TINJAUAN KOMENTAR PENGGUNA SIREKAP DI PLAY STORE: STUDI KOMPARASI MENGGUNAKAN NAIVE BAYES DAN SUPPORT VECTOR MACHINE

TUGAS AKHIR

Diajukan untuk Memenuhi Syarat Guna Memperoleh Gelar Sarjana Program Strata (S. 1) dalam Ilmu Teknologi Informasi

Diajukan oleh:

Istna A'la Mahmudah

NIM: 2108096030

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TENOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO

SEMARANG

2025

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Istna A'la Mahmudah

NIM : 2108096030

Prodi : Teknologi Informasi

Menyatakan bahwa tugas akhir yang berjudul :

TINJAUAN KOMENTAR PENGGUNA SIREKAP DI PLAY STORE: STUDI KOMPARASI MENGGUNAKAN NAIVE BAYES DAN SUPPORT VECTOR MACHINE

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

Semarang, 16 Juni 2025

Pembuat Pernyataan,

Istna A'la Mahmudah

NIM. 210809030

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

PENGESAHAN

Naskah skripsi berikut ini :

Judul	:Tinjauan Komentar Pengguna Sirekap Di Play Store: Studi Komparasi Menggunakan Naive Bayes dan Support Vectore Machine
Penulis	: Istna A'la Mahmudah
NIM	: 2108096030
Jurusan	: Teknologi Informasi

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

Semarang, 16 Juni 2025

DEWAN PENGUJI

Penguji I,

Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

Penguji II,

Mokhamad Iqbal Mustofa, M.Kom
NIP. 198808072019031010

Penguji III,

Dr. Masy Ari Ulinuha, ST., M.T
NIP. 198108122011011007

Penguji IV,

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

Pembimbing I,

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

Pembimbing II,

Mokhamad Iqbal Mustofa M.Kom
NIP. 1988080701931010

NOTA PEMBIMBING

Semarang, 28 April 2025

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	: Tinjauan Komentar Pengguna Sirekap Di Play Store Studi Komparasi Menggunakan Naive Bayes dan Support Vectore Machine
Nama	: Istna A'la Mahmudah
NIM	: 2108096030
Jurusan	: Teknologi Informasi

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

Wasaalamu'alaikum. Wr. Wb.

Pembimbing I,

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

NOTA PEMBIMBING

Semarang, 28 April 2025

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	: Tinjauan Komentar Pengguna Sirekap Di Play Store Studi Komparasi Menggunakan Naive Bayes dan Support Vectore Machine
Nama	: Istna A'la Mahmudah
NIM	: 2108096030
Jurusan	: Teknologi Informasi

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

Wasaalamu'alaikum. Wr. Wb.

Pembimbing II,

Mokhamad Iklil Mustofa M.Kom
NIP. 1988080701931010

ABSTRAK

Pesatnya perkembangan teknologi di Indonesia mendorong transformasi layanan publik melalui e-Government, salah satunya melalui aplikasi SIREKAP yang digunakan dalam proses rekapitulasi suara Pemilu 2024. Aplikasi ini mendapat berbagai tanggapan dari masyarakat, sehingga diperlukan analisis sentimen terhadap komentar pengguna di *Google Play Store.*

Proses awal pada penelitian ini melakukan pengambilan data komentar google play store dengan ID link dari aplikasi SIREKAP. Selanjutnya tanggapan tersebut dilakukan proses text prepocessing dan pembobotan kata TFIDF (Term Frequency Inverse Documnet Frequency). Setelah itu dilakukan klasifikasi menggunakan metode naive bayes dan support Vector machine Penelitian ini bertujuan untuk membandingkan performa dua algoritma, yaitu *naïve bayes* dan *support vector machine* (SVM), dalam mengklasifikasikan sentimen positif dan negatif. Proses analisis dilakukan untuk memastikan keakuratan informasi publik. Data yang diperoleh berasal dari *play store*, serta data yang diproses sebanyak 1116, yang mana sentimen positif sejumlah 558 dan sentimen negatif sejumlah 558.

Hasil analisis menunjukkan bahwa metode *naïve bayes* menghasilkan akurasi 69%, presisi 69%, recall 70%, dan f1-score 69%, sedangkan SVM menghasilkan akurasi 70%, presisi 70%, recall 70%, dan f1-score 70%. Dengan hasil evaluasi model, SVM menunjukkan performa lebih baik dalam analisis sentimen komentar pengguna. Hasil penelitian ini diharapkan dapat memberikan masukan bagi pengembangan aplikasi SIREKAP dan pengambilan keputusan berbasis data.

Kata Kunci: SIREKAP, Analisis Sentimen, Naive Bayes, Support Vector Machine

KATA PENGANTAR

Alhamdulillah, Puji Syukur atas kehadirat Allah SWT atas segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Tinjauan Komentar Pengguna Sirekap Di Play Store: Studi Komparasi Menggunakan Naive Bayes dan Support Vector Machine" dengan baik. Shalawat dan salam saya haturkan kepada baginda Rasulullah Muhammad *shallallahu 'alaihi wa sallam,* yang dengan kehadirannya, turun Rahmat dari Allah untuk semesta alam.

Dengan tujuan dibuat skripsi ini sebagai salah satu bentuk syarat kelulusan pada program sarjana (S1) prodi teknologi informasi di Universitas Islam Negeri Walisongo Kota 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 kesempatan ini penulis ingin mengucapkan banyak-banyak terimakasih kepada seluruh pihak yang memberi dukungan dan bantuan dari pelaksanaan skripsi hingga penyelesaian skripsi ini. Penulis mengakui bahwa apabila tanpa dibimbing, diberi arahan, serta dibina dan tanpa diberikan motivasi dari seluruh pihak, maka penulisan skripsi

ini tidak akan berjalan dengan baik. Maka dari itu penulis mengucapakan terimakasih yang tak terhingga kepada :

1. Allah SWT, atas segala rahmat-Nya dan kasih-Nya yang selalu mendampingi dari awal kuliah hingga saat ini dalam menyelesaikan skripsi.
2. Kedua Orang tua tercinta dan keluarga yang selalu menemani dalam membantu penulis, memotivasi dengan cara tersendiri, selalu mendo'akan serta memberikan dukungan baik kepada penulis.
3. Bapak Prof. Dr. H. Nizar, M.Ag, selaku Rektor Universitas Islam Negeri Walisongo Semarang.
4. Bapak Prof. Dr. H. Musahadi, M.Ag, selaku Dekan Fakultas Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
5. Bapak Khotibul Umam, ST., M.Kom, selaku ketua program studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.
6. Ibu Dr. Wenty Dwi Yuniarti, S.Pd., M.Kom dan Bapak Mokhamad Iklil Mustofa M.Kom, selaku dosen pembimbing skripsi saya yang selalu memberikan dukungan, arahan, bimbingan serta motivasi dalam pelaksanaan skripsi hingga pembuatan skripsi ini.
7. Seluruh dosen program studi Teknologi Informasi khususnya, serta Staff, karyawan dan dosen di

lingkungan Universitas Islam Negeri Walisongo Semarang.

8. Cemara AL – barokah, Umi, Aulia, Siti, yang sudah menemani selama penulis mengerjakan, tempat berbagi keluh kesah dan teman berkaryawisata penulis, jika penulis sudah bosan dalam mengerjakan.
9. Teman-teman yang selalu memberi dukungan dan motivasi, baik dari Teknologi Informasi, Saintek Sport, maupun teman dari luar kampus.
10. Semua pihak yang tidak bisa penulis sebutkan satu persatu yang terlibat dalam pembuatan skripsi ini sehingga dapat terselesaikan dengan baik.

Dalam pelaksanaan dan penyusunan skripsi, penulis menyadari bahwa tentunya masih jauh dari kata sempurna dan masih banyak kekurangan. Untuk itu, penulis sangat mengharapkan kritik serta saran yang membangun demi kesempurnaan penulisan skripsi ini, dan semoga skripsi ini dapat bermanfaat untuk semua pihak.

Semarang, 16 Juni 2025

penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

A. Latar Belakang

Pesatnya perkembangan teknologi di Indonesia menjadikan banyak perubahan di hampir seluruh aspek kehidupan. Diantaranya terjadi pada *e-government* atau pemerintahan elektronik (Artikel & Info, 2023). Dalam kasus Indonesia, e-Government telah menjadi salah satu solusi terbesar menghubungkan dan menjembatani antara pemerintah dan masyarakat dalam lingkup pelayanan publik. Penerapan e-Government di negara-negara berkembang, seperti Indonesia, mempunyai beberapa tantangan (Pokhrel, 2024).

Beberapa tantangan tersebut mencakup kesenjangan digital yang tinggi, infrastruktur elektronik yang tidak memadai dan kurangnya keterampilan dan kompetensi untuk desain, implementasi, penggunaan, dan pengelolaan sistem *e-Government*. Tentu saja *e– Government* ini berwujudkan aplikasi *mobile* yang mengikuti perkembangan era digital ini. Maka dari itu dengan adanya perkembangan teknologi ini penting untuk mengetahui seberapa puas dan tidak puas pengguna dengan aplikasi yang mereka gunakan. Google memiliki layanan yang disebut play

store, dimana berbagai jenis konten digital, termasuk game, aplikasi, film, musik dan buku, tersedia. Rating dan komentar merupakan fitur dari play store, di mana pengguna yang menggunakan produk di play store dapat menulis ulasan tentang produk yang telah mereka gunakan (Nurian et al., 2023). Komentar dari pengguna sering digunakan sebagai alat yang efektif dan efisien dalam menemukan informasi terhadap suatu produk atau jasa.

Bahwasannya penelitian yang sudah – sudah ini menemukan hampir 50 % dari pengguna internet bergantung pada rekomendasi word-of-mouth (opini) sebelum menggunakan produk, karena review merupakan salah satu bagian penting sebelum menggunakan produk tersebut (Nurjanah et al., 2023). Karena banyaknya konten digital (aplikasi) di play store yang dapat diakses, pada topik penelitian ini aplikasi yang menjadi studi penulis adalah aplikasi SIREKAP. Aplikasi yang digunakan sarana publikasi hasil perhitungan suara dan melakukan proses rekapitulasi hasil perhitungan suara dan juga alat pembantu anda pada saat pemilu tahun 2024 ini (Margokatonsid.slemankab.go.id, 2024). Aplikasi ini tentunya banyak mendapat sorotan dari berbagai belah

pihak terutama petugas pemilu dari tingkat rendah ketingkat yang lebih tinggi. dikarenakan penggunaan SIREKAP merupakan langkah penting untuk memantau perkembangan proses input suara dari setiap Tempat Pemungutan Suara (TPS), guna memastikan akurasi hasil dan mencegah penyimpangan (Natalianus et al., 2024).

Namun implementasi Sirekap tidak lepas dari tantangan. Penumpukan data yang belum diverifikasi menyebabkan keterlambatan dalam menampilkan hasil pemilu, sehingga KPU terpaksa menutup sementara diagram hasil Pemilu (Paris, 2024). Keterlambatan ini menunjukkan bahwa fitur-fitur dalam Sirekap belum dioptimalkan sepenuhnya. Oleh karena itu, diperlukan analisis terhadap saran, kritik, dan keluhan terkait aplikasi Sirekap untuk mencari solusi yang efektif. Pada penelitian ini penulis melihat komentar pengguna berdasarkan data yang ditemukan pada situs play store.

komponen evaluasi yang dipertimbangkan, yaitu komentar yang lebih tekstual berbicara tentang pendapat yang lebih mendalam (Insan et al., 2023). Akibatnya, dibutuhkan analisis sentimen yang merupakan proses otomatisasi pemrosesan data teks

untuk menentukan apakah pendapat yang terkandung dalam sebuah kalimat bersifat positif atau negatif (Kurniawan et al., 2019).

Analisis sentimen juga merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat (Buntoro, 2017). Permasalahan dalam pendapat pengguna aplikasi SIREKAP ini juga perlu diperhatikan dalam sudut pandang agama islam yang mana Allah berfirman dalam Al – Qur'an surah Al – Hujurat ayar 6:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ

Artinya : "Wahai orang-orang yang beriman, jika seorang fasik datang kepadamu membawa berita penting, maka telitilah kebenarannya agar kamu tidak mencelakakan suatu kaum karena ketidaktahuan(-mu) yang berakibat kamu menyesali perbuatanmu itu." (QS. Al-Hujurat 49: Ayat 6).

Dari ayat tersebut menjelaskan bahwa jika kita memperoleh suatu opini dari seseorang yang belum diketahui kejelasannya maka diharuskan melakukan *tabayyun*. Apabila kita tidak melakukan tabayyun maka dampak buruk akan kita rasakan sendiri juga orang

lain. Dari penjelasan ayat tersebut penulis melakukan penelitian ini sebagai salah satu usaha tabayyun dengan mengumpulkan berbagai macam komentar dari pengguna lalu menganalisisnya sehingga nantinya mendapatkan hasil yang berguna bagi pengguna lain.

Dikarenakan penulis menggunakan dua metode sebagai bahan dalam melakukan analisis sentimen aplikasi SIREKAP yaitu Algoritma *Naïve Bayes* dan *Support Vector Machine*. Didukung dengan studi sebelumnya yang menunjukkan bahwa algoritma *Naïve Bayes* dan SVM memiliki tingkat performa yang tinggi dalam analisis sentimen, meskipun hasil performa bisa bervariasi tergantung pada konteks dan metode yang digunakan (Syahputra et al., 2022). Oleh karena itu, penelitian ini akan membandingkan peforma algoritma *Naïve Bayes* dan SVM dalam analisis sentimen terhadap aplikasi SIREKAP Pemilu, untuk menentukan algoritma mana yang paling efektif

Setelah dilakukannya klasifikasi, dilakukan pencarian informasi dari klasifikasi sentimen kelas positif dan negatif. Tahapan analisis kemudian ditinjau dengan cara statistik deskriptif dan kata-kata untuk menemukan wawasan tentang aplikasi SIREKAP. Analisis membantu mengidentifikasi kekuatan dan

kelemahan aplikasi SIREKAP. Hasil analisis penelitian ini perlu mendukung pengolahan data uji semaksimal mungkin agar dapat memberikan informasi yang selengkap mungkin. Serta nantinya dapat bermanfaat bagi pemangku kepentingan yang memerlukan.

B. Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah yang dapat diangkat dalam penelitian ini yaitu:

1. Bagaimana klasifikasi tinjauan komentar aplikasi SIREKAP menggunakan metode *Naive Bayes* dan *Support Vector Machine*?
2. Bagaimana performa metode *Naive Bayes* dan *Support Vector Machine* dalam mengklasifikasi tinjauan komentar aplikasi SIREKAP?

C. Tujuan Penelitian

Tujuan dari penelitian ini yaitu :

1. Mengetahui cara pengklasifikasi tinjauan komentar aplikasi SIREKAP menggunakan metode *Naive Bayes* dan *Support Vector Machine*.
2. Mengetahui performa klasifiaksi yang dihasilkan dari kedua metode. yaitu, *Naive Bayes* dan *Support Vector Machine*.

D. Batasan Masalah

Agar penelitian dapat dilakukan secara objektif dan jelas, maka peneliti menerapkan batasan masalah yang diperlukan. Berikut batasan masalah pada penelitian ini, adalah:

1. Komentar pada *Play Store* akan diklasifikasikan menjadi dua sentimen yaitu sentimen positif dan negatif.
2. Data yang digunakan adalah data dari komentar yang ada di *Play store.*

E. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah :

1. Manfaat Akademis

   Penelitian ini dapat menambah studi literatur dibidang teknologi informasi khususnya dalam bidang pengetahuan klasifikasi machine learning dalam metode *Naive Bayes* dan *Support Vector Machine*.

2. Manfaat Praktis

   Penelitian ini diharapkan dapat bermanfaat bagi pengembang aplikasi SIREKAP untuk memotivasi dalam lebih mengupdate aplikasi tersebut.

F. Sistematika Penelitian

Sistematika penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan memuat latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penelitian.

BAB II LANDASAN TEORI

Bab ini memuat penelitian – penelitian yang terdahulu dan memuat penjelasan terkait landasan teori pada penelitian seperti analisis sentimen, python, aplikasi SIREKAP, *text preprocessing*, metode *Naive Bayes*, metode SVM, dan evaluasi model.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan, yaitu metode pengumpulan data, perangkat penelitian, alur penelitian serta ada uraian metodologi sebagai penjelas dari alur penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tantang hasil penelitian yang dilakukan berupa tinjauan komentar terhadap aplikasi SIREKAP berdasarkan penelitian yang sudah dilakukang dengan metode yang ditentukan.

BAB V KESIMPULAN

Bab ini berisi kesimpulan dari hasil penelitian yang diperoleh serta berisi saran untuk melakukan pengembangan penelitian selanjutnya.

BAB II
LANDASAN PUSTAKA

A. Landasan Teori

1. Analisis Sentimen

Analisis sentimen merupakan salah satu contoh dari bidang *Natural Language Processing* (NLP) yang paling populer. NLP merupakan bidang ilmiah yang membahas tentang bagaimana caranya agar komputer bisa bekerja dan berpikir seperti manusia, NLP juga bagian dari dari *Artificial Intelligence* (AI) merupakan salah satu dari empat cabang ilmu data mining, yaitu statistika, database, dan pencarian informasi.

Serta AI memerlukan *machine learning* sebagai alogaritma penyelesaian. *Machine learning* tidak mempunyai perasaan seperti manusia sehingga keputusan yang diambil berdasarkan data yang sudah diolah. Analisis sentimen adalah studi komputasi dari opini-opini, sentimen, seta emosi yang diekspresikan dalam teks. Tugas dasar dalam melakukan analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen, kalimat atau pendapat (Nugroho et al., 2015)

2. Python

Python merupakan salah satu bahasa pemprograman yang bersifat *open source,* menyediakan dukungan untuk pengelolaan data pada beragam implementsi, memiliki komunitas dan sumber belajar online yang banyak, serta menduduki peringkat atas pemprograman terpopuler mennurut beberapa komunitas pengindeks (W Dwi Yuniarti, 2019). Python juga dapat membantu dalam klasifikasi dalam berbagai algoritma, termasuk *naïve bayes* dan *support vector machine*, 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 serta beberapa *library* tambahan yang harus di *install* seperti *scikit-learn* (sklearn), *Natural Language Toolkit* (NLTK), *Journey to the Center of Python Machine Learning* (jcopml), Sastrawi. *Scikit-learn* merupakan *library* yang mencakup banyak algoritma untuk pembelajaran klasifikasi ataupun regresi, serta memiliki banyak algoritma yang menujang dalam pemrosesan data. NLTK merupakan library yang digunakan dalam komputasi linguistik.

NLTK mencakup pemrosesan bahasa alami berupa statistik, simbolik dan hubungan bertautan dengan copra. *Library* sastrawi mencakup pemrosesan bahasa alami yang berhubungan dengan bahasa yang digunakan untuk stemmer yang digunakan untuk mengatasi permasalahan perubahan kata imbuhan menjadi kata dasar (Rifano et al., 2020) .

3. Aplikasi SIREKAP

Sirekap adalah singkatan dari Sistem Informasi Rekapitulasi, yang dirancang oleh Komisi Pemilihan Umum (KPU) Republik Indonesia untuk mendukung pelaksanaan Pemilu 2024. Aplikasi SIREKAP sendiri telah diunduh lebih dari 1 juta, dengan memiliki rating 1,7 dan tercatat 36,4 ribu komentar pengguna aplikasi SIREKAP (Play.google.com, 2024). Aplikasi ini bertujuan untuk mempermudah proses perhitungan suara dan memastikan transparansi dalam pengumuman hasil pemungutan suara.

SIREKAP memiliki fungsi sebagai alat rekapitulasi hasil pemungutan suara, mempublikasikan hasil suara secara real-time, serata mengurangi kesalahan dalam pelaporan(Rismel, 2024). Aplikasi SIREKAP saat ini digunakan peneliti untuk menganalisis bagaimana tanggapan pengguna.

Dikarenakan banyak berita beredar tentang kendala aplikasi sirekap diantaranya; kesalahan konversi foto dokumen hasil perhitungan suara, petugas KPPS kesulitan akses, masalah keamanan aplikasi, performa SIREKAP yang menurun, kesalahan desain mekanisme verifikasi data, perbaikan data yang tidak dilakukan secara menyeluruh, dan aplikasi yang tidak dapat diakses serta tidak dapat digunakan pada hari pemilu (Paris, 2024).

