



**ANALISIS SENTIMEN TERHADAP KARIR
E-SPORTS DI INDONESIA PADA MEDIA
TWITTER MENGGUNAKAN METODE NAIVE
BAYES**

*diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana pada program studi Sistem
Informasi.*

Oleh :

Fendy Marzuki

172410101106

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2023

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT;
2. Ibu, Bapak, kakak, dan adik tercinta;
3. Pembimbing 1 Ibu Oktalia Juwita, S.Kom., M.MT. dan pembimbing 2 Bapak Priza Pandunata, S.Kom., M.Sc.;
4. Teman-teman saya semuanya.



MOTTO

“Yesterday is History, tomorrow is a mystery, but today is a gift, that is why it’s called a present”.

-Master Oogway-



PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini:

Nama : Fendy Marzuki

NIM : 172410101106

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Analisis Sentimen Terhadap Karir *E-Sports* Di Indonesia Pada Media *Twitter* Menggunakan Metode *Naive Bayes*”, adalah benar-benar hasil karya saya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada instansi manapun, dan bukti karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika dikemudian hari pernyataan ini tidak benar.

Jember, 25 Mei 2023

Yang menyatakan,

Fendy Marzuki

NIM 172410101106

HALAMAN PERSETUJUAN

Skripsi berjudul “Analisis Sentimen Terhadap Karir *E-Sports* Di Indonesia Pada Media *Twitter* Menggunakan Metode *Naive Bayes*” telah diuji dan disahkan oleh Fakultas Ilmu Komputer Universitas Jember pada:

Hari : Kamis

Tanggal : 25 Mei 2023

Tempat : Fakultas Ilmu Komputer Universitas Jember

Pembimbing Tanda Tangan

1. Pembimbing Utama

Nama : Oktalia Juwita, S.Kom., M.MT. (.....)

NIP : 198110202014042001

2. Pembimbing Anggota

Nama : Priza Pandunata, S.Kom., M.Sc. (.....)

NIP : 19830131201504001

Penguji

1. Penguji Utama

Nama : Achmad Maududie, ST., M.Sc. (.....)

NIP : 197004221995121001

2. Penguji Anggota

Nama : Muhamad Arief Hidayat M.Kom. (.....)

NIP : 198101232010121003

Dekan Fakultas Ilmu Komputer

1. Nama : Drs. Antonius Cahya Prihandoko, (.....)

M.App.Sc, Ph.D

NIP : 196909281993021001

ABSTRACT

The rapid growth of eSports has garnered significant attention worldwide, including in Indonesia. This study aims to perform sentiment analysis on the career of eSports in Indonesia using Twitter as the primary data source. The objective is to analyze the sentiment expressed by Twitter users towards eSports careers and identify the prevailing sentiment trends. The Naive Bayes method is employed as the classification algorithm for sentiment analysis. The study involves collecting a large dataset of tweets related to eSports careers in Indonesia and manually annotating them with sentiment labels. Preprocessing techniques are applied to clean the data and extract relevant features. The Naive Bayes classifier is trained on a labeled dataset to learn the relationships between features and sentiment labels. The trained classifier is then utilized to classify new tweets and determine their sentiment polarity. The results of this analysis provide insights into the sentiment towards eSports careers in Indonesia, including positive, negative, and neutral sentiments. The findings can be utilized by stakeholders in the eSports industry to gain a better understanding of public perception and make informed decisions regarding career development and marketing strategies. The evaluation of the sentiment analysis model is conducted using metrics such as accuracy, precision, recall, and F1-score. The achieved accuracy rate demonstrates the effectiveness of the Naive Bayes method in sentiment analysis of eSports careers on Twitter in the Indonesian context. This study contributes to the field of sentiment analysis in the eSports domain and provides valuable insights for industry professionals and researchers.

Keywords: sentiment analysis, eSports, career, Twitter, Naive Bayes, Indonesia.

RINGKASAN

Analisis Sentimen Terhadap Karir *E-Sports* Di Indonesia Pada Media *Twitter* Menggunakan Metode *Naive Bayes*; Fendy Marzuki, 172410101106, 2023; 62 halaman, Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Analisis sentimen karir esports pada Twitter menggunakan metode Naive Bayes adalah sebuah pendekatan yang digunakan untuk menentukan sentimen positif dan negatif terkait topik karir *esports* berdasarkan data tweet yang dianalisis.

Metode Naive Bayes adalah sebuah metode klasifikasi yang berdasarkan pada Teorema Bayes dengan asumsi bahwa setiap fitur (kata atau frasa) dalam data tweet adalah independen satu sama lainnya. Metode ini menghitung probabilitas masing-masing kelas sentimen (positif, negatif, atau netral) berdasarkan kemunculan fitur-fitur tersebut dalam set data pelatihan.

