Perbandingan Kinerja Metode Klasifikasi *Naïve Bayes* Dan *Random Forest* Dalam Analisis Sentimen Kasus Narkoba di Indonesia Pada Komentar YouTube

SKRIPSI

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)
dalam Teknologi Informasi

Diajukan oleh:
NAILUL 'INAYAH
NIM: 2008096008

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG

2023

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Nailul ‘Inayah
NIM : 2008096008
Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

Perbandingan Kinerja Metode Klasifikasi Naïve Bayes Dan Random Forest Dalam Analisis Sentimen Kasus Narkoba Di Indonesia Pada Komentar YouTube

Secara keseluruhan adalah penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 29 November 2023

Pembuat pernyataan,

Nailul ‘Inayah

NIM : 2008096008

KEMENTRIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601295 Fax.7615387

PENGESAHAN

Naskah skripsi berikut ini:

Judul	:	Perbandingan Kinerja Metode Klasifikasi Naïve Bayes Dan Random Forest Dalam Analisis Sentimen Kasus Narkoba Di Indonesia Pada Komentar YouTube
Penulis	:	**Nailul 'Inayah**
NIM	:	2008096008
Jurusan	:	Teknologi Informasi

Telah diujikan dalam siding tugas akhir oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam bidang ilmu Teknologi Informasi.

Semarang, Desember 2023

DEWAN PENGUJI

Penguji I,

Dr. Khothibul Umam, M.kom
NIP.197908272011011007

Penguji II,

Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP. 197312222006041001

Penguji III,

Wenty Dwi Yuniarti, S.Pd, M.Kom
NIP. 197706222006042005

Penguji IV,

Siti Nur'aini, M.Kom
NIP. 198401312018012001

Pembimbing I,

Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP. 197312222006041001

Pembimbing II,

Adzhal Arwani Mahfudh, M.Kom
NIP. 199107032019031006

NOTA PEMBIMBING

Semarang, 29 November 2023

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	:	Perbandingan Kinerja Metode Klasifikasi Naïve Bayes Dan Random Forest Dalam Analisis Sentimen Kasus Narkoba di Indonesia Pada Komentar YouTube
Penulis	:	Nailul 'Inayah
NIM	:	2008096008
Jurusan	:	Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing I,

Nur Cahyo Hendro Wibowo, S.T. M.Kom
NIP. 197312222006041001

NOTA PEMBIMBING

Semarang, 29 November 2023

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	:	Perbandingan Kinerja Metode Klasifikasi Naïve Bayes Dan Random Forest Dalam Analisis Sentimen Kasus Narkoba di Indonesia Pada Komentar YouTube
Penulis	:	Nailul 'Inayah
NIM	:	2008096008
Jurusan	:	Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing II,

Adzhal Arwani Mahfudh, M.Kom
NIP. 199107032019031006

ABSTRAK

Analisis sentimen merupakan suatu langkah atau proses untuk pengukuran suatu opini atau penilaian terhadap suatu permasalahan dengan tujuan nantinya dapat mencari jalan keluar dari permasalahan tersebut. Kasus narkoba di Indonesia merupakan permasalahan yang tidak ada habisnya untuk dibahas, karena keberadaannya yang semakin hari semakin mengkhawatirkan. Hal tersebut tentu mengundang banyak opini masyarakat. Untuk itu, analisis sentimen pada penelitian ini mengangkat tema tentang kasus narkoba di Indonesia yang opininya diambil dari komentar youtube.

Analisis sentimen dilakukan dengan menggunakan metode klasifikasi. Pada penelitian kali ini, peneliti akan mengukur dan membandingkan antara kinerja metode klasifikasi *naïve bayes* dan *random forest*.

Berdasarkan hasil penelitian yang dilakukan hasil akhir dari akurasi metode *naïve bayes* adalah 83% dan akurasi *random forest* 83%, meskipun terlihat sama, namun pada nilai akurasi *naïve bayes* lebih tinggi 0,0038 dari *random forest.* Hasil dari nilai presisi metode *naïve bayes* adalah 81% sedangkan *random forest* 65% dengan selisih 16% *naïve bayes* lebih tinggi. Hasil dari nilai *recall,* metode *naïve bayes* adalah 83%, sedangkan *random forest* 85% dengan selisih 2% *random forest* lebih tinggi. Hasil akhir *f1 score* pada metode *naïve bayes* adalah 81%, sedangkan *random forest* 73% dengan selisih 8% *naïve bayes* lebih tinggi. Pada nilai akurasi, presisi dan *f1 score,* metode *naïve bayes* lebih unggul *random forest,* sedangkan perhitungan *recall, random forest* lebih unggul dari *naïve bayes.*

Kata kunci: analisis sentimen, narkoba, *naïve bayes, random forest*

KATA PENGANTAR

Alhamdulillahi rabbil'aalamiin, puji syukur atas segala keadaan, ni'mat dan karunia terhaturkan kepada Allah *subhanahu wa ta'ala* yang dimana segala sesuatu yang dikehendaki-Nya pasti terjadi dan oleh karenanya, saya dapat menyelesaikan Skripsi yang berjudul "Perbandingan Kinerja Metode Klasifikasi *Naïve Bayes* Dan *Random Forest* Dalam Analisis Sentimen Kasus Narkoba di Indonesia Pada Komentar YouTube" dengan baik. Shalawat dan salam Sejahtera terhaturkan kepada Rasulullah Muhammad *shallallahu 'alaihi wa sallam*, yang dengan kehadirannya, turun Rahmat dari Allah untuk semesta alam.

Skripsi ini disusun guna sebagai salah satu bentuk syarat kelulusan pada program sarjana (S1) prodi teknologi informasi, fakultas sains dan teknologi di UIN Walisongo Semarang. Selain itu, tujuan penyusunan Skripsi ini juga sebagai bentuk pengaplikasian ilmu yang telah saya dapatkan di prodi teknologi informasi UIN Walisongo Semarang. Juga sebagai pengetahuan bagi pembaca agar bisa bermanfaat dan dimanfaatkan untuk kepentingan yang baik.

Pada lembar kata pengantar ini juga, penulis ingin menyampaikan bahwa masih banyak kekurangan dalam penelitian maupun penyusunan naskah ini. Untuk itu, salam

hormat dan terima kasih kepada para pembimbing yang telah mengarahkan dan membantu dalam proses penyusunan Skripsi ini sehingga Skripsi ini dapat tersusun dengan baik. Tidak lupa juga ucapan terima kasih yang tidak terhingga kepada:

1. Kedua orang tua tercinta dan segenap keluarga yang senantiasa *support* dan selalu mendoakan serta memberi dukungan.
2. Guru-guru mulia yang dengannya menambah motivasi dan semangat dalam melakukan kebaikan.
3. Prof. Dr. Nizar, M.Ag, selaku Plt. Rektor UIN Walisongo Semarang.
4. Bapak Dr. H. Ismail, M.Ag, selaku Dekan Fakultas Sains dan Teknologi UIN Walisongo Semarang.
5. Bapak Nur Cahyo Hendro Wibowo, S.T, M.Kom, selaku ketua program studi Teknologi Informasi UIN Walisongo Semarang dan selaku dosen pembimbing skripsi saya yang senantiasa memberi pengarahan dan motivasi dalam penyusunan skripsi ini, sehingga skripsi ini dapat diselesaikan dengan baik.
6. Bapak Adzhal Arwani Mahfudh, M.Kom, selaku dosen pembimbing skripsi saya yang senantiasa memberi pengarahan dan motivasi dalam penyusunan skripsi

ini sehingga skripsi ini dapat diselesaikan dengan baik.

7. Seluruh dosen program studi Teknologi Informasi khususnya, serta dosen dan pegawai di lingkungan UIN Walisongo Semarang.
8. Teman-teman yang selalu memberi dukungan dan motivasi, baik dari teman-teman teknologi informasi maupun teman saya di luar prodi dan kampus.
9. Semua pihak yang penulis tidak bisa sebutkan satu persatu yang membantu dalam tenaga, pikiran maupun doa.

Dalam penelitian dan penyusunan skripsi ini tentu banyak kekurangan dan tidak sempurna. Untuk itu, kritik dan saran yang membangun sangat diperlukan penulis untuk dapat memperbaiki kekurangan yang ada, dan semoga skripsi ini dapat bermanfaat.

Semarang, 19 November 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

A. Latar Belakang

Narkoba merupakan salah satu zat adiktif yang dapat menyebabkan kecanduan bagi penggunanya. Narkoba yang merupakan narkotika, psikotropika dan zat adiktif yang dapat menimbulkan sensasi imajinasi atau halusinasi, menjadikan stimulant dan dapat mengatasi depresi bagi penggunanya (Darwis et al., 2018). Namun, narkoba ini merupakan obat-obatan yang dilarang untuk dikonsumsi dari banyak segi, baik segi agama, kesehatan, dan negara pun tidak dilegalkan dalam penyalahgunaan konsumsi narkoba.

Dalam Islam, narkoba dilarang untuk dikonsumsi, karena walaupun narkoba tidak dalam bentuk minuman keras atau zat cair, namun narkoba memiliki efek memabukkan dan menghilangkan kesadaran. Karena dijelaskan dalam Hadist Riwayat Muslim berbunyi dalil:

عَنْ عُمَرَ بْنِ الخَطّابِ رَضِيَ اللهُ عَنْهُمَا : اَنَّ النَّبِيَّ صَلَّ اللهُ عَلَيْهِ وَسَلَّمَ قَالَ : << كُلُّ مُسْكِرٍ خَمْرٌوَكُلُّ مُسْكِرٍ حَرَامٌ >> رواه مسلم

> Artinya: *Dari 'Umar bin Khattab radhiyallahu 'anhuma: dari Nabi shallallahu 'alaihi wasallam bersabda: "Setiap yang memabukkan itu khamr, setiap yang memabukan itu haram"* (H.R. Muslim).

Khamr memang memiliki beberapa manfaat, namun ke-*madharat*-an atau kerusakannya lebih besar dari pada manfaatnya. Para pecandu menjadikan narkoba sebagai obat penenang diri dari permasalahannya, sekilas memang menenangkan, namun kerusakan yang didapatkannya lebih besar dari kemanfaatannya. Kerusakan akal, kerusakan tubuh, bahkan sampai bisa merenggut nyawa pengguna. Oleh sebab itu lah Islam mengharamkan atau tidak membolehkan umatnya untuk mengkonsumsi *khamr* salah satunya narkoba. Allah SWT berfirman dalam Q.S. Al-Baqarah (2) ayat 219 yang berbunyi:

يَسْئَلُوْنَكَ عَنِ الخَمْرِ وَالمَيْسِرِقُلْ فِيْهِمَا إِثْمٌ كَبِيْرٌ وَمَنَافِعُ لِلنَّاسِ وَإِثْمُهُمَا أَكْبَرُ مِمْ نَّفْعِهِمَا. وَيَسْأَلُوْنَكَ مَاذَا يُنْفِقُوْنَ قُلِ العَفْوَ. كَذَالِكَ يُبَيِّنُ اللهُ لَكُمُ الأَيتِ لَعَلَّكُمْ تَتَفَكَّرُوْنَ

> Artinya: *Mereka menanyakan kepadamu (Muhammad) tentang khamr dan judi. Katakanlah "Pada keduanya terdapat dosa yang besar dan beberapa manfaat bagi manusia, tetapi dosanya lebih besar dari pada manfaatnya." Dan mereka menanyakan kepadamu (tentang) apa yang harus mereka infakkan. Katakanlah "Kelebihan (dari apa yang diperlukan." Demikianlah Allah menerangkan ayat-ayat Nya kepadamu agar kamu memikirkannya"* (Q.S. Al-Baqarah (2) ayat 219)

Dengan adanya dalil tersebut, tentu sudah jelas bahwasanya narkoba bukanlah sesuatu yang dihalalkan untuk dikonsumsi. Islam melarang sesuatu tentu dengan tujuan yang baik. Di dalam sesuatu yang dilarang, tentu terdapat kerugian atau kerusakan di dalamnya. Kesadaran dimulai dari individu masing-masing, penegak hukum dan lembaga penyuluhan seperti BNN hanya bisa mensosialisasikan bahayanya narkoba dan menindak bagi penggunanya.

Kasus peredaran dan penyalahgunaan narkoba di Indonesia semakin hari semakin mengkhawatirkan. Semua lapisan elemen masyarakat tua dan muda telah terjerumus ke dalam zat adiktif illegal ini (Lukman et al., 2022). Kepala Badan Narkotika Nasional (BNN), Komjen Pol. Prof. Dr. Petrus R. Golose, M.M., menyatakan, tingkat preferensi penggunaan narkotika di Indonesia per-september 2023 adalah 1,95%, jika dipopulasikan mencapai 3,66 juta jiwa (MerdekaDotCom, 2023). Jutaan jiwa pengguna narkoba bukanlah angka yang sedikit. Perlunya penyelidik menyelidiki dan menangkap dalang dari pada penyeludupan narkoba di Indonesia ini.

Kehidupan yang modern seperti saat ini membuat peredaran narkoba dilakukan dengan cara yang canggih juga. Para sindikat narkoba tidak lagi mengimport dari luar, namun mereka memilih membuat pabrik sendiri. Pengadaan bahan, peracikan, pemilihan tempat, pengaturan komunikasi menggunakan aplikasi yang privat, dan tugas-tugas lain mereka rencanakan dengan baik. Pemertintah Indonesia hendaknya

memperkuat keamanan dan perhatian terhadap kasus narkoba ini. Pasalnya, Indonesia sebagai negara berkembang yang awal mulanya menjadi transit para pengedar narkoba, kini menjadi tujuan operasi oleh jaringan internasional narkoba (Hariyanto, 2018).

Akhir-akhir ini, Indonesia menemukan fakta baru mengenai narkoba. Telah diketahui gembong narkoba terbesar di Indonesia bernama Fredy Pratama. Pria asal Banjarmasin, Kalimantan Selatan ini telah melakukan aksinya sejak 2014. Kasus Fredy Pratama ini telah menjadi kasus internasional, pasalnya ia telah menyeludupkan narkoba ke beberapa negara seperti Thailand, Malaysia, Laos, Myanmar, dan Indonesia. Kini ia menjadi buronan dari polisi negara-negara tersebut (Sukabumi, 2023). Dari kasus tersebut, polisi telah mendapatkan laporan sebanyak 408 laporan polisi dan menemukan barang bukti berupa 10,2-ton sabu (2020-2023). Terdapat 884 orang ditetapkan sebagai tersangka namun baru 39 orang yang sudah tertangkap per Mei-September 2023.

Segala upaya dilakukan oleh gembong narkoba Fredy Pratama untuk melangsungkan aksi haramnya tersebut. Mulai dari mengganti identitasnya, seperti mengubah namanya menjadi inisial Miming, The Secret, Cassanova, Air Bag dan Mojopahit, hingga dikabarkan ia akan melakukan operasi plastik di Thailand agar identitasnya tidak lagi diketahui. Ia berhasil menyudupkan 100-599 kg narkoba per bulannya dengan modus menyamarkannya dalam kemasan teh. Dari hasil peredaran narkoba tersebut, ia mendapatkan perputaran uang sebanyak 51 trilliun rupiah (2020-2023). Polisi telah melakukan penyitaan terhadap asset yang dimiliki Fredy Pratama dengan nilai sebesar 10,5 trilliun rupiah (Sukabumi, 2023).

Penemuan narkoba dengan jumlah yang besar tentu menjadi pertanyaan masyarakat, kenapa bisa lolos peredaran sebesar itu. Uang merupakan salah satu penyebab orang gelap mata, termasuk para oknum penegak hukum yang rela mengorbankan kehormatan dan martabatnya dengan terlibat dalam kasus peredaran narkoba.

Para penegak hukum tentu harus menangkap para bandar narkoba, tidak hanya pengguna atau pecandunya. Karena jika bandar tidak ditangkap, maka proses penyeludupan narkoba ke Indonesia tidak akan berhenti.

Dalam penelitian ini, kami meneliti mengenai analisis sentimen komentar pada salah satu video youtube yang mengabarkan tentang penyeludupan narkoba yang terjadi di Indonesia baru-baru ini, yakni kasus Fredy Pratama yang merupakan gembong terbesar narkoba di Indonesia senilai 10,5 trilliun rupiah. Beragam asumsi masyarakat yang ditulis di kolom komentar. Dari komenar ini, kita ingin melihat dan mengukur bagaimana pendapat masyarakat Indonesia mengenai penyalahgunaan narkoba melalui respon komentar youtube.

Analisis dilakukan dengan menggunakan dua metode yakni *naïve bayes classifier* dengan *random forest classifier*, untuk membandingkan metode mana yang tingkat akurasi nya lebih tinggi. Peneliti mengambil metode *naïve bayes classifier* dengan *random forest classifier,* dikarenakan *naïve*

bayes merupakan metode klasifikasi yang sederhana dan popular banyak digunakan namun tetap memiliki nilai akurasi dan kecepatan yang tinggi dalam menghitung database yang besar (Yasar & Saritas, 2019), sedangkan *random forest* merupakan metode klasifikasi yang menjadi salah satu algoritma pembelajaran mesin terbaik yang digunakan di berbagai bidang yang bisa mengatasi *overfitting* (Apriliah et al., 2021). Selain itu, kelebihan lain dari metode *random forest* adalah memiliki hasil akurasi yang baik, relatif kuat terhadap *outliters* dan *nosie,* lebih cepat dibanding *bagging* dan *boosting,* serta sifatnya yang sederhana dan mudah diparalelkan (Fitri, 2020).

Algoritma *naïve bayes* dan *random forest* juga memiliki performasi yang berbeda yang terletak pada nilai akurasi. *Naïve bayes* memiliki performasi yang baik pada kasus klasifikasi jenis *text.* Sedangkan algoritma *random forest* memiliki performasi yang baik pada kasus klasifikasi kualitas kredit (Leonardo & Pratama, 2020). Oleh karena itu, perlu dilakukan perbandingan pada kedua algoritma untuk menentukan algoritma

yang tepat dalam menghitung sentimen opini masyarakat mengenai kasus narkoba di Indonesia pada komentar youtube.

B. Identifikasi Masalah

Dari uraian yang dijabarkan dalam latar belakang, maka dapat diidentifikasi beberapa masalah, diantaranya:

1. Penyalahgunaan narkoba di Indonesia semakin mengkhawatirkan, dan peredaran narkoba kini semakin canggih dan tidak dapat dengan mudah terdeteksi.
2. Penemuan kasus gembong narkoba terbesar di Indonesia dan jaringan terbesar yang melibatkan 4 negara lain yang diketuai oleh Fredy Pratama (Indonesia) dan melibatkan 884 orang dengan laporan sebanyak 408 laporan.
3. Perbandingan metode klasifikasi *naïve bayes* dan *random forest* dalam analisis sentimen kasus narkoba di Indonesia.

