1	1			
3	1	1	1			
4	1	1	1			
5	1	1	1			
6	1	1	1			
7	1	1	1			
8	1	1	1			
9	1	1	1			
10	1	1	1			
11	1	0			1	

12	1	1	1			
13	1	1	1			
14	1	1	1			
15	1	1	1			
16	1	1	1			
17	1	0			1	
18	1	1	1			
19	1	1	1			
20	1	0			1	
21	1	1	1			
22	1	1	1			
23	1	1	1			
24	1	0			1	
25	1	0			1	
26	1	0			1	
27	1	1	1			
28	1	1	1			
29	1	1	1			
30	0	0		1		
31	0	0		1		
32	0	0		1		
33	0	0		1		
34	0	0		1		
35	0	0		1		
36	0	0		1		
37	0	0		1		
38	0	0		1		
39	0	0		1		
40	0	0		1		
41	0	0		1		
42	0	0		1		
43	0	0		1		
44	0	0		1		
45	0	0		1		
46	0	0		1		
47	0	0		1		
48	0	0		1		
49	0	0		1		
50	0	0		1		

51	0	0		1		
52	0	0		1		
53	0	0		1		
54	0	0		1		
55	0	0		1		
56	0	0		1		
57	0	0		1		
58	0	0		1		
59	0	0		1		
60	0	0		1		
Jumlah			**22**	**31**	**7**	**0**

Dari hasil klasifikasi algoritma KNN dengan pengukuran jarak terdekat menggunakan metode *Cebyshev distance* pada Tabel 4.16 dapat dilihat bahwa dari 60 pasangan data ada 22 data yang termasuk dalam *true positif*, 31 data termasuk *true negative*, 7 data termasuk *false positif*, dan tidak ada data yang termasuk *false negative*. Maka berdasarkan Tabel 4.16 dapat dihitung nilai akurasi, presisi, *recall* dan *f1 measure*.

$$Akurasi = \frac{31 + 22}{31 + 22 + 7 + 0} * 100\% = 88\%$$

$$Presisi = \frac{22}{22 + 0} * 100\% = 100\%$$

$$Recall = \frac{22}{7 + 22} * 100\% = 70\%$$

$$f - measure = 2\frac{100x70}{100 + 70} = 82\%$$

4.3 Kesimpulan

Setelah melakukan implementasi dan uji coba pada penelitian ini dapat ditemukan beberapa kesimpulan antara lain. Pemilihan nilai K paling optimal menggunakan metode elbow menunjukan hasil yang valid. Karena pada pengujian

dilakukan percobaan nilai K terhadap 1 sampai dengan 15. Hasil elbow method menunjukan bahwa nilai K yang paling optimal adalah K=4. Hal ini terbukti dengan hasil akurasi dari nilai K=4 menghasilkan akurasi yang paling tertinggi.

Pada penelitian ini juga dilakukan pengukuran kedekatan jarak antar obyek dengan 4 metode, yaitu *Manhattan Distance, Euclidean Distance, Minkowsky, Distance,* dan *Chebyshev Distance*. Hasil menunjukan bahwa hasil perhitungan dari ke empat metode yang di uji coba menghasilkan jarak yang sama dalam setiap perhitungan jarak setiap pasangan kalimat. Maka dapat disimpulkan bahwa pada penelitian ini pemilihan pengukuran jarak tidak memberikan pengaruh yang signifikan terhadap performa algoritma knn dan menghasilkan performa yang sama pada setiap metode pengukuran jarak. Adapun hasil pengukuran jarak pada setiap metode pada pengujian penelitian ini di visualisasikan pada Tabel 4.17.

Tabel 4.17 Pengukuran *Euclidean, Manhattan, Minkowsky, Cebyshev Distance*

K	Euclidean	Manhattan	Minkowsky	Chebyshev
1	0.3400778	0.3400778	0.3400778	0.3400778
2	0.	0.	0.	0.
3	0.00461328	0.00461328	0.00461328	0.00461328
4	0.07810293	0.07810293	0.07810293	0.07810293

Berdasarkan hasil pengujian yang telah diuraikan maka dapat disimpulkan bahwa algoritma knn cukup optimal untuk mengklasifikasi dataset dalam penelitian ini karena menhasilkan akurasi yang memuaskan pada penelitian ini. Rangkuman nilai akurasi pada setiap percobaan nilai K dari setiap pengukuran jarak dapat dilihat pada Tabel 4.18.