4. *Text Preprocessing*

Text preprocessing merupakan suatu proses pengelolaan data set sebelum data di proses. Data set harus melalui tahap text preprocessing terlebih dahulu, karena fakta mengatakan kebanyakan dataset belum sepenuhnya bersih, seperti halnya format yang kurang beraturan, adanya data yang kosong, tipe data yang berbeda - beda dan masih banyak lainnya. Semakin bersih proses yang dilakukan, maka kemungkinan besar data tersebut semakin akurat. Pada penelitian ini *text preprocessing* terdiri dari beberapa langkah yaitu *case folding, cleansing, stopword, tokenized, steaming* (Fajrin & Maulana, 2018).

a. Cleansing

Tahap ini merupakan penghilangan suatu kalimat untuk menghilangkan simbol, tanda baca, dan bilangan angka. Serta menghapus data yang tidak valid dan tidak relevan. Seperti pada kata "sangat^{2}" disini angka 2 akan dihapus, "susah...." disini tanda baca "...." terlalu banyak atau berada ditengah kalimat makan akan dihapus.

b. Case Folding

Tahap ini mengubah huruf kapital yang ada dalam teks menjadi huruf kecil. Seperti halnya pada kalimat " susah nya setengah MATI" huruf yang kapital pada kalimat ini akan dirubah seperti " susah nya setengah mati"

c. Normalisasi (*Slang Word)*

Tahap ini mengubah *slang word* atau kata yang tidak resmi menjadi kata baku seperti halnya "aplikasi ini blm bagus" menjadi "aplikasi ini belum bagus". Pada tahap ini mengandalkan Kamus slang word Bahasa Indonesia.

d. Tokenized

Tahap ini yaitu memotong urutan karakter menjadi potongan – potongan yang disebut token (Hasanah et al., 2018). Token – token tersebut

nantinya akan diolah lebih lanjut dalam analisis data. Contoh dari tokenized pada kalimat ” aplikasi yang malah menyusahkan” akan dirubah menjadi ” ['aplikasi', 'yang', 'malah', 'menyusahkan']”.

e. Stopword

Stopword adalah tahapan untuk menghapus kata yang sering muncul tapi tidak memiliki arti penting dan maknanya tidak berpengaruh pada sistem, seperti oh, di, pada, dan sebagainya (Taufiqurrahman et al., 2021).

f. Steaming

Teknik stemming dilakukan untuk memotong term atau menghilangkan kata imbuhan untuk menjadi bentuk dasarnya (Khairunnisa et al., 2021). Contoh penerapan *steammig* adalah mengubah kata ”mendapatkan”, ”didapatkan” akan diubah menjadi kata dasar ”dapat”.

5. Metode Naive Bayes

Teorema yang dikenal sebagai "teorema Bayes" adalah teorema yang ditemukan oleh ilmuwan Inggris Thomas Bayes. Metode pengkasifikasian yang digunakan untuk klasifikasi *Naïve Bayes* didasarkan pada teorema ini. Teorema Bayes, bila dua peristiwa

yang berbeda (misalnya, X dan H), maka Teorema Bayes dapat dirumuskan sebagai berikut (Novriandy, 2023):

$$P(H|X) = \frac{P(X|H)}{P(X)} \; x \; P(H) \qquad (2.1)$$

Keterangan :

X =Class data yang dimasukkan

H =Data hipotesis

P(H|X) =Probabilitas hipotesis H yang mengacu pada kondisi X

P(H) =Probabilitas hipotesis H;

P(X|H) =Probabilitas X berdasarkan kondisi pada hipotesis H.

P(X) = Probabilitas X

Banyak orang menggunakan algoritma *Naïve Bayes* untuk melakukan analisis sentimen. Klasifikasi *Naive Bayes* dapat memprediksi kemungkinan keanggotaan kelas berdasarkan prediksi asumsi independen. Dengan demikian, pengalaman sebelumnya dapat digunakan untuk menentukan peluang masa depan. Keunggulan algoritma *Naïve Bayes* salah satunya adalah efisien, karena mampu menghasilkan proses analisis sentimen menjadi lebih singkat. Selain itu, algoritma ini cenderung lebih akurat walaupun menggunakan data latih yang relatif sedikit

(Herlinawati et al., 2020). Naïve Bayes adalah algoritma yang juga menghitung probabilitas untuk setiap kelas keputusan dengan asumsi bahwa fitur objek berbeda satu sama lain. Ini juga memperhitungkan jumlah frekuensi dalam tabel keputusan untuk menentukan probabilitas akhir (Sihombing, 2021).

6. Metode SVM

Metode Support Vector Machine, yang didasarkan pada gagasan Minimisasi Risiko Struktural, digunakan untuk klasifikasi dan regresi. SVM dapat mengubah data ke dalam hyperplane dan dapat mengklasifikasikan ruang input menjadi dua kelas (Sis.binus.ac.id, 2022). Dalam praktiknya, serta SVM menggunakan hyperplane sebagai pemisah dari kelompok kelas yang dibentuk berdasarkan dimensi vektor yang diukur jarak terjauh antara dua kelas atau margin sentimen (Iman & Wijayanto, 2021). Kasus ini merupakan SVM linier karena hanya menggunakan hyperplane sebaga pemisah data sehingga tidak perlu menjalani tranformasi apa pun untuk memisahkan data ke dalam berbagai kelas yang berbeda (Nelson, 2020). Untuk menjalankan hyperplane pemisah ini dapat dipresentasikan sebagai beirkut:

$$w \cdot x + b = 0 \qquad (2, 2)$$

di mana w adalah vektor bobot, x adalah vektor input, dan b adalah suku bias.

SVM juga memanfaatkan metode non-linear untuk mentransformasi data pelatihan ke dalam dimensi yang lebih tinggi, yang disebut sebagai pemetaan non-linear. Algoritma ini sering digunakan dalam industri pembangunan dan reparasi kapal, serta berbagai bidang lain seperti teknologi komputer, sektor keuangan, dan bidang kesehatan (Ritonga & Purwaningsih, 2018). Untuk mengatasi sifat tidak linier dengan menggunakan metode kernel. Kernel berguna untuk menggambarkan dimensi yang lebih kecil ke dimensi lebih besar.

Fungsi kernel yang biasa dipakai dalam SVM :

Tabel 2. 1 Kernel SVM

Kernel linier	k(x,y) = x.y
Kernel polinominal	k(x,y) = (x.y)
Kernel sigma	k(x,y) = tan(σ(x.y)+c
Kernel RBF	K(x,y) = exp $\frac{-x-y^2}{2\sigma^2}$

Kernel dimana:
c: constanta, d: degree, exp: eksponensial

7. Evaluasi Model

confusion matrix atau matriks kebingungan adalah alat visualisasi yang biasa digunakan untuk *supervised learning.* Setiap kolom pada matrix berisi

kelas prediksi dan setiap baris berisi kelas kejadian sebenarnya atau actual (Akhir et al., 2023). Adapun tabel *confusion matrix* di bawah ini:

Tabel 2. 2 Tabel *Confusion Matrix*

Kelas Aktual	Kelas prediksi	
	positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Tahap evaluasi menggunakan *matrix confusion* untuk mengevaluasi kinerja metode klasifikasi. Dimana, *True Positive* (TP) merupakan data yang diprediksi positif dan data sesungguhnya juga positif, *True Negative* (TN) merupakan data yang diprediksi negitif dan data sesungguhnya juga negatif, *False Positive* (FP) merupakan data yang diprediksi positif dan data sesungguhnya negatif, serta *False Negative* (FN) merupakan data yang diprediksi negatif dan data sesungguhnya positif. Keempat istilah diatas yang digunakan untuk menggambarkan hasil dari proses klasifikasi dalam pengukuran efisiensi (Karsito & Susanti, 2019). Adapun perhitungan untuk akurasi dapat dikalkulasi dan dilihat pada persamaan 2. 3, precision dilihat pada persemaan 2. 4 , recall dilihat

pada persamaan 2. 5, serta FI score dapat dilihat pada persamaan 2. 6.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \qquad (2.\ 3)$$

Akurasi adalah presentase prediksi yang benar dibandingkan dengan total jumlah prediksi yang dilakukan. Cara perhitungannya dengan cara membagi jumlah yang diprediksi sistem yang sesuai dengan jumlah keseluruhan data uji.

$$Precision = \frac{TP}{TP+FP}\ x\ 100\% \qquad (2.\ 4)$$

Precision adalah mengukur seberapa banyak dari prediksi positif yang benar – benar positif. Dalam hal lain mengindikasikan ketepatan model dalam memprediksi sentimen positif.

$$Recall = \frac{TP}{TP+FN}\ x\ 100\% \qquad (2.\ 5)$$

Recall adalah proporsi dari prediksi positif yang benar dibandingkan dengan total jumlah aktual dari kelas positif atau mengukur seberapa baik model dalam menangkap semua kasus positif.

$$F1\ Score = 2\ x\ \frac{Precision\ .\ Recall}{Precision+Recall}\ x\ 100\% \qquad (2.6)$$

F1 Score adalah rata – rata harmonis antara presisi dan recal, yang memberikan gambaran umum tentang keseimbangan antara keduanya.

B. Kajian Pustaka

Beberapa kajian penelitian sebelumnya yang penulis jadikan referensi dalam melakukan penelitian ini sebagai berikut :

Judul	Penulis dan Tahun	Tujuan dan Metode	Keterkaitan dan perbedaan
Analisis Sentimen Aplikasi Gojek Menggunakan *Support Vector Machine* dan *K Nearest Neighbor*	(Muttaqin & Kharisudin, 2021)	Analisis Sentimen Ulasan Aplikasi Gojek dengan *Support Vector Machine* dan *K Nearest Neighbor*	Keterkaitan pada penelitian ini sebagai sumber acuan menggunakan metode SVM, dengan perbedaan topik yang dianalisis berbeda.
Analisis Sentimen Opini Masyarakat Terkait Pelayanan Jasa Ekspedisi Antar aja Dengan	(Afandi et al., 2022)	Menganalisis kepuasan pelanggan ekspedisi antar aja melalui media sosial twitter menggunakan metode *Naive Bayes*	Keterkaitan pada penelitian ini adalah metode Naive Bayes yang digunakan, dengan perbedaan pemilihan topik dan media yang berbeda.

Metode *Naive Bayes*			
Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode *Naive Bayes* dan *Support Vector Machine*	(Petiwi et al., 2022)	Untuk mengelompokkan komentar masyarakat menjadi tiga kelas yakni positif, negatif, dan netral menggunakan metode *Naive Bayes* dan SVM.	Keterkaitan penelitian ini yaitu menggunakan kedua metode yang sama, dengan perbedaan topik dan media yang berbeda.
Analisis Sentimen Ulasan Aplikasi Ruangguru dengan Algoritma *Long Short Term Memory*	(Ramadhan, 2023)	Pada penelitian ini memiliki tujuan yaitu melakukan analisis sentimen pada ulasan aplikasi ruangguru serta membandingkan kinerja algoritma deep learning yaitu LSTM dengan algoritma *machine learning* klasik	Keterkaitan pada penelitian ini yaitu tahap yang digunakan memiliki urutan yang sama, dengan perbedaan menggunakan metode yang berbeda dan topik yang berbeda.

Perbandingan Akurasi Algoritma Naive Bayes dan Support Vector Machine dalam Analisis Sentimen Pengguna Twitter Terhadap Aplikasi Sirekap	(Natalianus et al., 2024)	Tujuan penelitian untuk menganalisis sentimen pengguna terhadap kinerja Aplikasi Sirekap serta membandingkan performa dua algoritma, yaitu Naive Bayes dan Support Vector Machine (SVM)	Keterkaitan penelitian yaitu aplikasi yang di analisis dan metode, dengan perbedaan media penelitian ini menggunakan media sosial Twitter. Serta sumber pengambilan data yang berbeda.

Kesimpulan dari tabel penelitian di atas menunjukkan bahwa terdapat sejumlah penelitian yang berkaitan dengan analisis sentimen menggunakan beberapa algoritma, seperti SVM, Naive Bayes, K-Nearest Neighbor, dan Long Short-Term Memory. Sebagian besar penelitian menggunakan SVM dan Naive Bayes, baik secara mandiri maupun dikombinasikan dengan metode lain. Penggabungan metode dengan tujuan meningkatkan akurasi dalam pengelompokkan sentimen. Walaupun memiliki

kesamaan penelitian – penelitian ini berbeda dalam hal topik dan media yang dianalisis.

Media yang digunakan juga bervariasi, dengan beberapa penelitian yang mengandalkan ulasan pengguna di platform aplikasi, sedangkan lainnya menganalisis data dari media sosial seperti Twitter. Perbedaan pendekatan ini menambah variasi pada cara analisis sentimen dilakukan di setiap penelitian, tergantung pada konteks dan tujuan yang ingin dicapai. Secara keseluruhan, penelitian-penelitian ini menunjukkan keterkaitan dalam penggunaan algoritma yang sama, terutama SVM dan Naive Bayes, namun tetap berbeda dalam konteks aplikasi atau platform yang menjadi objek kajian. Hal ini menunjukkan bahwa meskipun pendekatan umum yang sama dapat diterapkan, variasi dalam objek dan media penelitian mempengaruhi hasil dan relevansi analisis sentimen di berbagai bidang.

BAB III
METODOLOGI PENELITIAN

A. Metode Pengumpulan Data

Cara penulis mendapatkan data dan informasi yang menunjang dan dapat memenuhi kebutuhan penelitian ini, sebagai berikut.

1. Sumber Dataset
   Dataset berupa kumpulan komentar yang diperoleh menggunakan teknik scraping data dari API yang telah disediakan oleh *Google Play Store* menggunakan *library python* yaitu *google-play-scraper*. Total jumlah komentar yang penulis dapatkan adalah 2000 komentar, yang kemudian disimpan dalam file berbentuk csv.

2. Studi Literatur
   Penulis melakukan studi literatur dengan memanfaatkan jurnal, skripsi, buku – buku dan sejenisnya untuk mempelajari teori yang berhubungan dengan konsep, analisis, dan permasalahan yang diteliti oleh penulis. Selain itu, penulis juga melakukan pencarian data secara daring dengan menggunakan browser melalui internet dengan mengunjungi suatu website yang berhubungan dengan analisis sentimen, algoritma *naive bayes* dan algoritma SVM.

B. Perangkat Penelitian

Implementasi dalam melakukan penelitian ini membutuhkan beberapa perangkat keras dan perangkat lunak yang harus dipenuhi untuk berjalannya proses sistem. Adapun kebutuhan yang digunakan pada penelitian ini sebagai berikut:

1. Kebutuhan perangkat keras:

Tabel 3. 1 *Kebutuhan perangkat keras*

No.	Perangkat keras	Spesifikasi
1.	Device	ASUS Laptop X441UBR
2.	Processor	Intel(R) Core i3-7020U
3.	Memori (RAM)	4GB
4.	Monitor	14 inch
5.	Keyboard	Standard

2. Kebutuhan perangkat lunak:

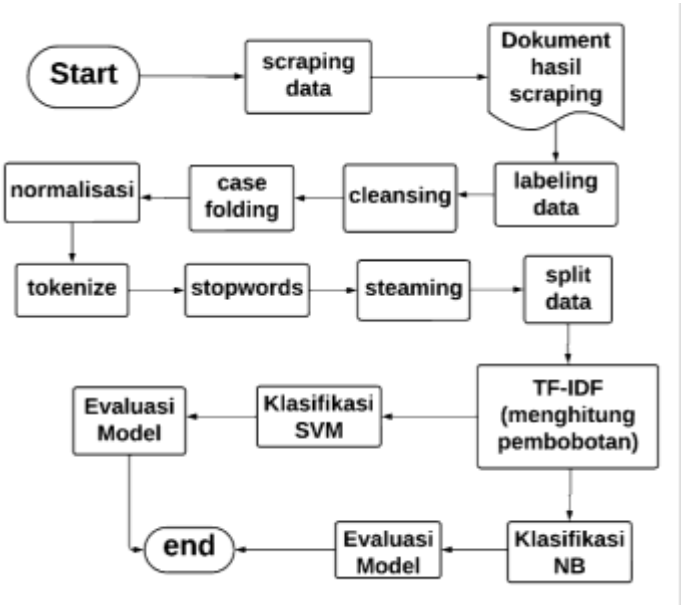
Tabel 3. 2 Kebutuhan perangkat keras

No.	Perangkat lunak	Spesifikasi
1.	Sistem operasi	Windows 10 64-bit
2.	Bahasa pemprograman	Python
3.	MS. Office	Ms. Word, Ms. Excel
4.	Browser	Chrome
5.	Google Drive	Google colab

C. Alur Penelitian

Alur penelitian ini membahas mengenai gambaran umum yang akan dilakukan penulis dalam tahapan pengerjaan yang dimulai dari awal hingga

akhir. Pada gambar 3.1 di bawah ini merupakan gambaran rangkain penelitian atau flowchart pengerjaan penelitian.



Gambar 3. 1 Flowchart alur pengerjaan penelitian

Tahapan awal dari pengerjaan penelitian ini adalah mengumpulkan data komentar dan rating yang ada di aplikasi melalui teknik web scraping yaitu dengan menggunakan library python yaitu google-play-scraper dengan menggunakan keyword id ”id.go.kpu.sirekap2024” untuk mendapatkan data yang penulis butuhkan. Pada tahap selanjutnya dilakukan preprocessing data terlebih dahulu sebelum dilakukan

tahap labeling (Ramadhan, 2023). Tahap *preprocessing* nantinya ada tahapan yang akan dilakukan yaitu, cleaning, case folding, normalisasi, tokenize, stopwords, dan steaming.

Tahapan tersebut yaitu berfungsi menghapus atau membenarkan kata yang kurang tepat atau emoticon yang tidak bisa didefinisikan dalam proses pelabelan nantinya. Setelah adanya preprocessing data maka selanjutnya ada tahap labeling yang mana, pada tahap ini nantinya komentar yang sudah diolah tadi akan diberikan label atau sentiment yang berupa komentar positif atau negatif. Setelah adanya pelabelan maka selanjutnya adalah splitting data yang digunakan sebagai pengembangan model *machine learning* yang nantinya dapat mengurangi resiko *overfitting* dengan melakukan proses pembagian dataset awal menjadi subset pelatihan, validasi, dan uji dengan proporsi yang sesuai untuk masing-masing subset.

Selanjutnya ada tahap TF-IDF (*Term Frequency Inverse Document Frequency*) yang digunakan untuk memberikan bobot terhadap teks. TF ialah jumlah sebuah kata dari tiap-tiap dokumen, sedangkan IDF ialah nilai invers dari jumlah dokumen yang mengandung kata tersebut (Fikri et al., 2020). Baru

nantinya setelah itu dilakukan klasifikasi menggunakan metode Naive Bayes dan Support Vector Machine. Setelah klasifikasi selesai nantinya ada evaluasi model yang digunakan dalam mengukur keberhasilan suatu sistem dengan membandingkan hasil pengujian pada sistem dengan standar yang telah ditentukan (Zidan, 2022) .

D. Uraian Metodologi

Pada tahap ini dilakukan penguraian pada alur penelitian untuk menjelaskan lebih dalam. Diantaranya sebagai berikut:

1. Pengambilan Data *Google Play Store*

   Data yang digunakan dalam penelitian ini adalah data dari komentar pengguna aplikasi SIREKAP pada google play store. Data yang diambil berupa data teks yang diambil dengan teknik *scraping* menggunakan API *google play store.* Cara melakukanya yaitu membuat program dengan memasukkan link id aplikasi SIREKAP yaitu "id.go.kpu.sirekap2024", kemudian tulis pada bagian berapa data yang diinginkan, penulis menulis data sebesar "2000" data komentar. Setelah itu akan muncul data yang akan digunakan dalam proses penelitian ini sebanyak data yang diinginkan. Kemudian data disimpan pada format

file csv. Tabel 3.3 dibawah merupakan data yang diambil dari hasil *scraping.*

Tabel 3. 3 Data hasil aplikasi SIREKAP

No	Content
1.	Aplikasi butut, herannya kpu dikasih waktu 5 tahun untuk buat ni aplikasi tapi hasilnya malah seadanya, kayak ga niat buat (lebih ke "hemat budget" mungkin). Gak bisa unggah foto dari galeri, kualitas foto jadi jelek banget + blur, gak ada flash, sudah diunggah suka minta foto ulang yang mana itu buat repot. Lain kali kalo diamanahkan untuk buat aplikasi tolong buatlah dengan sebaik mungkin, masukan dari teman" yang sudah diulas tolong dipikirkan.
2.	Segera perbaiki menu detail pemilihan, sebab gagal lakukan koreksi hasil pengambilan foto form C, periksa form C Hasil, dalam jendela periksa kesesuaian data, hanya bisa kasih x (cross) tidak sesuai, tp jumlah yg perlu dikoreksi tidak muncul editable box nya. Terima kasih.
3.	APLIKASI GAGAL. Suka nge bug, over akses bikin Lemot, kamera tidak bisa di fokuskan

	dan kualitas gambar tulisan jadi blur. Unggah gambar tulisan "tunggu sesaat" TAPI 3 hari blum selesai proses pengunggahannya. Cara pengaplikasianya cukup mudah. Tapi ya seperti tadi contohnya. Aplikasi PEMILU dari tahun 2019 sampe sekarang gak ada perubahan.
4.	Aplikasi belum layak untuk dioperasikan 1.bisa dilihat dari komenan kawan2,pas unggah selalu logout sendiri. 2.kualitas gambar,bukan kesalah spek android,camera saat poto biasa jernih dan fokus,diaplikasi ini hasil poto buram dan tidak dapat difokuskan secara manual,mungkin mengingat kualitas gambar HD memakan penyimpanan besar mungkin,tapi ini kualitasnya buram,terlalu kecil,tolong perbaiki secepatnya,!!!
5.	Jelek , dari segi tampilan ui nya terkesan jadul banget , terus juga untuk simulasi aplikasi sirekap ini ga terlalu jelas aneh, ngabisin anggaran doang ! Improvisasi lagi kedepannya pak agar lebih baik lagi !
6.	Aplikasi paling banyak minusnya. Kualitas gambar buram saat difoto (tidak bisa

	difokuskan manual), angka digital sering salah dibaca, susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi, dan masih banyak lagi minus yang lain.
7.	Aplikasi yang mlah menyusahkan. Fokus foto yang tidak bagus, scan manual, force close, dan jaringan yg sibuk. Aplikasi yang dicoba pada saat pemilu belum berlangsung saja sudah punya kualitas yang buruk, apalagi saat pemilu dilakukan.
8.	Perbaiki lagi kualitas aplikasinya kalau mau rekapan hasil pemilu berjalan lancar. Fix sangatÃ‚Â² membagongkan ni aplikasi mencetÃ‚Â² tombol sampai jari pegel pun tetep ga bisa kebaca tuh sidik jari.