C. Batasan Masalah

Analisis sentimen yang dilakukan oleh peneliti ini memiliki batasan masalah dalam penelitian, diantaranya:

1. Analisis sentimen ditujukan kepada kasus narkoba Fredy Pratama yang diambil dari komentar satu video youtube (13 September 2023-10 Oktober 2023).
2. Peneliti menggunakan 2 metode klasifikasi dalam penelitian, yakni *naïve bayes classifier* dan *random forest classifier.*
3. Bahasa pemrograman yang digunakan adalah *python*.

D. Rumusan Masalah

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

1. Bagaimana analisis sentimen mengenai respon warga Indonesia tentang penyalahgunaan narkoba di Indonesia?
2. Bagaimana hasil perbandingkan akurasi dari metode *naïve bayes classifier* dengan

random forest classifier dalam analisis sentiment?

E. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditulis di atas, maka tujuan penelitian ini adalah:

1. Mengetahui hasil analisis sentimen tentang penyalahgunaan narkoba di Indonesia yang diambil dari komentar salah satu video youtube.
2. Mengetahui metode yang lebih akurat antara *naïve bayes classifier* atau *random forest classifier* dalam analisis sentimen.

F. Manfaat Penelitian

Adapun setelah mengetahui tujuan dari penelitian ini, manfaat yang akan didapat dalam penelitian ini adalah:

1. Manfaat Praktis: dengan mengetahui tingkat sentimen mengenai penyalahgunaan narkoba kita akan mengetahui bagaimana warga Indonesia memandang tentang penyalahgunaan narkoba, dan diharapkan akan adanya kebijakan dari pemerintah untuk memberantas narkoba di Indonesia dan

memperketat keamanan agar tidak ada lagi penyeludupan narkotika di Indonesia.

2. Manfaat Teoretis: dengan membandingkan kedua metode yakni *naïve bayes* dan *random forest,* yang dimana belum banyak yang membandingkan kedua metode ini akan memiliki sumbang penelitian dari segi teori. Dan dari survey yang peneliti lakukan belum ada peneliti yang mengangkat tema analisis sentimen mengenai narkoba di Indonesia.

BAB II
TINJAUAN PUSTAKA

A. *Natural Language Processing* (NLP)

Natural Language Processing (NLP) adalah cabang *Artifical Intellegent* (AI) yang berfokus pada pemrosesan bahasa alami. Bahasa alami merupakan bahasa yang banyak digunakan oleh manusia untuk berkomunikasi satu sama lain (Cucus et al., 2019). NLP digunakan untuk proses penerjemahan bahasa alami ke bahasa komputer untuk dapat diolah data tersebut menghasilkan suatu analisis tertentu. Pada proses NLP pengolah kata, data yang akan diolah dapat disaring dengan cepat dan akurat sehingga menghasilkan satuan kata yang baik (Furqan & Shidqi, 2023).

NLP merupakan salah satu cabang AI atau kecerdasan buatan, dimana komputer dirancang untuk berkomunikasi dengan manusia menggunakan bahasa alami, seperti bahasa Indonesia (Furqan & Shidqi, 2023). NLP memudahkan manusia berinteraksi dengan

komputer sehingga komputer mengerti bahasa alami dari menusia.

Penggunaan NLP dalam penelitian ini tentu digunakan untuk menganalisis mengenai objek penelitian yakni komentar youtube mengenai penyalahgunaan narkoba di Indonesia pada salah satu video youtube. Melalui proses NLP, nantinya data komentar akan diproses menjadi bahasa yang dipahami komputer.

B. Text Mining

Text mining adalah penambangan data yang memecahkan masalah yang berkaitan dengan kebutuhan informasi melalui penerapan teknik penambangan data, pembelajaran mesin, pemrosesan bahasa alami, penyajian informasi dan manajemen informasi. Penambangan teks mencakup pemrosesan awal dokumen seperti klasifikasi teks, ekstraksi informasi, dan ekstraksi kata. Metode ini digunakan untuk mengekstraksi informasi dari sumber data dengan mengidentifikasi dan memeriksa pola yang menarik (Engelhart & Moughamian, 1968).

Text mining adalah bidang baru yang berkembang yang berupaya mengekstraksi informasi yang bermakna dari teks bahasa alami. Ini dapat secara kasar dicirikan sebagai proses menganalisis teks untuk mengekstraksi informasi yang berguna untuk tujuan tertentu. Teks memiliki fungsi untuk menyampaikan informasi atau opini faktual, dan insentif untuk secara otomatis mengekstraksi informasi dari teks (Witten, 2004).

Text mining secara umum merupakan teknik atau konsep yang digunakan untuk menganalisis teks. Dalam *text mining* terdapat beberapa hal yang mendukung seperti pemrosesan bahasa alami (NLP), tokenisasi, *text cleaning*, *stemming* dan *lemmatisasi,* representasi *vector* hingga menjadikannya sebagai analisis sentimen.

C. Analisis Sentimen

Analisis sentimen atau penambangan pendapat adalah bidang luas pemrosesan bahasa alami, linguistik komputasi, dan penambangan teks, yang tujuannya adalah untuk menemukan pendapat, perasaan, evaluasi, sikap, dan perasaan pembicara atau penulis tentang suatu topik,

produk, layanan atau organisasi, individu, atau aktivitas lainnya (Dale, 2010). Analisis sentimen juga dapat mengungkapkan kesedihan emosional, kegembiraan atau kemarahan. Kita dapat mencari pendapat tentang produk, merek, atau orang dan melihat apakah mereka memiliki ulasan positif atau negatif yang diambil datanya secara online (Sarjana et al., 2017).

Analisis sentimen merupakan salah satu NLP, yakni suatu proses agar komputer bisa berinteraksi dengan bahasa manusia dengan beberapa proses di dalamnya. Dalam analisis sentimen masih sangat memungkinkan untuk komputer tidak bisa membedakan antara ulasan positif dan negatif jika kita mengambil data dari sosial media, karena biasanya bahasa yang digunakan dalam sosial media sangat ambigu untuk dimengerti komputer. Sehingga memungkinkan untuk beberapa ratus data yang akan terambil mungkin hanya puluhan data. Namun, cara lain bisa dapat meningkatkan akurasi analisis seperti pelabelan ulasan secara manual

dan dihitung analisisnya dengan menggunakan *machine learning*.

D. YouTube API

Application Programming Interface (API) sebagai fungsi, kumpulan perintah dan juga protocol yang berguna untuk membantu programmer dalam membangun perangkat lunak dalam sistem operasi tertentu. Khususnya dalam youtube memungkinkan pengembang mengakses statistik video dan data saluran youtube melalui dua jenis panggilan, *Representational State Transfer* (REST), dan *Extensible Markup Language-Remote Procedure Call* (XML-RPC). Google sendiri mendeskripsikan sumber daya youtube API sebagai API dan alat yang memungkinkan Anda membawa pengalaman youtube ke laman web, aplikasi, atau perangkat Anda (Yoga Saputra et al., 2019).

YouTube API adalah antarmuka yang disediakan oleh yotube untuk mengakses dan mengintegrasikan fungsionalitas youtube ke dalam aplikasi atau layanan pihak ketiga. Dalam hal ini YouTube API digunakan untuk mengambil

komentar dari youtube atau disebut proses crawling atau pengambilan data yang akan diproses data nya untuk dianalisis (Yoga Saputra et al., 2019).

E. Python

Python merupakan salah satu bahasa pemrograman yang dapat digunakan untuk segala jenis analisis data. Di dalam python terdapat banyak *library* yang membantu *text preprocessing seperti text folding, tokenezing, convert emoticon, stemming* dan lain-lain sebagaimana dalam analisis sentimen diperlukan adanya pemrosesan teks (Syah & Witanti, 2022).

Python juga dapat membantu dalam klasifikasi dalam berbagai algoritma, termasuk naïve bayes dan juga random forest, yang mana kedua metode tersebut merupakan algoritma yang digunakan dalam penelitian ini. *Library* yang digunakan dalam menjunjang penelitian ini diantaranya berupa *library* bawaan dari instalasi awal python seperti pandas, numpy, matplotlib, sns dan sebagainya serta beberapa *library* tambahan yang harus di install seperti, *Google API*

Client Library for Python, scikit-learn (sklearn), Natural Language Toolkit (NLTK), *Journey to the Center of Python Machine Learning (jcopml), Sastrawi.*

Google API Client Library for Python merupakan library yang digunakan dalam python agar dapat mengakses data yang ada di google. *Scikit-learn* merupakan library yang mencakup banyak algoritma untuk pembelajaran klasifikasi ataupun regresi, serta memiliki banyak algoritma yang menujang dalam pemrosesan data (Gridin, 2022). NLTK merupakan *library* yang digunakan dalam komputasi linguistik. NLTK mencakup pemrosesan bahasa alami berupa statistik, simbolik dan hubungan bertautan dengan copra (Rifano et al., 2020). *Library jcopml* digunakan untuk pendekatan *workflow* yang terstruktur. *Library* sastrawi mencakup pemrosesan bahasa alami yang berhubungan dengan bahasa yang digunakan untuk *stemmer* yang digunakan untuk mengatasi permasalahan perubahan kata berimbuhan menjadi kata dasar (Rosid et al., 2020).

F. Crawling

Web crawling atau *web scrapping* adalah ekstraksi otomatis data dari situs web menggunakan perangkat lunak. Ini adalah proses yang sangat penting dalam bidang-bidang seperti intelijen bisnis saat ini. Pengikisan web adalah teknik yang memungkinkan kita mengekstrak data terstruktur dari teks seperti *Hypertext Markup Language* (HTML). Pengikisan web sangat berguna dalam situasi di mana data tidak disediakan dalam format yang dapat dibaca mesin seperti *Javascript Object Notation* (JSON) atau *Extensible Markup Language* (XML). Dengan mengumpulkan data melalui pengikisan web, harga dapat diperoleh dari situs web ritel hampir secara waktu nyata dan informasi tambahan dapat diberikan (Khder, 2021).

Crawling data adalah proses pengumpulan data secara otomatis dari berbagai sumber di web menggunakan perangkat lunak yang disebut *web crawler* atau *spider*. *Web crawler* bertugas untuk mengunjungi berbagai halaman web, mengidentifikasi tautan dan konten, dan

mengambil informasi yang diinginkan untuk diolah lebih lanjut (Khder, 2021).

G. Text Preprocessing

Text preprocessing merupakan level pertama dari *text processing*, yang berfungsi untuk mengubah format dokumen sesuai kebutuhannya menjadi data terstruktur agar dapat diproses lebih lanjut dalam proses *text mining* (Sarjana et al., 2017). Tahap *text preprocessing* klasifikasi bertujuan untuk meningkatkan akurasi klasifikasi data. *Preprocessing* dalam *text mining* cukup rumit, karena aturan penulisan kalimat dan pembentukan afiks dalam bahasa Indonesia berbeda. Adapun beberapa kata imbuhan yang dapat merubah makna adalah sebagai berikut:

1. Prefiks: Prefiks adalah morfem yang ditambahkan di awal sebuah kata untuk mengubah makna atau jenis kata tersebut. Prefiks biasanya terdiri dari satu atau lebih huruf atau fonem. Contoh prefiks dalam bahasa Indonesia adalah "di-" dalam kata "diambil" atau "ber-" dalam kata "berlari". Prefiks dapat mengubah verba menjadi

nomina, adjektiva menjadi verba, dan sebagainya.

2. Sufiks: Sufiks adalah morfem yang ditambahkan di akhir sebuah kata untuk mengubah makna atau jenis kata tersebut. Sufiks juga terdiri dari satu atau lebih huruf atau fonem. Contoh sufiks dalam bahasa Indonesia adalah "-kan" dalam kata "membaca" atau "-an" dalam kata "penulisan". Sufiks dapat mengubah verba menjadi verba kausatif, nomina menjadi adjektiva, dan sebagainya.
3. Infiks: Infiks adalah morfem yang dimasukkan di tengah-tengah sebuah kata untuk mengubah makna atau jenis kata tersebut. Infiks umumnya terdiri dari satu atau lebih huruf atau fonem. Infiks biasanya digunakan dalam bahasa-bahasa tertentu, seperti bahasa Tagalog atau bahasa Jawa. Contoh infiks dalam bahasa Indonesia adalah "-el-" dalam kata "kemilau" atau "-me-" dalam kata "gemetar". Infiks dapat mengubah nomina

menjadi verba, verba menjadi verba kausatif, dan sebagainya.

4. Konfiks: Konfiks adalah kombinasi dari prefiks dan sufiks yang digunakan bersama-sama untuk membentuk sebuah kata. Prefiks dan sufiks dalam konfiks tidak dapat digunakan secara terpisah. Contoh konfiks dalam bahasa Indonesia adalah "peng-" dan "-an" dalam kata "pengarahan". Konfiks dapat mengubah nomina menjadi verba atau adjektiva, atau mengubah verba menjadi nomina.

Text preprocessing adalah proses menghilangkan *noise* dan *unwanted attribute*. Langkah awalnya adalah menghapus teks yang tidak relevan, kemudian mengisi bagian yang kosong (*missing value*) dan menghapus kolom yang tidak berhubungan (Ratniasih et al., 2023). Dalam *text preprocessing* terdapat beberapa tahapan seperti:

1. *Remove Duplicate*

   Remove duplicate merupakan proses menghilangkan atau menghapus data yang sama dari kata-katanya, atau data yang terambil secara terulang pada saat proses pengambilan data (*crawling data*) (Adhi Putra, 2021). Data yang ter-*terduplicate* nantinya akan dihapus dan hanya tersisa satu jenis data.

2. *Case Foding*

   Case folding, adalah proses mengubah semua karakter dalam teks menjadi huruf kecil atau huruf besar, tergantung pada aturan yang ditentukan (Akbar et al., 2021). *Case folding* digunakan untuk menghapus perbedaan kasus (besar kecil huruf) dalam teks agar dapat dilakukan pemrosesan teks yang konsisten dan seragam.

3. *Cleansing*

   Cleansing adalah tahapan pembersihan teks, seperti menghapus tanda baca, menghapus enter, mengubah emoji

menjadi *string*, menghapus URL dan sebagainya.

4. *Normalization*

   Normalization adalah proses penormalan kata, dimana tidak setiap kata dapat terdeteksi oleh program maka akan dilakukan normalisasi. Seperti kata yang tidak baku, kata asing, *typo* dan proses penormalan lainnya.

5. *Stopword Removal*

   Stopwords adalah kata-kata umum yang sering muncul dalam teks namun cenderung tidak memiliki nilai informasi yang tinggi dalam analisis teks atau pemrosesan bahasa alami (Purwati, 2023). Contohnya adalah kata-kata seperti "di", "dan", "akan","untuk" dan lain sebagainya.

   Daftar stopwords umum biasanya disediakan oleh *library* atau *framework* pemrosesan teks dalam berbagai bahasa pemrograman, seperti *Python NLTK* (*Natural Language Toolkit*) atau library *scikit-learn*. Namun, daftar stopwords

dapat disesuaikan atau diperluas sesuai dengan kebutuhan dan konteks analisis teks yang spesifik.

6. *Stemming*

   Stemming adalah salah satu proses proses dalam pemrosesan teks yang dilakukan untuk mengubah kata-kata menjadi bentuk dasar atau kata dasar (*root word*) dengan cara menghilangkan akhiran atau awalan tertentu (Amin & Alfa Razaq, 2018). Proses *stemming* tidak mempertimbangkan konteks atau makna kata, tetapi hanya berfokus pada pola struktural kata. Algoritma stemming yang umum digunakan adalah algoritma *Porter stemming* atau algoritma *Snowball stemming*. Algoritma-algoritma ini menggunakan aturan-aturan berbasis aturan morfologi bahasa untuk melakukan pemangkasan akhiran atau awalan kata.

   Contoh penerapan *stemming* adalah mengubah kata-kata seperti "makanan", "makananmu", "makanannya" menjadi

bentuk dasar "makan". Dengan demikian, semua kata-kata tersebut dapat dianggap sama dan diperlakukan sebagai satu entitas dalam analisis teks atau pemrosesan bahasa alami.

7. *Tokenizing*

   Toekenizing adalah proses merubah teks menjadi token-token atau unit terkecil dari teks. Token merupakan unit dasar dalam pemrosesan teks, seperti kata, frasa, simbol, atau karakter tertentu.

   Tujuan utama dari tokenisasi adalah untuk mempersiapkan teks mentah agar dapat diolah lebih lanjut dalam analisis teks atau pemrosesan bahasa alami. Dengan memecah teks menjadi token-token yang lebih kecil, kita dapat mengidentifikasi dan mengisolasi unit-unit penting dalam teks, seperti kata-kata atau frasa, yang akan menjadi dasar dalam analisis lebih lanjut.

H. Split Validation Data

Split Validation Data adalah membagi data menjadi data *training* atau data latih dan data *testing* atau data uji dengan menggunakan *split validation*. *Split validation* dilakukan dengan jumlah data *testing* diambil 20% dari data *training.* Pengambilan data dilakukan secara random dengan bantuan *library python* (Turmudi Zy et al., 2021). *Split Validation* dilakukan dengan membagi data *training* dan data *testing* dengan perbandingan 80:20. Pembagian data dengan perbandingan tersebut merupakan strategi yang paling sederhana dan paling umum digunakan (Joseph & Vakayil, 2022).

I. TF-IDF (*Term Frequency-Inverse Document Frequency*)

TF-IDF merupakan gabungan dari 2 proses yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). TF-IDF digunakan untuk merubah teks menjadi *vector* namun tetap memperhatikan apakah kata tersebut cukup informatif atau tidak. Kata yang sering muncul memiliki nilai besar dan yang jarang muncul memiliki nilai yang kecil. Kata

yang sering muncul disebut juga *stopwords*, yang artinya kata tersebut dianggap kurang penting oleh model (Fatmawati, 2017).

TF adalah menghitung banyaknya kemunculan kata (t) pada sebuah dokumen (d). Disebabkan panjang kata setiap dokumen berbeda-beda, maka nilai TF dibagi dengan panjang dokumen (jumlah seluruh kata pada dokumen).

$$tf_{t,d} = \frac{n_{t,d}}{total\ number\ of\ terms\ in\ document} \qquad (2.1)$$

Dimana:

tf = kata/dokumen

n = frekuensi kemunculan pada kata/doc

total = jumlah seluruh kata di setiap dokumen

Jika TF yang menghitung banyaknya kata yang sering muncul, IDF akan menganggap kata yang sering muncul adalah kata yang kurang penting dan kata yang jarang muncul adalah kata yang penting. Rumusnya:

$$idf_d = \log\left(\frac{Number\ of\ document}{Number\ of\ document\ with\ term\ t'}\right) \qquad (2.2)$$

Kemudian hitung masing-masing TF-IDF di setiap korpus.

$$\left(tf_{idf}\right)_{t,d} = tf_{t,d} * idf_t \quad (2.3)$$

J. *Naïve Bayes Classifier*

Naïve Bayes Classifier (NBC) merupakan salah satu metode klasifikasi yang sangat popular. naïve bayes merupakan metode klasifikasi yang sederhana namun memiliki nilai akuransi yang tinggi. Algoritma NBC merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Engelhart & Moughamian, 1968).