Tabel 4.18 Hasil Akurasi Pengujian Algoritma KNN

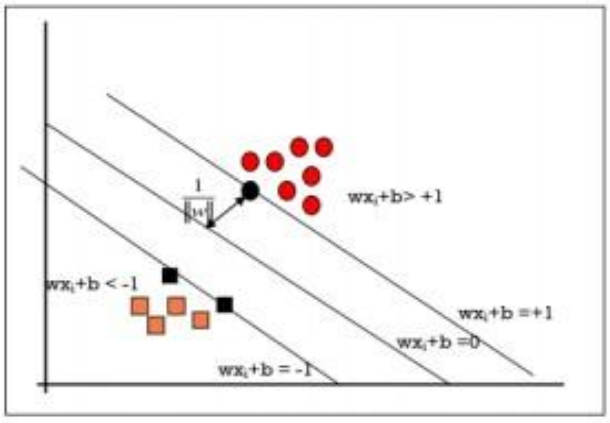
K	Euclidean	Manhattan	Minkowsky	Chebyshev
1	0.81	0.81	0.81	0.81
2	0.81	0.81	0.81	0.81
3	0.77	0.77	0.77	0.77
4	0.88	0.88	0.88	0.88
5	0.79	0.79	0.79	0.79
6	0.77	0.77	0.77	0.77
7	0.76	0.76	0.76	0.76
8	0.77	0.77	0.77	0.77
9	0.76	0.76	0.76	0.76
10	0.77	0.77	0.77	0.77
11	0.81	0.81	0.81	0.81
12	0.79	0.79	0.79	0.79
13	0.83	0.83	0.83	0.83
14	0.81	0.81	0.81	0.81
15	0.79	0.79	0.79	0.79

BAB V
METODE SUPPORT VECTOR MACHINE

5.1 Desain & Implementasi

Support Vector Machine (SVM) adalah metode yang mempelajari area yang memisahkan antar kategori dalam sebuah observasi. Dalam terminologi SVM, kita membahas jarak atau margin antar kategori. Setiap kategori memiliki observasi dimana nilai variabel targetnya sama (Williams, 2011). SVM juga dikenal sebagai sistem pembelajaran yang menggunakan hipotesis fungsi linear dalam ruang dimensi tinggi dan dilatih dengan algoritma berdasarkan teori optimasi dengan menerapkan learning bias yang berasal dari teori statistik. Tujuan dari metode ini adalah membangun pemisah optimum yang disebut OSH (*Optimal Separating Hyperplane*) sehingga dapat digunakan untuk klasifikasi.

Hyperplane terbaik antara kedua kelas dapat ditemukan dengan melakukan pengukuran margin *hyperplane* dan kemudian mencari titik maksimalnya. Margin adalah jarak antar *hyperplane* tersebut dengan data terdekat dari masing – masing kelas. Data yang paling dekat ini, disebut dengan *support vector*. Ilustrasi *hyperplane* ditunjukan pada Gambar 5.1.



Gambar 5.1 Margin *Hyperplane*

Seperti pada Gambar 5.1, *Support Vector Machine* bekerja menemukan *hyperplane* dengan margin yang maksimal. *Hyperplane* klasifikasi linear memisahkan kedua kelas dengan persamaan 5.1.

$$w.\ x_i + b = 0 \qquad (5.1)$$

Keterangan:

w = vector bobot

x = nilai masukan atribut

b = bias

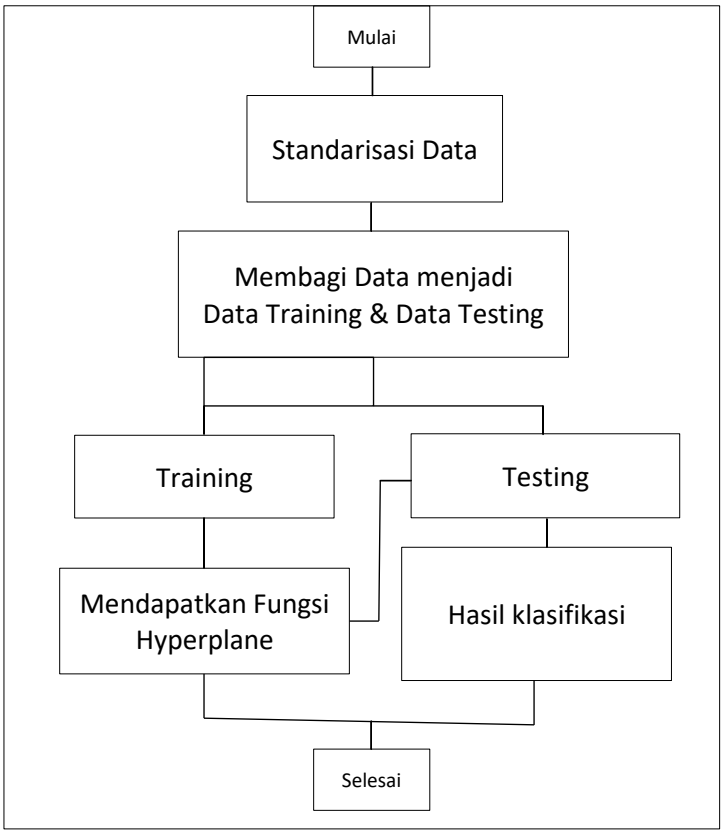
Sehingga didapatkan persamaan untuk kelas positif dan kelas negatif. Maka, suatu data x_i dapat diklasifikasikan sebagai kelas +1 jika :

$$w.\ x_i + b > 1 \qquad (5.2)$$

dan dapat diklasifikasikan kedalam kelas -1 jika :

$$w.\ x_i + b \leq -1 \qquad (5.3)$$

Adapun pada algoritma SVM *flowchart* proses klasifikasi *Support Vector Machine* pada penelitian ini ditunjukan pada Gambar 5.2.