.....

1999.	tolong apliaksi nya di perbaiki lagi..saat ingin login aplikasi eror trs menunggu di saat jam2 tidak sibuk pengguna nya...apa lagi saat pemilu nanti .tks
2000	Kenapa susah untuk proses sertifikat digital selalu time out..pdhl waktu pencoblosan sebentar lgi tapi aplikasinya tidak berfungsi dgn baik

2. Labeling

Pada proses pelabelan data berfungsi untuk menentukan sentimen/ label teks. Data yang diambil dari hasil scraping serta tersimpan dalam bentuk csv,

akan dilakukan pelabelan. Contoh proses tahap pelabelan dapat dilihat pada tabel 3.4.

Tabel 3. 4 Penerapan tahap cleansing

Teks	Label
Aplikasi paling banyak minusnya. Kualitas gambar buram saat difoto (tidak bisa difokuskan manual), angka digital sering salah dibaca, susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi, dan masih banyak lagi minus yang lain.	Negatif
Perbaiki lagi kualitas aplikasinya kalau mau rekapan hasil pemilu berjalan lancar. Fix sangatÃ‚Â² membagongkan ni aplikasi mencetÃ‚Â² tombol sampai jari pegel pun tetep ga bisa kebaca tuh sidik jari.	Positif

3. *Preprocessing*

Tujuan dari tahap ini agar data yang mentah dapat dibersihkan sehingga bisa dilakukan pengklasifikasian. Data yang dibersihkan berasal dari data yang diperoleh dalam proses pengambilan data diatas. Pada tahap cleansing hingga normalisasi jika menggunakan python menggunakan library yaitu:

```
from ast import Return
import re
```

```
import string
import nltk
```

berikut merupakan semua tahapan dalam preprocessing yang dilakukan diantaranya sebagai berikut:

a. *Cleansing*

Seringnya muncul data yang khas seperti halnya tanda retweet, bilangan angka, simbol dan tanda baca yang tidak digunakan. Merupakan komponen tidak memiliki peran penting dalam melakukan sentimen maka secara sistem akan dihapus. Dengan tujuan untuk mengurangi *noise*. Contoh proses tahap *cleansing* yang terdapat pada tabel 3.5.

Tabel 3. 5 Penerapan tahap cleansing

Teks asli	Hasil *cleansing*
Aplikasi paling banyak minusnya. Kualitas gambar buram saat difoto (tidak bisa difokuskan manual), angka digital sering salah dibaca, susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi, dan masih banyak lagi minus yang lain.	Aplikasi paling banyak minusnya Kualitas gambar buram saat difoto tidak bisa difokuskan manual angka digital sering salah dibaca susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi dan masih banyak lagi minus yang lain

b. Case folding

Pada tahap ini teks yang mengandung huruf kapital disamaratakan menjadi huruf kecil atau lowercase. Contoh proses tahap case folding yang terdapat pada tabel 3.6.

Tabel 3. 6 Penerapan tahap case folding

Hasil *cleansing*	Hasil *case folding*
Aplikasi paling banyak minusnya Kualitas gambar buram saat difoto tidak bisa difokuskan manual angka digital sering salah dibaca susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi dan masih banyak lagi minus yang lain	aplikasi paling banyak minusnya kualitas gambar buram saat difoto tidak bisa difokuskan manual angka digital sering salah dibaca susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi dan masih banyak lagi minus yang lain

c. Normalisasi

Pada bagian ini kata yang tidak resmi diganti dengan kata baku. Pada proses ini nantinya memerlukan kamus kata baku, yang mana kamus tersebut bernama kamus slag (kata baku dan tidak baku) (Akhir et al., 2023). Berikut merupakan contoh normalisasi bisa dilihat pada tabel 3.7.

Tabel 3. 7 Penerapan tahap normalisasi

Hasil *case folding*	Hasil Normalisasi
aplikasi paling banyak minusnya kualitas gambar buram saat difoto tidak bisa difokuskan manual angka digital sering salah dibaca susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi dan masih banyak lagi minus yang lain	aplikasi paling banyak minusnya kualitas gambar buram saat difoto tidak bisa difokuskan manual angka digital sering salah dibaca susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi dan masih banyak lagi minus yang lain

Pada proses tahap ini kata “blum”, “sampe”, serta ”gak” diubah menjadi ”belum”, ”sampai”, serta ”tidak”. Bisa tahap ini dapat dilihat perubahan dari kata yang tidak resmi menjadi kata yang baku.

d. Tokenize

Pada tahap ini yaitu proses memecah kalimat menjadi kata yang dinamakan token. Dengan menghilangkan whitespace. Contoh proses pada tokenize dapat dilihat pada tabel 3.8.

Tabel 3. 8 Penerapan tahap tokonize

Hasil normalisasi	Hasil *Tokenize*
aplikasi paling banyak minusnya kualitas gambar buram saat	['aplikasi', 'paling', 'banyak', 'minusnya', 'kualitas', 'gambar',

difoto tidak bisa difokuskan manual angka digital sering salah dibaca susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi dan masih banyak lagi minus yang lain	'buram', 'saat', 'difoto', 'tidak', 'bisa', 'difokuskan', 'manual', 'angka', 'digital', 'sering', 'salah', 'dibaca', 'susah', 'untuk', 'mengedit', 'hasil', 'yang', 'tidak', 'sesuai', 'dibaca', 'oleh', 'aplikasi', 'dan', 'masih', 'banyak', 'lagi', 'minus', 'yang', 'lain']

e. Stopword

Tahap ini dilakukan penghapusan daftar kata umum yang tidak memiliki arti penting. Dengan tujuan untuk mengurangi jumlah kata yang disimpan oleh sistem. Contoh proses stopword dapat dilihat pada tabel 3.9.

Tabel 3. 9 Penerapan tahap stopword

Hasil *tokenize*	Hasil *Stopword*
['aplikasi', 'paling', 'banyak', 'minusnya', 'kualitas', 'gambar', 'buram', 'saat', 'difoto', 'tidak', 'bisa', 'difokuskan', 'manual', 'angka', 'digital', 'sering', 'salah', 'dibaca', 'susah', 'untuk', 'mengedit', 'hasil', 'yang', 'tidak',	['aplikasi', 'minusnya', 'kualitas', 'gambar', 'buram', 'difoto', 'difokuskan', 'manual', 'angka', 'digital', 'salah', 'dibaca', 'susah', 'mengedit', 'hasil', 'sesuai', 'dibaca', 'aplikasi', 'minus']

'sesuai', 'dibaca', 'oleh', 'aplikasi', 'dan', 'masih', 'banyak', 'lagi', 'minus', 'yang', 'lain']	

Pada proses ini juga kata yang sering muncul dan tidak terlalu penting dihilangkan seperti halnya pada kata ”paling”, ”bisa”, ”saat”, ”tidak”, dan masih banyak lainnya. Hasil dari tabel diatas jika menggunakan pemprograman python yaitu menggunakan *library* yaitu:

```
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')
```

f. Steaming

Pada proses tahap steaming yaitu mengganti kata yang memiliki imbuhan menjadi bentuk kata dasar, dan mengubah kata perulangan menjadi kata dasar. Contoh proses steaming dapat dilihat pada tabel 3.10.

Tabel 3. 10 Penerapan tahap steaming.

Hasil *stopword*	Hasil *Steaming*
['aplikasi', 'minusnya', 'kualitas', 'gambar', 'buram', 'difoto', 'difokuskan', 'manual', 'angka', 'digital', 'salah',	aplikasi minus kualitas gambar buram foto fokus manual angka digital salah baca susah edit

'dibaca', 'susah', 'mengedit', 'hasil', 'sesuai', 'dibaca', 'aplikasi', 'minus']	hasil sesuai baca aplikasi minus

Pada proses ini yang kata yang terdapat kata imbuhan akan dihapus, seperti kata ”minusnya” dirubah menjadi ”minus”, ”difoto” menjadi ”foto”, ”mengedit” menjadi ”edit” serta masih banyak lainnya. Hasil dari tabel diatas jika menggunakan pemprograman python yaitu menggunakan *library* yaitu:

```
!pip install Sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
```

4. Split data

Pada penerapan proses ini yaitu membagi dataset menjadi 2 bagian yaitu traning set dan test set dengan ukuran 80% sebagai data training dan 20% untuk data testing (Turmudi Zy et al., 2021).

5. TF-IDF

Setelah dilakukan split data, fitur diekstraksi menggunakan metode TF-IDF untuk mengidentifikasi dan menilai pentingnya kata -kata dalam teks. metode

ini juga dapat meningkatkan akurasi dalam proses analisis. Proses menghitung bobot kata dalam metode TF-IDF merupakan hasil dari penggabungan dua konsep, yaitu *Term Frequence* (TF) dan *Inverse Document Frequency* (IDF). Di mana TF merupakan banyakanya kemunculan sebuah kata dalam suatu dokumen dan jumlahnya sebanding dengan besaran bobot kata tersebut karena semakin sering muncul, maka bobotnya semakin besar. Sedangkan, pada konsep IDF, suatu kata yang sering muncul memiliki bobot kecil (Yutika et al., 2021).

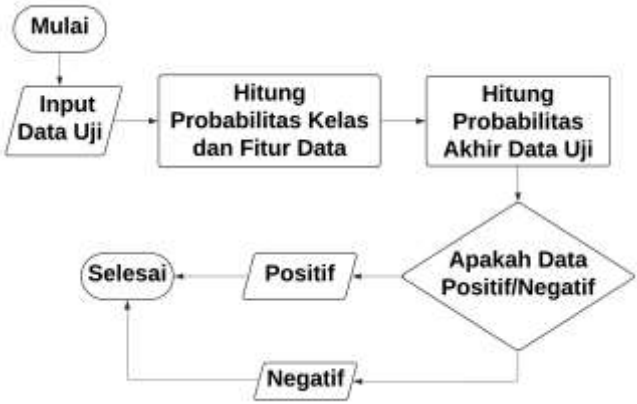
Untuk mendapatkan nilai IDF dapat dilakukan dengan menggunakan persamaan logaritma dari jumlah dokumen dalam suatu kumpulan data (n) dibagi jumlah dokumen yang di dalamnya terdapat suatu term (df^t) tertentu. Sehingga secara keseluruhan pembobotan TF-IDF dapat dilakukan menggunakan Persamaan 1.

$$TF - IDF_{std}(t) = tf_d^t \, x \, log \, \frac{n}{df^t} \qquad (3.1)$$

8. Klasifikasi *Naive bayes*

Setelah dilakukannya TF-IDF maka dilakukan lah klasifikasi menggunakan metode *naive bayes* yang mana data yang sudah dilakukan preprocessing akan

dilakukan perhitungan probabilitas kelas dan fitur data. Kemudian menghitung probabilitas akhir data yang kemudian diklasifikasikan menjadi positif atau negatif. Pada perencanaan perhitungan menggunakan metode *naive bayes* dibuat diagram alir pada gambar 3.2 dibawah ini.

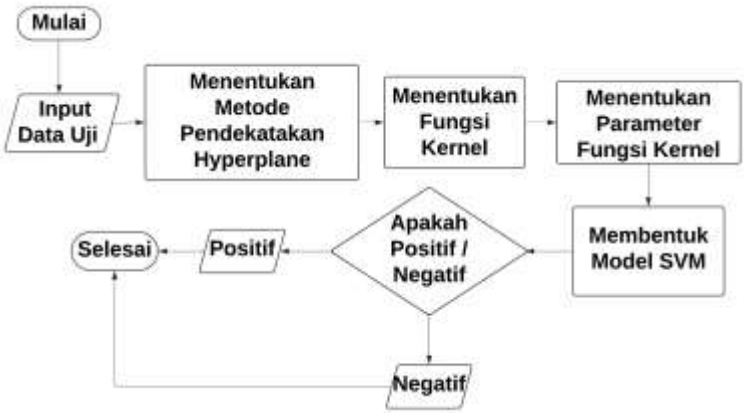


Gambar 3. 2 *flowchart* metode *Naive Bayes*

9. Klasifikasi Support Vector Machine

Penggunaan klasifikasi SVM juga memIliki tahapan yang perlu diperhatikan. Dimulai dengan menginput data, setelah itu dilakukan penentuan pendekatan metode *hyperplane*. Setelah ditemukan *hyperplane*, menentukan fungsi kernel untuk dapat mengolah data di antara keempat fungsi kernel. Dilanjut dengan menentukan parameter fungsi kernel yang sudah ditentukan di tahap sebelumnya. Kemudian

dilanjut membentuk model SVM, dengan menentukan klasifikasi SVM apakah termasuk positif atau negatif. Pada gambar flowchart 3.3 dibawah ini digambarkan bagaimana urutan dari kerja klasifikasi SVM.



Gambar 3. 3 *Flowchart* Metode SVM

10. Evaluasi Model

Proses ini dilakukan untuk mengetahui kinerja suatu model. Dilakukan dengan cara melihat tingkat performa metode melalui dua kelas *confusion matrix.* Data test diujikan terhadap data *training,* maka akan menghasilkan daftar kelas-kelas dari data test, sebut saja prediksi kelas. Kemudian prediksi kelas dibandingkan dengan kelas yang sebenarnya dari data test yang disembunyikan sebelumnya. Sehingga dapat

dilihat pada performa model naive bayes dan SVM yang berupa tingkat akurasi, precision, recall dan f1 score.

BAB IV
HASIL DAN PEMBAHASAN

Pada bab ini, akan membahas hasil dari tinjauan komentar pengguna aplikasi SIREKAP yang diambil dari *Google Play Store*. Analisis ini dilakukan dengan menggunakan dua metode, yaitu *Naive Bayes* dan *Support Vector Machine* (SVM). Hasil yang diperoleh akan dibandingkan untuk menentukan metode mana yang lebih efektif dalam mengklasifikasikan sentimen komentar pengguna.

A. Scraping Data Komentar *Play Store*

Proses pengambilan data dilakukan dengan metode *scraping* menggunakan *library google-play-scraper* untuk mendapatkan data ulasan pada *google play store. Library* ini dibuat untuk mempermudah melakukan pengambilan data melalui *google play store, library* ini sangat efektif dikarenakan kita hanya perlu menambahakan *link id* untuk mendapatkan data yang kita butuhkan, tanpa mengistal aplikasi yang dibutuhkan. Maka untuk menggunakan *library* tersebut dilakukan instalasi terlebih dahulu menggnakan perintah ”pip”.

Berikut instalasi library *google-play-scraper* menggunakan Source code seperti di bawah ini.

```
!pip install google-play-scraper
```

Selain menggunakan *library* tersebut peneliti menggunakan *tools* pendukung untuk menganalisis sentimen yaitu *google colab* sebagai pendistribusi *python*. *Google colab* sangat efektif digunakan karena tidak perlu melakukan pengisntalan sehingga tidak memakan memori, hanya perlu menggunakan jaringan internet serta akun *google* sebagai sarana akses yang berguna dalam menyimpan hasil codingan (Carneiro et al., 2018). *Software* ini pada dasarnya serupa dengan jupyter notebook gratis berbentuk cloud yang dijalankan menggunakan *browser*. Berikut tampilan dari *google colab* pada Gambar 4.1 di bawah.

Gambar 4. 1 Tampilan *Google Colab*

Untuk melakukan proses *scraping* data maka harus siapkan atau meng-*import* data yang diperlukan dengan mengguakan source code di bawah ini.

```
from google_play_scraper import app
import pandas as pd
import numpy as np
```

Pada gambar 4.3 pada bagian atas untuk mengimpor fungsi "app" dari *library* yang digunakan untuk mengambil data aplikasi dari *google play store*. Serta *library* pandas berfungsi sebagai manipulasi data dan analisis data, terutama dalam bentuk *DataFrame*. Serta ada *library numpy* yang berfungsi untuk operasi numerik dan manipulasi *array*, yang biasa digunakan dalam analisis data dan pemprograman ilmiah yang melakukan perhitungan matematis yang kompleks.

Pada tahap selanjunya, pengambilan data yang mana nantinya akan menyantumkan *id link* yang ada pada aplikasi. Sedangkan pada kasus ini peneliti menggunakan aplikasi SIREKAP, dengan ini peneliti menggunakan *id link* aplikasi tersebut yaitu " id.go.kpu.sirekap2024". Data yang didapatkan berupa *username*, ulasan/*content*, *rating,* waktu dan tanggal, dan lainnya sebanyak 2000 data. Dari banyaknya data yang diambil, penulis melakukan pemilahan data dan yang tersisa hanya data pada bagian *content*.

Pengambilan data komentar menggunakan *python* menggunakan source code seperti di bawah ini.

```
from google_play_scraper import Sort, reviews
result, continuation_token = reviews(
        'id.go.kpu.sirekap2024',
        lang='id',
        country='id',
        sort=Sort.MOST_RELEVANT,
        count=2000,
        filter_score_with=None
)
```

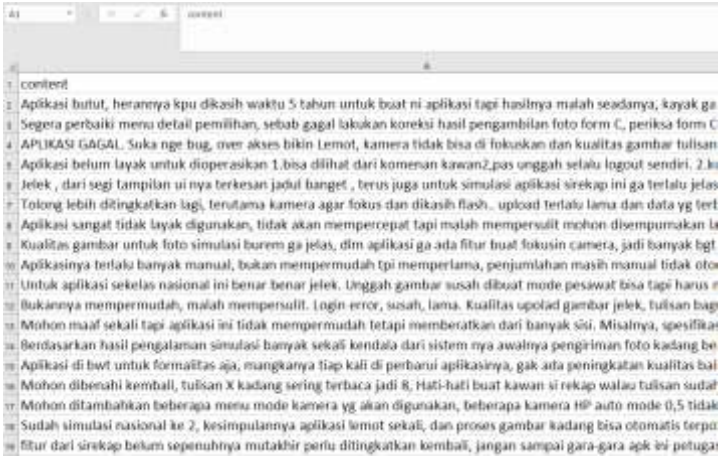
Setelah dilakukan pengambilan data, data yang sudah berada dalam *DataFrame* kemudian disimpan dalam bentuk CSV dengan *source code* yaitu:

```
my_data.to_csv("komentar.csv", index = False)
```

Data yang sudah berbentuk file cvs, bisa di lakukan pemanggilan kembali nantinya, agar data tidak berubah ubah lagi. Dikarenakan setiap menjalan source sode untuk mendapatkan data, data pasti akan berubah, maka dilakukan pemanggilan data yang sudah disimpan terlebih dahulu. Pemanggilan data tersebut dapat menggunakan source code seperti dibawah ini.

```
df_data = pd.read_csv("data_komentar.csv")
df_data.head()
```

Setelah proses tersebut maka data komentar telah tersimpan dalam bentuk file CSV. Gambar 4.2 merupakan data yang telah tersipan dalam bentuk CSV.