Naive bayes adalah pengklasifikasi probabilitas sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari kumpulan data yang diberikan. Algoritma menggunakan teorema Bayes dan mengasumsikan bahwa semua atribut independen atau tidak bergantung satu sama lain berdasarkan nilai variabel kelas (Yasar & Saritas,

2019). Definisi lain *naïve bayes* adalah klasifikasi yang menggunakan probabilitas dan metode statistik yang diusulkan oleh ilmuwan Inggris Thomas Bayes, yang memprediksi kemungkinan masa depan berdasarkan pengalaman masa lalu (Bustami, 2014). Adapun persamaan *naïve bayes* adalah:

$$P(H|X) = \frac{P(X|H)*P(H)}{P(X)} \quad (2.4)$$

Dimana:

X : Data dengan class yang belum diketahui

H : Hipotesis data X merupakan suatu class spesifik

P(H|X) : Probabilitas hipotesis H berdasar kondisi X (*posteriori probability*)

P(H) : Probabilitas hipotesis H (*prior probability*)

P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H (*likelihood probability*)

P(X) : Probabilitas X (*evidence probability*)

Ide landasan dari aturan Bayes adalah hasil dari hipotesis (H) dapat diperkirakan berdasarkan beberapa *evidence* (X) yang diamati. Hal yang ada perlu diperhatikan juga dalam Bayes adalah sebuah probabilitas awal atau P(H) adalah probabilitas dari hipotesis yang belum ada bukti yang diamati. Sebuah probabilitas *posterior* H atau P(H|X) adalah probabilitas dari hipotesis setelah ada bukti yang diamati (Yuniarti et al., 2020).

K. Random Forest Classifier

Random Forest Classifier (RFC) adalah metode klasifikasi yang terdiri dari kumpulan pohon keputusan yang nantinya digunakan sebagai voting untuk mendapatkan hasil akhir deteksi sarkasme berupa data pelatihan dan variabel acak independen dengan karakteristik yang berbeda. Pohon keputusan dibangun dengan menentukan simpul akar dan memilih beberapa simpul daun untuk mendapatkan hasil akhir. Pendekatan *random forest* yang diusulkan oleh Breiman adalah algoritma pembelajaran mesin dengan banyak pohon keputusan (Alita & Isnain, 2020). *Random forest* adalah kombinasi dari metode *bagging* dan

subruang acak. Metode ini telah membuktikan nilainya dalam masalah regresi dan klasifikasi dalam beberapa tahun terakhir dan merupakan salah satu algoritma pembelajaran mesin terbaik yang digunakan di berbagai bidang (Apriliah et al., 2021).

Random forest memiliki keunggulan dalam mengatasi *overfitting* (*overfitting bias*), tahan terhadap *noise* atau data yang tidak relevan, serta mampu menangani data yang memiliki banyak fitur. Selain itu, *random forest* juga memberikan kemampuan untuk mengukur pentingnya fitur-fitur dalam model (Apriliah et al., 2021).

Random forest merupakan implementasi *homogenius ensamble* atau pembelajaran *ensamble* dengan model yang sama dengan menerapkan pohon keputusan. Selain *bagging*, *random forest* juga menerapkan fitur acak dimana setiap *bagging* yang dihasilkan menerapkan fitur sampel pelatihan secara acak. Yang kami maksud adalah setiap model pohon keputusan akan dilatih dengan bagian yang berisi berbagai dataset (Awangga & Khonsa', 2022).

L. Ketepatan Klasifikasi

Pengukuran ketepatan klasifikasi penting untuk dilakukan untuk melihat apakah metode klasifikasi yang digunakan sudah tepat. Cara untuk melihat ketepatan klasifikasi adalah dengan menghitung nilai akurasi, *recall, precision,* dan *f1* score (Grandini et al., 2020). Nilai-nilai tersebut nantinya akan dituangkan ke dalam *matrix confused*.

Matrix confused atau matriks kebingungan adalah alat visualisasi yang biasa digunakan untuk *supervised learning.* Setiap kolom pada matris berisi kelas prediksi dan setiap baris berisi kelas kejadian sebenarnya atau *actual* (Mahfudh & Mustofa, 2019). Tabel *matrix confused* ada pada tebel 2.1

Tabel 2. 1 Tabel *Matrix Confused* 2x2

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

Dimana:

1. TP (*true positif*), adalah data yang diprediksi positif dan data sesungguhnya juga positif.
2. TN (*true negatif*), adalah data yang diprediksi negitif dan data sesungguhnya juga negatif.
3. FP (*false positif*), adalah data yang diprediksi positif dan data sesungguhnya negatif.
4. FN (*false negatif*), adalah data yang diprediksi negatif dan data sesungguhnya positif.

Adapun rumus untuk perhitungannya adalah sebagai berikut:

$$Akurasi = \frac{TN+TP}{TN+TP+FN+FP} \quad (2.5)$$

$$Precision = \frac{TP}{TP+FN} \quad (2.6)$$

$$Recall = \frac{TN}{TN+FP} \quad (2.7)$$

$$f1\ score = 2x\frac{Precision\ .\ Recall}{Precision+Recall}\ x\ 100\% \qquad (2.8)$$

Confusion matrix atau *matrix confused* adalah metode yang umum digunakan untuk menghitung akurasi, *recall, presicion*, dan tingkat kesalahan (*error rate*). Yang mana, akurasi berarti mengevaluasi kemampuan sistem untuk menemukan kecocokan terbaik dan ditentukan oleh persentase dokumen yang diambil dan benar-benar relevan dengan kueri. *Recall* berarti mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dalam sekumpulan dokumen dan didefinisikan sebagai persentase dokumen yang relevan dengan kueri. Akurasi adalah perbandingan kasus yang teridentifikasi dengan benar dengan jumlah total kasus, dan tingkat kesalahan adalah perbandingan kasus yang teridentifikasi secara salah dengan jumlah total kasus. Presisi adalah ukuran seberapa besar tingkat kebenaran antara informasi yang diminta dengan respon yang diberikan system (Arini et al., 2020). Adapun *f1 score* adalah parameter tunggal yang digunakan untuk

mengukur keberhasilan gabungan dari *recall* dan *precision* (Zidan, 2022).

M. Narkoba

Menurut Undang-Undang Nomor 22 Tahun 1997 tentang Narkotika, disebutkan bahwa narkotika adalah "zat atau obat yang berasal dari tanaman atau bukan tanamanbaik sintetis maupun semisintetis yang dapat menyebabkan penurunan atau perubahan kesadaran, hilangnya rasa, mengurangi sampai menghilangkan rasa nyeri, dan dapat menimbulkan ketergantungan".

Narkoba merupakan narkotika, psikotropika dan zat adiktif lainnya yang membuat orang lain merasakan kecanduan (Lukman et al., 2022). Narkoba adalah obat, bahan atau zat dan bukan tergolong makanan jika diminum, dihisap, dihirup, ditelan atau disuntikkan, yang dapat mempengaruhi fungsi otak (sistem saraf pusat) dan seringkali menimbulkan ketergantungan. Akibatnya kerja otak berubah (bertambah atau berkurang), begitu pula fungsi vital organ tubuh

lainnya (jantung, peredaran darah, pernafasan dan lain-lain) (Hariyanto, 2018).

Narkoba merupakan sekelompok senyawa yang mempunyai efek dan risiko menimbulkan kecanduan pada penggunanya. Tujuan penggunaan obat yang mengandung senyawa psikotropika ini, sebenarnya untuk menjadikan obat tersebut sebagai obat bius, yaitu untuk membius pasien pada saat pembedahan atau sebagai obat untuk mengobati penyakit tertentu. Namun penggunaan narkoba saat ini kurang memahami fungsinya karena adanya penyalahgunaan obat, misalnya obat ini digunakan oleh pasien bedah atau untuk penyakit tertentu, namun pengguna saat ini sering menggunakan obat tersebut untuk disalahgunakan (Perkembangan & Masyarakat, 2013).

Narkoba yang termasuk di dalamnya adalah narkotika, bekerja dengan mengikat reseptor di otak dan menghalangi rasa sakit. Sebab, obat ini ampuh meredakan rasa sakit dalam waktu singkat. Namun obat ini bisa menimbulkan efek adiktif atau kecanduan. Kecanduan menjadikan pengguna tidak dapat mengendalikan

penggunaan narkoba tersebut sehingga dikonsumsi secara terus-menerus. Kecanduan narkoba dapat menimbulkan keinginan yang kuat untuk terus menggunakan obat tersebut (Darwis et al., 2018).

Tertuang dalam Undang-Undang Narkoba Nomor 35 Tahun 2009, narkoba dibagi menjadi 3 jenis yakni narkotika, psikotropika dan zat adiktif lainnya.

1. Narkotika

   Menurut *World Health Organization* (WHO), narkotika adalah "zat psikoaktif yang digunakan tanpa resep dokter atau indikasi medis. Penggunaan narkotika dapat menyebabkan ketergantungan psikologis atau fisik dan dapat memberikan dampak negatif pada Kesehatan fisik, mental dan sosial". Penggunaan narkotika dapat menyebabkan khayalan, halusinasi, menghilangkan rasa sakit, menambah semangat, sehingga menyebabkan orang akan kecanduan dalam mengonsumsinya (Hariyanto, 2018).

Narkotika dibagi menjadi 3 golongan:

a. Golongan I, yakni narkotika yang paling berbahaya, golongan ini memiliki zat adaktif yang tinggi. Contohnya adalah kokain, ganja, heroin, morfin dan opium. Narkotika jenis ini biasa digunakan untuk penelitian ilmiah.
b. Golongan II, yakni narkotika yang memiliki zat adiktif yang kuat juga, namun tidak sekuat narkotika golongan I. Contohnya adalah petidin, benzetidin, dan betametadol. Narkotika jenis ini bisa digunakan untuk penelitian ilmiah dan pengobatan.
c. Golongan III, yakni narkotika yang memiliki zat adiktif yang ringan. Contohnya adalah kodein dan turunannya. Narkotika golongan ini juga bermanfaat untuk penelitian dan pengobatan.

2. Psikotropika

   Menurut BNN, psikotropika adalah "zat atau obat yang berasal dari tanaman atau bukan tanaman, baik sintetis maupun semisintetis, yang dapat mempengaruhi fungsi mental dan perilaku manusia. Psikotropika dapat menyebabkan penurunan atau perubahan kesadaran, hilangnya rasa, mengurangi atau menghilangkan nyeri, dan menyebabkan ketergantungan". Psikotropika dibagi menjadi beberapa golongan juga, diantaranya:

   a. Golongan I, merupakan psikotropika dengan pengaruh adiktif yang sangat kuat. Contohnya adalah MDMA, *Lysergic Acid Diethylamide* (LSD), STP, dan ekstasi. Kemanfaatan dari psikotropika golongan ini belum diketahui.
   b. Golongan II, merupakan psikotropika dengan pengaruh

adiktif yang kuat. Contohnya adalah amfetamin, metamfetamin, dan metakualon. Psikotropika golongan ini memiliki kemanfaatan di bidang penelitian dan pengobatan.

c. Golongan III, merupakan psikotropika dengan pengaruh adiktif yang sedang. Contohnya adalah lumibal, buprenosina, dan fleenitrazepam. Psikotropika golongan ini memiliki kemanfaatan di bidang penelitian dan pengobatan.
d. Golongan IV, merupakan psikotropika yang memiliki daya aktif ringan. Contohnya nitrazepam (BK, mogadon, dumolid) dan diazepam.

3. Zat adiktif lainnya

   Zat adiktif lainnya adalah zat adiktif yang dapat menimbulkan ketergantungan selain narkoba dan psikotropika. Diantaranya, rokok, beberapa jenis alcohol dan zat yang

menimbulkan bau menyengat yang membuat ketagihan seperti thiner, lem, bensin, cat dan zat lain yang memabukkan (Hariyanto, 2018).

N. Kajian Penelitian Terkait

Tabel 2. 2 Tabel Penelitian Terdahulu

No	Pustaka	Metode	Hasil
1	(R. Leonardo et al, 2022) "Perbandingan Metode *Random Forest* Dan *Naïve Bayes* Dalam Prediksi Keberhasilan Klien Telemarketing"	Random Forest dan Naïve Bayes	Akurasi random forest lebih tinggi 5% dari pada naïve bayes.
2	(R. Wasono et al, 2019) "Perbandingan Metode *Random Forest* dan *Naïve Bayes* untuk Klasifikasi Debitur Berdasarkan Kualitas Kredit"	Random Forest dan Naïve Bayes	Akurasi random forest yang memiliki tingkat akurasi 98,16% dan naïve bayes memiliki tingkat akurasi 95,93%.

3	(G.M. Momole et al, 2022) "Perbandingan *Naïve Bayes* Dan *Random Forest* Dalam Klasifikasi Bahasa Daerah"	Random Forest dan Naïve Bayes	Akurasi naïve bayes lebih baik dengan nilai akurasi sebesar 0,9922 dibandingkan dengan nilai akurasi random forest sebesar 0,6544.
4	(A. Miftahusalam et al, 2022) "Perbandingan Metode *Random Forest* dan *Naive Bayes* pada Analisis Sentimen Review Aplikasi BCA Mobile"	Random Forest dan Naïve Bayes	Metode random forest menghasilkan prediksi yang lebih baik daripada metode naïve bayes dengan dengan nilai akurasi sebesar 93,93%, nilai presisi 93,02%, nilai recall 89,89%, dan nilai F1-score 91,43%.

Topik penelitian mengenai analisis sentimen narkoba di Indonesia belum ada di penelitian sebelumnya. Beberapa jurnal membahas tentang penjelasan atau deskripsi mengenai penyalahgunaan narkoba di Indonesia. Namun, mengenai perbandingan metode naïve bayes dengan random forest, sudah ada beberapa jurnal penelitian yang membahasnya.

Seperti pada penelitian yang dilakukan oleh R. Leonardo dan teman-temannya pada penelitian yang berjudul "Perbandingan Metode Random Forest Dan Naïve Bayes Dalam Prediksi Keberhasilan Klien Telemarketing" tersebut menggunakan random forest dan naïve bayes sebagai metode penelitiannya. Adapun objek yang diteliti adalah telemarketing dengan klasifikasi berupa positif, negatif dan netral. Dengan hasil akhirnya disimpulkan bahwa algoritma random forest lebih cocok digunakan dalam hal memprediksi keputusan pelanggan. Hal ini terlihat ketika akurasi random forest yang diperoleh sebesar 90%, lebih tinggi 5% dibandingkan algoritma naïve bayes. Nilai AUC pada algoritma random forest sebesar 0.97, lebih tinggi 1.3 dibandingkan algoritma naïve bayes (Leonardo & Pratama, 2020).

Penelitian lain dilakukan oleh B. Wasono dan kawan-kawan dengan judul penelitian "Perbandingan Metode Random Forest dan Naïve Bayes untuk Klasifikasi Debitur Berdasarkan Kualitas Kredit". Adapun metodenya yaitu random

forest dan naïve bayes dengan objek dalam penelitiannya yakni salah satu jenis bank yang ada di Indonesia dengan klasifikasi positif dan negatif. Adapun nilai akhirnya yakni berupa akurasi random forest yang memiliki tingkat akurasi 98,16% dan naïve bayes memiliki tingkat akurasi 95,93%. Sehingga pada penelitian ini, metode random forest memiliki nilai akurasi yang lebih tinggi dari naïve bayes (Wasono, 2022).

Penelitian selanjutnya yakni penelitian yang dilakukan oleh G.M. Momole dan teman-teman dengan judul penelitian "Perbandingan Naïve Bayes Dan Random Forest Dalam Klasifikasi Bahasa Daerah". Adapun objek dari penelitian ini adalah bahasa daerah dengan 3 klasifikasi yakni Halma, Kali dan Tora dengan metode random forest dan naïve bayes. Hasil akhir dari penelitian ini, metode naïve bayes dalam pengenalan bahasa sangat baik karena menghasilkan nilai akurasi diatas 0,90 dibandingkan random forest yang hanya mencapai nilai akurasi dibawah 0,70. Dalam perhitungan matriks konfusi, metode naïve bayes lebih baik dengan nilai akurasi sebesar 0,9922

dibandingkan dengan nilai akurasi random forest sebesar 0,6544 (Momole, 2022).

Penelitian lain dilakukan oleh A. Miftahussalam dan teman-temannya dengan penelitian yang berjudul "Perbandingan Metode *Random Forest* dan *Naive Bayes* pada Analisis Sentimen Review Aplikasi BCA Mobile" dengan metode random forest dan naïve bayes dan objek penelitian yakni Aplikasi BCA Mobile. Klasifikasi dibagi menjadi 3 kelas yakni positif, negatif dan netral. Hasilnya, metode random forest menghasilkan prediksi yang lebih daripada metode naïve bayes dengan dengan nilai akurasi sebesar 93,93%, nilai presisi 93,02%, nilai recall 89,89%, dan nilai F1-score 91,43% (Miftahusalam et al., 2023).

BAB III

METODOLOGI PENELITIAN

A. Sumber Data

Penelitian ini merupakan penelitian kuantitatif yang nantinya hasil akhirnya berupa nilai akurasi dari analisis sentimen dengan 2 metode yakni naïve bayes *classifier* dan random forest *classifier*. Data yang digunakan dalam penelitian ini adalah data primer dan data sekunder. Data primer diambil dari komentar dari salah satu video youtube dari akun youtube tvOneNews yang diupload pada tanggal 13 September 2023, dengan video berjudul "Polisi Ungkap Jaringan Narkoba Internasional Terbesar Senilai 10 Triliun Rupiah" yang data diambil per 10 Oktober 2023, komentar youtube memuat sebanyak 3000 komentar, namun data hanya dapat di-*crawling* sebanyak 2.702 data. Sedangkan data sekunder diambil dari literasi jurnal sebagai penunjang penelitian dan juga referensi informasi dari beberapa *report* atau berita dari youtube.