Gambar 5.2 *Flowchart* Proses Klasifikasi SVM

5.1.1. Standarisasi Data

Standarisasi data adalah teknik statistik yang berfungsi untuk menormalisasikan range pada fitur—fitur data sehingga seluruh fitur berada pada range yang sama. Standarisasi data dilakukan supaya hasil prediksi lebih akurat sehingga seluruh fitur dapat berkontribusi mempengaruhi hasil prediksi. Dalam melakukan standarisasi dapat dihitung menggunakan rumus *Z-Score* seperti Rumus 5.4.

$$X = \frac{x-\mu}{\sigma} \quad (5.4)$$

Keterangan :

μ = Nilai rata-rata fitur

σ = Standar deviasi

Adapun hasil implementasi standarisasi data pada penelitian ini menggunakan persamaan rumus *Z-Score* dapat dilihat pada Tabel 5.1.

Tabel 5.1 Standarisasi Data

No	X_train	Hasil Standarisasi
1	0,696343	0,97885132
2	0	-1,32669202
3	0,342372	-0,19312049
4	0,73685	1,11296845
5	0,731068	1,09382347
…	…	…
56	0,856064	1,50767957
57	0,108919	-0,96606855
58	0	-1,32669202
59	0,191932	-0,69121692
60	0,740766	1,12593273

5.1.2. Membagi Data menjadi Data Training & Data Testing

Pembagian data ini berfungsi untuk mengevaluasi performa model *support vector machine* yang akan dibangun. *Data training* adalah sekumpulan data yang akan dilatih untuk membuat model dari algoritma *support vector machine*. Sedangkan *data testing* adalah sekumpulan data untuk mengevaluasi performa dari model yang telah dihasilkan apakah memiliki hasil yang akurat atau sebaliknya. Rasio pembagian *data training* dan *data testing* dalam penelitian ini adalah 80% *data training* dan 20% *data testing*.

5.1.3 Training

Proses training / pelatihan diawali dengan nilai fitur dari semua fitur yang telah didapatkan dari proses *preprocessing*. Kemudian nilai bobot parameter diinisiasi terlebih dahulu menggunakan Rumus 5.5.

$$w_{\text{awal}} = 1/total \text{ fitur} \quad (5.5)$$

Pada penelitian ini terdapat 2 fitur maka nilai *w* untuk inisiasi bobot awal adalah 1/total fitur. Kemudian tahap selanjutnya yaitu menghitung nilai parameter pada setiap fitur yang tersedia, seperti langkah-langkah berikut ini :

1. Menghitung fungsi *hyperplane* menggunakan persamaan 5.6.

$$f(x) = \sum_{i=1}^{totalfitur} w_i * x_i \quad (5.6)$$

Keterangan :

w_i : bobot dari setiap fitur

x_i : nilai parameter dari masing-masing fitur

2. Kemudian dari hasil persamaan 5.6 maka akan dihasilkan :

   a. Jika hasil dari $f(y_i) > 0$ maka $y_i = +1$

   b. Jika hasil dari $f(y_i) < 0$ maka $y_i = -1$

3. Jika nilai *f(y)* sama dengan target maka nilai *w* tidak perlu diperbaharui

5. Jika nilai *f(y)* tidak sama dengan target, maka nilai *w* diperbaharui dengan menggunakan Rumus 5.7.

$$w_{baru} = w_{awal} + y_i \times x_{ij} \quad (5.7)$$

5. Iterasi berlanjut sampai nilai *w* tidak berubah.

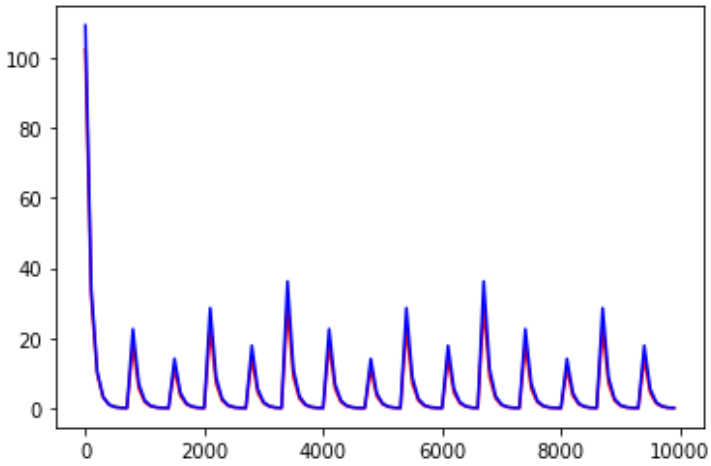
6. Hasil dari pelatihan tersebut akan digunakan dalam mencari nilai *hyperplane* sebagai batas nilai positif (+1) dan negatif (-1). Dari batas tersebut terdapat jarak atau *margin*. *Margin* tersebut diformulasikan pada Rumus 5.8.

$$C = \frac{(a+b)}{2} \quad (5.8)$$

Dimana a merupakan nilai $(y_i) > 0$ dan b merupakan nilai $(y_i) < 0$. Hasil dari tahap pelatihan ini adalah nilai dari masing-masing parameter (w). nilai setiap parameter akan bervariasi. Dengan parameter tersebut, fungsi *hyperplane* dapat dibuat. Fungsi *hyperplane*, dapat ditulis seperti pada persamaan 5.9.

$$y\left(\sum_{i=1}^{totalfitur} w_i * x_i\right) \geq C \quad (5.9)$$

Adapun visualisasi proses *training* / pelatihan algoritma *support vector machine* pada penelitian ini disajikan pada Gambar 5.3.