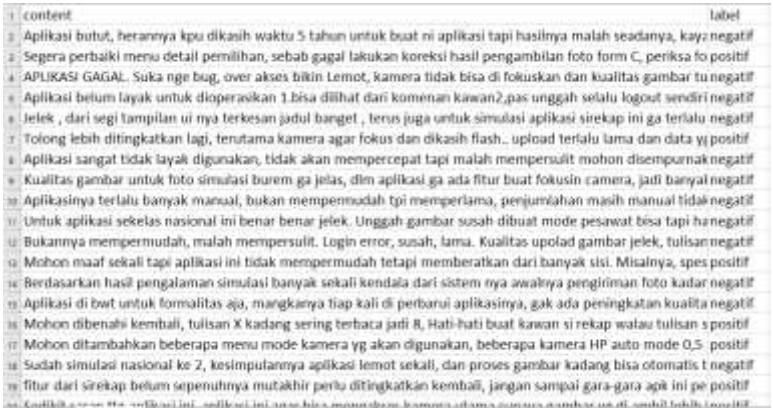
	content
1	content
2	Aplikasi butut, heranmya kpu dikasih waktu 5 tahun untuk buat ni aplikasi tapi hasilnya malah seadanya, kayak ga
3	Segera perbaiki menu detail pemilihan, sebab gagal lakukan koreksi hasil pengambilan foto form C, periksa form C
4	APLIKASI GAGAL. Suka nge bug, over akses bikin Lemot, kamera tidak bisa di fokuskan dan kualitas gambar tulisan
5	Aplikasi belum layak untuk dioperasikan 1.bisa dilihat dari komenan kawan2,pas unggah selalu logout sendiri. 2.ku
6	Jelek , dari segi tampilan ui nya terkesan jadul banget , terus juga untuk simulasi aplikasi sirekap ini ga terlalu jelas
7	Tolong lebih ditingkatkan lagi, terutama kamera agar fokus dan dikasih flash.. upload terlalu lama dan data yg terl
8	Aplikasi sangat tidak layak digunakan, tidak akan mempercepat tapi malah mempersulit mohon disempurnakan la
9	Kualitas gambar untuk foto simulasi burem ga jelas, dlm aplikasi ga ada fitur buat fokusin camera, jadi banyak bgt
10	Aplikasinya terlalu banyak manual, bukan mempermudah tpi memperlama, penjumlahan masih manual tidak oto
11	Untuk aplikasi sekelas nasional ini benar benar jelek. Unggah gambar susah dibuat mode pesawat bisa tapi harus r
12	Bukannya mempermudah, malah mempersulit. Login error, susah, lama. Kualitas upolad gambar jelek, tulisan bag
13	Mohon maaf sekali tapi aplikasi ini tidak mempermudah tetapi memberatkan dari banyak sisi. Misalnya, spesifikas
14	Berdasarkan hasil pengalaman simulasi banyak sekali kendala dari sistem nya awalnya pengiriman foto kadang be
15	Aplikasi di bwt untuk formalitas aja, mangkanya tiap kali di perbarui aplikasinya, gak ada peningkatan kualitas bal
16	Mohon dibenahi kembali, tulisan X kadang sering terbaca jadi 8, Hati-hati buat kawan si rekap walau tulisan sudah
17	Mohon ditambahkan beberapa menu mode kamera yg akan digunakan, beberapa kamera HP auto mode 0,5 tidak
18	Sudah simulasi nasional ke 2, kesimpulannya aplikasi lemot sekali, dan proses gambar kadang bisa otomatis terpo
19	fitur dari sirekap belum sepenuhnya mutakhir perlu ditingkatkan kembali, jangan sampai gara-gara apk ini petuga

Gambar 4. 2 Data dalam bentuk CSV.

B. Pelabelan

Tahap selanjutnya yaitu proses pelabelan yang mana data yang sudah diambil akan diberikan sentimen berupa positif dan negatif. Dalam proses ini seharusnya untuk menentukan nilai tersebut dilakukan oleh pakar bahasa atau yang terkait dibidangnya dan juga setidaknya membutuhkan dua orang atau lebih untuk menghindari perbedaan pendapat yang sama, serta menghindari subjektivitas dalam menentukan sentimen (Imron, 2019).

Pada proses pelabelan peneliti didampingi dua mahasiswi jurusan bahasa sastra dan pendidikan untuk membantu proses pelabelan manual ini. Untuk waktu yang dibutuhkan lumayan lama, dikarenakan banyaknya data. Berikut pada gambar 4.3 adalah data yang sudah diberi label sentimen.



	content	label
2	Aplikasi butut, herannya kpu dikasih waktu 5 tahun untuk buat ni aplikasi tapi hasilnya malah seadanya, kayz	negatif
3	Segera perbaiki menu detail pemilihan, sebab gagal lakukan koreksi hasil pengambilan foto form C, periksa fo	positif
4	APLIKASI GAGAL. Suka nge bug, over akses bikin Lemot, kamera tidak bisa di fokuskan dan kualitas gambar tu	negatif
5	Aplikasi belum layak untuk dioperasikan 1.bisa dilihat dari komenan kawan2,pas unggah selalu logout sendiri	negatif
6	Jelek , dari segi tampilan ui nya terkesan jadul banget , terus juga untuk simulasi aplikasi sirekap ini ga terlalu	negatif
7	Tolong lebih ditingkatkan lagi, terutama kamera agar fokus dan dikasih flash.. upload terlalu lama dan data yg	positif
8	Aplikasi sangat tidak layak digunakan, tidak akan mempercepat tapi malah mempersulit mohon disempurnak	negatif
9	Kualitas gambar untuk foto simulasi burem ga jelas, dlm aplikasi ga ada fitur buat fokusin camera, jadi banyal	negatif
10	Aplikasinya terlalu banyak manual, bukan mempermudah tpi memperlama, penjumlahan masih manual tidak	negatif
11	Untuk aplikasi sekelas nasional ini benar benar jelek. Unggah gambar susah dibuat mode pesawat bisa tapi ha	negatif
12	Bukannya mempermudah, malah mempersulit. Login error, susah, lama. Kualitas upolad gambar jelek, tulisan	negatif
13	Mohon maaf sekali tapi aplikasi ini tidak mempermudah tetapi memberatkan dari banyak sisi. Misalnya, spes	positif
14	Berdasarkan hasil pengalaman simulasi banyak sekali kendala dari sistem nya awalnya pengiriman foto kadar	negatif
15	Aplikasi di bwt untuk formalitas aja, mangkanya tiap kali di perbarui aplikasimya, gak ada peningkatan kualita	negatif
16	Mohon dibenahi kembali, tulisan X kadang sering terbaca jadi 8, Hati-hati buat kawan si rekap walau tulisan s	positif
17	Mohon ditambahkan beberapa menu mode kamera yg akan digunakan, beberapa kamera HP auto mode 0,5	positif
18	Sudah simulasi nasional ke 2, kesimpulannya aplikasi lemot sekali, dan proses gambar kadang bisa otomatis t	negatif
19	fitur dari sirekap belum sepenuhnya mutakhir perlu ditingkatkan kembali, jangan sampai gara-gara apk ini pe	positif

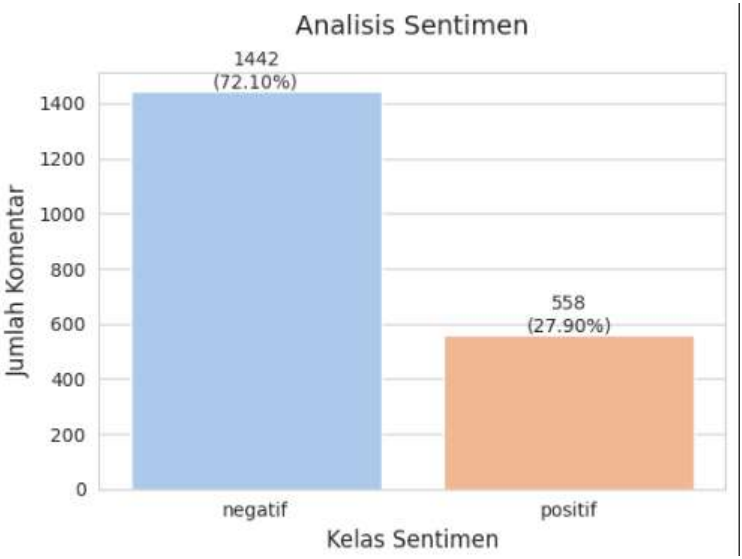
Gambar 4. 3 Data yang sudah sudah diberi label.

Adapun visualisasi untuk memperjelas berapa jumlah sentimen positif dan negatif yang telah diberikan label. Peneliti melakukan visualisasi dengan bantuan pemprograman menggunakan bahasa *python*. Pada bagian ini menggunakan *library "matplotlib"* serta *library "serborn"* yang digunakan dalam memvisualisasikan data yang ada. Berikut merupakan *source code* yang digunakan dalam visualisasi data

pada penelitian ini, yang mana akan dimulai dari *penginstallan library* terlebih dahulu.

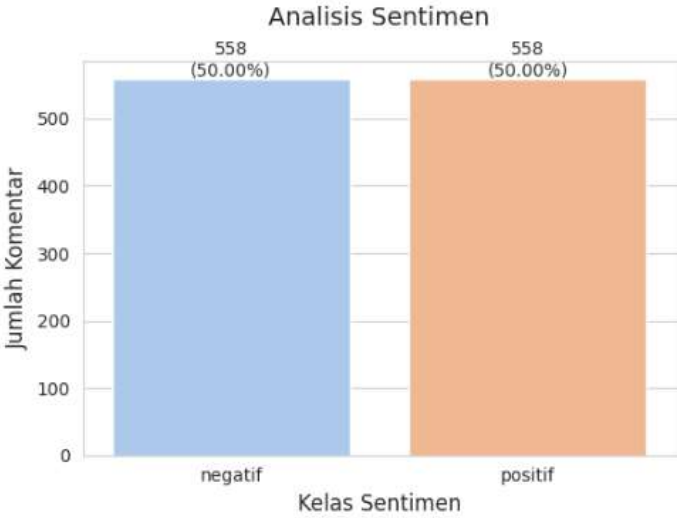
```
import matplotlib.pyplot  as plt
import seaborn as sns

sentiment_count = my_data['label'].value_counts()
sns.set_style('whitegrid')
fig, ax = plt.subplots(figsize=(6,  4))
ax = sns.barplot(x=sentiment_count.index,
y=sentiment_count.values,  palette='pastel')
plt.title('Analisis  Sentimen',  fontsize=14, pad=20)
plt.xlabel('Kelas  Sentimen',  fontsize=12)
plt.ylabel('Jumlah  Komentar', fontsize=12)

total = len(my_data['label'])

for i, count in enumerate(sentiment_count.values):
  Percentage = f'{100 * count / total:.2f}%'
  ax.text(i, count + 0.10, f'{count}\n({Percentage})',
   ha='center',  va='bottom')

plt.show()
```

pada *source code* ini nantinya akan membentuk diagram batang yang mana akan menunjukkan banyaknya jumlah data positif dan negatif. Adapun hasil dari visualisasi menggunakan pemprograman bahasa python dapat dilihat pada gambar 4.4.



Gambar 4. 4 *Output* dari Visualisasi

Dikarnekan data tidak seimbang, yang mana jumlah data negatif lebih banyak dari data positif. Maka, penulis melakukan penyeimbangan data dengan cara melakukan pengurangan data negatif, dengan jumlah awal data yaitu 1442 menjadi 558. Agar data negatif dan positif dapat dikatakan data seimbang. Jadi data yang awalnya berjumlah 2000, dengan adanya proses pengurangan tersebut data secara keseluruhan berubah menjadi 1116 data. Yang mana visualisasi diagram dapat dilihat pada gambar 4. 5.



Gambar 4. 5 Visualisasi setelah penyeimbangan data

C. Preprocessing

Tahapan ini memiliki beberapa proses yaang akan dilakukan dalam pengolahan data. Dikarenakan data komentar memilki karakteristik yang tidak tersetruktur yang banyak memuat noise. Maka, pada tahap ini memiliki tujuan untuk mengubah suatu data yang mentah, yang akan dilolah menjadi data yang bersih sehingga dapat dilakukan pengklasifikasian. Tahap *preprocessing* data pada penilitian ini dilakukan dengan melakukan 6 proses secra urut, diantaranya:

1. *Cleansing*

   Pada proses ini digunakan untuk membersihkan *noise* yaitu keslahan acak atau

varian dalam variable terukur (Buntoro, 2017), untuk itu kita harus menghilangkan *noise* tersebut. Seperti adanya emoji, tanda *hashtag* (#), simbol, angka, html, serta URL. Berikut merupakan kumpulan code yang digunakan dalam proses cleansing .

```
from ast import Return
import re
import string
import nltk
```

Pada *source code* diatas merupakan *import* fungsi dari berbagai *library* diantaranya 'ast' digunakan dalam pemprosesan pohon sintaks abstrak dengan menggunakan kelas 'return', 're' yang bertujuan untuk melakukan tahapan regular epression (regex), 'string' mengidentifikasi karakter khusus seperti huruf besar, kecil, angka dan lainnya, terakhir 'nltk' digunakan sebagai alat dalam melakukan pemprosesan data. Source code selanjutnya yaitu tahap clenasing itu sendiri yaitu sebagai berikut:

```
def remove_URL(tweet):
  if tweet is not None and isinstance(tweet, str):
    url = re.compile(r'https?://\S+|www\.\S+')
    return url.sub(r'', tweet)
  else:
```

```
  return tweet

def remove_html(tweet):
  if tweet is not None and isinstance(tweet, str):
    html = re.compile(r'<.*?>')
    return html.sub(r'', tweet)
  else:
    return tweet
```

Pada source code digunakan dalam proses penghapusan jika ada URL dan HTML yang ada pada DataFrame.

```
def remove_emoji(tweet):
  if tweet is not None and isinstance(tweet, str):
    emoji_pattern = re.compile("["
      u"\U0001F600-\U0001F64F"
      u"\U0001F300-\U0001F5FF"
      u"\U0001F680-\U0001F6FF"
      u"\U0001F700-\U0001F77F"
      u"\U0001F780-\U0001F7FF"
      u"\U0001F800-\U0001F8FF"
      u"\U0001F900-\U0001F9FF"
      u"\U0001FA00-\U0001FA6F"
      u"\U0001FA70-\U0001FAFF"
      u"\U0001F004-\U0001F0CF"
      u"\U0001F1E0-\U0001F1FF"
                        "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', tweet)
  else:
    return tweet
```

```
def remove_symbols(tweet):
  if tweet is not None and isinstance(tweet, str):
    tweet = re.sub(r'[^a-zA-Z0-9\s]', '', tweet)
  return tweet
def remove_numbers(tweet):
  if tweet is not None and isinstance(tweet, str):
    tweet = re.sub(r'\d', '', tweet)
  return tweet

my_data['cleaning']=my_data['content'].apply(lambd
a x: remove_URL(x))
my_data['cleaning']=my_data['cleaning'].apply(lamb
da x: remove_html(x))
my_data['cleaning']=my_data['cleaning'].apply(lamb
da x: remove_emoji(x))
my_data['cleaning']=my_data['cleaning'].apply(lamb
da x: remove_symbols(x))
my_data['cleaning']=my_data['cleaning'].apply(lamb
da x: remove_numbers(x))

my_data.head()
```

Pada bagian ini juga masih termasuk proses *cleansing*, yang mana fungsi code dari menghapus emoji dengan mencocokkan rentang karakter *unicode* yang mewakili emoji, penghapusan simbol, serta penghapusan angka yang tidak diperlukan pada *dataframe*. Kemudian setelah data sudah diberihkan dengan cara dipanggil dalam perintah *my_data ['cleansing']*, maka data akan ditampulkan

dalam perintah *'my_data.head()'*. Gambar 4.6 menunjukkan hasil dari data *cleansing*.

	content	label	cleaning
0	Aplikasi butut, herannya kpu dikasih waktu 5 t...	negatif	Aplikasi butut herannya kpu dikasih waktu tah...
1	Segera perbaiki menu detail pemilihan, sebab g...	positif	Segera perbaiki menu detail pemilihan sebab ga...
2	APLIKASI GAGAL. Suka nge bug, over akses bikin...	negatif	APLIKASI GAGAL Suka nge bug over akses bikin L...
3	Aplikasi belum layak untuk dioperasikan 1.bisa...	negatif	Aplikasi belum layak untuk dioperasikan bisa d...
4	Jelek , dari segi tampilan ui nya terkesan jad...	negatif	Jelek dari segi tampilan ui nya terkesan jadu...

Gambar 4. 6 Hasil dari *Cleansing*

2. *Case Folding*

Proses selanjutnya adalah *case folding* dimana proses ini mengganti huruf kapital ke huruf kecil dari *DataFrame* yang peneliti dapatkan. Tujuan dari *case folding* untuk menyamakan penggunaan huruf kapital, sehingga kata-kata yang sama tidak terdeteksi berbeda hanya karena perbedaan huruf kapital. Proses ini membantu meningkatkan konsistensi dan akurasi dalam analisis teks (Ardiani et al., 2020). *Source code* yang digunakan dalam melakukan *case folding* sebagai berikut:

```
def case_folding(text):
  if isinstance(text, str):
    lowercase_text = text.lower()
    return lowercase_text
```

```
  else:
    return text

my_data['case_folding']=my_data['cleaning'].apply(case_folding)

my_data.head()
```

Untuk hasil yang dihasilkan dalam proses tersebut bisa dilihat dalam gambar 4. 7 di bawah ini.

	content	label	cleaning	case_folding
0	Aplikasi butut, herannya kpu dikasih waktu 5 t...	negatif	Aplikasi butut herannya kpu dikasih waktu tah...	aplikasi butut herannya kpu dikasih waktu tah...
1	Segera perbaiki menu detail pemilihan, sebab g...	positif	Segera perbaiki menu detail pemilihan sebab ga...	segera perbaiki menu detail pemilihan sebab ga...
2	APLIKASI GAGAL. Suka nge bug, over akses bikin...	negatif	APLIKASI GAGAL Suka nge bug over akses bikin L...	aplikasi gagal suka nge bug over akses bikin l...
3	Aplikasi belum layak untuk dioperasikan 1.bisa...	negatif	Aplikasi belum layak untuk dioperasikan bisa d...	aplikasi belum layak untuk dioperasikan bisa d...
4	Jelek , dari segi tampilan ui nya terkesan jad...	negatif	Jelek dari segi tampilan ui nya terkesan jadu...	jelek dari segi tampilan ui nya terkesan jadu...

Gambar 4. 7 Hasil dari *Case Folding*

3. Normalisasi

Proses selanjutnya yaitu normalisasi *(slang word)* tahap ini nantinya akan digunakan dalam memperbaiki kata yang disingkat, pengurangan variasi, mengkatkan kualitas data (SHELEMO, 2023). Dengan tujuan mengubah teks menjadi bentuk yang konsisten serta seragam, seperti memperbaiki kesalahan penulisan. *Source code* yang peneliti gunakan sebagai berikut:

```
def replace_taboo_words(text, kamus_tidak_baku):
```

```
if isinstance(text, str):
  words = text.split()
  replaced_words = []
  kalimat_baku = []
  kata_diganti = []
  kata_tidak_baku_hash = []

  for word in words :
    if word in kamus_tidak_baku:
      baku_word = kamus_tidak_baku[word]
      if isinstance(baku_word, str) and
      all(char.isalpha() for char in baku_word):
        replaced_words.append(baku_word)
        kalimat_baku.append(baku_word)
        kata_diganti.append(word)
        kata_tidak_baku_hash.append(hash(word))

    else:
      replaced_words.append(word)
  replaced_text = ' '.join(replaced_words)
else:
  replaced_text = ''
  kalimat_baku = []
  kata_diganti = []
  kata_tidak_baku_hash = []

return replaced_text, kalimat_baku, kata_diganti,
kata_tidak_baku_hash

data = pd.DataFrame(my_data[['content', 'label',
'cleaning', 'case_folding']])
data.head()
```

```
kamus_data=pd.read_excel("kamuskatabaku.xls
x")
kamus_tidak_baku=dict(zip(kamus_data['tidak_b
aku'], kamus_data['kata_baku']))
```

Fungsi *'replace_taboo_words'* untuk membersihkan teks dari kata-kata tidak baku atau tabu, menggantinya dengan bentuk baku yang lebih formal. Dengan memuat kamus kata baku dan tidak baku. Hal ini dapat meningkatkan kualitas analisis teks, seperti analisis sentimen atau pemodelan bahasa, dengan memastikan bahwa kata-kata yang digunakan adalah konsisten dan sesuai dengan norma bahasa yang baku. Hasil dari pemprosesan data pada tahap normalisasi dapat dilihat pada gambar 4.8.

case_folding	normalisasi
aplikasi butut herannya kpu dikasih waktu tah...	aplikasi butut herannya kpu dikasih waktu tahu...
segera perbaiki menu detail pemilihan sebab ga...	segera perbaiki menu detail pemilihan sebab ga...
aplikasi gagal suka nge bug over akses bikin l...	aplikasi gagal suka nge bug over akses bikin l...
aplikasi belum layak untuk dioperasikan bisa d...	aplikasi belum layak untuk dioperasikan bisa d...
jelek dari segi tampilan ui nya terkesan jadu...	jelek dari segi tampilan ui ya terkesan jadul ...
tolong lebih ditingkatkan lagi terutama kamera...	tolong lebih ditingkatkan lagi terutama kamera...