B. Kebutuhan Perangkat Penelitian

Dalam penelitian ini diperlukan perangkat yang mendukung untuk kebutuhan penilitian, diantaranya perangkat keras dan perangkat lunak yang digunakan peneliti dengan spesifikasi seperti berikut:

1. Kebutuhan Perangkat Keras

Tabel 3. 1 Tabel Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Device	Lenovo 81WO
2	Processor	AMD Athlon Silver 3050U with Radeon Graphic
3	Memori (RAM)	8 GB
4	Monitor	14”
5	Keyboard dan Mouse	Normal

2. Kebutuhan Perangkat Lunak

Tabel 3. 2 Tabel Kebutuhan Perangkat Lunak

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	Windows 11 64-bit
2	Bahasa Pemrograman	Python
3	Ms. Office	Ms. Word, Ms. Exel 2019
4	Anaconda Navigator	Anaconda3 2022.10
5	Browser	Chrome

C. Langkah Analisis Data

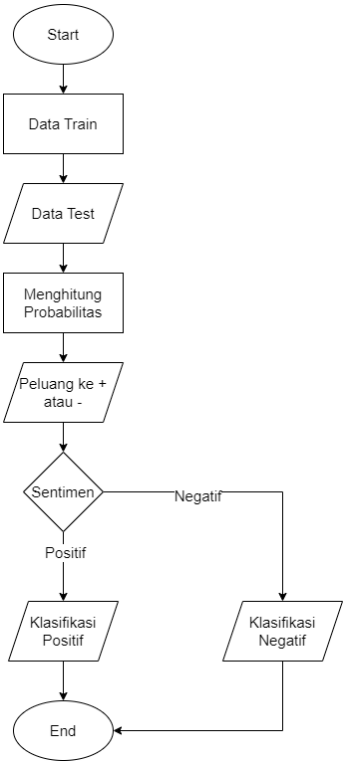
Tahapan dalam analisis data untuk penelitian ini adalah sebagai berikut:

1. *Crawling* data dari YouTube API
    a. Mengambil API key dari YouTube API
    b. Melakukan *crawling* data dari YouTube API menggunakan *python*.
    c. Menyimpan data dalam bentuk csv
2. Menyiapkan data komentar dari youtube
    a. Membaca data csv dalam python
    b. Melakukan pelabelan pada data komentar dengan nilai positif dan negatif
    c. Mencari data *duplicate* dan menghapusnya jika ada
    d. Mencari *missing value* dan menghapusnya jika ada
3. *Text Processing*
    a. Menghapus URL
    b. Menghapus tanda baca (*punctual removal*)
    c. Melakukan *case folding* yaitu merubah semua huruf menjadi huruf kecil

d. Melakukan *normalization*, yakni melakukan normalisasi pada kata-kata agar dapat dimengerti oleh komputer
e. Membuang *stopwords*, atau kata yang memiliki rasa tetapi tidak memiliki arti
f. Melakukan *stemming*, yakni merubah kata menjadi kata dasar
g. Melakukan *tokenizing*, merubah teks menjadi token-token
h. Membagi data *training* dan data *testing*
i. Melakukan TF-IDF pada data

4. Klasifikasi
   a. Naïve Bayes Classifier

      Melakukan klasifikasi naïve bayes dengan bantuan *library* python dan beberapa fungsi yang terdapat dalam modul python yang menunjang dalam perhitungan klasifikasi naïve bayes. Adapun alur klasifikasi naïve bayes digambarkan dalam Gambar 3.1.



Gambar 3. 1 Flowchart Klasifikasi Naïve Bayes

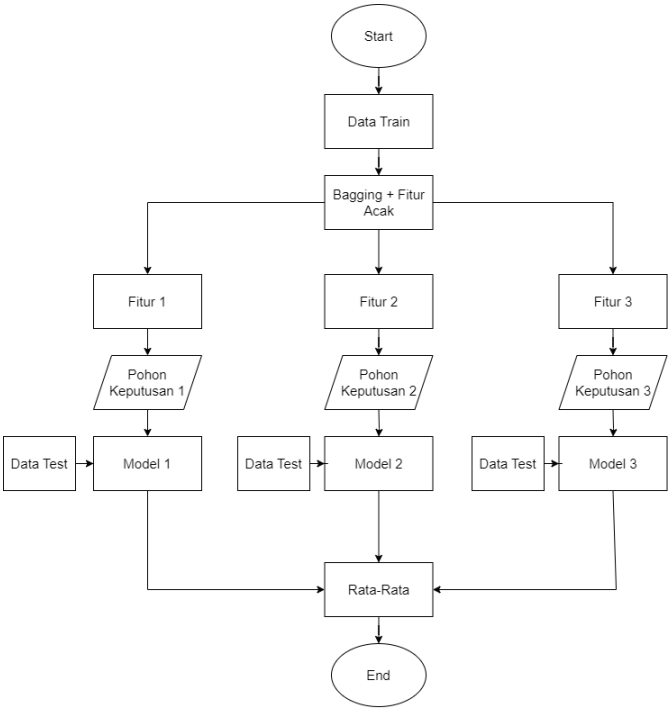
Alur pengklasifian naïve bayes dalam analisis sentiment yakni terdapat beberapa tahapan:

1) Pemilihan fitur berupa kata atau frasa yang menjadi indicator sentimen.

2) Perhitungan probabilitas, yakni menghitung probabilitas yang muncul pada suatu ulasan.
3) Asumsi independensi, artinya naïve bayes menganggap setiap kata tidak saling mempengaruhi.
4) Perhitungan probabilitas kategori, artinya setiap kategori akan dihitung pada setiap ulasan, baik itu positif ataupun negatif.

b. Random Forest Classifier

Melakukan klasifikasi random forest dengan bantuan *library* python. dan beberapa fungsi yang terdapat dalam modul python yang menunjang dalam perhitungan klasifikasi random forest. Adapun alur klasifikasi random forest digambarkan dalam Gambar 3.2.



Gambar 3. 2 Flowchart Random Forest

Alur pengklasifian random forest dalam analisis sentiment yakni terdapat beberapa tahapan:

1) Pemilihan sampel atau pemilihan subset secara acak.
2) Pembentukan pohon keputusan, yakni pohon dibangun dengan

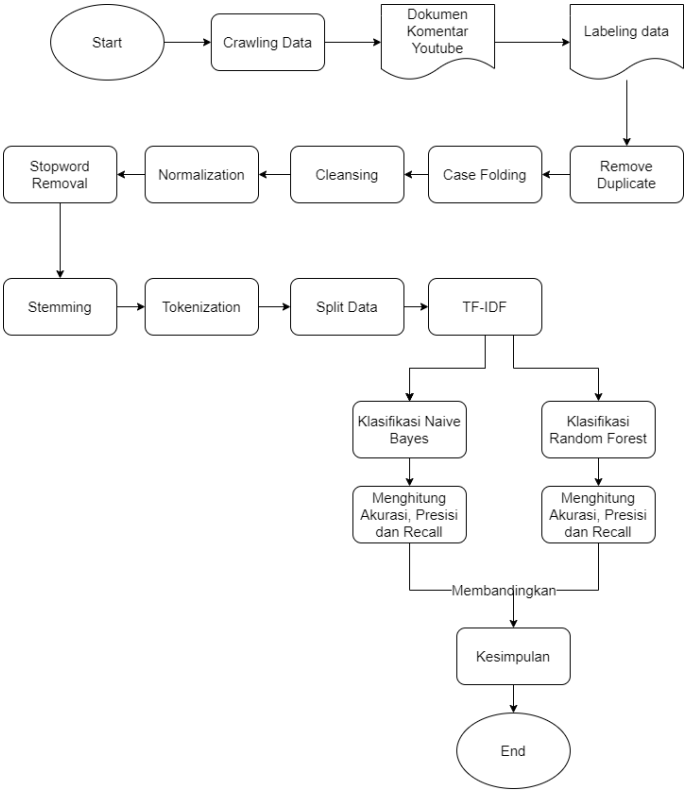
pemilihan fitur-fitur yang relavan secara acak dari dataset. Pemilihan membantu mengurangi korelasi antar pohon, sehingga setiap pohon memiliki pola yang berbeda.

3) Melakukan prediksi pada setiap pohon. Setiap pohon akan memberikan prediksi berdasarkan fitur-fitur yang dipilih. Fitur mungkin berupa kata atau frasa yang menunjukkan sentiment positif ataupun negatif.
4) Voting, yakni hasil setiap pohon akan dijumlahkan. Jika sebagian besar pohon memiliki nilai positif, maka sentimen dianggap positif. Begitupun apabila sebagian besar pohon memiliki nilai negatif, maka sentimen dianggap negatif.

5. Menghitung akurasi, presisi, dan *f1 score* dengan bantuan *library python* untuk masing-masing klasifikasi dan membadingkan hasilnya.

6. Menemukan hasil klasifikasi yang paling baik diantara kedua metode klasifikasi.

Untuk lebih jelasnya, bisa dilihat untuk diagram gambar 3.3 mengenai alur metodologi penelitian:



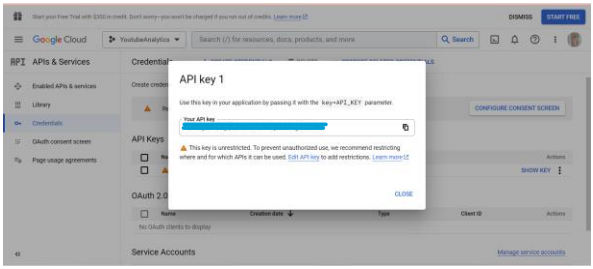
Gambar 3. 3. Diagram Alur Metodologi

D. Penjelasan Metodologi

1. Pengambilan Data (*Crawling Data*)

   Data yang digunakan dalam penelitian ini adalah data komentar youtube, maka untuk mengambil datanya peneliti menggunakan Google Cloude untuk dapat mendapatkan code token untuk dapat melakukan *crawling* data dari API Keys untuk dapat mengambil data dari YouTube API.

   Untuk mendapatkan API Keys dari Google Cloude perlu untuk mendaftar terlebih dahulu dengan mengisi beberapa form. Berikut tampilan apabila telah mendapatkan API Key dari YouTube API ada pada gambar 3.2



Gambar 3. 4 *Crawler* YouTube API

Setelah mendapatkan API Key, key tersebut nantinya akan dapat dijadikan sebagai pengambilan data melalui bahasa pemrograman *python.*

2. Pelabelan Data

   Setelah dilakukan *crawling,* langkah selanjutnya adalah pelabelan. Data yang telah disimpan dalam bentuk csv, akan dilakukan pelabelan. Contoh proses pelabelan ada pada tabel 3.3

Tabel 3. 3 Tabel Pelabelan Data

Komentar	Sentimen
Langsung exekusi mati orangnya.	Negatif
Mantap pak pol wahyu dlm menangani kasus narkoba. doa kami menyertai kalian semua	Positif

3. *Remove Duplicate*

   Dalam data terkadang terdapat kata-kata yang sama yang terkadang dikirim orang lebih dari sekali pada komentar. Oleh karena itu, perlunya dilakukan pembersihan duplikat atau pembuangan duplikat. Berikut contoh komentar yang sama dengan jumlah lebih dari satu pada gambar 3.4

Gambar 3. 5 Contoh data duplikat

4. *Case Folding*

   Case Folding merupakan salah satu tahapan *Text Preprocessing* yang berfungsi untuk merubah huruf besar menjadi huruf kecil atau *lowercase.* Hal ini bertujuan untuk menyamakan semua huruf menjadi kecil agar memudahkan dalam pengelolaan teks. Contohnya ada pada tabel 3.3

Tabel 3. 4 Tabel Contoh *Case Folding*

Input Proccess	*Output Proccess*
Keberhasilan Polisi menangkap Narkoba terbesar di Indonesia adalah keberhasilan Negara, untuk mengamankan generasi muda dari bahaya narkoba, saat ini polisi di angkat Jempol, kemudian Polisi ada di Kecamatan, dan sekarang nakoba sudah menyebar sampai tingkat kecamatan.HATI2.	keberhasilan polisi menangkap narkoba terbesar di indonesia adalah keberhasilan negara, untuk mengamankan generasi muda dari bahaya narkoba, saat ini polisi di angkat jempol, kemudian polisi ada di kecamatan, dan sekarang nakoba sudah menyebar sampai tingkat kecamatan.hati2.

5. *Cleansing*

   Cleansing merupakan salah satu tahapan *Text Preprocessing* yang berfungsi untuk menghapus URL atau tautan web, menghapus perulangan karakter yang berurutan, menghapus tanda baca, dan mengubah emoji menjadi teks. Contoh *cleansing* ada pada tabel 3.4

Tabel 3. 5 Tabel Contoh *Cleansing*

Input Proccess	*Output Proccess*
keberhasilan polisi menangkap narkoba terbesar di indonesia adalah keberhasilan negara, untuk mengamankan generasi muda dari bahaya narkoba, saat ini polisi di angkat jempol, kemudian polisi ada di kecamatan, dan sekarang nakoba sudah menyebar sampai tingkat kecamatan.hati2.	keberhasilan polisi menangkap narkoba terbesar di indonesia adalah keberhasilan negara untuk mengamankan generasi muda dari bahaya narkoba saat ini polisi di angkat jempol kemudian polisi ada di kecamatan dan sekarang nakoba sudah menyebar sai tingkat kecamatan hati

6. *Normalization*

   Normalization juga merupakan salah satu tahapan *text preprocessing* yang berfungsi untuk menormalkan kata yang tidak sesuai dengan KBBI. Selain itu *normalization* juga menormalkan kata-kata gaul zaman sekarang seperti adanya singkatan dan bahasa anak muda yang bukan merupakan bahasa Indonesia baku. Contohnya ada pada tabel 3.5

Tabel 3. 6 Tabel Contoh *Normalization*

Input Proccess	*Output Proccess*
keberhasilan polisi menangkap narkoba terbesar di indonesia adalah keberhasilan negara untuk mengamankan generasi muda dari bahaya narkoba saat ini polisi di angkat jempol kemudian polisi ada di kecamatan dan sekarang nakoba sudah menyebar sai tingkat kecamatan hati	keberhasilan polisi menangkap narkotika, psikotropika, dan obat terlarang terbesar di indonesia adalah keberhasilannegara untuk mengamankan generasi muda dari bahaya narkotika, psikotropika, dan obat terlarang saat ini polisi di angkat jempol kemudian polisi ada di kecamatan dan sekarang narkoba sudah menyebar sai tingkat kecamatan hati

7. *Stopword Removal*

   Stopword Removal juga merupakan salah satu tahapan *text preprocessing. Stopword Removal* berfungsi menghilangkan kata yang memiliki fungsi tapi tidak memiliki arti atau makna. *Stopword Removal* membantu dalam pengolahan teks agar sistem tidak menyimpan banyak kata untuk diproses. Contoh *Stopword Removal* ada pada tabel 3.6

Tabel 3. 7 Tabel Contoh *Stopword Removal*

Input Proccess	*Output Proccess*
keberhasilan polisi menangkap narkotika, psikotropika, dan obat terlarang terbesar di indonesia adalah keberhasilannegara untuk mengamankan generasi muda dari bahaya narkotika, psikotropika, dan obat terlarang saat ini polisi di angkat jempol kemudian polisi ada di kecamatan dan sekarang narkoba sudah menyebar sai tingkat kecamatan hati	keberhasilan polisi menangkap narkotika, psikotropika, obat terlarang terbesar indonesia keberhasilan negara mengamankan generasi muda bahaya narkotika, psikotropika, obat terlarang polisi angkat jempol polisi kecamatan narkoba menyebar sai tingkat kecamatan hati

8. *Stemming*

   Stemming merupakan salah satu tahap *text preprocessing. Stemming* berfungsi untuk menghilangkan imbuhan kata seperti prefiks, sufiks dan infiks menjadi kata dasar. Contohnya ada pada tabel 3.7

Tabel 3. 8 Tabel Contoh *Stemming*

Input Proccess	*Output Proccess*
keberhasilan polisi menangkap narkotika, psikotropika, obat terlarang terbesar indonesia keberhasilan negara mengamankan generasi muda bahaya narkotika, psikotropika, obat terlarang polisi angkat jempol polisi kecamatan narkoba menyebar sai tingkat kecamatan hati	hasil polisi tangkap narkotika psikotropika obat larang besar indonesia hasil negara aman generasi muda bahaya narkotika psikotropika obat larang polisi angkat jempol polisi camat narkoba sebar sai tingkat camat hati

9. *Tokenization*

   Tokenization merupakan salah satu tahap *preprocessing* yang digunakan untuk merubah kalimat menjadi token-token. *Tokenization* adalah tahap untuk menghilangkan *whitespace* sehingga kalimat dipecah menjadi per kata. Contohnya ada pada tabel 3.8

Tabel 3. 9 Tabel Contoh *Tokenization*

Input Proccess	*Output Proccess*
hasil polisi tangkap narkotika psikotropika obat larang besar indonesia hasil negara aman generasi muda bahaya narkotika psikotropika obat larang polisi angkat jempol polisi camat narkoba sebar sai tingkat camat hati	['hasil', 'polisi', 'tangkap','narkotika', 'psikotropika', 'obat', 'larang', 'besar', 'indonesia', 'hasil', 'negara', 'aman', 'generasi', 'muda', 'bahaya', 'narkotika', 'psikotropika', 'obat', 'larang', 'polisi', 'angkat', 'jempol', 'polisi', 'camat', 'narkoba', 'sebar', 'sai', 'tingkat', 'camat', 'hati']

10. *Split Data*

    Split Data adalah membagi data menjadi data *training* atau data latih dan data *testing* atau data uji dengan menggunakan *split validation*. *Split validation* dilakukan dengan jumlah data *testing* diambil 20% dari data *training.* Pengambilan data dilakukan secara random dengan bantuan *library python* (Turmudi Zy et al., 2021). Proses pengujian model dilakukan setelah proses pelatihan data. Pengujian terhadap model sendiri dilakukan untuk mengetahui kinerja model. Setelah pengujian

model dilakukan, kinerja metode akan muncul (Zidan, 2022).

11. Ekstraksi Fitur

    Output dari klasifikasi nantinya akan muncul tiga nilai yakni positif, negatif, dan netral. Untuk melakukan klasifikasi sentimen, peneliti akan menggunakan data mulai dari *preprocessing* hingga pembobotan kata menggunakan TF-IDF. TF-IDF digunakan untuk merubah teks menjadi *vector.* Kata yang sering muncul cenderung memiliki nilai yang kecil sedangkan nilai yang jarang muncul memiliki nilai yang besar. Setelah data berhasil dilatih kemudian akan dilakukan pengujian data uji untuk menguji ketepatan klasifikasi yang dilakukan (Zidan, 2022).

12. Klasifikasi Naïve Bayes

    Data yang telah dilakukan *preprocessing* dan ekstraksi fitur, tahap selanjutnya yakni proses pengklasifikasian menggunakan naïve bayes. Peneliti akan menerapkan metode naïve bayes untuk mengukur ketepatan klasifikasi dengan dibagi menjadi tiga output yakni positif,

negatif, dan netral. Setelah dilakukan tahapan *preprocessing* dan ekstraksi fitur dengan menggunakan TF-IDF kemudian data diuji menggunakan data *training* dan data *testing* untuk mengetahui ketepatan klasifikasi menggunakan metode naïve bayes. Kelas dengan kemunculan skor terbanyak akan dianggap menjadi kelas tersebut (Fitri, 2020).