Gambar 5.3 Grafik Tahap Pelatihan

5.1.4 Testing

Proses *testing*/ pengujian sama dengan proses pelatihan, namun pada proses pengujian menggunakan nilai parameter untuk fitur yang diperoleh dari proses pelatihan. Alur proses pada tahap pengujian adalah seperti berikut ini :

1. Memasukan teks yang akan diuji (data t*esting*)

2. Kemudian ekstrak fitur tersebut

3. Nilai hasil ekstrak fitur diolah menggunakan Rumus 5.9

4. Jika hasil f(y) < margin maka dokumen tersebut tergolong dokumen non-parafrase, jika sebaliknya, maka dokumen tergolong dokumen parafrase.

5.2 Uji Coba

Tahap uji coba algoritma *support vector machine* pada penelitian ini menggunakan fungsi kernel linear untuk melakukan klasifikasi data. Kernel linear adalah fungsi kernel yang paling sederhana. Kernel linear digunakan ketika data yang dianalisis sudah terpisah secara linear. Rumus persamaan untuk fungsi kernel linear dapat dilihat pada persamaan 5.1

Berikut ini adalah cara kerja klasifikasi algoritma SVM dalam penelitian ini. Tabel 5.3 adalah sample dataset pasangan *input* dan *output* pada penelitian ini.

Tabel 5.2 Pasangan *input* dan *output*

No	**X**	**Y_i**
1	0.97885132	1
2	-1.32669202	1
3	-0.19312049	-1
…	…	…

58	0.09695081	-1
59	1.36382549	1
60	0.22459681	1

Pada Tabel 5.2 terdapat atribut X yang akan menghasilkan nilai bobot w. Kemudian margin diminimalkan menggunakan Rumus 5.10.

$$y_i(x_i \ . \ w + b) - 1 \ \geq 0 \tag{5.10}$$

dengan syarat

$$y_i(x_i \ \cdot w + b) \geq 1, i = 1,2,3 \ldots, N \tag{5.11}$$

Sehingga dari persamaan diatas didapatkan persamaan berikut ini :

(1) 1(0.97885132 w + b) ≥1→ (0.97885132 w + b) ≥1 (5.12)

(2) 1(-1.32669202 w + b) ≥1→ (-1.32669202 w + b) ≥1 (5.13)

(3) -1(-0.19312049 w + b) ≥1→ (0.19312049 w - b) ≥1 (5.14)

(4) -1(0.09695081 w + b) ≥1→ (-0.09695081 w - b) ≥1 (5.15)

(5) 1(1.36382549 w + b) ≥1→ (1.36382549 w + b) ≥1 (5.16)

(6) 1(0.22459681w + b) ≥1→ (0.22459681 w + b) ≥1 (5.17)

Selanjutnya mencari nilai w dan b dari persamaan 5.12 dan 5.14.

(0.97885132 w + b) ≥1

(0.19312049 w - b) ≥1

——————————— +

1,17197181 w = 2

w = 2/1,17197181

w=1,72

Dari persamaan diatas dihasilkan nilai w sebesar 1,72. Kemudian Langkah selanjutnya adalah mencari nilai b dengan mensubstitusikan nilai w ke dalam persamaan 5.12.

(0.97885132 w + b) ≥1

(0.97885132 (1,72) + b) ≥1

1,66 + b = 1

b = 1-1,66

b = 0,66

Maka Persamaan hyperplane menjadi seperti berikut ini :

$$w.x + b = 0 \qquad (5.18)$$

$$1{,}72\ x + 0{,}66 = 0 \qquad (5.19)$$

Setelah mendapatkan garis hyperplane, maka langkah selanjutnya yaitu mengklasifikasikan data uji melalui hyperplane dengan menggunakan persamaan *hyperplane* pada persamaan 5.19 dengan $(x) \coloneqq sgn(f(x))$. Hasil klasifikasi menggunakan algoritma *Support Vector Machine* dalam penelitian ini dapat dilihat pada Tabel 5.3.

Tabel 5. 3 Hasil Klasifikasi Algoritma SVM

Data Ke-	Y Aktual	Y Pred	TP	TN	FP	FN
1	1	1	1			
2	1	1	1			
3	1	0			1	
4	1	0			1	
5	1	1	1			
6	1	1	1			
7	1	1	1			
8	1	1	1			

9	1	1	1			
10	1	1	1			
11	1	1	1			
12	1	1	1			
13	1	1	1			
14	1	1	1			
15	1	1	1			
16	1	1	1			
17	1	0			1	
18	1	1	1			
19	1	1	1			
20	1	1	1			
21	1	1	1			
22	1	1	1			
23	1	1	1			
24	1	0			1	
25	1	0			1	
26	1	1	1			
27	1	1	1			
28	1	1	1			
29	1	1	1			
30	0	0		1		
31	0	0		1		
32	0	0		1		
33	0	0		1		
34	0	0		1		
35	0	0		1		
36	0	0		1		
37	0	0		1		
38	0	0		1		
39	0	0		1		
40	0	1				1
41	0	0		1		
42	0	1				1
43	0	0		1		
44	0	1				1
45	0	0		1		
46	0	0		1		
47	0	0		1		