Gambar 4. 8 Hasil dari Normalisas

4. *Tokenize*

Tahap tokenization merupakan tahap dimana memecahakan kalimat/teks menjadi beberapa

bagian-bagian kata. Berikut *source code* yang dari tahap *tokenize*.

```
def tokenize(text):
  tokens = text.split()
  return tokens

df['tokenize']  = df['normalisasi'].apply(tokenize)

df.head()
```

Sehingga, hasil dari tahapan tokenization dapat ditunjukkan pada gambar 4. 9 dibawah ini.

normalisasi	tokenize
aplikasi butut herannya kpu dikasih waktu tahu...	[aplikasi, butut, herannya, kpu, dikasih, wakt...
segera perbaiki menu detail pemilihan sebab ga...	[segera, perbaiki, menu, detail, pemilihan, se...
aplikasi gagal suka nge bug over akses bikin l...	[aplikasi, gagal, suka, nge, bug, over, akses,...
aplikasi belum layak untuk dioperasikan bisa d...	[aplikasi, belum, layak, untuk, dioperasikan, ...
jelek dari segi tampilan ui ya terkesan jadul ...	[jelek, dari, segi, tampilan, ui, ya, terkesan...

Gambar 4. 9 Hasil dari *Tokenize*

5. *Stopword*

Stopword merupakan proses menghapus kata yang memiliki fungsi tetapi tidak memiliki arti atau makna. Untuk melakukan *stopword* membutuhkan *library* NLTK dan *corpus stopword* dari *library* NLTK. Adapun code untuk penginstallan library stopword sebagai berikut.

```
from nltk.corpus import stopwords

nltk.download('stopwords')
```

```
stop_words = stopwords.words('indonesian')
```

setelah *library* terinstal dilanjutkan dengan source code untuk menjalakan data yang sudah ada, source codenya sebagai berikut:

```
def stopwords(text):
  return [word for word in text if word not in stop_words]

df['stopwords'] = df['tokenize'].apply(lambda x:stopwords(x))

df.head(20)
```

dari *source code* tersebut akan didapatkan hasil seperti pada gambar 4. 10.

	tokenize	stopwords
	[aplikasi, butut, herannya, kpu, dikasih, wakt...	[aplikasi, butut, herannya, kpu, dikasih, nih,...
	[segera, perbaiki, menu, detail, pemilihan, se...	[perbaiki, menu, detail, pemilihan, gagal, lak...
	[aplikasi, gagal, suka, nge, bug, over, akses,...	[aplikasi, gagal, suka, nge, bug, over, akses,...
	[aplikasi, belum, layak, untuk, dioperasikan, ...	[aplikasi, layak, dioperasikan, komentar, kawa...
	[jelek, dari, segi, tampilan, ui, ya, terkesan...	[jelek, segi, tampilan, ui, ya, terkesan, jadu...

Gambar 4. 10 Hasil dari *Stopword*

6. *Steaming*

Tahap *stemming* adalah tahap mengubah kata yang memiliki imbuhan menjadi bentuk kata dasar. Pada proses tahapan stemming perlu disiapkan *library* yang akan digunakan yaitu *library sastrawi*. Maka untuk menggunakan library sastrawi perlu dilakukan instalasi terlebih dahulu. Adapun code

untuk membuat fungsi *steaming* dan mengaplikasikannya pada data komentar sebagai berikut:

```
!pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer

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

def stem_text(text):
    return [stemmer.stem(word) for word in text]

df['steaming'] = df['stopwords'].apply(lambda x: ' '.join(stem_text(x)))

df.head()
```

Selain adanya penginstalan *library* sastrawi juga adanya mengimpor *'StemmerFactory'* dari pustaka Sastrawi. Kelas ini digunakan untuk membuat objek stemmer yang dapat digunakan untuk melakukan *stemming* pada teks dalam bahasa Indonesia. kelas *'PorterStemmer'* digunakan untuk bahasa Inggris. Ini menghapus akhiran dari kata-kata dalam bahasa Inggris untuk mendapatkan bentuk dasarnya. Kelas '*SnowballStemmer*' adalah algoritma yang lebih canggih dibandingkan Porter Stemmer dan mendukung beberapa bahasa, termasuk bahasa

Inggris dan bahasa lainnya. *Snowball Stemmer* juga dapat digunakan untuk bahasa Indonesia jika diatur dengan benar. Gambar 4. 11 merupakan hasil dari proses steaming.

stopwords	steaming
[aplikasi, butut, herannya, kpu, dikasih, nih,...	aplikasi butut heran kpu kasih nih aplikasi ha...
[perbaiki, menu, detail, pemilihan, gagal, lak...	baik menu detail pilih gagal laku koreksi hasi...
[aplikasi, gagal, suka, nge, bug, over, akses,...	aplikasi gagal suka nge bug over akses bikin l...
[aplikasi, layak, dioperasikan, komentar, kawa...	aplikasi layak operasi komentar kawanpas ungga...
[jelek, segi, tampilan, ui, ya, terkesan, jadu...	jelek segi tampil ui ya kes jadul banget simul...

Gambar 4. 11 Hasil dari *Steaming*

D. Split Data

Spilt data merupakan 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 latih 80% 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 sebagai berikut.

```
!pip install scikit-learn

from sklearn.model_selection import train_test_split
```

```
X = df ['normalisasi']
y = df ['label']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, stratify=y, random_state=42)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Output yang dihasilkan pada source code tersebut terdapat pada gambar 4.12.

```
((892,), (224,), (892,), (224,))
```

Gambar 4. 12 Hasil dari split data

Data yang awalnya 2000 dibagi menjadi data training dengan jumlah 1600 dan utuk data test yaitu dengan jumlah 400. Sebelum dilakukannya TF – IDF peneliti melakukan beberapa *impor library* yang akan digunakan dalam melakukan tahap yang dilakukan saat TF – IDF, *library yang digunakan sebagai berikut:*

```
from sklearn. preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import
RandomizedSearchCV
from sklearn.metrics import classification_report
```

Library diatas nantinya akan di gunakan pada saat melakukan proses TF – IDF, juga adanya

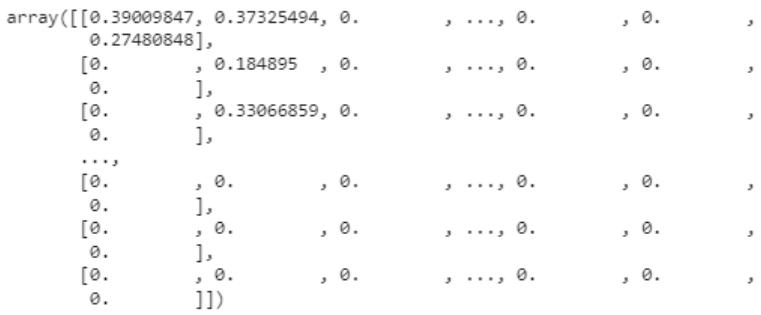
pengkalsifikasian metode yang penulis gunakan serta adanya penggunaan saat evaluasi model dan pembentukan matric.

E. TF – IDF

Tahap ini merupakan proses merubah kata menjadi vector, TF-IDF juga proses menghitung kemunculan kata dalam kalimat. Dalam python, TF - IDF memerlukan *library sklearn* dengan beberapa kelas seperti *TfidfVectorizer* dan juga kelas *LabelEncoder*. Adapun *source code* yang digunakan dalam melakukan TF – IDF yaitu sebagai berikut:

```
Encoder = LabelEncoder()
y_train = Encoder.fit_transform(y_train)
y_test = Encoder.fit_transform(y_test)

Tfidf_vect = TfidfVectorizer(max_features=50)
Tfidf_vect.fit(df['normalisasi'])
Train_x_Tfidf = Tfidf_vect.transform(X_train)
Test_x_Tfidf = Tfidf_vect.transform(X_test)


Train_x_Tfidf.toarray()
```

Hasil dari source code tersebut dapat dilihat pada gambar 4. 13.



Gambar 4. 13 Hasil dari TF - IDF

Hasil dari pembobotan dengan TF – IDF menghasilkan kalimat yang sudah menjadi kumpulan array yang menjadi suatu matriks, dimana setiap baris mewakili setiap dokumen, sedangkan pada setiap kolom mewakili seluruh kata yang ada pada seluruh teks. Sebagai contoh peneliti menggunakan tiga komentar untuk perhitungan manualisasi untuk pembobotan TF – IDF sebagai berikut:

(Doc1)=" Aplikasi paling banyak minusnya. Kualitas gambar buram saat difoto (tidak bisa difokuskan manual), angka digital sering salah dibaca, susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi, dan masih banyak lagi minus yang lain."

(Doc2)=" Aplikasi yang mlah menyusahkan. Fokus foto yang tidak bagus, scan manual, *force close*, dan jaringan yg sibuk. Aplikasi yang dicoba pada saat pemilu belum

berlangsung saja sudah punya kualitas yang buruk, apalagi saat pemilu dilakukan."

(Doc3)=" Perbaiki lagi kualitas aplikasinya kalau mau rekapan hasil pemilu berjalan lancar. Fix sangatÃ‚Â² membagongkan ni aplikasi mencetÃ‚Â² tombol sampai jari pegel pun tetep ga bisa kebaca tuh sidik jari."

Setelah sistem melakukan preprocessing maka menjadi seperti ini:

(Doc1)=['aplikasi', 'minus', 'kualitas', 'gambar', 'buram', 'foto', 'fokus', 'manual', 'angka', 'digital', 'salah', 'baca', 'susah', 'edit', 'hasil', 'sesuai', 'baca', 'aplikasi', 'minus']

(Doc2)=['aplikasi', 'susah', 'fokus', 'foto', 'bagus', 'scan', 'manual', *'force'*, '*close*', 'jaring', 'sibuk', 'aplikasi', 'coba', 'milu', 'kualitas' 'buruk', 'milu']

(Doc3)=['baik', 'kualitas', 'aplikasi', 'rekap', 'hasil', 'milu', 'jalan', 'lancar', 'fix', 'bagong', 'nih', 'aplikasi', 'mencet', 'tombol', 'jari', 'gel', 'baca', 'tuh', 'sidik', 'jari']

Pada tahap selanjutnya dilakukan perhitungan dengan metode TF - IDF dengan tujuan membentuk *word vector* yang telah diberi bobot nilai. Pada metode TF - IDF terdapat dua kata yaitu TF (*Term Frequency*) dan IDF (*Invers Document Frequency*), dimana TF sendiri merupakan jumlah sebuah kata dari tiap dokumen, sedangkan IDF berfungsi mengurangi bobot

suatu kata apabila kemunculannya banyak tersebar diseluruh dokumen (Zidan, 2022). Dimana pada perhitungan pembobotan TF - IDF langkah awalnya dilakukan pencarian nilai TF terlebih dahulu. Berikut contoh perhitungan TF secara manualisasi yang terdapat pada tabel 4.1

Tabel 4. 1 Contoh perhitungan TF

Token	TF		
	D1	D2	D3
Aplikasi	2	2	2
Minus	2	0	0
Kualitas	1	1	1
Gambar	1	0	0
Buram	1	0	0
Foto	1	1	0
Fokus	1	1	0
Manual	1	1	0
Angka	1	0	0
Digital	1	0	0
Salah	1	0	0
Baca	2	0	1
Susah	1	1	0
Edit	1	0	0
Hasil	1	0	1

Sesuai	1	0	0
Bagus	0	1	0
Scan	0	1	0
Force	0	1	0
Close	0	1	0
Jaring	0	1	0
Sibuk	0	1	0
Coba	0	1	0
Milu	0	2	1
Buruk	0	1	0
Baik	0	0	1
Rekap	0	0	1
Jalan	0	0	1
Lancar	0	0	1
Fix	0	0	1
Bagong	0	0	1
Nih	0	0	1
Mencet	0	0	1
Tombol	0	0	1
Jari	0	0	2
Gel	0	0	1
Tuh	0	0	1
Sidik	0	0	1

Setelah didapatkan TF, supaya dapat masuk ke perhitungan IDF maka harus menentukan DF *(Document Frequency)* terlebih dahulu. Df sendiri merupakan banyaknya jumlah dokumen yang mengandung kata tersebut. Berikut contoh perhitungan DF yang terdapat pada tabel 4.2.

Tabel 4. 2 contoh Perhitungan DF

Token	TF			DF
	D1	D2	D3	
Aplikasi	2	2	2	6
Minus	2	0	0	2
Kualitas	1	1	1	3
Gambar	1	0	0	1
Buram	1	0	0	1
Foto	1	1	0	2
Fokus	1	1	0	2
Manual	1	1	0	3
Angka	1	0	0	1
Digital	1	0	0	1
Salah	1	0	0	1
Baca	2	0	1	3
Susah	1	1	0	2
Edit	1	0	0	1
Hasil	1	0	1	1

Sesuai	1	0	0	1
Bagus	0	1	0	1
Scan	0	1	0	1
Force	0	1	0	1
Close	0	1	0	1
Jaring	0	1	0	1
Sibuk	0	1	0	1
Coba	0	1	0	1
Milu	0	2	1	3
Buruk	0	1	0	1
Baik	0	0	1	1
Rekap	0	0	1	1
Jalan	0	0	1	1
Lancar	0	0	1	1
Fix	0	0	1	1
Bagong	0	0	1	1
Nih	0	0	1	1
Mencet	0	0	1	1
Tombol	0	0	1	1
Jari	0	0	2	2
Gel	0	0	1	1
Tuh	0	0	1	1
Sidik	0	0	1	1

Setelah menghitung nilai DF, maka langkah selanjunya adalah menghitung nilai IDF. Hitungan nilai IDF dapat di lihat pada tabel 4.3.

Tabel 4. 3 Menghitung IDF

Token	DF	D/DF	IDF	IDF + 1
			Log(D/DF)	
Aplikasi	6	0.5	-0.301	0.699
Minus	2	1.5	0.176	1.176
Kualitas	3	1	0	1
Gambar	1	3	0.477	1.477
Buram	1	3	0.477	1.477
Foto	2	1.5	0.176	1.176
Fokus	2	1.5	0.176	1.176
Manual	3	1	0	1
Angka	1	3	0.477	1.477
Digital	1	3	0.477	1.477
Salah	1	3	0.477	1.477
Baca	3	1	0	1
Susah	2	1.5	0.176	1.176
Edit	1	3	0.477	1.477
Hasil	1	3	0.477	1.477
Sesuai	1	3	0.477	1.477
Bagus	1	3	0.477	1.477

Scan	1	3	0.477	1.477
Force	1	3	0.477	1.477
Close	1	3	0.477	1.477
Jaring	1	3	0.477	1.477
Sibuk	1	3	0.477	1.477
Coba	1	3	0.477	1.477
Milu	3	1	0	1
Buruk	1	3	0.477	1.477
Baik	1	3	0.477	1.477
Rekap	1	3	0.477	1.477
Jalan	1	3	0.477	1.477
Lancar	1	3	0.477	1.477
Fix	1	3	0.477	1.477
Bagong	1	3	0.477	1.477
Nih	1	3	0.477	1.477
Mencet	1	3	0.477	1.477
Tombol	1	3	0.477	1.477
Jari	2	1.5	0.176	1.176
Gel	1	3	0.477	1.477
Tuh	1	3	0.477	1.477
Sidik	1	3	0.477	1.477

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

Tabel 4. 4 Hasil Perhitungan Tf – IDF

Token	TF			IDF	IDF +1	TF *IDF		
	D1	D2	D3	Log (D/DF		D1	D2	D3
Aplikasi	2	2	2	-0.301	0.699	1.398	1.398	1.398
Minus	2	0	0	0.176	1.176	2.352	0	0
Kualitas	1	1	1	0	1	1	1	1
Gambar	1	0	0	0.477	1.477	1.477	0	0
Buram	1	0	0	0.477	1.477	1.477	0	0
Foto	1	1	0	0.176	1.176	1.176	1.176	0
Fokus	1	1	0	0.176	1.176	1.176	1.176	0
Manual	1	1	0	0	1	1	1	0
Angka	1	0	0	0.477	1.477	1.477	0	0
Digital	1	0	0	0.477	1.477	1.477	0	0
Salah	1	0	0	0.477	1.477	1.477	0	0
Baca	2	0	1	0	1	2	0	1
Susah	1	1	0	0.176	1.176	1.176	1.176	0
Edit	1	0	0	0.477	1.477	1.477	0	0
Hasil	1	0	1	0.477	1.477	1.477	0	1.477
Sesuai	1	0	0	0.477	1.477	1.477	0	0

Bagus	0	1	0	0.477	1.477	0	1.477	0
Scan	0	1	0	0.477	1.477	0	1.477	0
Force	0	1	0	0.477	1.477	0	1.477	0
Close	0	1	0	0.477	1.477	0	1.477	0
Jaring	0	1	0	0.477	1.477	0	1.477	0
Sibuk	0	1	0	0.477	1.477	0	1.477	0
Coba	0	1	0	0.477	1.477	0	1.477	0
Milu	0	2	1	0	1	0	2	1
Buruk	0	1	0	0.477	1.477	0	1.477	0
Baik	0	0	1	0.477	1.477	0	0	1.477
Rekap	0	0	1	0.477	1.477	0	0	1.477
Jalan	0	0	1	0.477	1.477	0	0	1.477
Lancar	0	0	1	0.477	1.477	0	0	1.477
Fix	0	0	1	0.477	1.477	0	0	1.477
Bagong	0	0	1	0.477	1.477	0	0	1.477
Nih	0	0	1	0.477	1.477	0	0	1.477
Mencet	0	0	1	0.477	1.477	0	0	1.477
Tombol	0	0	1	0.477	1.477	0	0	1.477
Jari	0	0	2	0.176	1.176	0	0	2.352
Gel	0	0	1	0.477	1.477	0	0	1.477
Tuh	0	0	1	0.477	1.477	0	0	1.477
Sidik	0	0	1	0.477	1.477	0	0	1.477

Sehingga apabila hasi TF – IDF dituliskan dalam bentuk array menjadi seperti di bawah ini:

Array([

[1.398, 2.352, 1, 1.477, 1.477, 1.176, 1.176, 1, 1.477, 1.477, 1.477, 2, 1.176, 1.477, 1.477, 1.477, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 , 0, 0],

[1.398, 0, 1, 0, 0, 1.176, 1.176, 1, 0, 0, 0, 0, 1.176, 0, 0, 0, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 2, 1.477, 0, 0, 0, 0, 0, 0, 0, 0, 0 ,0 , 0, 0, 0],

[1.398, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1.477, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 2.352, 1.477, 1.477, 1.477]

])

Dimana setiap barisnya mengimplementasikasikan proses TF – IDF setiap kata pada dokumen. Terbukti bahwa TF menunjukkan banyaknya jumlah data, sedangkan IDF adalah menunjukkan sering tidaknya kata tersebut muncul pada setiap dokumen, semakin banyak muncul kata, nilai IDFnya semakin kecil.

F. Klasifikasi *Naïve Bayes*

Setelah dilakukannya pemprosesan 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 alogaritma *naive bayes* secara manual sebagai berikut:

(Doc (negatif))=['aplikasi', 'minus', 'kualitas', 'gambar', 'buram', 'foto', 'fokus', 'manual', 'angka', 'digital', 'salah', 'baca', 'susah', 'edit', 'hasil', 'sesuai', 'baca', 'aplikasi', 'minus']

(Doc (positif))=['baik', 'kualitas', 'aplikasi', 'rekap', 'hasil', 'milu', 'jalan', 'lancar', 'fix', 'bagong', 'nih', 'aplikasi', 'mencet', 'tombol', 'jari', 'gel', 'baca', 'tuh', 'sidik', 'jari']

Data tersebut merupakan data yang sudah di lakukan preprocessing. Yang mana nanti akan dihitung term yang berada pada kalimat positif dan negatif, ditampilkan pada tabel 4. 5.

Tabel 4. 5 Hasil vektor dan term pada klasifikasi NB

Token	Doc 1 (negatif)	Doc 2 (positif)
aplikasi	2	2
minus	2	0
kualitas	1	1
gambar	1	0

buram	1	0
foto	1	0
fokus	1	0
manual	1	0
angka	1	0
digital	1	0
salah	1	0
baca	2	1
susah	1	0
edit	1	0
hasil	1	1
sesuai	1	0
baik	0	1
rekap	0	1
milu	0	1
jalan	0	1
lancar	0	1
fix	0	1
bagong	0	1
nih	0	1
mencet	0	1
tombol	0	1
jari	0	2

gel	0	1
tuh	0	1
sidik	0	1
Term	**19**	**20**

Diperoleh: count positif 20, count negatif 19, dengan jumlah kata 30.

1. Hitung probabilitas prior yang mana setiap kategori ada 2, yaitu positif dan negatif :

   $P\left(\frac{positif}{negatif}\right) = \frac{x\,(positif/negatif}{|C|}$

   P (positif) = $\frac{fx\,(positif)}{|C|} = \frac{1}{2} = 0{,}5$

   P (negatif) = $\frac{fx\,(negatif)}{|C|} = \frac{1}{2} = 0{,}5$

2. Hitung probabilitas likelihood (dari 31 kata yang nanti di bagi menjadi 2 yaitu negatif (19) dan positif (20)) ;

   P (w(positif/negatif)) = $\frac{nk\,(positif/negatif)+1}{n\,(positif/negatif)+|T|}$

   1. Probabilitas (apilikasi)

      P (aplikasi/positif) = $\frac{2+1}{20+30} = \frac{3}{50} = 0{,}06$

      P (aplikasi/negatif) = $\frac{2+1}{19+30} = \frac{3}{49} = 0{,}0612$

   2. Probabilitas (minus)

      P (minus/positif) = $\frac{0+1}{20+30} = \frac{1}{50} = 0{,}02$

      P (minus/negatif) = $\frac{2+1}{19+30} = \frac{3}{49} = 0{,}0612$

3. Probabilitas (kualitas)

   P (kualitas/positif) = $\frac{1+1}{20+30} = \frac{2}{50}$ = 0,04

   P (kualitas/negatif) = $\frac{1+1}{19+30} = \frac{2}{49}$ = 0,0408

4. Probabilitas (gambar)

   P (gambar/positif) = $\frac{0+1}{20+30} = \frac{1}{50}$ = 0,02

   P (gambar/negatif) = $\frac{1+1}{19+30} = \frac{2}{49}$ = 0,0408

5. Probabilitas (buram)

   P (buram/positif) = $\frac{0+1}{20+30} = \frac{1}{50}$ = 0,02

   P (buram/negatif) = $\frac{1+1}{19+30} = \frac{2}{49}$ = 0,0408

6. Probabilitas (foto)

   P (foto/positif) = $\frac{0+1}{20+30} = \frac{1}{50}$ = 0,02

   P (foto/negatif) = $\frac{1+1}{19+30} = \frac{2}{49}$ = 0,0408

7. Probabilitas (fokus)

   P (fokus/positif) = $\frac{0+1}{20+30} = \frac{1}{50}$ = 0,02

   P (fokus/negatif) = $\frac{1+1}{19+30} = \frac{2}{49}$ = 0,0408

8. Probabilitas (manual)

   P (manual/positif) = $\frac{0+1}{20+30} = \frac{1}{50}$ = 0,02

   P (manual/negatif) = $\frac{1+1}{19+30} = \frac{2}{49}$ = 0,0408

9. Probabilitas (angka)

   P (angka/positif) = $\frac{0+1}{20+30} = \frac{1}{50}$ = 0,02

P (angka/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

10. Peobabilitas (digital)

    P (digital/positif) = $\frac{0+1}{20+30} = \frac{1}{50} = 0{,}02$

    P (digital/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

11. Probabilitas (salah)

    P (salah/positif) = $\frac{0+1}{20+30} = \frac{1}{50} = 0{,}02$

    P (salah/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

12. Probabilitas (baca)

    P (baca/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

    P (baca/negatif) = $\frac{2+1}{19+30} = \frac{3}{49} = 0{,}0612$

13. Probabilitas (susah)

    P (susah/positif) = $\frac{0+1}{20+30} = \frac{1}{50} = 0{,}02$

    P (susah/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

14. Probabilitas (edit)

    P (edit/positif) = $\frac{0+1}{20+30} = \frac{1}{50} = 0{,}02$

    P (edit/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

15. Probabilitas (hasil)

    P (hasil/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

    P (hasil/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

16. Probabilitas (sesuai)

P (sesuai/positif) = $\frac{0+1}{20+30} = \frac{1}{50} = 0{,}02$

P (sesuai/negatif) = $\frac{1+1}{19+30} = \frac{2}{49} = 0{,}0408$

17. Probabilitas (baik)

P (baik/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (baik/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

18. Probabilitas (rekap)

P (rekap/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (rekap/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

19. Probabilitas (milu)

P (milu/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (milu/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

20. Probabilitas (jalan)

P (jalan/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (jalan/negatif) = $\frac{0+1}{19+31} = \frac{1}{50} = 0{,}02$

21. Probabilitas (lancar)

P (lancar/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (lancar/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

22. Probabilitas (fix)

P (fix/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (fix/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

23. Probabilitas (bagong)

P (bagong/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (bagong/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

24. Probabilitas (nih)

P (nih/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (nih/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

25. Probabilitas (mencet)

P (mencet/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (mencet/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

26. Probabilitas (tombol)

P (tombol/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (tombol/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

27. Probabilitas (jari)

P (jari/positif) = $\frac{2+1}{20+30} = \frac{3}{50} = 0{,}06$

P (jari/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

28. Probabilitas (gel)

P (gel/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (gel/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

29. Probabilitas (tuh)

P (tuh/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (tuh/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

30. Probabilitas (sidik)

P (sidik/positif) = $\frac{1+1}{20+30} = \frac{2}{50} = 0{,}04$

P (sidik/negatif) = $\frac{0+1}{19+30} = \frac{1}{49} = 0{,}0204$

Setelah mendapat semua perhitungan dari probabilitas prior maupun probabilitas likelihood. Kita coba dalam menentukan sentimen dengan data uji yang mana sentimen data tersebut belum diketahui. Data uji yang sudah dilakukan preprocessing:['aplikasi', 'susah', 'fokus', 'foto', 'bagus', 'scan', 'manual', 'force', 'close', 'jaring', 'sibuk', 'aplikasi', 'coba', 'milu', 'kualitas' 'buruk', 'milu']

P $\left(\frac{positif}{negatif}\ d\right)$ = $\left(\frac{positif}{negatif}\right)$ x $\pi p\ (w|positif/negatif)$

Pada data uji kata yang masuk kedalam data training yaitu:

(Aplikasi, susah, fokus, foto, manual, milu, kualitas)

P (uji | positif)

= P (positif) x P (aplikasi) x P (susah) x P (fokus) x P (foto) x P (manual) x P (milu) x P (kualitas)

= 0,5 x 0,06 x 0,02 x 0,02 x 0,02 x 0,02 x 0,04 x 0,04

=7,68E-12

P (uji | negatif)

= P (negatif) x P (aplikasi) x P (susah) x P (fokus) x P (foto) x P (manual) x P (milu) x P (kualitas)

= 0,5 x 0,0612 x 0,0408 x 0,0408 x 0,0408 x 0,0408 x 0,0204 x 0,0408 = 7,06E-11

Dapat disimpulkan dalam menghitung probabilitas dalam data uji antara probabilitas positif dan negatif, yang mana nilai probabilitas tertinggi yaitu sebesar 7,06E-11 pada P (uji | negatif) sehingga komentar tersebut diklasifikasikan kedala sentimen **"negatif"**.

Dalam penelitian ini, perhitungan nilai sentimen juga dilakukan dengan menggunakan alogaritma *naive bayes* Dibantu menggunakan bahasa pemprograman python dengan model multinominalNB dan metode RandomizedSearchCV pada *library skelarn*. Pada pengujian pertama menggunakan salah satu contoh dari kalimat yang ada pada data yang digunakan seperti data di atas di uji menggunakan model alogaritma *naive bayes* ditambah menggunakan salah satu library untuk menjalankannya yaitu *scikit-learn* yang *CountVectorizer* yang digunakan untuk kumpulan teks yang diubah menjadi token seperti halnya pada tahap tokenize.

Adapun code yang digunakan sebagai berikut:

```
from sklearn.feature_extraction.text import CountVectorizer

doc_negatif = ['aplikasi', 'minus', 'kualitas', 'gambar', 'buram', 'foto', 'fokus', 'manual', 'angka', 'digital','salah', 'baca', 'susah', 'edit', 'hasil', 'sesuai', 'baca', 'aplikasi', 'minus']
doc_positif = ['baik', 'kualitas', 'aplikasi', 'rekap', 'hasil', 'milu', 'jalan', 'lancar', 'fix', 'bagong', 'nih' 'mencet', 'tombol','jari', 'gel', 'baca', 'tuh', 'sidik', 'jari']

data = doc_negatif + doc_positif
labels = [0] * len(doc_negatif) + [1] * len(doc_positif

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data)

model = MultinomialNB()

model.fit(X, labels)

test_data = ['aplikasi', 'susah', 'fokus', 'foto', 'susah', 'scan', 'manual', 'force', 'close', 'jaringan', 'sibuk', 'aplikasi', 'coba', 'minus', 'kualitas', 'buruk', 'milu']

X_test = vectorizer.transform(test_data)

prediksi = model.predict(X_test)

probabilitas = model.predict_proba(X_test)

print("Hasil Prediksi untuk Data Uji:")
```

```
for i, word in enumerate(test_data):
        print(f'Kata: {word}, Prediksi: {"Positif" if prediksi[i] ==
        1 else "Negatif"}, '
              f'Probabilitas Negatif:
              {probabilitas[i][0]:.4f}, '
              f'Probabilitas Positif: {probabilitas[i][1]:.4f}')

final_prediction = np.bincount(prediksi).argmax()
print(f'\nPrediksi Akhir untuk Data Uji: {"Positif" if
final_prediction == 1 else "Negatif"}')
```

Output dapat dilihat pada gambar 4. 14 dibawah ini:

```
Hasil Prediksi untuk Data Uji:
Kata: aplikasi, Prediksi: Negatif, Probabilitas Negatif: 0.6079, Probabilitas Positif: 0.3921
Kata: susah, Prediksi: Negatif, Probabilitas Negatif: 0.6740, Probabilitas Positif: 0.3260
Kata: fokus, Prediksi: Negatif, Probabilitas Negatif: 0.6740, Probabilitas Positif: 0.3260
Kata: foto, Prediksi: Negatif, Probabilitas Negatif: 0.6740, Probabilitas Positif: 0.3260
Kata: susah, Prediksi: Negatif, Probabilitas Negatif: 0.6740, Probabilitas Positif: 0.3260
Kata: scan, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: manual, Prediksi: Negatif, Probabilitas Negatif: 0.6740, Probabilitas Positif: 0.3260
Kata: force, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: close, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: jaringan, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: sibuk, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: aplikasi, Prediksi: Negatif, Probabilitas Negatif: 0.6079, Probabilitas Positif: 0.3921
Kata: coba, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: minus, Prediksi: Negatif, Probabilitas Negatif: 0.7561, Probabilitas Positif: 0.2439
Kata: kualitas, Prediksi: Negatif, Probabilitas Negatif: 0.5083, Probabilitas Positif: 0.4917
Kata: buruk, Prediksi: Negatif, Probabilitas Negatif: 0.5135, Probabilitas Positif: 0.4865
Kata: milu, Prediksi: Positif, Probabilitas Negatif: 0.3407, Probabilitas Positif: 0.6593

Prediksi Akhir untuk Data Uji: Negatif
```

Gambar 4. 14 Hasil Output dari Meode naive bayes (contoh kasus pada data)

Selanjutnya untuk *source code* diabwah ini akan menunjukkan menggunakan metode *naïve bayes* dengan keseluruhan data yang digunakan. Hasil yang akan ditampilkan yaitu *best params, train score, best*

score dan *test score*. Adapun code yang digunakan dalam klasifikasi *naive bayes* sebagai berikut:

```
# Menentukan parameter untuk pencarian
hyperparameter
param = {
   'fit_prior': [True, False],
   'alpha': np.logspace(-3, 3, 7)
}
# Menggunakan RandomizedSearchCV untuk pencarian
hyperparameter
model_nb = RandomizedSearchCV(MultinomialNB(),
param, cv=4, n_iter=50, n_jobs=-2, verbose=1,
random_state=42)
model_nb.fit(X_resampled, y_resampled)
# Menampilkan hasil terbaik
print("Best Parameters:", model_nb.best_params_)
print(model_nb.score(X_train_tfidf, y_train),
model_nb.best_score_, model_nb.score(X_test_tfidf,
y_test))
```

Output yang dihasilkan dalam klasifikasi menggunakan pemprogman bahasa *python* dapat dilihat pada gambar 4. 15.

```
Fitting 4 folds for each of 14 candidates, totalling 56 fits
Best Parameters: {'fit_prior': False, 'alpha': np.float64(1.0)}
0.9282511210762332 0.6827354260089686 0.6919642857142857
```

Gambar 4. 15 Hasil Klasifikasi *Naive Bayes*

Hasil pada gambar diatas menunjukkan bahwa best parameter model yang menghitung probabilitas

prior dari kelas. Untuk hasil train score yang memperoleh 0,9283 (92,83%) menunjukkan bahwa model memiliki akurasi yang baik pada data pelatihan. Untuk *best score* pada metode *naive bayes* diperoleh 0,6827 (68,27%) menunjukkan bahwa model diharapkan bekerja dengan baik dalam pemprosesan data yang tidak terlihat, berdasarkan data pelatihan.Sedangkan hasil test score yang memperoleh 0,6919 (69,19%) menunjukkan bahwa model memilki akurasi yang baik pada data pengujian.

G. Klasifikasi Support Vector Machine

Sebagaimana klasifikasi *naïve bayes* yang diproses setelah adanya *text preprocessing*, split validation data dan proses TF-IDF, klasifikasi SVM pun demikian. Klasifikasi SVM bekerja dengan mencari *hyperplane* yang memisahkan kelas-kelas dalam ruang fitur dengan margin maksimum. Jika persamaan *hyperplane* telah ditemukan maka persamaan yang > 0 komentar tersebut merupakan sentimen positif, akan tetapi < 0 komentar tersebut merupakan sentimen negatif. Contoh proses perhitungan sentimen dalam algoritma SVM secara manual sebagai berikut:

1. Langkah pertama yaitu menentukan vektor (titik data), yang mana sudah diketahui dalam label 4.5,

pada langkah ini di bagi dalam bentuk positif dan negatif, berikut vektor titik data:

a. Negatif

(x_1)= [2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

b. Positif

(x_2)= [2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1]

2. Menentukan *hyperplane* dengan mencari vektor bobot (w) dan bias (b). Di misalkan dengan ; [w . x + b = 0]. Selanjutnya yaitu menentukan w terlebih dahulu yaitu dengan : [w = x_2 – x_1]

w=[2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1] - [2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

w=[0, -2, 0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1]

kemudian dilanjut menentukan b dengan menggunakan salah satu titik data, yang peneliti gunakan yaitu titik data positif (x_2)

[b = - (w . x_2) + 1]

[b = - (0 . 2 + (-2) . 0 + 0 . 1 + (-1) . 0 + (-1) . 0 + (-1) . 0 + (-1) . 0 + (-1) . 0 + (-1) . 0 + (-1) . 0 + (-1) . 1 + (-1) . 0 + (-1) . 0 + 0 . 1 + (-1) . 0 + 1 . 1 + 1 .

1 + 1 . 1 + 1 . 1 + 1 . 1 + 1 . 1 + 1 .1 + 1 . 1 + 1 . 1 + 2 . 2 + 1 . 1 + 1 . 1 + 1 . 1 + 1 . 1) + 1]

[b = - [0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + (-1) + 0 + 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 4 + 1 + 1 + 1] + 1]

[b = - (-1 + 20) = -19 + 1] = -18

Jadi persamaan hyperplane yaitu: [w . x -18 = 0]

3. Tahap selanjutnya yaitu mencoba menentuka sentimen pada data uji menggunakan klasifikasi SVM. Peneliti memiliki data uji yang sudah melalui proses preprocessing, yaitu sebagai berikut:

   Data uji: ['aplikasi', 'susah', 'fokus', 'foto', 'bagus', 'scan', 'manual', 'force', 'close', 'jaring', 'sibuk', 'aplikasi', 'coba', 'milu', 'kualitas' 'buruk', 'milu']

   Langkah selanjutnya menentukan titik data:

   [2, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

   Setelah tititk data sudah ditentukan selanjutnya yaitu menghitung (w . x - 18), yang mana w menggunakan perhitungan sebelumnya.

   (w . x - 18) = (0 . 2 + (-2) . 0 + 0 . 1 + (-1) . 0 + (-1) .0 + (-1) . 1 + (-1) . 1 + (-1) . 1 + (-1) . 0 + (-1) . 0 + (-1) .0 + (-1) . 0 + (-1) . 1 + (-1) . 0 + 0 . 0 + (-1) . 0 + (-

1) . 0 + 1 . 0 + 1 . 0 + 1 . 2 + 1 . 0 + 1 . 0 + 1 . 0 + 1 . 0 + 1 . 0 + 1 . 0 + 2 . 0 + 1 . 0 + 1 . 0 + 1 . 0)

= (0 + 0 + 0 + 0 + 0 + 0 + 0 + (-1) + (-1) + (-1) + + (-1) + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 1)

= (-4) + (2) = -2

Jadi w . x -18 = -2 – 18 = -20. Dapat disimpulkan bahwa hasil dari data baru merupakan sentimen **"negatif"** dikarenakan hasil akhir didapatkan -20 yang mana hasilnya kurang dari 0 (-20 < 0).

Dalam penelitian ini, perhitungan nilai sentimen juga dilakukan dengan menggunakan alogaritma SVM. Pada pengujian pertama menggunakan salah satu contoh dari kalimat yang ada pada data yang digunakan seperti data di atas di uji menggunakan model alogaritma SVM. Menggunakan salah satu library untuk menjalankannya yaitu *scikit-learn* yang *CountVectorizer* yang digunakan untuk kumpulan teks yang diubah menjadi token seperti halnya pada tahap tokenize. Adapun code yang digunakan sebagai berikut:

```
doc_negatif = ['aplikasi', 'minus', 'kualitas', 'gambar', 'buram', 'foto', 'fokus', 'manual', 'angka', 'digital','salah', 'baca', 'susah', 'edit', 'hasil', 'sesuai', 'baca', 'aplikasi', 'minus']
```

```
doc_positif = ['baik', 'kualitas', 'aplikasi', 'rekap', 'hasil',
'milu', 'jalan', 'lancar', 'fix', 'bagong', 'nih' 'mencet',
'tombol', 'jari', 'gel', 'baca', 'tuh', 'sidik', 'jari']

data = doc_negatif + doc_positif
labels = [0] * len(doc_negatif) + [1] * len(doc_positif)


vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data)

model = SVC(probability=True) # Mengatur
probability=True untuk mendapatkan probabilitas

model.fit(X, labels)

test_data = ['aplikasi', 'susah', 'fokus', 'foto', 'susah',
'scan', 'manual', 'force', 'close', 'jaringan', 'sibuk',
'aplikasi', 'coba', 'minus', 'kualitas', 'buruk', 'milu']

X_test = vectorizer.transform(test_data)

prediksi = model.predict(X_test)

probabilitas = model.predict_proba(X_test)

print("Hasil Prediksi untuk Data Uji:")
for i, word in enumerate(test_data):
    print(f'Kata: {word}, Prediksi: {"Positif" if prediksi[i]
    == 1 else "Negatif"}, '
        f'Probabilitas Negatif:
        {probabilitas[i][0]:.4f}, '
         f'Probabilitas Positif: {probabilitas[i][1]:.4f}')
```

```
final_prediction = np.bincount(prediksi).argmax()
print(f'\nPrediksi Akhir untuk Data Uji: {"Positif" if final_prediction == 1 else "Negatif"}')
```

Output dapat dilihat pada gambar 4.15 dibawah ini:

```
Hasil Prediksi untuk Data Uji:
Kata: aplikasi, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: susah, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: fokus, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: foto, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: susah, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: scan, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: manual, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: force, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: close, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: jaringan, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: sibuk, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: aplikasi, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: coba, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: minus, Prediksi: Negatif, Probabilitas Negatif: 0.4590, Probabilitas Positif: 0.5410
Kata: kualitas, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: buruk, Prediksi: Negatif, Probabilitas Negatif: 0.5140, Probabilitas Positif: 0.4860
Kata: milu, Prediksi: Positif, Probabilitas Negatif: 0.5711, Probabilitas Positif: 0.4289

Prediksi Akhir untuk Data Uji: Negatif
```

Gambar 4. 16 Hasil Output dari Meode SVM (contoh kasus pada data)

Sedangkan untuk keseluruhan pengolahan data menggunkan metode SVM dibantu menggunakan salah satu bahasa pemprograman python dengan model SVC dan metode RandomizedSearchCV pada library skelarn. Hanya saja sebelum masuk model peneliti menentukan fungsi karnel yang mana menggunakan jenis linear dan RBF, serta fungsi 'gamma' yang digunakan dalam mengontrol seberapa jauh pengaruh suatu titik data. Hasil yang akan ditampilkan dalam code tersebut berupa best_params_, train score dan test

score. Adapun code yang digunakan dalam klasifikasi SVM sebagai berikut:

```
param = {
    C': np.logspace(-3, 3, 7), # Regularization parameter
    'kernel': ['linear', 'rbf'], # Jenis kernel
    'gamma': ['scale', 'auto']    # Parameter kernel
}

model_svm = RandomizedSearchCV(SVC(), param, cv=4, n_iter=50, n_jobs=-2, verbose=1, random_state=42)
model_svm.fit(X_resampled, y_resampled)
print("Best Parameters:", model_svm.best_params_)
print(model_svm.score(X_train_tfidf, y_train), model_svm.best_score_,
model_svm.score(X_test_tfidf, y_test))
```

hasil dari proses SVM dapat dilihat pada gambar 4. 17.