13. Klasifikasi Random Forest

    Sama dengan klasifikasi naïve bayes, dalam klasifikasi random forest juga melewati tahapan yang seperti *preprocessing,* ekstraksi fitur, pembobotan TF-IDF dan pengujian data *training* dan data *testing.* Dan kelas dengan kemunculan skor terbanyak akan dianggap menjadi kelas tersebut (Fitri, 2020).

14. Evaluasi Model

    Langkah selanjutnya adalah evaluasi model untuk masing-masing model klasifikasi. Evaluasi model dilakukan dengan menguji tingkat kinerja metode melalui matriks konfusi multi-layer (*matrix confusion multiclass*). Intinya, matriks konfusi berisi informasi yang

membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang diperlukan (Turmudi Zy et al., 2021). Dengan kata lain, data uji yang diuji dengan data latih akan menghasilkan daftar kelas dari data uji tersebut, yang disebut prediksi kelas. Prediksi kelas tersebut kemudian dibandingkan dengan kelas sebenarnya dari data pengujian yang sebelumnya disembunyikan (Zidan, 2022). Sehingga dapat dilihat pada performa masing-masing model baik naïve bayes maupun random forest yang berupa tingkat akurasi, pressicion, recall dan f1 score. Dengan hasil nilai-nilai akurasi, presisi, *recall* dan *f1 score* antara naïve bayes *classifier* dan random forest *classifier* nantinya akan dibandingkan manakah dari kedua model tersebut yang memiliki nilai yang lebih tinggi sebagai jawaban dari perbandingan performa klasifikasi dalam penelitian ini.

BAB IV
HASIL DAN PEMBAHASAN

A. *Crawling* Data Komentar YouTube

Pengambilan data komentar youtube menggunakan bantuan *google cloud* untuk mendapatkan youtube API *key* nya, setelah didapatkan token API key seperti pada gambar 3.4, langkah selanjutnya untuk melakukan pengambilan data dibantu dengan menggunakan bahasa pemrograman *python* menggunakan *Jupyter nootebook.* Untuk dapat menghubungkan *google cloud* ke *python* adalah dengan meng-*install library google-api-python-client. Google-api-python-client* adalah *library* yang digunakan untuk dapat mengakses berbagai layanan google ke dalam *python* termasuk *google cloud.* Adapun cara *install google-api-python-client* adalah dengan menggunakan code pada gambar 4.1.

```
In [2]: pip install google-api-python-client
```

Gambar 4. 1 *Source Code to install google-api-python-client*

Setelah dilakukan instalasi pada *google-api-python-client,* langkah selanjutnya adalah pengambilan data komentar youtube. Pengambilan data komentar menggunakan *python* menggunakan *code* pada gambar 4.2.

```python
from googleapiclient.discovery import build
# Menentukan kunci API Anda
api_key = [illegible]
# Membangun objek YouTube Data API
youtube = build('youtube', 'v3', developerKey=api_key)

def video_comments(video_id):
    comments = []
    next_page_token = None
    while True:
        # retrieve youtube video result
        video_response = youtube.commentThreads().list(
            part='snippet,replies',
            videoId=video_id,
            maxResults=None,  # Ubah nilai maxResults sesuai kebutuhan Anda
            pageToken=next_page_token
        ).execute()
        # iterate video response
        for item in video_response['items']:
            comment = item['snippet']['topLevelComment']['snippet']['textDisplay']
            comments.append(comment)
        # cek jika ada halaman berikutnya
        if 'nextPageToken' in video_response:
            next_page_token = video_response['nextPageToken']
        else:
            break
    return comments

# Menentukan ID video yang ingin Anda dapatkan komentarnya
video_id = 'P01IloHn1pk'
# Mengambil komentar-komentar dari video
comments = video_comments(video_id)
# Menampilkan komentar-komentar yang didapatkan
for comment in comments:
    print(comment)
```

Gambar 4. 2 *Source Code Crawling* Data Komentar

Pengambilan data komentar dimulai dengan meng-*import library google-api-python-*

client, kemudian memasukan API key yang telah didapat dari *google cloud.* Langkah selanjutnya yakni membangun objek youtube data API dengan menggunakan fungsi *'build'*, objek ini akan berinteraksi dengan API youtube data. Langah berikutnya yakni membangun fungsi dengan nama *'video_comments'*, ini adalah fungsi yang digunakan untuk mengambil ID video tersebut dengan menggunakan objek variable 'youtube' pada code sebelumya. Kemudian menentukan ID video yang akan diambil komentarnya, pada gambar 4.2 ada pada variable 'video_id'. Setelah itu diambilah data komentar yang ada pada video youtube yang telah ditentukan ID nya dengan variable bernama 'comments. Data komentar yang telah diambil kemudian membuat *dataframe* yang berisi komentar-komentar sebelum disimpan dalam bentuk csv seperti pada gambar 4.3.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [11]: df = pd.DataFrame(comments,columns=['komentar'])
         print(df)
```

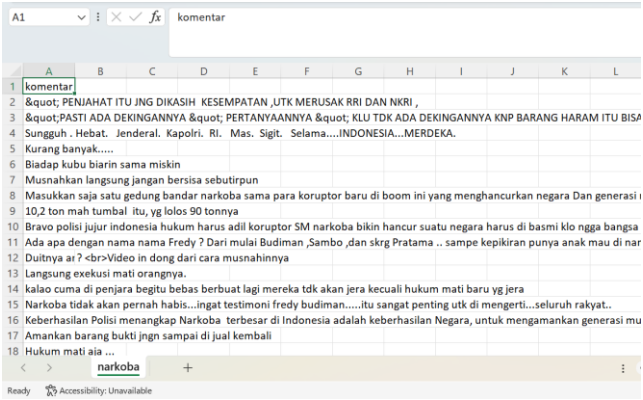
Gambar 4. 3 *Source Code* Pembuatan *Dataframe*

Data komentar yang sudah tertata dalam *dataframe* kemudian disimpan dalam bentuk csv dengan *source code* pada gambar 4.4.

```
In [13]: df.to_csv('narkoba.csv', index=False)
```

Gambar 4. 4 *Source Code Save* Data Komentar pada CSV

Setelah proses tersebut maka data komentar telah tersimpan dalam bentuk file CSV pada gambar 4.5.



Gambar 4. 5 Data Komentar yang Disimpan dalam File CSV

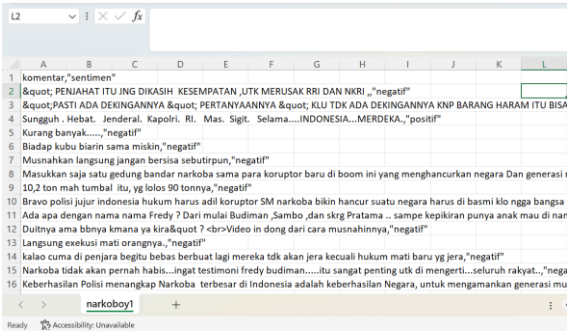
B. Pelabelan

Langkah selanjutnya setelah pengumpulan data adalah pelabelan. Pelabelan di sini bermaksud melabeli sentimen pada setiap komentar apakah data termasuk ke dalam sentimen positif atau negatif. Dalam proses pelabelan sentimen komentar ini, peneliti akan menentukan nilai sentimen positif apabila di dalam komentar terdapat kata-kata yang bermakna afirmasi ataupun kata-kata positif yang lebih dominan dari kata negatifnya dan kalimat tersebut bermakna mendukung terhadap narkoba. Sedangkan nilai negatif dianggap apabila kalimat tersebut memiliki kata-kata negatif yang dominan.

Dalam proses pelabelan seharusnya bisa dilakukan dengan memanfaatkan *polarity* dengan *library textblob* yang ada pada *python*. Namun, dikarenakan pada kasus penelitian ini menggunakan bahasa Indonesia yang kompleks dan banyak data yang tidak menggunakan bahasa baku sehingga hasil percobaan program tidak memunculkan hasil yang valid pada nilai sentimen, sehingga proses pelabelan dilakukan dengan

manual. Dalam proses pelabelan manual seharusnya dilakukan oleh pakar bahasa ataupun seseorang yang ahli di bidangnya, dan setidaknya membutuhkan dua orang atau lebih dalam pendiskusian mengenai pelabelannya untuk menghindari nilai subjektiv dalam melakukan sentimen terhadap data (Imron, 2019).

Pada pelabelan sentimen penelitian ini, peneliti didampingi oleh salah satu mahasiswa jurusan bahasa untuk membantu proses pelabelan manual ini. Namun, dalam proses pelabelan manual ini membutuhkan waktu yang cukup lama karena jumlah datanya yang banyak. Berikut pada gambar 4.6. adalah data yang sudah diberi label sentimen.



Gambar 4. 6 Data yang Sudah Diberi Label Sentimen

C. *Remove Duplicate*

Remove duplicate adalah menghapus data yang memiliki kesamaan sehingga menjadi satu data saja. Oleh karena itu, nantinya jumlah datanya juga akan berkurang setelah proses penghapusan *duplicate.* Untuk mengetahui berapa jumlah data yang terduplikat dapat menggunakan *code* pada gambar 4.7.

```
In [6]: df.duplicated().sum()
Out[6]: 50
```

Gambar 4. 7 *Source Code* Jumlah Data Duplikat

Untuk mengetahui data-data mana saja yang terduplikat dapat menggunakan *code* pada gambar 4.8.

```
duplicates = df[df.duplicated()]
print(duplicates)
```

```
                                                komentar sentimen
181                                          Hukum mati  negatif
186                                          Hukum mati  negatif
354                                          Hukum mati  negatif
358   Nah gitu itu baru josss institusi kepolisian h...  negatif
379   Modarr miskin kan sj yg melakukan Marlina temb...  negatif
566                                        ðŸ‘ðŸ‘ðŸ‘  positif
618                                          Hukum mati  negatif
652                                          Hukum mati  negatif
674                                                ðŸ˜Š  positif
675                                                ðŸ˜Š  positif
725                                          Hukum mati  negatif
771                                          Hukum mati  negatif
791                                         Hukum mati.  negatif
836                                          Hukum mati  negatif
847   CAAIIIIIRRRRRRR ðŸ¤£<br>SEKILO AJA BANYAK DUIT...  negatif
848   CAAIIIIIRRRRRRR ðŸ¤£<br>SEKILO AJA BANYAK DUIT...  negatif
849   CAAIIIIIRRRRRRR ðŸ¤£<br>SEKILO AJA BANYAK DUIT...  negatif
850   CAAIIIIIRRRRRRR ðŸ¤£<br>SEKILO AJA BANYAK DUIT...  negatif
997                   Tolong di morowali,,  di basmi,,  negatif
1157                                         Hukum mati  negatif
1204                                        Hukuman mati  negatif
```

Gambar 4. 8 *Source Code* dan Tampilan Hasil Data Duplikat

Setelah muncul data-data yang terduplikat, langkah selanjutnya adalah menghapus data yang terduplikat menjadi satu data saja, dengan menggunakan *code* pada gambar 4.9.

```
In [9]: df.drop_duplicates(inplace = True)
```

Gambar 4. 9 *Source Code* Hapus Data Duplikat

D. *Case Folding dan Cleansing*

Dalam penelitian kali ini, proses *case folding* dan *cleansing* peneliti jadikan satu dalam *source code* nya untuk meringkas proses *text*

preprocessing. Dalam *text preprocessing* diperlukan beberapa *library* untuk prosesnya. Sebelum dilakukannya langkah-langkah *text preprocessing* pada tahap-tahap selanjutnya, peneliti melakukan import *library* nya di awal proses. Adapun *source code* untuk pemanggilan beberapa *library* nya ada pada gambar 4.10.

```
import nltk
nltk.download("punkt")

import nltk
nltk.download("stopwords")

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import re
import emoji
from string import punctuation
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\NAILUL\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\NAILUL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Gambar 4. 10 Import Library Text Preprocessing

Proses *case folding* dan *cleansing* merupakan proses *text preprocessing* untuk menyeragamkan semua huruf menjadi huruf kecil, membersihkan teks dari tanda baca, mengubah emoji menjadi *string,* mengapus angka, menghapus URL dan sebagainya. Adapun *code* untuk

melakukan *case folding* dan *cleansing* ada pada gambar 4.11.

```
def cleansing(text):
    text = emoji.demojize(str(text))
    text = text.lower()
    text = re.sub(r'^https?://.*[\r\n]*', '', text, flags=re.MULTILINE)
    text = re.sub(r'(.)\1+', r'\1\1', text)
    text =  re.sub(r'quot', '', text)
    text = re.sub(r'br', '', text)
    text = re.sub(r'amp', '', text)
    re.sub(r'[?\.\!]+', '', text)
    text = re.sub(r'[^a-zA-Z]', ' ', text)
    return text
```

Gambar 4. 11 *Source Code Case Folding* dan *Cleansing*

Untuk mengaplikasikan fungsi *cleansing* seperti pada gambar 4.11 pada data komentar adalah dengan menggunakan *code* pada gambar 4.12.

```
In [14]: df['komentar']=df.komentar.apply(cleansing)
```

Gambar 4. 12 *Cose Folding* dan *Cleansing* Komentar

Untuk melihat hasil proses fungsi *cleansing* adalah dengan menggunakan *code* pada gambar 4.13.

```
df.head()
```

	komentar	sentimen
0	penjahat itu jng dikasih kesempatan utk m...	negatif
1	pasti ada dekingannya pertanyaannya kl...	negatif
2	sungguh hebat jenderal kapolri ri ma...	positif
3	kurang banyak	negatif
4	biadap kubu biarin sama miskin	negatif

Gambar 4. 13 Hasil Proses *Case Folding* dan *Cleansing*

E. *Normalization*

Normalization adalah proses pernormalan kata, seperti kata tidak baku diubah menjadi kata baku, mengubah kata singkatan menjadi kata yang sebenarnya dan membenarkan kata yang *typo* atau kesalahan dalam penulisan. Dalam *normalization* ini, peneliti menggunakan 5 data tambahan untuk menunjang proses pernormalan ini. Berikut adalah *code* untuk *import* data tambahan pada gambar 4.14.

```
slang_dictionary = pd.read_csv('colloquial-indonesian-lexicon.csv')
s_d = pd.Series(slang_dictionary['formal'].values,index=slang_dictionary['slang']).to_dict()

slang_dictionary1 = pd.read_csv('kbba.txt' , sep='\t')
s_d1 = pd.Series(slang_dictionary1['tujuan'].values,index=slang_dictionary1['7an']).to_dict()

slang_dictionary2 = pd.read_csv('slangword.txt' , sep=':')
s_d2 = pd.Series(slang_dictionary2['dan'].values,index=slang_dictionary2['&']).to_dict()

slang_dictionary3 = pd.read_csv('formalizationDict.txt' , sep='\t')
s_d3 = pd.Series(slang_dictionary3['tujuan'].values,index=slang_dictionary3['7an']).to_dict()

slang_dictionary4 = pd.read_csv('typo.txt' , sep=':')
s_d4 = pd.Series(slang_dictionary4['jangan'].values,index=slang_dictionary4['jng']).to_dict()
```

Gambar 4. 14 *Source Code Import* Data Pendukung

Setelah dilakukan import data pendukung *normalization* peneliti akan menampilkan data teratas dari file pendukung tersebut pada gambar 4.15 - 4.17.

```
slang_dictionary.head()
```

	slang	formal	In-dictionary
0	woww	wow	1
1	aminn	amin	1
2	met	selamat	1
3	netaas	menetas	1
4	keberpa	keberapa	0

```
slang_dictionary1.head()
```

	7an	tujuan
0	@	di
1	ababil	abg labil
2	abis	habis
3	acc	accord
4	ad	ada

Gambar 4. 15 Data Pendukung 1 dan 2

```
slang_dictionary2.head()
```

	&	dan
0	+	tambah
1	/	atau
2	=	sama dengan
3	ababil	anak ingusan
4	abal2	palsu

```
slang_dictionary3.head()
```

	7an	tujuan
0	@	di
1	ababil	abg labil
2	abis	habis
3	acc	accord
4	ad	ada

Gambar 4. 16 Data Pendukung 3 dan 4

```
slang_dictionary4.head()
```

	jng	jangan
0	biadap	biadab
1	ama	sama
2	kmana	kemana
3	exekusi	eksekusi
4	nakoba	narkoba

Gambar 4. 17 Data Pendukung 5

Langkah selanjutnya yaitu membuat fungsi untuk mengimplementasikan data penunjang tersebut dengan menggunakan *code* pada gambar 4.18.

```
def Slangwords(text):
    for word in text.split():
        if word in s_d.keys():
            text = text.replace(word, s_d[word])
    return text

def Slangwords1(text):
    for word in text.split():
        if word in s_d1.keys():
            text = text.replace(word, s_d1[word])
    return text

def Slangwords2(text):
    for word in text.split():
        if word in s_d2.keys():
            text = text.replace(word, s_d2[word])
    return text

def Slangwords3(text):
    for word in text.split():
        if word in s_d3.keys():
            text = text.replace(word, s_d3[word])
    return text

def Slangwords4(text):
    for word in text.split():
        if word in s_d.keys():
            text = text.replace(word, s_d[word])
    return text
```

Gambar 4. 18 *Source Code* Data Penunjang Normalisasi

Langkah selanjutnya yakni menerapkan fungsi-fungsi pada gambar 4.18 pada data komentar, dengan menggunakan *code* pada gambar 4.19.

```
df['Text_Clean'] = df['komentar'].apply(Slangwords)
df['Text_Clean'] = df['Text_Clean'].apply(Slangwords1)
df['Text_Clean'] = df['Text_Clean'].apply(Slangwords2)
df['Text_Clean'] = df['Text_Clean'].apply(Slangwords3)
df['Text_Clean'] = df['Text_Clean'].apply(Slangwords4)
df['Text_Clean'].head()
```

```
0       penjahat itu jangan dikasih  kesempatan  un...
1      pasti ada dekingannya    pertanyaannya    ka...
2    sungguh   hebat   jenderal   kapolrepublik ind...
3                                      kurang banyak
4                      biadap kubu biarkan sama miskin
Name: Text_Clean, dtype: object
```

Gambar 4. 19 Normalisasi Pada Data Komentar

F. *Stopword Removal*

Stopword Removal merupakan proses menghapus kata yang memiliki fungsi tetapi tidak memiliki arti atau makna. Untuk melakukan *stopword removal* membutuhkan *library* NLTK dan *corpus stopword* dari *library* NLTK. Untuk *code import* NLTK dan *download stopword* ada pada gambar 4.10. *Code* untuk meng-*install library* NLTK ada pada gambar 4.20.

```
pip install nltk
```

Gambar 4. 20 Source Code Instalasi library NLTK

Adapun *code* untuk membuat *fungsi stopword* dan mengaplikasikannya pada data komentar ada pada gambar 4.21.

```
sw_indo = stopwords.words('indonesian') + list(punctuation)

def stopword(text):  # Menerima argumen teks
    word_list = [word for word in text.split() if word.lower() not in sw_indo]
    # Gabungkan kata-kata yang tersisa
    text = ' '.join(word_list)
    return text

df['Text_Clean'] = df['Text_Clean'].apply(stopword)
df['Text_Clean'].head()
```

```
0         penjahat dikasih kesempatan merusak rri nkri
1    dekingannya pertanyaannya dekingannya barang h...
2    sungguh hebat jenderal kapolrepublik indonesia...
3
4                           biadap kubu biarkan miskin
Name: Text_Clean, dtype: object
```

Gambar 4. 21 *Source Code Stopword Removal*

G. *Stemming*

Stemming adalah proses merubah kata berimbuhan menjadi kata dasar. Untuk melakukan proses *stemming* memerlukan *library sastrawi* dan memerlukan modul *stemmerFactory.* Untuk code pengimportan *sastrawi* ada pada gambar 4.10. *Code* untuk melakukan install *library sastrawi* ada pada gambar 4.22.

```
pip install sastrawi
```

Gambar 4. 22 *Source Code* Instalasi *Library Sastrawi*

Adapun *code* untuk melakukan *stemming* dan pengaplikasiannya terhadap komentar ada pada gambar 4.23.