48	0	0		1		
49	0	0		1		
50	0	0		1		
51	0	1				1
52	0	0		1		
53	0	0		1		
54	0	0		1		
55	0	0		1		
56	0	1				1
57	0	0		1		
58	0	0		1		
59	0	0		1		
60	0	0		1		
JUMLAH			24	26	5	5

Dari hasil klasifikasi algoritma SVM menggunakan kernel linear pada Tabel 5.3 dapat dilihat bahwa dari 60 pasangan data ada 24 data yang termasuk dalam *true positif*, 26 data termasuk *true negative*, 5 data termasuk *false positif*, dan 5 data termasuk *false negative*. Maka berdasarkan Tabel 5.4 dapat dihitung nilai akurasi, presisi, *recall* dan *f1 measure*.

$$Akurasi = \frac{26 + 24}{26 + 5 + 5 + 24} * 100\% = 83\%$$

$$Presisi = \frac{24}{24 + 5} * 100\% = 82\%$$

$$Recall = \frac{24}{5 + 24} * 100\% = 82\%$$

$$f - measure = 2\frac{82.82}{82+82} = 82\%$$

5.4 Kesimpulan

Setelah melakukan implementasi algoritma SVM dan uji coba pada penelitian ini dapat diambil beberapa kesimpulan antara lain yaitu Algoritma SVM

dapat diimplementasikan dalam melakukan klasifikasi terhadap dokumen yang di parafrase. Berdasarkan uji coba yang telah dilakukan terhadap dataset, menunjukan bahwa hasil klasifikasi menggunakan algoritma *support vector machine* linear yaitu sebesar 83%.

Berdasarkan hasil pengujian yang telah diuraikan maka dapat disimpulkan bahwa algoritma SVM cukup optimal untuk mengklasifikasi *dataset* dalam penelitian ini karena menhasilkan akurasi yang memuaskan pada penelitian ini. Rangkuman hasil pengujian algoritma support vector machine dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Akurasi Pengujian Algoritma SVM

Pengujian	**Hasil**
Akurasi	83%
Precision	82%
Recall	82%
f-measure	82%

BAB VI
PEMBAHASAN

6.1 Pembahasan Kompleksitas Deteksi Parafrase

Dalam melakukan deteksi parafrase pada penelitian ini dilakukan tahap preprocessing untuk meningkatkan kualitas data. Tahap *preprocessing* pada penelitian ini terdiri dari *remove punctuation, case folding, tokenizing, filtering,* dan *stemming.* Kemudian setelah data melewati tahap *preprocessing* selanjutnya dilakukan tahap pembobotan setiap *term* dalam dokumen menggunakan algoritma TF IDF. Setelah mendapatkan nilai TF IDF setiap term, selanjutnya dihitung nilai *cosine similarity* untuk mengetahui kesamaan antara dokumen kemudian setiap pasangan data di klasifikasi menggunakan algoritma *K-Nearest Neighbor* dan *Support Vector Machine* untuk menentukan dokumen parafrase dan non parafrase.

Berdasarkan hasil uji coba yang telah dilakukan pada algoritma K-Nearest Neighbor dapat diketahui bahwa penggunaan nilai K yang berbeda mempengaruhi klasifikasi pada algoritma KNN. Hal ini dikarenakan pemilihan nilai K harus mempertimbangkan jumlah data dan dimensi data. Pada pengujian dilakukan percobaan nilai K terhadap 1 sampai dengan 15 menggunakan *elbow method*. Hasil dari *elbow method* menunjukan bahwa nilai K yang paling optimal adalah K=4. Hal ini terbukti dengan hasil akurasi dari nilai K=4 menghasilkan akurasi yang paling tertinggi yaitu sebesar 88%. Ramkuman nilai akurasi setiap nilai K dapat dilihat pada Tabel 6.1.

Tabel 6.1 Nilai Akurasi Setiap Nilai K

Nilai K-	Nilai Akurasi
1	81%
2	81%
3	84%
4	88%
5	83%
6	83%
7	83%
8	83%
9	81%
10	77%
11	81%
12	83%
13	83%
14	83%
15	83%

Pada uji coba algoritma KNN juga dilakukan pengukuran kedekatan jarak antar obyek dengan 4 metode, yaitu *Manhattan Distance*, *Euclidean Distance, Minkowsky, Distance,* dan *Chebyshev Distance*. Hasil menunjukan bahwa hasil perhitungan dari ke empat metode yang di uji coba menghasilkan jarak yang sama dalam setiap perhitungan jarak setiap pasangan kalimat. Maka dapat disimpulkan bahwa pada penelitian ini pemilihan pengukuran jarak tidak memberikan pengaruh yang signifikan terhadap performa algoritma knn dan menghasilkan performa yang sama pada setiap metode pengukuran jarak.

Selanjutnya, uji coba dilakukan pada algoritma *support vector machine* menggunakan kernel linear mendapatkan hasil performa yang cukup memuaskan dengan nilai akurasi 83%, *precision* 82% , *recall* 82%, dan *f-measure* 82%.