```
Fitting 4 folds for each of 28 candidates, totalling 112 fits
Best Parameters: {'kernel': 'rbf', 'gamma': 'auto', 'C': np.float64(1000.0)}
0.8957399103139013 0.6849775784753364 0.6964285714285714
```

Gambar 4. 17 Hasil Klasifikasi SVM

Hasil pada gambar diatas menunjukkan bahwa best parameter fungsi karnel yaitu linier. Untuk hasil train score yang memperoleh 0,8957 (89,57%) menunjukkan bahwa model memiliki akurasi yang baik pada data pelatihan. Untuk *best score* pada metode SVM diperoleh 0,6849 (68,49%)

menunjukkan bahwa model diharapkan bekerja dengan baik dalam pemprosesan data yang tidak terlihat, berdasarkan data pelatihan. Untuk hasil test score yang memperoleh 0,6964 (69,68%) menunjukkan bahwa model memilki akurasi yang baik pada data pengujian. Terdapat perbedaan dari cara manual dan pemprograman, yang mana cara manual tidak menggunakan fungsi karnel karena saat menentukan hyperplane sudah menemukan batas keputusan yang memisahkan dua kelas dalam ruang fitur. Maka peneliti tidak memerlukan fungsi karnel lagi dikarenakan fungsi ini digunakan saat mengubah data kedalam ruang fitur yang lebih tinggi, jika data tidak dapat dipisahkan secara linear dalam ruang asli (Arifin et al., 2021).

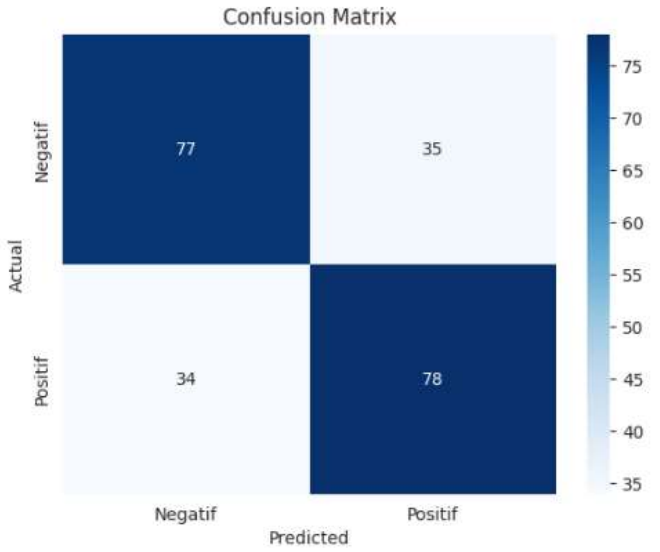
H. Evaluasi Model

Setelah melalui proses uji model, selanjutnya dilakukan tahap evaluasi model untuk menghetauhi performa pada model tersebut. Pada penelitian ini, peneliti melakukan perhitungan performa yang terdiri dari nilai akurasi, *presicion*, *recall* dan *f1 score* yang dihitung dari dua kelas, yang dimana nantinya nilai dari *confusion matrix* tersebut dapat digunakan untuk menghitung nilainya. Adapun untuk melihat *confusion*

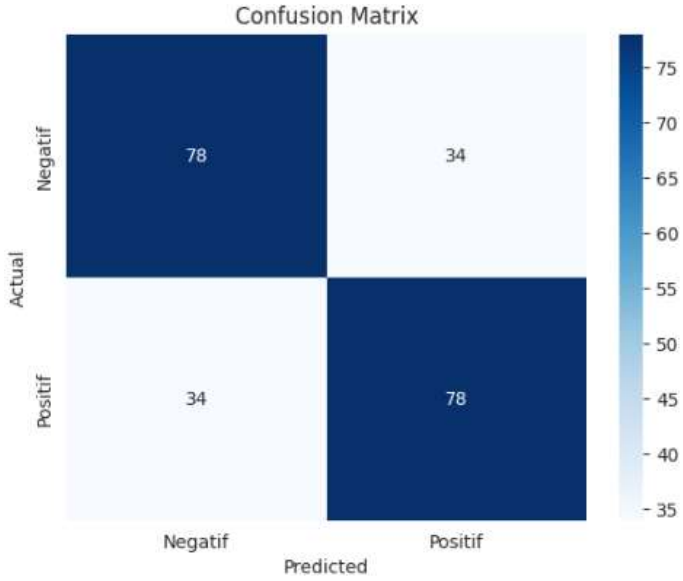
matrix ada pada tabel 2.2 pada bab sebelumnya. Sebelum melakukan tahap tersebut, dilakukan penginstallan library terlebih dahulu. Library yang digunakan yaitu "jcopml" yang digunakan dalam proses pemodelan dan evaluasi. Maka nantinya proses evaluasi model dapat berjalan sesuai yang diinginkan. Sebelum masuk perhitungan peneliti akan menampilkan visualisasi *confusion matrix* yang nantinya digunakan untuk mengetahui jumlah dari data yang dibutuhkan. Maka visualisasi dilakukan dengan menggunakan modul confusion_matrix. Adapun code untuk menampilkan confusion matrix NB dan SVM sebagai berikut:

```
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Negatif', 'Positif'], yticklabels=['Negatif', 'Positif'])
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion Matrix')
plt.show()
```

Hasil dari *confusion matrix* metode NB dan SVM tidak sama. Gambar 4. 18 merupakan tampilan dari hasil *confusion matrix* metode NB sedangkan gambar 4. 19 merupakan tampilan dari hasil *confusion matrix* metode SVM.



Gambar 4. 18 *Confusion Matrix Naïve Bayes*



Gambar 4. 19 *Confusion Matrix* SVM

Confusion matrix tersebut diambil dari data uji dengan jumlah total data uji yakni 224 dari 1116 data komentar yang telah diolah, yang mana perbandingan data latih dan data uji adalah 80:20. Data uji yang diproses sebanyak 224 dan sisa dari data tersebut merupakan data latih. Dalam *confusion matrix* terdapat informasi yang digunakan untuk menghitung nilai akurasi, presisi, *racall* dan *f1 score*, untuk klasifikasi *naive bayes* yakni 77 merupakan *true negative* di kelas aktual negatif dan diprediksi negatif, 78 *true positive* di kelas aktual positif dan diprediksi positif, 35 *false negative* di kelas aktual negatif namun diprediksi positif, serta 34 *false positif* di kelas aktual positif namun diprediksi negatif. Sedangkan dalam klasifikasi *support vector machine* yakni 78 merupakan *true negative* di kelas aktual negatif dan diprediksi negatif, 78 *true positive* di kelas aktual positif dan diprediksi positif, 34 *false negative* di kelas aktual negatif namun diprediksi positif, serta 34 *false positif* di kelas aktual positif namun diprediksi negatif. Untuk lebih mudah dalam memahami peneliti akan menjabarkan dalam tabel 4.6.

Tabel 4. 6 Hasil Evaluasi Model

Klasifikasi	TN	TP	FN	FP
Naive Bayes	77	78	34	35
SVM	78	78	34	34

Langkah selanjutnya yakni menghitung nilai akurasi, presisi, *recall*, dan *f1 score* dari klasifikasi *naive bayes* terlebih dahulu:

A $= \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% = \frac{78+77}{78+77+35+34} \text{ x } 100\%$

$= \frac{155}{224} \text{ x } 100\% = 0{,}69 \text{ x } 100\% = 69\%$

P $= \frac{TP}{TP+FP}\ x\ 100\% = \frac{78}{78+35} \text{ x } 100\% = \frac{78}{113} \text{ x } 100\%$

$= 0{,}69 \text{ x } 100\% = 69\%$

R $= \frac{TP}{TP+FN}\ x\ 100\% = \frac{78}{78+34} \text{ x } 100\% = \frac{78}{112} \text{ x } 100\%$

$= 0,\ 696 \text{ x } 100\% = 69{,}6\% = 70\%$

F1 score $= 2\ x \frac{Precision\ .\ Recall}{Precision+Recall}\ x\ 100\%$

$= 2 \text{ x } \frac{0{,}69\ .\ 0{,}696}{0{,}69+0{,}696} \text{ x } 100\%$

$= 2 \text{ x } \frac{0{,}48024}{1{,}386} \text{ x } 100\%$

$= 2 \text{ x } 0{,}3464 \text{ x } 100\%$

$= 0{,}69 \text{ x } 100\ \%$

$= 69\ \%$

Dilanjutkan dengan menghitung nilai akurasi, presisi, *recall,* dan *f1 score* dari klasifikasi *support vector machine* sebagai berikut:

A $= \frac{TP+TN}{TP+TN+FP+FN} \; x \; 100\% = \frac{78+78}{78+78+34+34} \text{ x } 100\%$

$= \frac{156}{224} \text{ x } 100\% = 0{,}696 \text{ x } 100\% = 69{,}6\% = 70\%$

P $= \frac{TP}{TP+FP} \; x \; 100\% = \frac{78}{78+34} \text{ x } 100\% = \frac{78}{112} \text{ x } 100\%$

$= 0{,}696 \text{ x } 100\% = 69{,}6\% = 70\%$

R $= \frac{TP}{TP+FN} \; x \; 100\% = \frac{78}{78+34} \text{ x } 100\% = \frac{78}{112} \text{ x } 100\%$

$= 0,\ 696\text{x } 100\% = 69{,}6\ \% = 70\%$

F1 score $= 2 \; x \frac{Precision \; . \; Recall}{Precision+Recall} \; x \; 100\%$

$= 2 \text{ x } \frac{0{,}696 \; . \; 0{,}696}{0{,}696+0{,}696} \text{ x } 100\%$

$= 2 \text{ x } \frac{0{,}484415}{1{,}392} \text{ x } 100\%$

$= 2 \text{ x } 0{,}384 \text{ x } 100\%$

$= 0{,}696 \text{ x } 100\ \%$

$= 69{,}6\%$

$= 70\%$

Hasil perhitungan manual pada peforma klasifikasi *naive bayes* dan kalsifikasi *support vector machine* dirangkum dalam tabel 4.7

Tabel 4. 7 Hasil Performa Klasifikasi

Klasifikasi	Akurasi	Presisi	Recall	F1 Score
Naive Bayes	69%	69%	70%	69%
SVM	70 %	70%	70%	70%

Peneliti juga melakukan perhitungan evaluasi model menggunakan sistem dengan menggunakan code yang ditunjukkan pada gambar 4. 20 yaitu klasifikasi *naive bayes* dan gambar 4. 21 yaitu klasifikasi SVM.

```
[ ]  # Naive Bayes
     y_pred = model_nb.predict(X_test_tfidf)
     print(classification_report(y_test, y_pred))
```

Gambar 4. 20 *Source Code* Evaluasi Model *Naïve Bayes*