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

df['Text_Clean'] = df['Text_Clean'].apply(stemmer.stem)
df['Text_Clean'].head()
```

```
0                    jahat kasih sempat rusak rri nkri
1    deking tanya deking barang haram masuk nkri an...
2    sungguh hebat jenderal kapolrepublik indonesia...
3
4                              biadap kubu biar miskin
Name: Text_Clean, dtype: object
```

Gambar 4. 23 *Source Code Stemming*

H. *Tokenization*

Tokenization adalah merubah kalimat menjadi token-token atau per-kata. Tokenisasi dilakukan nantinya untuk memudahkan proses TF-IDF yakni proses setelah *text preprocessing*. Adapun untuk proses tokenisasi membutuhkan *library* NLTK dengan fungsi *word_tokenize*, pengimportan fungsi ada pada gambar 4.10. Adapun untuk *code* melakukan proses tokenisasi

dan pengaplikasiannya pada komentar youtube ada pada gambar 4.24.

```
#tokenizen
def tokek(teks):
    text_list = []
    for txt in teks.split(" "):
        text_list.append(txt)
    return text_list
```

```
df['Text_token'] = df['Text_Clean'].apply(tokek)
df['Text_token'].head()
```

```
0              [jahat, kasih, sempat, rusak, rri, nkri]
1    [deking, tanya, deking, barang, haram, masuk, ...
2    [sungguh, hebat, jenderal, kapolrepublik, indo...
3                                                   []
4                          [biadap, kubu, biar, miskin]
Name: Text_token, dtype: object
```

Gambar 4. 24 *Source Code Tokenization*

Setelah dilakukannya beberapa proses dari *text preprocessing* di atas, maka tampilan dari hasil *text preprocessing* adalah seperti pada gambar 4.25.

df

	komentar	sentimen	Text_Clean	Text_token
0	penjahat itu jng dikasih kesempatan utk m...	negatif	jahat kasih sempat rusak rri nkri	[jahat, kasih, sempat, rusak, rri, nkri]
1	pasti ada dekingannya pertanyaannya kl...	negatif	deking tanya deking barang haram masuk nkri an...	[deking, tanya, deking, barang, haram, masuk, ...
2	sungguh hebat jenderal kapolri ri ma...	positif	sungguh hebat jenderal kapolrepublik indonesia...	[sungguh, hebat, jenderal, kapolrepublik, indo...
3	kurang banyak	negatif		[]
4	biadap kubu biarin sama miskin	negatif	biadap kubu biar miskin	[biadap, kubu, biar, miskin]
...	...	...	...	...
2697	mau tanya ni apa fungsi bea cukai harus ...	negatif	fungsi bea cukai selidik labuh laut bandara ba...	[fungsi, bea, cukai, selidik, labuh, laut, ban...
2698	barang sita an di bakar di live biar tau rakya...	negatif	barang sita an bakar live biar rakyat jual	[barang, sita, an, bakar, live, biar, rakyat, ...
2699	pasti ada oknum yg bantu gk mungkin sendiri	negatif	oknum bantu muntidakin	[oknum, bantu, muntidakin]
2700	tolong awasi barang bukti jangan sai kaya si t...	negatif	tolong awasih barang bukti sai kayak sih teddy...	[tolong, awasih, barang, bukti, sai, kayak, si...
2701	ini terjadi semakin marak dan banyak karena hu...	negatif	marak hukum mati palsu palsu negara	[marak, hukum, mati, palsu, palsu, negara]

2652 rows × 4 columns

Gambar 4. 25 Hasil Setelah *Text Preprocessing*

I. *Split Validation Data*

Split Validation Data adalah membagi data komentar menjadi data latih (*training data*) dan data uji (*testing data*). Dalam penelitian ini *split data* dilakukan dengan perbandingan 80:20, yang berarti data uji sebesar 20% dari data komentar. *Library* yang digunakan adalah *sklearn* yang di dalamnya terdapat kelas *train_test_split.* Adapun untuk *code* proses *split data* ada pada gambar 4.26.

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(df['Text_Clean'],df['sentimen'],test_size = 0.2,random_state = 42)
```

Gambar 4. 26 *Source Code Split Validation Data*

J. TF-IDF

TF-IDF (*Term Frequency-Invers Document Frequency*) adalah proses merubah kata menjadi *vector*. TF-IDF adalah proses menghitung kemunculan kata dalam kalimat. Dalam *python* TF-IDF memerlukan *library sklearn* dengan beberapa kelas seperti TfidfVectorizer, CountVectorizer yang ada pada gambar 4.10. Dan juga kelas LabelEncoder pada *library sklearn* seperti pada

gambar 4.27 berikut dengan *code* proses TF-IDF dan pengaplikasiannya pada komentar youtube.

```
#TFIDF
from sklearn.preprocessing import LabelEncoder
Encoder = LabelEncoder()
y_train = Encoder.fit_transform(y_train)
y_test = Encoder.fit_transform(y_test)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
Tfidf_vector = TfidfVectorizer(max_features = 50)
Tfidf_vector.fit(df['Text_Clean'])
Train_x_Tfidf = Tfidf_vector.transform(x_train)
Test_x_Tfidf = Tfidf_vector.transform(x_test)
```

```
Train_x_Tfidf.toarray()
```

```
array([[0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.40792698, 0.        , 0.        , ..., 0.        , 0.445808  ,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ]])
```

Gambar 4. 27 Pembobotan TF-IDF dan Hasilnya

Dalam penelitian ini, untuk contoh cara kerja perhitungan TF-IDF peneliti mengambil 3 contoh data yakni sebagai berikut:

(Doc 1) = “Hukum mati dia merusak anak bangsa”

(Doc 2) = “Perusak generasi bangsa itu setelah disita aset para pengedar narkoba ini hukum mati semuanya biar tak ada lagi bandar besar ini”

(Doc 3) = “Hukum mati semua yg usaha produksi narkoba”

Setelah dilakukan *text preprocessing* maka data tersebut menjadi seperti:

(Doc 1) = [‘hukum’ , ’mati’, ‘rusak’ , ‘anak’ , ‘bangsa’]

(Doc 2) = [‘rusak’ , ‘generasi’ , ‘bangsa’ , ‘sita’ , ‘aset’ , ‘edar’ , ‘narkoba’ , ‘hukum’ , ‘mati’ , ‘bandar’ , ‘besar’]

(Doc 3) = [‘hukum’ , ‘mati’ , ‘usaha’ , ‘produk’ , ‘narkoba’]

Tahap selanjutnya yakni perhitungan TF-IDF menggunakan *word vector*. Perhitungan jumlah kata pada setiap dokumen dinamakan proses TF, sedangkan IDF adalah mengurangi bobot suatu kata apabila kemunculan kata tersebut tersebar banyak pada setiap dokumen. Adapun perhitungan TF ada pada tabel 4.1.

Tabel 4. 1 Contoh Perhitungan TF (*Term-Frequency*)

Kata	TF		
	D1	D2	D3
Hukum	1	1	1
Mati	1	1	1
Rusak	1	1	0
Anak	1	0	0
Bangsa	1	1	0
Generasi	0	1	0
Sita	0	1	0
Aset	0	1	0
Edar	0	1	0
Narkoba	0	1	1
Bandar	0	1	0
Besar	0	1	0
Usaha	0	0	1
Produk	0	0	1

Setelah disusunnya tabel TF, kemudian langkah selanjutnya adalah menentukan DF. DF adalah jumlah dari banyaknya dokumen yang mengandung kata tertentu, Adapun penjabarannya ada pada tabel 4.2.

Tabel 4. 2 Contoh Perhitungan DF

Kata	TF			DF
	D1	D2	D3	
Hukum	1	1	1	3
Mati	1	1	1	3
Rusak	1	1	0	2
Anak	1	0	0	1
Bangsa	1	1	0	2
Generasi	0	1	0	1
Sita	0	1	0	1
Aset	0	1	0	1
Edar	0	1	0	1
Narkoba	0	1	1	2
Bandar	0	1	0	1
Besar	0	1	0	1
Usaha	0	0	1	1
Produk	0	0	1	1

Setelah dihitungnya nilai DF, langkah selanjutnya adalah menghitung nilai IDF. Adapun perhitungan nilai IDF terjabar dalam tabel 4.3.

Tabel 4. 3 Contoh Perhitungan IDF

Kata	DF	D/DF (D=3)	IDF (log(D/DF))	IDF+1
Hukum	3	1	Log 1 = 0	1
Mati	3	1	Log 1 = 0	1
Rusak	2	1,5	Log 1,5 = 0,176	1,176
Anak	1	3	Log 3 = 0,447	1,447
Bangsa	2	1,5	Log 1,5 = 0,176	1,176
Generasi	1	3	Log 3 = 0,447	1,447
Sita	1	3	Log 3 = 0,447	1,447
Aset	1	3	Log 3 = 0,447	1,447
Edar	1	3	Log 3 = 0,447	1,447
Narkoba	2	1,5	Log 1,5 = 0,176	1,176
Bandar	1	3	Log 3 = 0,447	1,447
Besar	1	3	Log 3 = 0,447	1,447
Usaha	1	3	Log 3 = 0,447	1,447
Produk	1	3	Log 3 = 0,447	1,447

Setelah diketahui nilai TF dan IDF, langkah selanjutnya yaitu menghitung nilai TF-IDF dengan penjabaran pada tabel 4.4.

Tabel 4. 4 Contoh Perhitungan TF-IDF

Kata	TF			IDF (Log (D/DF))	IDF+1	TF*IDF		
	D1	D2	D3			D1	D2	D3
Hukum	1	1	1	0	1	1	1	1
Mati	1	1	1	0	1	1	1	1
Rusak	1	1	0	0,176	1,176	1,176	1,176	0
Anak	1	0	0	0,447	1,447	1,447	0	0
Bangsa	1	1	0	0,176	1,176	1,176	1,176	0
Generasi	0	1	0	0,447	1,447	0	1,447	0
Sita	0	1	0	0,447	1,447	0	1,447	0
Aset	0	1	0	0,447	1,447	0	1,447	0
Edar	0	1	0	0,447	1,447	0	1,447	0
Narkoba	0	1	1	0,176	1,176	0	1,176	1,176
Bandar	0	1	0	0,447	1,447	0	1,447	0
Besar	0	1	0	0,447	1,447	0	1,447	0
Usaha	0	0	1	0,447	1,447	0	0	1,447
Produk	0	0	1	0,447	1,447	0	0	1,447

Hasil dari TF-IDF tersebut jika dijabarkan dalam bentuk *array,* maka akan seperti berikut:

Array([

[1, 1, 1.176, 1.447, 1.176, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[1, 1, 1.176, 0, 1.176, 1.447, 1.447, 1.447, 1.447, 1.176, 1.447, 1.447, 0, 0],

[1, 1, 0, 0, 0, 0, 0, 0, 0, 1.176, 0, 0, 1.447, 1.447]

])

Pada *array* tersebut setiap barisnya mengimplementasikan TF-IDF setiap kata pada dokumen. Dari hasil tersebut dapat terbukti bahwa TF menunjukkan banyaknya jumlah data, sedangkan IDF adalah menunjukkan sering tidaknya kata tersebut muncul pada setiap dokumen, semakin banyak muncul kata, nilai IDF-nya semakin kecil.

K. Klasifikasi *Naïve Bayes*

Setelah dilakukannya pemrosesan teks pada data komentar, kemudian dibagi datanya menjadi data latih dan data uji lalu diubahnya teks menjadi *vector* pada proses TF-IDF langkah selanjutnya adalah proses klasifikasi. Dalam algoritma klasifikasi *naive bayes,* sentiment dilakukan dengan menghitung nilai probabilitas, mana yang nilainya lebih tinggi antara positif dan negatifnya nantinya akan menjadi label sentimen pada komentar tersebut. Contoh proses perhitungan sentimen dalam algoritma *naïve bayes* secara manual sebagai berikut:

(Doc) = "Kalau polisi tidak berani memutuskan hukuman mati maka akan lebih hancur negara ini."

1. Data:
    a. Tidak (negatif)
    b. Berani (positif)
    c. Hukuman mati (negatif)
    d. Hancur (negatif)
2. Perhitungan *Prior*
    a. Jumlah *prior* positif (N_{pos}) = 1
    b. Jumlah *prior* negative (N_{neg}) = 3
    c. Probabilitas *prior* positif

        P (Positif) = $\frac{N_{pos}}{N_{pos}+N_{neg}} = \frac{1}{4}$

    d. Probabilitas *prior* negatif

        P (Positif) = $\frac{N_{neg}}{N_{pos}+N_{neg}} = \frac{3}{4}$
3. Menghitung probabilitas *likelihood*
    a. P ("Tidak" |pos) = $\frac{N_a}{N_{pos}} = \frac{0}{1} = 0$
    b. P ("Tidak" |neg) = $\frac{N_a}{N_{neg}} = \frac{1}{3}$
    c. P ("Berani" |pos) = $\frac{N_a}{N_{pos}} = \frac{1}{1} = 1$
    d. P ("Berani" |neg) = $\frac{N_a}{N_{neg}} = \frac{0}{3} = 0$
    e. P ("Hukuman mati"|pos) = $\frac{N_a}{N_{pos}} = \frac{0}{1} = 0$
    f. P ("Hukuman mati"|neg) = $\frac{N_a}{N_{neg}} = \frac{1}{3}$

g. P (“Hancur” |pos) = $\frac{N_a}{N_{pos}} = \frac{0}{1} = 0$

h. P (“Hancur” |neg) = $\frac{N_a}{N_{neg}} = \frac{1}{3}$

4. Perhitungan probabilitas keseluruhan:

   a. P (Positif | "Tidak Berani Hukuman mati Hancur") ∝ P (Positif) × P ("Tidak" |Positif) × P ("Berani" |Positif) × P ("Hukuman mati" |Positif) × P ("Hancur"|Positif) =

      P (Positif|"Tidak Berani Hukuman mati Hancur") $\propto \frac{1}{4} * 0 * 1 * 0 * 0 * 0$

   b. P (Negatif | "Tidak Berani Hukuman mati Hancur") ∝ P(Negatif) × P ("Tidak" |Negatif) × P ("Berani" |Negatif) × P ("Hukuman mati" |Negatif) × P ("Hancur"|negatif)

      P (Negatif |"Tidak Berani Hukuman mati Hancur") $\propto \frac{3}{4} * \frac{1}{3} * 0 * \frac{1}{3} * \frac{1}{3}$

   c. Nilai P (Negatif |"Tidak Berani Hukuman mati Hancur") lebih tinggi dari nilai positifnya.

Dari perhitungan menggunakan rumus *bayes* diatas maka dikatakan bahwa kalimat "Kalau polisi tidak berani memutuskan hukuman mati maka akan lebih hancur negara ini" Memiliki sentimen negatif.

Dalam penelitian ini, perhitungan nilai sentimen menggunakan algoritma *naïve bayes* dibantu menggunakan bahasa pemrograman *python* dengan bantuan modul *multinominalNB* pada *library skelarn.* Selain itu digunakan juga *modul accuracy_score, precision_score, recall_score, f1_score* untuk mencari skor akurasi, presisi, *recall,* dan *f1 score* nya. Adapun *code* untuk perhitungan klasifikasi *naïve bayes* ada pada gambar 4.28.

```
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Membuat dan melatih model Naive Bayes
naive_bayes_model = MultinomialNB()
naive_bayes_model.fit(Train_x_Tfidf.toarray(), y_train)

predicted = naive_bayes_model.predict(Test_x_Tfidf.toarray())
print("Akurasi Naive Bayes:", accuracy_score(y_test,predicted))
print("Presisi Naive Bayes:", precision_score(y_test,predicted,average='weighted'))
print("Recall Naive Bayes:", recall_score(y_test,predicted,average='weighted'))
print("F1 Score Naive Bayes:", f1_score(y_test,predicted,average='weighted'))
```

```
Akurasi Naive Bayes: 0.8361581920903954
Presisi Naive Bayes: 0.8324125264357202
Recall Naive Bayes: 0.8361581920903954
F1 Score Naive Bayes: 0.8020121487513223
```

Gambar 4. 28 *Source Code Naïve Bayes Classifier*

Pada perhitungan klasifikasi menggunakan *code* tersebut hasil dari klasifikasi *naïve bayes* untuk akurasi bernilai 0,8361, presisi bernilai 0,8324, *recall* bernilai 0,8361 dan F1 *score* bernilai 0,8020.