6.2 Pembahasan Komparasi Performa Algoritma

Pada bab ini dilakukan komparasi atau perbandingan hasil performa dari algoritma K-Nearest Neighbor dengan algoritma Support Vector Machine. Pengukuran kinerja algoritma K-Nearest Neighbor dan algoritma Support Vector Machine pada penelitian ini dengan membandingkan adalah nilai akurasi, *precision, recall* dan *f1-score.*

Berdasarkan hasil uji coba pada penelitian ini algoritma KNN menghasilkan nilai akurasi 88%, *precision* 100% , *recall* 70%, dan *f-measure* 82%. Selanjutnya, pada uji coba yang telah dilakukan pada algoritma *support vector machine* menghasilkan performa klasifikasi yang lebih rendah dari hasil performa klasifikasi algoritma KNN dengan nilai akurasi 83%, *precision* 82% , *recall* 82%, dan *f-measure* 82%.

Berdasarkan pembahasan performa klasifikasi algoritma KNN dan *support vector machine* yang telah diuraikan pada bagian sebelumnya, didapatkan bahwa algoritma KNN merupakan algoritma yang lebih baik dalam melakukan klasifikasi pada dataset dalam penelitian ini. Rangkuman pengukuran performa algoritma KNN dan support vector machine disajikan pada Tabel 6.2.

Tabel 6.2 Hasil perbandingan performa algoritma

No	Algoritma	Akurasi	Precision	Recall	F-Measure
1	K- Nearest Neighbor	88%	100%	70%	82%
2	Support Vector Machine	83%	82%	82%	82%

Berdasarkan pada Tabel 6.2 hasil performa algoritma *K-Nearest Neighbor* dan *Support Vector Machine* cukup optimal untuk mengklasifikasi dokumen yang di parafrase dalam penelitian ini, karena menghasilkan akurasi yang memuaskan pada penelitian ini. Namun, algoritma K-Nearest Neighbor menunjukan hasil performa klasifikasi yang lebih unggul dari algoritma *support vector machine*.

Dengan nilai akurasi sebesar 88% dapat disimpulakan bahwa algoritma KNN dapat digunakan untuk membantu mengklsifikasi dokumen parafrase dan dokumen non-parafrase. Parafrase adalah bentuk pengungkapan kembali suatu tata bahasa, kalimat, atau pernyataan dengan menggunakan diksi yang lebih sederhana tanpa mengubah makna dari bahasa tersebut. Dengan adanya sistem klasifikasi dokumen parafrase ini diharapkan dapat mendorong sikap jujur dan tanggung jawab setiap orang dalam melakukan penulisan penelitian, karya ilmiah atau penulisan lainnya. Allah SWT memerintahkan hambanya untuk selalu berbuat baik kepada orang lain dan selalu bersikap jujur. Sebagaimana firman Allah s.w.t dalam surat al-Ahzab ayat 70-71:

(يُصْلِحْ لَكُمْ أَعْمَالَكُمْ وَيَغْفِرْ 70يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَقُولُوا قَوْلا سَدِيدًا)
لَكُمْ ذُنُوبَكُمْ وَمَنْ يُطِعِ اللَّهَ وَرَسُولَهُ فَقَدْ فَازَ فَوْزًا عَظِيم (71)

Artinya : "*Hai orang-orang yang beriman, bertakwalah kamu kepada Allah dan katakanlah perkataan yang benar, niscaya Allah memperbaiki bagimu amalan-amalanmu dan mengampuni bagimu dosa-dosamu. Dan barang siapa menaati Allah dan Rasul-Nya, maka sesungguhnya ia telah mendapat kemenangan yang besar*."

Merujuk pada tafsir Ibnu Katsir, surat al-Ahzab ayat 70-71 menjelaskan bahwa Allah Swt. memerintahkan kepada hamba-hamba-Nya yang beriman agar tetap bertakwa kepada-Nya dan menyembah-Nya dengan penyembahan sebagaimana seseorang yang melihat-Nya, dan hendaklah mereka mengucapkan perkataan yang benar, yang jujur, tidak bengkok, tidak pula menyimpang. Lalu Allah menjanjikan kepada mereka jika mereka melakukan perintah-perintah-Nya ini, Dia akan memberi mereka pahala dengan memperbaiki amal perbuatan mereka. Yakni Allah memberi mereka taufik untuk mengerjakan amal-amal yang saleh, dan bahwa Allah akan mengampuni dosa-dosa mereka yang terdahulu. Sedangkan dosa yang akan mereka lakukan di masa mendatang, Allah akan memberi mereka ilham untuk bertobat darinya.