Proses Analisis sentimen dilakukan setelah data didapatkan pada tahapan crawling data, dilanjutkan dengan pembersihan kata pada proses preprocessing, dan pembobotan kata. Analisis sentimen dengan mengimplementasikan metode naive bayes classifier bertujuan agar mendapatkan klasifikasi mengenai opini publik terhadap karir *esports* di Indonesia. Terdapat dua kelas pada penelitian ini yaitu kelas positif dan negatif. Dari 2000 unggahan tweet dilakukan pelabelan sentimen positif dan negatif lalu akan dibagi antara data training dan data testing dengan perbandingan sebesar 60%:40%, 70%:30%, 80%:20%, dan 90%:10%. Dari hasil evaluasi pada Analisis sentimen mengenai opini publik terhadap karir *esports* pada twitter didapatkan nilai akurasi tertinggi sebesar 68% dengan data training yang dipakai sebesar 90%, data testing sebesar 10%.

PRAKATA

Puji dan syukur penulis panjatkan atas segala Rahmat dan Karunia kepada Tuhan Yang Maha Esa, sehingga Penulis kemudian bisa menyelesaikan skripsi ini dengan judul “Analisis Sentimen Terhadap Karir *E-Sports* Di Indonesia Pada Media *Twitter* Menggunakan Metode *Naive Bayes*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember. Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Allah SWT yang senantiasa memberikan rahmad dan hidayah-Nya untuk mempermudah dan memberikan melancarkan dalam proses pengerjaan skripsi;
2. Drs. Antonius Cahya Prihandoko, M.App.Sc, Ph.D., selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
3. Oktalia Juwita, S.Kom., M.MT. selaku Dosen Pembimbing Utama dan Priza Pandunata, S.Kom., M.Sc. selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
4. Yanuar Nurdiansyah, ST,. M.Cs selaku Dosen Pembimbing Akademik yang sudah membimbing penulis selama menjadi mahasiswa;
5. Ibu, Bapak, kakak, dan adik tercinta yang selalu mendukung dan mendoakan;
6. Semua teman – teman PojokStayKalem yang selalu ada dalam suka maupun duka;
7. Kakak – kakak dan teman – teman Dota saya yang selalu senantiasa memberikan dukungan , arahan, dan bantuan selama masa perkuliahan dan juga masa pengerjaan skripsi;

8. Teman - teman SEKTE.
9. Teman – teman satu angkatan 2017 Fakultas Ilmu Komputer.

Penulis memang menyadari betul bahwasanya skripsi ini juga masih sangat jauh terlepas dari kata sempurna, oleh karena itu penulis juga mengharapkan dan menerima segala kritik maupun saran yang sangat membangun dari semua pihak. Penulis tentunya juga berharap adanya skripsi ini bisa berguna dan juga bermanfaat bagi semua pihak.

Jember, 25 Mei 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERSEMBAHAN	ii
MOTTO	iii
PERNYATAAN ORISINALITAS	iv
HALAMAN PERSETUJUAN	v
ABSTRACT	vi
RINGKASAN	vii
PRAKATA	viii
DAFTAR ISI	x
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	4
1.5. Batasan Masalah	4
BAB 2. TINJAUAN TEORI	5
2.2. Sentimen Analisis	6
2.3. Text Mining	8
2.3.1 Preprocessing	9
2.4. Algoritma <i>Term Frequency-Inverse Document Frequency (TF-IDF)</i>	10
2.5. Metode Naive Bayes	12
2.6. Gaussian Naive Bayes	14
2.7. Confusion Matrix	14
2.8. <i>Wordcloud</i>	15
2.9. Penelitian Terdahulu	15
BAB 3. METODOLOGI PENELITIAN	17
3.2. Tahapan Penelitian	17
3.3. Metode Pengumpulan Data	19
3.4. Pelabelan Data	20
3.5. Preprocessing	21
3.6. Penyusunan Model	22
3.7. Pengujian Model	23
3.7.1. Confusion Matrix	23
3.7.2. Pengukuran Akurasi	23

3.8 <i>Wordcloud</i>	24
BAB 4. HASIL DAN PEMBAHASAN.....	25
4.1. Hasil Dataset.....	25
4.2. Hasil Implementasi Sistem.....	26
4.2.1 Implementasi Crawling Data.....	26
4.2.2 Implementasi Preprocessing.....	27
4.2.3 Implementasi Penyusunan Model.....	29
4.2.4 Implementasi Penyusunan <i>Wordcloud</i>	30
4.3. Hasil Pengujian.....	31
4.3.1 Hasil Pengujian Implementasi <i>Preprocessing</i>	31
4.3.2 Hasil Implementasi Penyusunan Model.....	35
4.3.3 Hasil Uji Evaluasi.....	43
4.3.4 Hasil Uji <i>Wordcloud</i>	44
BAB 5. KESIMPULAN DAN SARAN.....	47
5.1. Kesimpulan.....	47
5.2. Saran.....	48
DAFTAR PUSTAKA.....	49
LAMPIRAN.....	51

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Pada tahun 2019, nilai pasar game global diharapkan mencapai US\$152 miliar atau setara dengan Rp.2,15 kuadriliun, meningkat sebesar 9,6% dibandingkan tahun sebelumnya. Indonesia termasuk salah satu pasar video game terbesar di wilayah Asia Pasifik, dengan nilai sebesar US\$941 juta atau setara dengan Rp.13 triliun (*newzoo.com*, 2019). Kenaikan ini memberikan kontribusi signifikan pada pertumbuhan industri e-sport yang sangat pesat, yang mana e-sport merupakan bentuk permainan kompetitif yang dilakukan dengan memainkan video game.