```
[ ]  # SVM
     y_pred = model_svm.predict(X_test_tfidf)
     print(classification_report(y_test, y_pred))
```

Gambar 4. 21 *Source Code* Evaluasi Model SVM

Hasil yang didapatkan dalam pemprogaman diatas dapat dilihat dalam gambar 4.22 yang menampilkan hasil dari metode naive bayes dan 4. 23 yang menampilkan hasil dari SVM.

	precision	recall	f1-score	support
0	0.69	0.69	0.69	112
1	0.69	0.70	0.69	112
accuracy			0.69	224
macro avg	0.69	0.69	0.69	224
weighted avg	0.69	0.69	0.69	224

Gambar 4. 22 Hasil Dari Evaluasi Mode NB

	precision	recall	f1-score	support
0	0.70	0.70	0.70	112
1	0.70	0.70	0.70	112
accuracy			0.70	224
macro avg	0.70	0.70	0.70	224
weighted avg	0.70	0.70	0.70	224

Gambar 4. 23 Hasil Dari Evaluasi Mode SVM

Dari perhitungan yang dilakukan secara manual ataupun dengan sistem di dapatkan hasil yang sama. Akurasi yang merupakan proporsi dari prediksi yang benar dibandingkan dengan total jumlah prediksi. Pada klasifikasi *naive bayes nilai* akurasi mendapatkan 69% serta pada metode SVM nilai akurasi didapatkan 70%, yang mana nilai akurasi pada metode SVM lebih tinggi 1% dibanding dengan nilai akurasi pada metode *naive bayes*. Perhitungan presisi yang merupakan proporsi dari prediksi positif yang benar dibandingkan dengan total prediksi positif yang dibuat oleh model,

atau menghitung seberapa akurat model dalam mengidentifikasi kelas positif.

Dalam klasifikasi *naive bayes* mendapatkan 69%, sedangkan untuk klasifikasi SVM mendapatkan 70%. Selanjtnya yaitu ada perhitungan *recall* yang merupakan proporsi dari prediksi positif yang benar dibandingkan dengan total jumlah aktual dari kelas positif atau mengukur seberapa baik model dalam menangkap semua kasus positif. Pada klasifikasi *naive bayes* dan SVM didapatkan hasil yang sama besar yaitu 70%. Terakhir yaitu perhitungan *f1 score* yang merupakan harmonisasi antara presisi dan recall serta membantu dalam mengevaluasi model secara komprehensif dalam memberikan keseimbangan antara presisi dan *recall*.

Pada klasifikasi *naive bayes* didapatkan sebesar 69%, sedangkan pada klasifikasi SVM didapatkan sebesar 70%. Hasil perbandingan metode analisis sentimen tidak selalu bersifat mutlak, masing – masing tergantung dengan data yang diolah. Dalam kasus analisis sentimen aplikasi SIREKAP menggunakan komentar di *play store*, dengan membandingkan hasil peforma dari metode klasifikasi *naïve bayes* dan SVM, menunjukkan hasil akhir bahwa, metode klasifikasi

SVM lebih baik dari pada metode klasifikasi naive bayes, yang mana dari nilai akurasi, presisi, serta *f1-score* pada SVM lebih tinggi dari performa dari metode naive bayes walaupun pada hasil nilai recalnya seimbang.

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa:

1. Analisis sentimen menggunakan metode klasifikasi *naïve bayes* dan SVM untuk analisis sentimen pada aplikasi SIREKAP menggunakan komentar di *play store* dapat dilakukan dengan baik. Dari data komentar awal berjumlah 2000, data dengan jumlah sentiment 1442 sentimen negatif dan 558 sentimen positif. Dikarenakan data tidak seimbang maka penulis melakukan pengurangan pada data negatif, yang mana data menjadi 1116, dengan jumlah sentimen negatif dan sentimen positif masing – masing berjumlah 558. Setelah itu dilakukan proceses preprocessing. Pada proses ini tidak ada yang berkurang.
2. Dalam penelitian digunakan komposisi data sebesar 80% pada data latih dan 20% data uji dari jumlah keseluruhan data diambil secara acak. Proses setelah pembagian data tersebut akan dilakukan pembobotan dengan TF – IDF, dilanjut dengan menentukan klasifikasi *naive bayes* dan SVM serta adanya evaluasi model.

3. Hasil nilai akhir pada analisis sentimen pada metode *naive bayes* menghasilkan akurasi 69%, presisi 69%, *recall* 70%, dan *f1 score* 69%, sedangkan nilai akhir pada metode SVM mendapatkan akurasi 70%, presisi 70%, *recall* 70%, dan *f1 score* 70%. Dalam hal ini jika tujuan awal dari peneliti yaitu menentukan peforma yang terbaik dari kedua metode tersebut, maka hasilnya yaitu metode SVM.

B. Saran

Berdasarkan penelitian yang dilaksanakan, penulis berharap agar peneliti selanjutnya dapat mengembangkan penelitian serupa dengan beberapa saran yang telah disampaikan:

1. Pada penelitian ini menggunakan perbandingan metode klasifikasi *naïve bayes* dan SVM dalam melakukan analisis sentiment. Penelitian selanjutnya mungkin dapat membandingkan dengan metode klasifikasi yang lain seperti membandingkan metode K-NN dengan *Naïve Bayes, Random forest* dengan SVM ataupun perbandingan metode klasifikasi yang lain.
2. Pada penelitian ini, data diambil dari *play store* pada penelitian berikutnya diharapkan data dapat

diperoleh dari media sosial seperti youtube, facebook, instagram atau tik tok dll.

3. Penambahan koleksi kamus untuk kata-kata yang tidak baku atau gaul sangat penting, mengingat banyaknya komentar di *Play Store* yang menggunakan bahasa yang kurang baku.

DAFTAR PUSTAKA

Afandi, I. R., Noor Hasan, F., Rizki, A. A., Pratiwi, N., & Halim, Z. (2022). Analisis Sentimen Opini Masyarakat Terkait Pelayanan Jasa Ekspedisi Anteraja Dengan Metode Naive Bayes. *Jurnal Linguistik Komputasional*, *5*(2), 63–70. https://t.co/2HAdwg1drL

Akhir, M. T., Syarat, M., Memperoleh, G., Sarjana, G., Satu, S., & Informasi, T. (2023). *Perbandingan Kinerja Metode Klasifikasi Naïve Bayes Dan Random Forest Dalam Analisis Sentimen Kasus Narkoba di Indonesia Pada Komentar YouTube SKRIPSI Diajukan oleh : NAILUL ' INAYAH PROGRAM STUDI TEKNOLOGI INFORMASI*.

Ardiani, L., Sujaini, H., & Tursina, T. (2020). Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, *8*(2), 183. https://doi.org/10.26418/justin.v8i2.36776

Arifin, N., Enri, U., & Sulistiyowati, N. (2021). Penerapan Algoritma Support Vector Machine (SVM) dengan TF-IDF N-Gram untuk Text Classification. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *6*(2), 129. https://doi.org/10.30998/string.v6i2.10133

Artikel, I., & Info, A. (2023). *ANALISIS SENTIMEN PENGGUNA APLIKASI JMO (JAMSOSTEK MOBILE) PADA GOOGLE PLAY STORE MENGGUNAKAN METODE NAIVE BAYES*. *2*(3), 109–117.

Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *INTEGER: Journal of Information Technology*, *2*(1), 32–41. https://doi.org/10.31284/j.integer.2017.v2i1.95

Carneiro, T., Da Nobrega, R. V. M., Nepomuceno, T., Bian, G. Bin, De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*,

6, 61677–61685. https://doi.org/10.1109/ACCESS.2018.2874767

Fajrin, A. A., & Maulana, A. (2018). Penerapan_Data_Mining_Untuk_Analisis_Pol. *Kumpulan Jurnal, Ilmu Komputer*, *05*(01), 27–36.

Fikri, M. I., Sabrila, T. S., & Azhar, Y. (2020). Comparison of Naïve Bayes and Support Vector Machine Methods in Twitter Sentiment Analysis. *Smatika Jurnal*, *10*(02), 71–76.

Hasanah, U., Astuti, T., Wahyudi, R., Rifai, Z., & Pambudi, R. A. (2018). An experimental study of text preprocessing techniques for automatic short answer grading in Indonesian. *Proceedings - 2018 3rd International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2018*, *August 2024*, 230–234. https://doi.org/10.1109/ICITISEE.2018.8720957

Herlinawati, N., Yuliani, Y., Faizah, S., Gata, W., & Samudi, S. (2020). Analisis Sentimen Zoom Cloud Meetings di Play Store Menggunakan Naïve Bayes dan Support Vector Machine. *CESS (Journal of Computer Engineering, System and Science)*, *5*(2), 293. https://doi.org/10.24114/cess.v5i2.18186

Iman, Q., & Wijayanto, A. W. (2021). Klasifikasi Rumah Tangga Penerima Beras Miskin (Raskin)/Beras Sejahtera (Rastra) di Provinsi Jawa Barat Tahun 2017 dengan Metode Random Forest dan Support Vector Machine. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, *9*(2), 178. https://doi.org/10.26418/justin.v9i2.44137

Imron, A. (2019). Analisis Sentimen Terhadap Tempat Wisata di Kabupaten Rembang Menggunakan Metode Naive Bayes Classifier. *Teknik Informatika*, 10–13. https://dspace.uii.ac.id/handle/123456789/14268

Karsito, & Susanti, S. (2019). Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes

Di Perumahan Azzura Residencia. *Jurnal Teknologi Pelita Bangsa*, *9*, 43–48.

Khairunnisa, S., Adiwijaya, A., & Faraby, S. Al. (2021). Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19). *Jurnal Media Informatika Budidarma*, *5*(2), 406. https://doi.org/10.30865/mib.v5i2.2835

Muttaqin, M. N., & Kharisudin, I. (2021). Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor. *UNNES Journal of Mathematics*, *10*(2), 22–27. http://journal.unnes.ac.id/sju/index.php/ujm

Natalianus, G., Kurniawan, V., & Feta, N. R. (2024). *Perbandingan Akurasi Algoritma Naïve Bayes Dan Support Vector Machine Dalam Analisis Sentimen Pengguna Twitter Terhadap Aplikasi Sirekap Abstrak*. *9*(2).

Nelson, D. (2020). *Apa itu Support Vector Machines? - Bersatu.AI*. IMB. https://www.unite.ai/id/what-are-support-vector-machines/

Novriandy, A. (2023). Implementasi Algoritma Naive Bayes dan Algoritma C4. 5 dalam Klasifikasi Kelayakan Bantuan UMKM. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *4*(1), 208–217. https://doi.org/10.30865/klik.v4i1.1099

Nugroho, D. G., Chrisnanto, Y. H., & Wahana, A. (2015). *Analisis Sentimen Pada Jasa Ojek Online ... (Nugroho dkk.)*. 156–161.

Paris. (2024). *Menilai Integrasi Pemilu 2024 Melalui Sirekap*. 14 Maret 2024. https://ugm.ac.id/id/berita/menilai-integritas-pemilu-2024-melalui-sirekap/

Petiwi, M. I., Triayudi, A., & Sholihati, I. D. (2022). Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode Naïve Bayes dan Support Vector Machine. *Jurnal Media Informatika Budidarma*, *6*(1), 542. https://doi.org/10.30865/mib.v6i1.3530

Pokhrel, S. (2024). No TitleΕΛΕΝΗ. *Αγαη*, *15*(1), 37–48.

Ramadhan, G. A. (2023). Analisis sentimen ulasan aplikasi ruangguru dengan algoritma long short term memory. In *Repository.Uinjkt.Ac.Id*. https://repository.uinjkt.ac.id/dspace/handle/123456789/71510%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/71510/1/GILANG AMBANG RAMADHAN-FST.pdf

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

Rismel. (2024). *Manfaat Pengguna Aplikasi SIREKAP dan Pentingnya Lakukan Simulasi*. 03 Februari 2024. https://ugm.ac.id/id/berita/menilai-integritas-pemilu-2024-melalui-sirekap/

Ritonga, A. S., & Purwaningsih, E. S. (2018). *PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) DALAM KLASIFIKASI KUALITAS PENGELASAN SMAW (SHIELD METAL ARC WELDING)*. *5*(1), 17–25.

SHELEMO, A. A. (2023). No Titleيليب. In *Nucl. Phys.* (Vol. 13, Issue 1).

Sihombing, J. (2021). Klasifikasi Data Antroprometri Individu Menggunakan Algoritma Naïve Bayes Classifier. *BIOS : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, *2*(1), 1–10. https://doi.org/10.37148/bios.v2i1.15

Sis.binus.ac.id. (2022). *Support Vector Machine Alogaritma*.

Syahputra, R., Yanris, G. J., & Irmayani, D. (2022). SVM and Naïve Bayes Algorithm Comparison for User Sentiment Analysis on Twitter. *Sinkron*, *7*(2), 671–678. https://doi.org/10.33395/sinkron.v7i2.11430

Taufiqurrahman, F., Faraby, S. Al, & Purbolaksono, M. D. (2021). Klasifikasi Teks Multi Label pada Hadis

Terjemahan Bahasa Indonesia Menggunakan Chi Square dan SVM. *E-Proceeding of Engineering*, *8*(5), 10650–10659.

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.

W Dwi Yuniarti. (2019). *Dasar - Dasar Pemprograman dengan Python*.

Yutika, C. H., Adiwijaya, A., & Faraby, S. Al. (2021). Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes. *Jurnal Media Informatika Budidarma*, *5*(2), 422. https://doi.org/10.30865/mib.v5i2.2845

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: Lembar Pengesahan Proposal Skripsi

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

PENGESAHAN

Naskah proposal skripsi berikut ini :

Judul	:Tinjauan Komentar Pengguna Sirekap Di Play Store Studi Komparasi Menggunakan Naive Bayes dan Support Vectore Machine
Penulis	: Istna A'la Mahmudah
NIM	: 2108096030
Jurusan	: Teknologi Informasi

Telah diujikan dalam seminar proposal skripsi oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Teknologi Informasi.

Semarang, 16 Januari 2025

DEWAN PENGUJI

Penguji I	Penguji II
Dr. Khotibul Umam, M.Kom	Dr. Wenty Dwi Yuniarti, S.Pd.,M.Kom
NIP. 197908272011011007	NIP. 197706222006042005
Penguji III	**Penguji IV**
Dr. Masy Ari Ulinuha, ST.,M.T	Siti Nur 'Aini, M.Kom
NIP. 198108122011011007	NIP. 198401312018012001

LAMPIRAN 2: Lembar Persetujuan Proposal Skripsi

HALAMAN PERSETUJUAN

Proposal skripsi ini telah disetujui oleh pembimbing untuk dilaksanakan.

Disetujui pada

Hari : 19 Nov 2024

Tanggal :

Pembimbing I,	Pembimbing II,
Dr. Wenty Dwi Yuniarti, S. Pd., M. Kom	Mokhamad Iklil Mustofa M.Kom
NIP. 197312222006041001	NIP. 1988080701931010

Mengetahui,
Ketua Jurusan Teknologi Informasi

Khoirbul Umam, S. T, M. Kom
NIP. 197908272011011007

ii

LAMPIRAN 3: Contoh Data Scraping Data Play Store

No	Content
1.	Aplikasi butut, herannya kpu dikasih waktu 5 tahun untuk buat ni aplikasi tapi hasilnya malah seadanya, kayak ga niat buat (lebih ke "hemat budget" mungkin). Gak bisa unggah foto dari galeri, kualitas foto jadi jelek banget + blur, gak ada flash, sudah diunggah suka minta foto ulang yang mana itu buat repot. Lain kali kalo diamanahkan untuk buat aplikasi tolong buatlah dengan sebaik mungkin, masukan dari teman" yang sudah diulas tolong dipikirkan.
2.	Segera perbaiki menu detail pemilihan, sebab gagal lakukan koreksi hasil pengambilan foto form C, periksa form C Hasil, dalam jendela periksa kesesuaian data, hanya bisa kasih x (cross) tidak sesuai, tp jumlah yg perlu dikoreksi tidak muncul editable box nya. Terima kasih.
3.	APLIKASI GAGAL. Suka nge bug, over akses bikin Lemot, kamera tidak bisa di fokuskan dan kualitas gambar tulisan jadi blur. Unggah gambar tulisan "tunggu sesaat" TAPI 3 hari blum selesai proses pengunggahannya. Cara pengaplikasianya cukup mudah. Tapi ya seperti tadi contohnya. Aplikasi PEMILU dari tahun 2019 sampe sekarang gak ada perubahan.
4.	Aplikasi belum layak untuk dioperasikan 1.bisa dilihat dari komenan kawan2,pas unggah selalu logout sendiri. 2.kualitas gambar,bukan kesalah spek android,camera saat poto biasa jernih dan fokus,diaplikasi ini hasil poto buram dan tidak dapat difokuskan secara manual,mungkin mengingat kualitas gambar HD memakan penyimpanan besar mungkin,tapi ini kualitasnya buram,terlalu kecil,tolong perbaiki secepatnya,!!!

5.	Jelek , dari segi tampilan ui nya terkesan jadul banget , terus juga untuk simulasi aplikasi sirekap ini ga terlalu jelas aneh, ngabisin anggaran doang ! Improvisasi lagi kedepannya pak agar lebih baik lagi !
6.	Aplikasi paling banyak minusnya. Kualitas gambar buram saat difoto (tidak bisa difokuskan manual), angka digital sering salah dibaca, susah untuk mengedit hasil yang tidak sesuai dibaca oleh aplikasi, dan masih banyak lagi minus yang lain.
7.	Aplikasi yang mlah menyusahkan. Fokus foto yang tidak bagus, scan manual, force close, dan jaringan yg sibuk. Aplikasi yang dicoba pada saat pemilu belum berlangsung saja sudah punya kualitas yang buruk, apalagi saat pemilu dilakukan.
8.	Perbaiki lagi kualitas aplikasinya kalau mau rekapan hasil pemilu berjalan lancar. Fix sangatÃ‚Â² membagongkan ni aplikasi mencetÃ‚Â² tombol sampai jari pegel pun tetep ga bisa kebaca tuh sidik jari.
9.	Tolong lebih ditingkatkan lagi, terutama kamera agar fokus dan dikasih flash.. upload terlalu lama dan data yg terbaca salah jd harus edit sendiri.. sangat membuang" waktu dan memperlambat kinerja. Fitur edit ditambahkan, jangan skali pencet g bsa berubah lagi Ã°Å¸Â¥Â² agar bisa di edit sampai data bner bner sesuai.. Cukup membantu dr pada yg sebelumnya tapi banyak nyusahinnya ini aplikasi Ã°Å¸Â¥Â²Ã°Å¸Â¤Â£
10.	Aplikasi sangat tidak layak digunakan, tidak akan mempercepat tapi malah mempersulit mohon disempurnakan lagi, sangat banyak kekurangan: 1. Pada saat hp posisi online di aplikasi terdeteksi offline 2. Saat unggah foto aplikasi tidak benar " Bisa membaca angka di c. Plano 3. Unggah foto pada

	menu DPR waktunya sangat lama Dll.....!!!!! Masih banyak kekurangan ! !!!!!!
11.	Kualitas gambar untuk foto simulasi burem ga jelas, dlm aplikasi ga ada fitur buat fokusin camera, jadi banyak bgt kesalahan baca data dan ada beberapa bug juga untuk kunci data & buat dokumen pdf loadingnya lama banget, bahkan ada yang ga bisa sampai berhari2 mungkin itu bug tolong diperbaiki, pemilu tinggal berapa hari lagi.. ini aplikasi bisa dibilang belum berfungsi secara optimal!
12.	Aplikasinya terlalu banyak manual, bukan mempermudah tpi memperlama, penjumlahan masih manual tidak otomatis, jika terjadi ketidak sesuaian suara hanya diberi tanda saja dan tidak bisa mengubah angka yang sebenarnya.ketika kirim foto mental, ketika foto C hasil meskipun buram tapi tetap lolos verifikasi, tidak bisa hapus foto dan edit jumlah suara sebelum mengunci dokumen. Ketika login sulit. Ketika mau kirim pdf ke saksi/pengawas mental lagi, Kalau belum siap jangan dipaksakan
13.	Untuk aplikasi sekelas nasional ini benar benar jelek. Unggah gambar susah dibuat mode pesawat bisa tapi harus masukkan angka secara manual. Belum lagi untuk membuat dokumen pdf nya lama mulai sore sampai pagi tetap tidak bisa dibuat. Untuk tim pembuat aplikasi kalau ada komen komen gini tolong dibaca dan diperbaiki karena yang memakai aplikasi ini kan kpps untuk kelancaran pemilu juga. Besok yang buat lama bukan penghitungan suara tapi sirekap bermasalah. Bukan mempermudah jadi mempersulit
14.	Bukannya mempermudah, malah mempersulit. Login error, susah, lama. Kualitas upolad gambar jelek, tulisan bagus malah tidak terbaca, verifikasi gambar juga lama, membuat berkas ke pdf juga lama.

	Tolong segera diperbaiki jika memang berniat mempermudah !!
15.	Mohon maaf sekali tapi aplikasi ini tidak mempermudah tetapi memberatkan dari banyak sisi. Misalnya, spesifikasi harus Ram 4 dgn sistem android 12, koneksi stabil, terdapat sidik jari, dgn posisi penggunaan serentak seluruh Indonesia, scan menunggu tanda hijau. Kesabaran diuji sekali bahkan ketika kesulitan login.

.....

1999.	tolong apliaksi nya di perbaiki lagi..saat ingin login aplikasi eror trs menunggu di saat jam2 tidak sibuk pengguna nya...apa lagi saat pemilu nanti .tks
2000	Kenapa susah untuk proses sertifikat digital selalu time out..pdhl waktu pencoblosan sebentar lgi tapi aplikasinya tidak berfungsi dgn baik

LAMPIRAN 4 : Contoh Data yang Sudah Diberi Label

No	Content	Label
1.	Aplikasi butut, herannya kpu dikasih waktu 5 tahun untuk buat ni aplikasi tapi hasilnya malah seadanya, kayak ga niat buat (lebih ke "hemat budget" mungkin). Gak bisa unggah foto dari galeri, kualitas foto jadi jelek banget + blur, gak ada flash, sudah diunggah suka minta foto ulang yang mana itu buat repot. Lain kali kalo diamanahkan untuk buat aplikasi tolong buatlah dengan sebaik mungkin, masukan dari teman" yang sudah diulas tolong dipikirkan.	Negatif
2.	Segera perbaiki menu detail pemilihan, sebab gagal lakukan koreksi hasil pengambilan foto form C, periksa form C Hasil, dalam jendela periksa kesesuaian data, hanya bisa kasih x (cross) tidak sesuai, tp jumlah yg perlu dikoreksi tidak muncul editable box nya. Terima kasih.	Positif
3.	APLIKASI GAGAL. Suka nge bug, over akses bikin Lemot, kamera tidak bisa di fokuskan dan kualitas gambar tulisan jadi blur. Unggah gambar tulisan "tunggu sesaat" TAPI 3 hari blum selesai proses	Negatif

	pengunggahannya. Cara pengaplikasianya cukup mudah. Tapi ya seperti tadi contohnya. Aplikasi PEMILU dari tahun 2019 sampe sekarang gak ada perubahan.	
4.	Aplikasi belum layak untuk dioperasikan 1.bisa dilihat dari komenan kawan2,pas unggah selalu logout sendiri. 2.kualitas gambar,bukan kesalah spek android,camera saat poto biasa jernih dan fokus,diaplikasi ini hasil poto buram dan tidak dapat difokuskan secara manual,mungkin mengingat kualitas gambar HD memakan penyimpanan besar mungkin,tapi ini kualitasnya buram,terlalu kecil,tolong perbaiki secepatnya,!!!	Negatif
5.	Jelek , dari segi tampilan ui nya terkesan jadul banget , terus juga untuk simulasi aplikasi sirekap ini ga terlalu jelas aneh, ngabisin anggaran doang ! Improvisasi lagi kedepannya pak agar lebih baik lagi !	Negatif
6.	Aplikasi paling banyak minusnya. Kualitas gambar buram saat difoto (tidak bisa difokuskan manual), angka digital sering salah dibaca, susah untuk mengedit hasil yang tidak sesuai dibaca oleh	Negatif

	aplikasi, dan masih banyak lagi minus yang lain.	
7.	Aplikasi yang mlah menyusahkan. Fokus foto yang tidak bagus, scan manual, force close, dan jaringan yg sibuk. Aplikasi yang dicoba pada saat pemilu belum berlangsung saja sudah punya kualitas yang buruk, apalagi saat pemilu dilakukan.	Negatif
8.	Perbaiki lagi kualitas aplikasinya kalau mau rekapan hasil pemilu berjalan lancar. Fix sangatÃ‚Â² membagongkan ni aplikasi mencetÃ‚Â² tombol sampai jari pegel pun tetep ga bisa kebaca tuh sidik jari.	Positif
9.	Tolong lebih ditingkatkan lagi, terutama kamera agar fokus dan dikasih flash.. upload terlalu lama dan data yg terbaca salah jd harus edit sendiri.. sangat membuang" waktu dan memperlambat kinerja. Fitur edit ditambahkan, jangan skali pencet g bsa berubah lagi Ã°Å¸Â¥Â² agar bisa di edit sampai data bner bner sesuai.. Cukup membantu dr pada yg sebelumnya tapi banyak nyusahinnya ini aplikasi Ã°Å¸Â¥Â²Ã°Å¸Â¤Â£	Positif

10.	Aplikasi sangat tidak layak digunakan, tidak akan mempercepat tapi malah mempersulit mohon disempurnakan lagi, sangat banyak kekurangan: 1. Pada saat hp posisi online di aplikasi terdeteksi offline 2. Saat unggah foto aplikasi tidak benar " Bisa membaca angka di c. Plano 3. Unggah foto pada menu DPR waktunya sangat lama Dll.....!!!!! Masih banyak kekurangan ! !!!!!!	Negatif
11.	Kualitas gambar untuk foto simulasi burem ga jelas, dlm aplikasi ga ada fitur buat fokusin camera, jadi banyak bgt kesalahan baca data dan ada beberapa bug juga untuk kunci data & buat dokumen pdf loadingnya lama banget, bahkan ada yang ga bisa sampai berhari2 mungkin itu bug tolong diperbaiki, pemilu tinggal berapa hari lagi.. ini aplikasi bisa dibilang belum berfungsi secara optimal!	Negatif
12.	Aplikasinya terlalu banyak manual, bukan mempermudah tpi memperlama, penjumlahan masih manual tidak otomatis, jika terjadi ketidak sesuaian suara hanya diberi tanda saja dan tidak bisa mengubah angka yang sebenarnya.ketika	Negatif

	kirim foto mental, ketika foto C hasil meskipun buram tapi tetap lolos verifikasi, tidak bisa hapus foto dan edit jumlah suara sebelum mengunci dokumen. Ketika login sulit. Ketika mau kirim pdf ke saksi/pengawas mental lagi, Kalau belum siap jangan dipaksakan	
13.	Untuk aplikasi sekelas nasional ini benar benar jelek. Unggah gambar susah dibuat mode pesawat bisa tapi harus masukkan angka secara manual. Belum lagi untuk membuat dokumen pdf nya lama mulai sore sampai pagi tetap tidak bisa dibuat. Untuk tim pembuat aplikasi kalau ada komen komen gini tolong dibaca dan diperbaiki karena yang memakai aplikasi ini kan kpps untuk kelancaran pemilu juga. Besok yang buat lama bukan penghitungan suara tapi sirekap bermasalah. Bukan mempermudah jadi mempersulit	Negatif
14.	Bukannya mempermudah, malah mempersulit. Login error, susah, lama. Kualitas upolad gambar jelek, tulisan bagus malah tidak terbaca, verifikasi gambar juga lama, membuat berkas ke pdf juga	Negatif

	lama. Tolong segera diperbaiki jika memang berniat mempermudah !!	
15.	Mohon maaf sekali tapi aplikasi ini tidak mempermudah tetapi memberatkan dari banyak sisi. Misalnya, spesifikasi harus Ram 4 dgn sistem android 12, koneksi stabil, terdapat sidik jari, dgn posisi penggunaan serentak seluruh Indonesia, scan menunggu tanda hijau. Kesabaran diuji sekali bahkan ketika kesulitan login.	Positif
.....		
1999.	tolong apliaksi nya di perbaiki lagi..saat ingin login aplikasi eror trs menunggu di saat jam2 tidak sibuk pengguna nya...apa lagi saat pemilu nanti .tks	Positif
2000	Kenapa susah untuk proses sertifikat digital selalu time out..pdhl waktu pencoblosan sebentar lgi tapi aplikasinya tidak berfungsi dgn baik	Negatif

LAMPIRAN 5 : Contoh Data Pendukung "kamuskatabaku"

No.	tidak_baku	kata_baku
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
.....		
15184.	lu	kamu
15185.	bang	abang
15186.	kaka	kakak

DAFTAR RIWAYAT HIDUP

A. Identitas Diri

Nama	: Istna A'la Mahmudah
Tempat, Tanggal Lahir	: Pati, 11 Agustus 2003
Alamat	: Desa Kedalingan, RT 02 / RW 01, Kec. Tambakromo, Kab. Pati
HP	: 082132181626
Email	: isnaalamahmudah@gmail.com

B. Riwayat Pendidikan

1. SD Negeri Kedalingan 01
2. MTS Salafiyah Kajen
3. SMA N 1 Kayen