Maka berdasarkan surat al-Ahzab ayat 70-71, telah jelas bahwa Allah telah memerintahkan hambanya untuk selalu berprilaku jujur dalam hal apapun. Bahkan pada surat al-Ahzab ayat 70-71 Allah juga telah menyebutkan ganjaran yang besar bagi orang yang jujur. Allah pun ridha terhadap orang-orang yang jujur karena perilaku jujurnya merupakan bentuk ketaatan kepada Rabb-Nya. Maka ketika seseorang membuat sebuah karya tulis dengan jujur bahwa karya yang ditulis berdasarkan idenya sendiri dan informasi yang dituangkan dalam tulisan bisa dipertanggung jawabkan oleh penulisnya, maka orang itu akan mendapatkan kebermanfaatan dari karya nya dan mendapatkan ganjaran pahala seperti yang telah

Allah janjikan dala surat al-Ahzab ayat 70-71. Begitupula sebaliknya, ketika seseorang penulis membuat karya tulis dengan tidak jujur, maka karya nya itu tidak akan membawa manfaat kepadanya di akhirat kelak dan akan mendapatkan siksa yang pedih sebagaimana yang telah disabdakan oleh Rasulullah SAW :

رَأَيْتُ رَجُلَيْنِ أَتَيَانِي، قَالاَ: اَلَّذِى رَأَيْتَهُ يُشَقُّ شِدْقُهُ فَكَذَّابٌ، يَكْذِبُ بِالْكَذْبَةِ تُحْمَلُ عَنْهُ حَتَّى تَبْلُغُ الآفَاقَ، فَيُصْنَعُ بِهِ إِلَى يَوْمِ الْقِيَامَةِ

"Aku melihat dalam mimpi dua orang Malaikat, keduanya berkata: "Orang yang engkau lihat mulutnya dikoyak hingga telinga, adalah seorang pembohong. Ia berbohong hingga kebohongannya tersebut dibebankan kepadanya hingga mencapai ufuk, maka dibuatlah ia diberi beban seperti itu hingga hari kiamat." (HR. Bukhari).

Dengan hadirnya sistem ini diharapkan mampu melakukan klasifikasi terhadap dokumen parafrase dengan hasil yang lebih baik dan akurat sehingga sistem yang telah dibangun diharapkan dapat dijadikan alat bantu seorang penulis dalam mendapatkan kebenaran mengenai hasil pengukuran parafrase pada dokumen. Sistem ini juga diharapkan dapat membantu seorang penulis untuk melakukan pengecekan apakah karyanya adalah termasuk kategori dokumen parafrase atau tidak, sehingga dapat terhindar dari praktik plagiarisme. Dan mejadikan seorang penulis lebih kreatifitas dalam membuat sebuah karya tulis sebagai idenya sendiri juga lebih hati – hati dalam menuangkan idenya sendiri pada karya tulis agar tidak termasuk dalam plagiarisme parafrase dan bisa mempertanggung jawabkan apa yang ditulis oleh dirinya sendiri

BAB VII
KESIMPULAN

7.1 Kesimpulan

Berdasarkan serangkaian uji coba yang telah dilakukan dalam deteksi plagiarisme berdasarkan paraphrase pada teks Bahasa Indonesia menggunakan algoritma *K-Neirest Neighbor* dan *Support Vector Machine* dapat disimpulkan beberapa hal berikut ini :

1. Pada uji coba yang telah dilakukan pada penelitian ini dapat diketahui bahwa deteksi parafrase sulit diidentifikasi karena beberapa faktor, diantaranya yaitu setiap kalimat harus dilakukan ekstraksi supaya dapat mengambil makna yang terkandung dalam suatu kalimat. Dari percobaan yang telah dilakukan masih ditemukan kesalahan /*error* dalam mengklasifikasikan dokumen parafrase. Hal ini disebabkan kekurangan pada tahap *preprocessing* dalam melakukan proses penghapusan *stopword.* Pada tahap penghapusan *stopwords* terdapat kata -kata penting yang terhapus karena terdeteksi dalam kata-kata tidak penting. Sehingga sistem tidak optimal dalam melakukan klasifikasi. Kemudian proses *stemming* pada penelitian ini memiliki kekurangan yaitu penyamarataan makna variasi kata. Sehingga kata yang memiliki makna sama tidak teridentifikasi sebagai kalimat parafrase. Selanjutnya, algoritma *K-Neirest Neighbor* terdapat parameter yang mempengaruhi kinerja performa algoritma KNN yaitu parameter K. Penggunaan nilai K yang berbeda akan menghasilkan akurasi yang berbeda pula pada setiap pengujian. Pada penelitian ini digunakan *elbow method* untuk

mendapatkan nilai K yang optimal, dari uji coba yang telah dilakukan didapatkan bahwa nilai K yang paling optimal adalah ketika K=4.

2. Pada penelitian ini klasifikasi dokumen parafrase dan non parafrase dilakukan menggunakan algoritma *K-Neirest Neighbor* dan *Support Vector Machine*. Berdasarkan uji coba yang telah dilakukan *K-Nearest Neighbor* lebih unggul dalam melakukan identifikasi dokumen parafrase dengan nilai akurasi sebesar 88%, *precision* 100% , *recall* 70%, dan *f-measure* 82%. Adapun algoritma *Support Vector Machine* menghasilkan performa yang lebih rendah dengan presentasi nilai akurasi sebesar 83%, *precision* 82% , *recall* 82%, dan *f-measure* 82%.