L. Klasifikasi *Random Forest*

Sebagaimana klasifikasi *naïve bayes* yang diproses setelah adanya *text preprocessing, split validation data* dan proses TF-IDF, klasifikasi *random forest* pun demikian. Klasifikasi *random forest* adalah klasifikasi yang menggunakan konsep pohon keputusan. *Random forest* membentuk beberapa pohon berisi fitur-fitur yang berbeda, kemudian hasil dari pohon-pohon tersebut dipadukan menjadi satu hingga menghasilkan nilai akhir pada *random forest.*

Random forest merupakan algoritma klasifikasi yang dilakukan dengan menggunakan pohon keputusan, dan pohon keputusan tersebut dipisahkan dengan menggunakan fitur-fitur yang ditentukan secara acak. Dengan adanya fitur-fitur berbentuk pohon keputusan yang dibentuk pada algoritma *random forest* nantinya akan lebih

menghasilkan perhitungan yang baik. *Random forest* juga juga memberikan kemampuan untuk mengukur pentingnya fitur-fitur dalam model (Apriliah et al., 2021). Dalam *python* kita dapat melihat seberapa pentingnya fitur yang ada dalam pengolahan data komentar dengan menggunakan modul *RandomForestClassifier* pada *library sklearn* dan menggunakan code seperti pada gambar 4.29.

```
from sklearn.ensemble import RandomForestClassifier

# Inisialisasi dan melatih model Random Forest
random_forest_model = RandomForestClassifier()
random_forest_model.fit(Train_x_Tfidf, y_train)

# Mendapatkan tingkat kepentingan fitur dari model Random Forest
feature_importances = random_forest_model.feature_importances_

# Mendapatkan indeks fitur yang paling penting
top_n = 20  # Ganti sesuai kebutuhan
top_features = feature_importances.argsort()[-top_n:][::-1]

# Menampilkan nama fitur dan tingkat kepentingannya
feature_names = TFidf_vector.get_feature_names_out()
for i in range(top_n):
    print(f"{i + 1}. Fitur: {feature_names[top_features[i]]}, Kepentingan: {feature_importances[top_features[i]]}")
```

```
1. Fitur: polri, Kepentingan: 0.09885504863419806
2. Fitur: mantap, Kepentingan: 0.06829421776401699
3. Fitur: avo, Kepentingan: 0.06714900275681684
4. Fitur: moga, Kepentingan: 0.06530838200544404
5. Fitur: polisi, Kepentingan: 0.06432901397931202
```

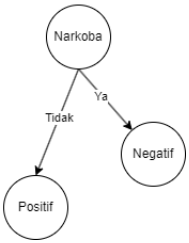
Gambar 4. 29 Tingkat Kepentingan Fitur *Random Forest*

Setelah kita mengetahui tingkat kepentingan fitur-fitur dalam memisahkan kelas target, kali ini peneliti akan memberikan contoh perhitungan manual pada fitur “narkoba” sebagai berikut:

Tabel 4. 5 Perhitungan Fitur Random Forest

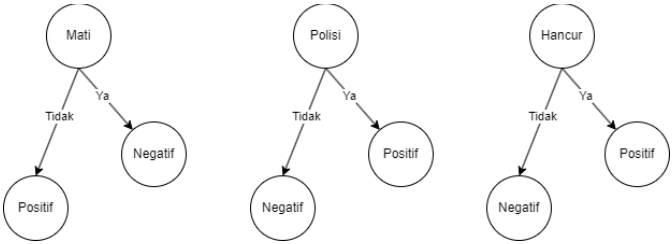
Komentar	Sentimen
Masukkan saja satu gedung bandar narkoba sama para koruptor baru di boom ini yang menghancurkan negara Dan generasi mudah Indonesia	Negatif
I love you narkoba	Positif
Susah payah bongkar gembong narkoba di uangkan Rp 10 triliun, hukuman mati termasuk anak buah	Negatif
Perusak generasi bangsa itu setelah di sita Aset para pengedar Narkoba ini hukum mati semuanya biar tak ada lagi bandar bandar besar ini	Negatif
Alhamdulillah pelaku narkoba itu jangan tanggung hukum gantung sampai mati	Negatif
Wahay para bndar2 narkoba tetaplah berjuang terus jangan pernah takut krena semua orang munafik	Positif

Tabel di atas menunjukkan sentimen negatif lebih dominan dari pada sentimen positif. Oleh karena itu, model *random forest* mempelajari bahwa teks yang memiliki kata 'narkoba' kemungkinan memiliki nilai sentimen negatif. Jika digambarkan dalam bentuk pohon keputusan maka akan berbentuk seperti gambar 4.30.



Gambar 4. 30 Contoh Pohon Keputusan

Random forest merupakan algoritma berupa Kumpulan pohon keputusan yang berisi fitur-fitur secara acak (*bagging*). Pada gambar 4.30 adalah contoh pembentukan 1 pohon. Pada gambar 4.31 berikut terdapat contoh penerapan klasifikasi *random forest* dari beberapa pohon keputusan sebagai berikut:



Gambar 4. 31 Contoh beberapa fitur pohon keputusan

Dari beberapa pohon tersebut, peneliti ambil contoh untuk pengaplikasian pada komentar "Kalau polisi tidak berani memutuskan hukuman mati maka akan lebih hancur negara ini." peneliti jabarkan pada tabel 4.6.

Tabel 4. 6 Perhitungan Fitur

Fitur	Jumlah	Sentimen
Mati	1	Negatif
Polisi	1	Positif
Hancur	1	Negatif

Pada komentar "Prestasi polri luar biasa ...kami bangga polri pemberantas narkoba" memiliki 1 fitur 'mati', yang berarti terdapat kata 'mati' dalam komentar tersebut yang berarti bernilai negatif. Kemudian memiliki 1 fitur 'polisi' yang berarti terdapat 1 kata 'polisi' dalam kalimat tersebut dan bernilai positif. Terdapat juga 1 fitur 'hancur' yang berarti terdapat kata 'hancur' dalam komentar tersebut dan bernilai negatif. Maka, dalam komentar tersebut terdapat 1 nilai positif dan 2 nilai negatif sehingga komentar tersebut bersentimen negatif.

Penelitian ini memanfaatkan bahasa pemrograman *python* sebagai alat bantu dalam meneliti kasus analisis sentimen ini. Termasuk perhitungan klasifikasi *random forest* yang memanfaatkan *library python* yaitu salah satunya *sklearn* dengan modul *RandomForestClassifier* yang berfungsi untuk melakukan klasifikasi *random forest*. Adapun *code* pada gambar 4.32 berikut:

```
from sklearn.ensemble import RandomForestClassifier

# Membuat dan melatih model Random Forest
random_forest_model = RandomForestClassifier(random_state=42)
random_forest_model.fit(Train_x_Tfidf.toarray(), y_train)

predicted_rf = random_forest_model.predict(Test_x_Tfidf.toarray())
print("Akurasi Random Forest:", accuracy_score(y_test, predicted_rf))
print("Presisi Random Forest:", precision_score(y_test, predicted_rf, average='weighted'))
print("Recall Random Forest:", recall_score(y_test, predicted_rf, average='weighted'))
print("F1 Score Random Forest:", f1_score(y_test, predicted_rf, average='weighted'))
```

```
Akurasi Random Forest: 0.832391713747646
Presisi Random Forest: 0.8161114600385952
Recall Random Forest: 0.832391713747646
F1 Score Random Forest: 0.8161043548575866
```

Gambar 4. 32 *Source Code* Klasifikasi *Random Forest*

Pada perhitungan klasifikasi menggunakan *code* tersebut hasil dari klasifikasi *random forest* untuk akurasi bernilai 0,8323, presisi bernilai 0,8161, *recall* bernilai 0,8323 dan F1 *score* bernilai 0,8161.

M. Ketepatan Klasifikasi

Ketepatan klasifikasi penting dalam analisis sentimen, yang mana pada tahap ini akan memunculkan hasil akhir berupa akurasi, presisi, *recall,* dan *f1 score*. Nilai-nilai tersebut merepresentasikan ketepatan model klasifikasi atau algoritma dalam menebak sentimen pada komentar. Pada data komentar youtube mengenai kasus narkoba di Indonesia, sentimen negatif lebih mendominasi dari pada sentimen positifnya, dimana sentimen negatif berjumlah 2083 dan sentimen positif berjumlah 568 hal ini dapat dilihat pada gambar 4.33.

Gambar 4. 33 Visualisasi jumlah sentimen

Ketepatan klasifikasi dalam menghitung nilai akurasi, presisi, *recall,* dan *f1 score* pada penelitian ini diwujudkan dengan *matrix confused* (matriks kebingungan). Dalam *matrix confused* terdapat dua kelas, yakni kelas aktual dan kelas prediksi yang dimana nantinya nilai dari *matrix confused* tersebut dapat digunakan untuk menghitung nilainya. Adapun untuk melihat *matrix confused* ada pada tabel 2.1 pada bab sebelumnya.

Visualisasi *matrix confused* pada penelitian ini dilakukan dengan menggunakan *library python* yakni *sklearn* dengan modul *confusion_matrix.* Adapun *code* untuk menampilkan *matrix confused naïve bayes* ada pada gambar 4.34 dan *matrix confused random forest* ada pada gambar 4.35.

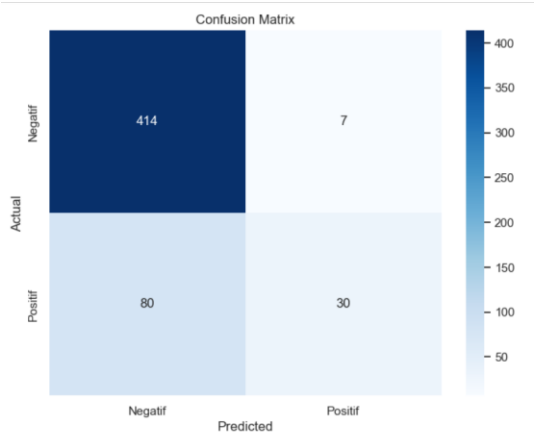
```
# Membuat prediksi menggunakan model
predicted = naive_bayes_model.predict(Test_x_Tfidf.toarray())

# Menghitung dan mencetak matriks kebingungan
cm = confusion_matrix(y_test, predicted)
print("Confusion Matrix Naive Bayes:")
print(cm)

# Menampilkan matriks kebingungan dalam bentuk plot
plt.figure(figsize=(8, 6))
classes = set(y_test)  # Menggunakan set untuk memastikan kelas unik
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=classes, yticklabels=classes)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Naive Bayes')
plt.show()
```

Gambar 4. 34 *Source Code Matrix Confused* NB

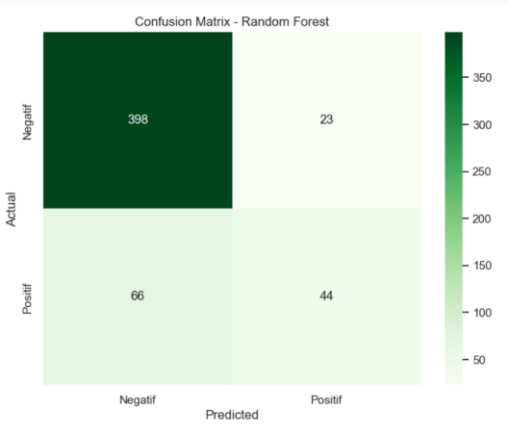
```python
# Membuat prediksi menggunakan model
predicted_rf = random_forest_model.predict(Test_x_Tfidf.toarray())

# Menghitung dan mencetak matriks kebingungan
cm_rf = confusion_matrix(y_test, predicted_rf)
print("Confusion Matrix Random Forest:")
print(cm_rf)

# Menampilkan matriks kebingungan dalam bentuk plot
plt.figure(figsize=(8, 6))
classes_rf = set(y_test)  # Menggunakan set untuk memastikan kelas unik
sns.heatmap(cm_rf, annot=True, fmt='d', cmap='Greens', xticklabels=classes_rf, yticklabels=classes_rf)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Random Forest')
plt.show()
```

Gambar 4. 35 *Source Code Matrix Confused* RF

Adapun hasil *matrix confused* untuk metode *naïve bayes* dan *random forest* adalah tidak sama. Adapun hasil tampilan *matrix confused naïve bayes* pada gambar 4.36 dan tampilan *matrix confused random forest* pada gambar 4.37.



Gambar 4. 36 *Matrix confused Naïve Bayes*



Gambar 4. 37 *Matrix Confused Random Forest*

Matrix confused tersebut diambil dari data uji dengan jumlah total data uji yakni 531 dari 2651 data komentar yang telah diolah, yang mana perbandingan data latih dan data uji adalah 80:20 diproses data uji sebanyak 531 dan sisanya adalah data latih.

Dalam *matrix confused* tersebut terdapat informasi yang digunakan untuk menghitung nilai akurasi, presisi, *recall* dan *f1 score*, untuk klasifikasi *naïve bayes* yakni 414 merupakan *true negative* dimana kelas aktual sama dengan

prediksi yakni negatif, 30 *true positive* dimana kelas aktual sama dengan predikasi yakni positif, 7 *false negative* dimana kelas aktual negatif namun diprediksi positif dan 80 *false positif* dimana kelas aktual positif namun diprediksi negatif. Sedangkan untuk klasifikasi *random forest* yakni 398 merupakan *true negative* dimana kelas aktual sama dengan prediksi yakni negatif, 44 *true positive* dimana kelas aktual sama dengan predikasi yakni positif, 23 *false negative* dimana kelas aktual negatif namun diprediksi positif dan 66 *false positif* dimana kelas aktual positif namun diprediksi negatif. Lebih jelasnya akan dijabarkan pada tabel 4.7.

Tabel 4. 7 Hasil *Matrix Confused*

Klasifikasi	TN	TP	FN	FP
Naïve Bayes	414	30	7	80
Random Forest	398	44	23	66

Langkah selanjutnya yakni menghitung nilai aktual, presisi, *recall* dan *f1 score* dari klasifikasi *naïve bayes* berikut:

$$A = \frac{TN+TP}{TN+TP+FN+FP} = \frac{414+30}{414+30+7+80} = \frac{444}{531} = 0{,}8361$$

$$P = \frac{TP}{TP+FN} = \frac{30}{30+7} = \frac{30}{37} = 0{,}8108$$

$$R = \frac{TN}{TN+FP} = \frac{414}{414+80} = \frac{414}{494} = 0{,}8380$$

$$f1 = 2x\frac{Precision\ .\ Recall}{Precision+Recall}\ x\ 100\%$$
$$= 2x\frac{0{,}81\ .\ 0{,}83}{0{,}81+0{,}83}x\ 100\% = 2x\frac{0{,}67}{1{,}64}x\ 100\%$$
$$= 0{,}8170$$

Kemudian menghitung nilai aktual, presisi, *recall* dan *f1 score* dari klasifikasi *random forest* berikut:

$$A = \frac{TN+TP}{TN+TP+FN+FP} = \frac{398+44}{394+44+23+66} = \frac{442}{531} = 0{,}8323$$

$$P = \frac{TP}{TP+FN} = \frac{44}{44+23} = \frac{44}{67} = 0{,}6567$$

$$R = \frac{TN}{TN+FP} = \frac{398}{398+66} = \frac{398}{464} = 0{,}8577$$

$$f1 = 2x\frac{Precision\ .\ Recall}{Precision+Recall}\ x\ 100\%$$
$$= 2x\frac{0{,}65\ .\ 0{,}85}{0{,}65+0{,}85}x\ 100\% = 2x\frac{0{,}55}{1{,}5}x\ 100\%$$
$$= 0{,}7366$$

Hasil perhitungan manual pada performa klasifikasi *naïve bayes* dan klasifikasi *random forest* dirangkum pada tabel 4.8.

Tabel 4. 8 Hasill performa klasifikasi

Klasifikasi	Akurasi	Presisi	*Recall*	*F1 Score*
Naïve Bayes	83%	81%	83%	81%
Random Forest	83%	65%	85%	73%

Akurasi adalah kemampuan untuk menghitung sejauh mana model klasifikasi memberikan prediksi positif dan negatif yang benar dibanding dengan semua data baik positif maupun negatif. Dari tabel 4.8 didapatkan jawaban hasil akurasi pada perhitungan algoritma *naïve bayes* dan *random forest* sebesar 83%. Namun, jika dilihat hasil dari perhitungan manual, algoritma

naïve bayes lebih unggul 0,0038 dari perhitungan akurasinya.

Presisi adalah kemampuan untuk menghitung sejauh mana model klasifikasi memberikan prediksi positif yang benar dibanding dengan total prediksi positif yang diberikan oleh model. Dari tabel 4.8 didapatkan jawaban hasil presisi pada perhitungan algoritma *naïve bayes* sebesar 81% dan presisi pada perhitungan menggunakan algoritma *random forest* juga sebesar 65%. Selisih hasil presisi *naïve bayes* dan *random forest* 16% *naïve bayes* lebih tinggi dari *random forest*. Sehingga dapat disimpulkan bahwa model klasifikasi *naïve bayes* lebih baik dari *random forest* pada nilai presisi untuk penelitian pada kasus ini.

Recall adalah kemampuan untuk menghitung sejauh mana model klasifikasi memberikan prediksi negatif yang benar dibanding dengan total prediksi negatif yang diberikan oleh model. Dari tabel 4.8 didapatkan jawaban hasil *recall* pada perhitungan algoritma *naïve bayes* sebesar 83% dan *recall* pada

perhitungan menggunakan algoritma *random forest* juga sebesar 85%. Selisih hasil *recall naïve bayes* dan *random forest* 2% *random forest* lebih tinggi dari *naïve bayes*. Sehingga dapat disimpulkan bahwa model klasifikasi *random forest* lebih baik dari *naïve bayes* pada nilai *recall* untuk penelitian pada kasus ini.

F1 score adalah gabungan dari presisi dan *recall* dimana berfungsi untuk menyeimbangkan nilai dari keduanya. Dari tabel 4.8 didapatkan jawaban hasil *f1 score* pada perhitungan algoritma *naïve bayes* sebesar 81% dan *f1 score* pada perhitungan menggunakan algoritma *random forest* juga sebesar 73%. Selisih hasil *f1 score naïve bayes* dan *random forest* 8% *naïve bayes* lebih tinggi dari *random forest*. Sehingga dapat disimpulkan bahwa model klasifikasi *naïve bayes* lebih baik dari *random forest* pada nilai *f1 score* untuk penelitian pada kasus ini.

Sebagaimana yang telah dijelaskan dalam latar belakang pada BAB I bahwa, *Naïve bayes* memiliki performasi yang baik pada kasus klasifikasi jenis *text.* Sedangkan algoritma *random*

forest memiliki performasi yang baik pada kasus klasifikasi kualitas kredit (Leonardo & Pratama, 2020), dan beberapa kajian penelitian terkait, dapat disimpulkan bahwa klasifikasi *naïve bayes* memiliki keunggulan dalam klasifikasi jenis teks, sedangkan *random forest* memiliki keunggulan dalam keterlibatan fitur-fitur yang ada dalam data. Sehingga pada kasus penelitian ini, *naïve bayes* lebih unggul dari *random forest.* Hal ini menunjukkan bahwa metode klasifikasi *naïve bayes* lebih sesuai dengan struktur dan sifat data pada komentar youtube yakni berupa teks.