7.2 Saran

Peneliti menyadari bahwa penelitian ini masih memiliki kekurangan dan masih harus dilakukan penelitian lebih lanjut, guna meningkatkan hasil yang lebih baik. Adapun saran untuk penelitian selanjutnya adalah :

1. Perlunya *dataset* yang lebih banyak, sehingga dapat membandingkan dengan berbagai sumber yang lebih bervariatif dan hasil yang didapatkan menjadi lebih akurat.
2. Pada penelitan berikutnya dapat digunakan tahap preprocessing yang dapat melakukan ekstraksi suatu kalimat dengan lebih tepat.
3. Deteksi plagiarisme berdasarkan parafrase pada teks Bahasa Indonesia dapat dilakukan dengan menggunakan metode *machine learning* lainnya

Daftar Pustaka

A. Chitra and A. Rajkumar, "Evolutionary approach for building efficient Paraphrase Recognizers," 2011 World Congress on Information and Communication Technologies, 2011, pp. 661-666, doi: 10.1109/WICT.2011.6141324.

Alfikri, Z. F., & Purwarianti, A. (2014). Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach (Naive Bayes and SVM). *TELKOMNIKA Indonesian Journal of Electrical Engineering*, *12*(11), 7794–7804. https://doi.org/10.11591/telkomnika.v12i11.6652

Alvi, F., Stevenson, M., & Clough, P. (2021). Parafrase type identification for plagiarism detection using contexts and word embeddings. *International Journal of Educational Technology in Higher Education*, *18*(1). https://doi.org/10.1186/s41239-021-00277-8

Alzahrani, S. M., Salim, N., & Abraham, A. (2012). Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *42*(2), 133–149. https://doi.org/10.1109/TSMCC.2011.2134847

Chitra, A., & Rajkumar, A. (2013). Genetic algorithm based feature selection for paraphrase recognition. *International Journal on Artificial Intelligence Tools*, *22*(2), 1–17. https://doi.org/10.1142/S0218213013500073

Chitra, A., & Rajkumar, A. (2016). Plagiarism Detection Using Machine Learning-Based Parafrase Recognizer. *Journal of Intelligent Systems*, *25*(3), 351–359. https://doi.org/10.1515/jisys-2014-0146

Clough, Paul & Gaizauskas, Rob & Piao, Scott. (2002). Building and annotating a corpus for the study of journalistic text reuse.

Clough, Paul & Stevenson, Mark. (2011). Developing a corpus of plagiarised short answers. Language Resources and Evaluation. 45. 5-24. 10.1007/s10579-009-9112-1.

D. Wang and H. Zhang, “Inverse-Category-Frequency based supervised term weighting scheme for text categorization,” J. Inf. Sci. Eng., vol. 29, no. 2, pp. 209– 225, Dec. 2010.

F. Ren and M. G. Sohrab, “Class-indexing-based term weighting for automatic text classification,” Inf. Sci., vol. 236, pp. 109–125, Jul. 2013

Foltýnek, T., Meuschke, N., & Gipp, B. (2019). Academic plagiarism detection: A systematic literature review. *ACM Computing Surveys*, *52*(6). https://doi.org/10.1145/3345317

Hunt, E., Janamsetty, R., Kinares, C., Koh, C., Sanchez, A., Zhan, F., Ozdemir, M., Waseem, S., Yolcu, O., Dahal, B., Zhan, J., Gewali, L., & Oh, P. (2019). Machine learning models for parafrase identification and its applications on plagiarism detection. *Proceedings - 10th IEEE International Conference on Big Knowledge, ICBK 2019*, 97–104. https://doi.org/10.1109/ICBK.2019.00021

Julianto, B. I., Mubarok, M. S., Batu, T. B., & Barat, J. (2017). Identifikasi Parafrasa Bahasa Indonesia Menggunakan Naïve Bayes. *E-Proceeding of Engineering*, *4*(3), 4978–4982.

Kodinariya, Trupti M. & Makwana, Prashant R., (2013). Review on determining number of cluster in K-Means Clustering. International Journal of Advance Research in Computer Science and Management Studies, I(6), pp. 90-95

M. Lan, C. L. Tan, J. Su, and Y. Lu, ―Supervised and traditional term weighting

Madhulatha, T.S., 2012. An Overview On Clustering Methods. IOSR Journal of Engineering, II(4), pp.719-725

methods for automatic text categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 4, 2009, pp.721-735.

P. Vigneshvaran, E. Jayabalan and A. V. Kathiravan, "An Eccentric Approach for Paraphrase Detection Using Semantic Matching and Support Vector Machine," 2014 International Conference on Intelligent Computing Applications, 2014, pp. 431-434, doi: 10.1109/ICICA.2014.94.

Ren, F., & Sohrab, M. G. (2013). Class-indexingbased term weighting for automatic text classification. 236, 109–125.

Sabbah, T., Selamat, A., Selamat, M. H., Al-Anzi, F. S., Viedma, E. H., Krejcar, O., & Fujita, H. (2017). Modified frequency-based term weighting schemes for text classification. Applied Soft Computing Journal, 58, 193–206. https://doi.org/10.1016/j.asoc.2017. 04.069

Subroto, Imam & Selamat, Ali. (2014). Plagiarism Detection through Internet using Hybrid Artificial Neural Network and Support Vectors Machine. TELKOMNIKA (Telecommunication Computing Electronics and Control). 12. 209. 10.12928/TELKOMNIKA.v12i1.648.

Thompson, V. (2017). *Methods for Detecting Parafrase Plagiarism*. 1–21. http://arxiv.org/abs/1712.10309