Hasil perbandingan metode analisis sentimen tidak selalu bersifat mutlak, masing-masing tergantung dengan data yang diolah. Dalam kasus analisis sentimen kasus narkoba di Indonesia menggunakan komentar youtube, dengan membandingkan metode klasifikasi *naïve bayes* dan *random forest*, menunjukkan hasil akhir metode klasifikasi *naïve bayes* lebih baik dari pada metode klasifikasi *random forest.*

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Analisis sentimen menggunakan metode klasifikasi *naïve bayes* dan *random forest* untuk analisis sentimen pada kasus narkoba di Indonesia pada komentar youtube dapat dilakukan dengan baik. Dari data komentar awal berjumlah 2702 data, setelah dilakukan *text preprocessing* jumlah data menjadi 2651 data dengan jumlah sentiment 2083 sentimen negatif dan 568 sentimen positif.
2. Hasil nilai akhir pada analisis sentiment pada metode klasifikasi *naïve bayes* menghasilkan akurasi 83%, presisi 81%, *recall* 83% dan *f1 score* 81%. Sedangkan nilai akhir pada metode klasifikasi *random forest* dengan nilai akurasi 83%, presisi 65%, *recall* 85% dan *f1 score* 73%.

3. Hasil perbandingan kinerja klasifikasi *naïve bayes* dan klasifikasi *random forest* dalam analisis sentimen kasus narkoba di Indonesia pada komentar youtube pada perhitungan akurasi, keduanya memiliki performa yang sama-sama baik yakni 83% namun, akurasi *naïve bayes* memiliki nilai 0,0038 lebih unggul *random forest.* Kemudian, dalam perhitungan presisi metode klasifikasi *naïve bayes* menghasilkan nilai presisi 81% sedangkan *random forest* menghasilkan nilai presisi 65%, sehingga pada perhitungan presisi *naïve bayes* lebih unggul 16% dari *random forest.* Selanjutnya, pada perhitungan nilai *recall* pada klasifikasi *naïve bayes* sebesar 83%, sedangkan klasifikasi *random forest* sebesar 85%, sehingga pada perhitungan *recall, random forest* lebih unggul 2% dari *naïve bayes.* Lalu, pada perhitungan *f1 score* hasil dari klasifikasi *naïve bayes* adalah 81%, sedangkan klasifikasi *random forest* 73%, sehingga pada perhitungan *f1 score, naïve*

bayes lebih unggul 8% dari *random forest.* Sehingga dapat disimpulkan secara keseluruhan, metode klasifikasi *naïve bayes* lebih baik dari pada metode klasifikasi *random forest* pada kasus ini.

B. Saran

Setelah dilakukannya penelitian, saran dari peneliti untuk dapat mengembangkan penelitian serupa terdapat beberapa saran diantaranya:

1. Pada penelitian ini menggunakan perbandingan metode klasifikasi *naïve bayes* dan *random forest* dalam melakukan analisis sentiment. Penelitian selanjutnya mungkin dapat membandingkan dengan metode klasifikasi yang lain seperti membandingkan metode *Support Vector Machine* dengan *Naïve Bayes* ataupun perbandingan metode klasifikasi yang lain.
2. Membandingkan klasifikasi *naïve bayes* dan *random forest* pada tugas yang lain selain analisis sentimen, mungkin klasifikasi penyakit, klasifikasi gambar ataupun yang lain.

3. Mengambil komentar dari beberapa video youtube tidak hanya pada satu video apabila ingin mengembangkan penelitian den.
4. Menemukan cara atau langkah yang lebih cepat dan efektif dalam sentimenisasi data atau opini yang menggunakan Bahasa Indonesia.
5. Mengembangkan tahap normalisasi yang lebih baik sehingga hasil *text preprocessing* lebih optimal.

DAFTAR PUSTAKA

Adhi Putra, A. D. (2021). Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *8*(2), 636–646. https://doi.org/10.35957/jatisi.v8i2.962

Akbar, M. N., Darmatasia, D., Mustikasari, M., & Syahwal, M. (2021). Analisis Clustering Teks Tanggapan Masyarakat Di Twitter Terhadap Pembatasan Sosial Berskala Besar Menggunakan Algoritma K-Means. *Jurnal INSYPRO (Information System and Processing)*, *6*(1), 1–9. https://doi.org/10.24252/insypro.v6i1.23325

Alita, D., & Isnain, A. R. (2020). Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier. *Jurnal Komputasi*, *8*(2), 50–58. https://doi.org/10.23960/komputasi.v8i2.2615

Amin, F., & Alfa Razaq, J. (2018). Implementasi Stemmer Bahasa Jawa dengan Metode Rule Base Approach pada Sistem Temu Kembali Informasi Dokumen Teks Berbahasa Jawa. *Prosiding SENDI_U*, 199–206.

Apriliah, W., Kurniawan, I., Baydhowi, M., & Haryati, T. (2021). SISTEMASI: Jurnal Sistem Informasi Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest. *Jurnal Sistem Informasi*, *10*(1), 163–171. http://sistemasi.ftik.unisi.ac.id

Arini, A.-, Wardhani, L. K., & Octaviano, D.-. (2020). Perbandingan Seleksi Fitur Term Frequency & Tri-Gram Character Menggunakan Algoritma Naïve Bayes Classifier (Nbc) Pada Tweet Hashtag #2019gantipresiden. *Kilat*, *9*(1), 103–114. https://doi.org/10.33322/kilat.v9i1.878

Awangga, R. M., & Khonsa', N. H. (2022). Analisis Performa Algoritma Random Forest dan Naive Bayes Multinomial pada Dataset Ulasan Obat dan Ulasan Film. *InComTech : Jurnal Telekomunikasi Dan Komputer*, *12*(1), 60. https://doi.org/10.22441/incomtech.v12i1.14770

Bustami. (2014). Penerapan Algoritma Naive Bayes. *Jurnal Informatika*, *8*(1), 884–898.

Cucus, A., Endra, R. Y., & Naralita, T. (2019). Chatter Bot Untuk Konsultasi Akademik Di Perguruan Tinggi. *Explore: Jurnal Sistem Informasi Dan Telematika*, *10*(1). https://doi.org/10.36448/jsit.v10i1.1214

Dale, R. (2010). Classical approaches to natural language processing. In *Handbook of Natural Language Processing, Second Edition*.

Darwis, A., Dalimunthe, G. I., & Riadi, S. (2018). Narkoba, Bahaya Dan Cara Mengantisipasinya. *Amaliah: Jurnal Pengabdian Kepada Masyarakat*, *1*(1), 36–45. https://doi.org/10.32696/ajpkm.v1i1.14

Engelhart, M. D., & Moughamian, H. (1968). Book Reviews : Book Reviews. *Educational and Psychological Measurement*, *28*(2), 619–620. https://doi.org/10.1177/001316446802800256

Fatmawati, M. (2017). *Pengklasteran Laporan Tugas Akhir Berdasarkan Abstrak Menggunakan Metode Rapid Automatic Keyphrase Extraction Dan Average Linkage Hierarchical Clustering*.

Fitri, E. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. *Jurnal Transformatika*, *18*(1), 71. https://doi.org/10.26623/transformatika.v18i1.2317

Furqan, M., & Shidqi, M. N. (2023). Chatbot Telegram

Menggunakan Natural Language Processing. *Walisongo Journal of Information Technology*, *5*(1), 15–26. https://doi.org/https://doi.org/10.21580/wjit.2023.5.1.14793

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. 1–17. http://arxiv.org/abs/2008.05756

Gridin, I. (2022). Hyperparameter Optimization. In *Automated Deep Learning Using Neural Network Intelligence*. https://doi.org/10.1007/978-1-4842-8149-9_2

Hariyanto, B. P. (2018). Pencegahan Dan Pemberantasan Peredaran Narkoba Di Indonesia. *Jurnal Daulat Hukum*, *1*(1), 201–210. https://doi.org/10.30659/jdh.v1i1.2634

Imron, A. (2019). *Kabupaten Rembang Menggunakan Metode Naive Bayes Classifier*.

Joseph, V. R., & Vakayil, A. (2022). SPlit: An Optimal Method for Data Splitting. *Technometrics*, *64*(2), 166–176. https://doi.org/10.1080/00401706.2021.1921037

Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, *13*(3), 144–168. https://doi.org/10.15849/ijasca.211128.11

Leonardo, R., & Pratama, J. (2020). Perbandingan Metode Random Forest Dan NaïveBayes Dalam Prediksi Keberhasilan Klien Telemarketing. *Jurnal Penelitian Teknik Informatika*, *3*(123), 455–459.

Lukman, G. A., Alifah, A. P., Divarianti, A., & Humaedi, S. (2022). Kasus Narkoba Di Indonesia Dan Upaya Pencegahannya Di Kalangan Remaja. *Jurnal Penelitian*

Dan Pengabdian Kepada Masyarakat (JPPM), *2*(3), 405. https://doi.org/10.24198/jppm.v2i3.36796

Mahfudh, A. A., & Mustofa, H. (2019). Klasifikasi Pemahaman Santri Dalam Pembelajaran Kitab Kuning Menggunakan Algoritma Naive Bayes Berbasis Forward Selection. *Walisongo Journal of Information Technology*, *1*(2), 101. https://doi.org/10.21580/wjit.2019.1.2.4529

MerdekaDotCom, Y. (2023). *BNN blak-blakan 10 daerah "zona merah" narkoba di Indonesia , pakai penanganan luar biasa*. https://youtu.be/tcDWGsH01QE?si=wAXqO5YP9Fmwrk-f

Miftahusalam, A., Pratiwi, H., Slamet, I., Statistika, P. S., & Maret, U. S. (2023). *Perbandingan Metode Random Forest dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA Mobile*. 1–8.

Momole, G. M. (2022). Perbandingan Naïve Bayes dan Random Forest Dalam Klasifikasi Bahasa Daerah. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(2), 855–863. https://doi.org/10.35957/jatisi.v9i2.1857

Perkembangan, S. D. A. N., & Masyarakat, I. K. (2013). Sejarah dan perkembangan. *Ilmuti.Org*, *X*, 13–19. https://doi.org/http://www.sejarahkita.web.id/2013/01/sejarah-dan-perkembangan-microsoft.html

Purwati, K. Y. (2023). *Analisis Sentimen Berita Vaksin Covid-19 dengan Rebustly Optimized Bert Pre-training Approach (Roberta)* [UIN Syarif Hidayatullah]. https://repository.uinjkt.ac.id/dspace/handle/123456789/73319

Ratniasih, N. L., Larasati, I., Putri, N., & Kepentingan, K. P. (2023). *ANALISIS SENTIMEN KEPUASAN PEMANGKU KEPENTINGAN MENGGUNAKAN METODE NAÏVE*

BAYES CLASSIFIER DAN K-NEAREST. *2*, 103–109.

Rifano, E. J., Fauzan, A. C., Makhi, A., Nadya, E., Nasikin, Z., & Putra, F. N. (2020). Text Summarization Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, *2*(1), 8–17. https://doi.org/10.28926/ilkomnika.v2i1.32

Rosid, M. A., Fitrani, A. S., Astutik, I. R. I., Mulloh, N. I., & Gozali, H. A. (2020). Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi. *IOP Conference Series: Materials Science and Engineering*, *874*(1). https://doi.org/10.1088/1757-899X/874/1/012017

Sarjana, P. S., Statistika, D., Matematika, F., Ilmu, D. A. N., & Alam, P. (2017). *291465666*.

Sukabumi, Y. K. T. (2023). *Jaringan Narkoba Terbesar di Indonesia Fredy Pratama Terbongkar Laporan Khusus*. https://www.youtube.com/watch?v=6MQJjXgBrew

Syah, H., & Witanti, A. (2022). Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (Svm). *Jurnal Sistem Informasi Dan Informatika (Simika)*, *5*(1), 59–67. https://doi.org/10.47080/simika.v5i1.1411

Turmudi Zy, A., Adji Ardiansyah, L., & Maulana, D. (2021). Implementasi Algoritma Naïve Bayes Dalam Mendiagnosa Penyakit Angin Duduk. *Jurnal Pelita Teknologi*, *16*(1), 52–65.

Wasono, R. (2022). *Bayes Untuk Klasifikasi*.

Witten, I. H. (2004). Text mining. *The Practical Handbook of Internet Computing*, 14-1-14–22. https://doi.org/10.1201/9780203507223

Yasar, A., & Saritas, M. M. (2019). Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification. *International Journal of Intelligent Systems and Applications in Engineering*, *7*(2), 88–91. https://doi.org/10.18201/ijisae.2019252786

Yoga Saputra, P., Hanifudin Subhi, D., & Zain Afif Winatama, Z. (2019). Implementasi Sentimen Analisis Komentar Channel Video Pelayanan Pemerintah Di Youtube Menggunakan Algoritma Naïve Bayes. *Jurnal Informatika Polinema*, *5*(3), 209–213. http://jip.polinema.ac.id/ojs3/index.php/jip/article/view/259

Yuniarti, W. D., Faiz, A. N., & Setiawan, B. (2020). Identifikasi Potensi Keberhasilan Studi Menggunakan Naïve Bayes Classifier. *Walisongo Journal of Information Technology*, *2*(1), 1. https://doi.org/10.21580/wjit.2020.2.1.5204

Zidan, M. (2022). Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter Di Indonesia Menggunakan Metode Naive Bayes. In *Skripsi. Semarang: UIN Walisongo*.

DAFTAR LAMPIRAN

LAMPIRAN 1 : Contoh Dokumen Hasil Crawling

Data Komentar YouTube

No	Komentar YouTube
1	penjahat itu jangan dikasih kesempatan
2	pasti ada dekingannya pertanyaannya klu tdk ada dekingannya knp barang haram itu bisa masuk ke nkri seandainya dijaga ketat di jln tikus di daerah daerah manapun seperti sarawak tentu barang haram itu tdk bisa masuk ke nkridan harus jaga diperketat di beacukai dan cargo cargo
3	sungguh hebat jenderal kapolri ri mas sigit selama indonesia merdeka
4	kurang banyak
5	biadap kubu biarin miskin
6	musnahkan langsung jangan bersisa sebutirpun
7	masukkan saja satu gedung bandar narkoba sama para koruptor baru di boom ini yang menghancurkan negara dan generasi mudah Indonesia
8	10,2 ton mah tumbal yang lolos 90 tonnya
9	bravo polisi indonesia hukum harus aduk koruptor sama narkoba bikin hancur satu negara harus di basmi klo ngga bangsa indonesia tinggal tunggu kehancurannya penegak hukum harus tegas dong
10	ada apa dengan nama nama fredy dari mulai budiman sambo dan skrg pratama sampe kepikiran punya anak mau di namain fredy escobar
....	
2702	ini terjadi semakin marak dan banyak karena hukum mati abal abal negara ini

LAMPIRAN 2 : Contoh Dokumen yang Sudah Diberi Label

No	Komentar YouTube	Sentimen
1	penjahat itu jangan dikasih kesempatan	Negatif
2	pasti ada dekingannya pertanyaannya klu tdk ada dekingannya knp barang haram itu bisa masuk ke nkri seandainya dijaga ketat di jln tikus di daerah daerah manapun seperti sarawak tentu barang haram itu tdk bisa masuk ke nkridan harus jaga diperketat di beacukai dan cargo cargo	Negatif
3	sungguh hebat jenderal kapolri ri mas sigit selama indonesia merdeka	Positif
4	kurang banyak	Negatif
5	biadap kubu biarin miskin	Negatif
6	musnahkan langsung jangan bersisa sebutirpun	Negatif
7	masukkan saja satu gedung bandar narkoba sama para koruptor baru di boom ini yang menghancurkan negara dan generasi mudah Indonesia	Negatif
8	10,2 ton mah tumbal yang lolos 90 tonnya	Negatif
9	bravo polisi indonesia hukum harus aduk koruptor sama narkoba bikin hancur satu negara harus di basmi klo ngga bangsa indonesia tinggal tunggu kehancurannya penegak hukum harus tegas dong	Positif
10	ada apa dengan nama nama fredy dari mulai budiman sambo dan skrg pratama sampe kepikiran punya anak mau di namain fredy escobar	Negatif
....		
2702	ini terjadi semakin marak dan banyak karena hukum mati abal abal negara ini	Negatif

LAMPIRAN 3 : Data Pendukung Proses *Normalization*

"colloquial-indonesian-lexicon"

No	Slang	Formal
1	woww	wow
2	aminn	amin
3	met	selamat
4	netaas	menetas
5	keberpa	keberapa
6	eeeehhhh	eh
7	Kata2nyaaa	Kata-katanya
8	hallo	halo
9	kaka	kakak
10	ka	kak
11	daah	dah
12	aaaaahhhh	ah
13	yaa	ya
14	smga	semoga
15	slalu	selalu
16	amiin	amin
17	kk	kakak
18	trus	terus
19	kk	kakak
20	sii	sih
....		
15397	gaharus	enggak harus

LAMPIRAN 4 : Data Pendukung Proses *Normalization*

"kbba"

No	Slang	Formal
1	7an	tujuan
2	@	di
3	ababil	abg labil
4	abis	habis
5	acc	accord
6	ad	ada
7	adlah	adalah
8	adlh	adalah
9	adoh	aduh
10	afaik	as far as i know
11	aha	tertawa
12	ahaha	haha
13	aing	saya
14	aj	saja
15	aja	saja
16	ajep-ajep	dunia gemerlap
17	ajj	saja
18	ak	saya
19	aka	dikenal juga sebagai
20	akika	aku
....		
1319	mgkin	mungkin

LAMPIRAN 5 : Data Pendukung Proses *Normalization*

"Slangword"

No	Slang	Formal
1	&	dan
2	+	tambah
3	/	atau
4	=	sama dengan
5	ababil	anak ingusan
6	abal2	palsu
7	abal	palsu
8	ad	ada
9	akooh	aku
10	alay	norak
11	albm	album
12	ampe	sampai
13	anjir	waw
14	anyway	ngomong-ngomong
15	aq	aku
16	asap	secepatnya
17	ato	atau
18	atw	atau
19	ava	foto profil
20	baget	keras kepala
....		
286	yg	yang

LAMPIRAN 6 : Data Pendukung Proses *Normalization*

"FormalizationDict"

No	Slang	Formal
1	7an	tujuan
2	@	di
3	ababil	abg labil
4	abis	habis
5	acc	accord
6	ad	ada
7	adlah	adalah
8	adlh	adalah
9	adoh	aduh
10	afaik	as far as i know
11	aha	tertawa
12	ahaha	haha
13	aing	saya
14	aj	saja
15	aja	saja
16	ajep-ajep	dunia gemerlap
17	ajj	saja
18	ak	saya
19	aka	dikenal juga sebagai
20	akika	aku
....		
1294	irit	hemat

LAMPIRAN 7 : Data Pendukung Proses *Normalization*

"Typo"

No	Slang	Formal
1	jng	jangan
2	biadap	biadab
3	ama	sama
4	kmana	kemana
5	exekusi	eksekusi
6	nakoba	narkoba
7	kalo	kalau
8	jngn	jangan
9	cidrai	ciderai
10	brantaskan	berantaskan
11	lgi	lagi
12	tingi	tinggi
13	bb	barangbukti
14	merintaj	pemerintah
15	bnyak	banyak
16	bagget	banget
17	j	saja
18	lom	belum
19	aja	saja
20	b	biasa
....		
979	dpm	dalam