Pemerintah Indonesia memberikan respon positif terhadap perkembangan industri e-sport di negara ini. Indonesia e-Sport Association (IeSPA) merupakan asosiasi resmi yang didirikan oleh pemerintah untuk mengawasi dan mendukung komunitas e-sport di Indonesia. IeSPA membantu para pemain dalam mengembangkan kemampuan dan bakat mereka dalam bermain game kompetitif. Selain itu, pemerintah juga mengadakan berbagai kompetisi seperti IEC University Series 2019, Youth National eSport Championship, dan Piala Presiden. Ekosistem industri e-sport di Indonesia berkembang pesat dan memiliki potensi untuk menciptakan banyak lapangan kerja. Tidak harus ahli dalam bermain game, *e-sport* juga membutuhkan peran-peran seperti pelatih, tim yang memiliki struktur organisasi yang cukup lengkap, *streamer*, *caster*, *event organizer* agar keberlangsungan *e-sport* dapat berjalan dengan baik. *E-sport* ini juga dilirik sebagai industri bisnis yang menjanjikan, oleh karena itu cukup banyak sponsor maupun investor yang turut meramaikan industri ini.

Pandangan masyarakat Indonesia terhadap *e-sports* masih sebagian besar terbatas pada persepsi bahwa itu hanya sekadar bermain game. Karena *e-sports* masih dianggap sebagai cabang olahraga yang relatif baru, masih banyak orang yang belum sepenuhnya memahami berbagai dampak yang ditimbulkan oleh aktivitas ini. Beberapa orang bahkan masih menganggap bahwa *e-sports* hanya duduk diam di depan layar monitor tanpa memahami

risiko yang mungkin dihadapi oleh para pemainnya. Selain itu, ada juga orang-orang yang cenderung melihat *e-sports* dari sudut pandang negatif tanpa melihat aspek positifnya, atau sebaliknya. Meskipun demikian, kenyataannya esports tidak hanya sebatas bermain game, melainkan juga menawarkan berbagai peluang karir di baliknya. Hal ini menjadi salah satu keprihatinan dalam industri *e-sports* yang saat ini semakin populer, karena tingkat literasi mengenai *e-sports* masih sangat rendah di kalangan masyarakat umum. Orang-orang biasanya hanya menganggap *e-sports* sebagai cara menghabiskan waktu luang dengan bermain game. Oleh karena itu dilakukannya penelitian ini agar dapat menjadi informasi untuk banyak orang terkait karir *e-sports* di Indonesia.

Pelaku e-sport sering menggunakan media sosial twitter sebagai tempat mereka bertukar informasi dan melakukan komunikasi yang berkaitan tentang game ketimbang media sosial lainnya. Twitter pun juga mengklaim dirinya sebagai rumahnya para gamers baik untuk mendapatkan informasi, menonton potongan video pertandingan yang sedang berlangsung, atau berinteraksi dengan atlet e- sports favorit mereka. Sebanyak 91 Juta pengguna aktif twitter diseluruh dunia telah melakukan tweet terkait game per detik dan telah menghasilkan 156 Juta tweet tentang game sepanjang tahun ini (marketing.twitter.com, 2019).

Twitter sering digunakan untuk mengungkapkan emosi mengenai sesuatu hal, baik dalam hal memuji ataupun mencela dalam bentuk emosi. Tidak hanya itu, *Twitter* sering digunakan oleh penggunanya sebagai tempat untuk mendapatkan informasi berisi teks maupun link berita yang *up to date*. Sebagian pengguna percaya bahwa konten informasi di media sosial Indonesia bermula dari *Twitter*, kemudian kontentersebut disebar ulang ke media sosial *Facebook*, *Instagram*, dan *Youtube*.

Terkait dengan opini tentang karir *e-sport* di media sosial Twitter, polaritas opini positif atau negatif akan dilakukan menggunakan metode *Naïve Bayes Classifier*. *Naive Bayes Classifier* adalah salah satu metode yang sering digunakan untuk pengklasifikasian dokumen. Metode ini memiliki akurasi dan kecepatan analisis yang tinggi ketika digunakan untuk mengolah data pada basis data yang besar dan data yang bermacam-macam.

Penelitian ini telah dilakukan oleh Simorangkir dan Lhaksana (2018) (Simorangkir & Lhaksana, 2018) dengan judul “Analisis Sentimen pada *Twitter* untuk *Games Online Mobile Legends* dan *Arena of Valor* dengan Metode *Naive Bayes Classifier*”. Hasil penelitian beliau memiliki akurasi cukup baik. Berdasarkan apa yang dipaparkan pada latar belakang di atas serta mengacu pada penelitian-penelitian sebelumnya, maka peneliti tertarik untuk melakukan penelitian yang berjudul “**Analisis Sentimen Terhadap Karir E-Sports Di Indonesia Pada Media *Twitter* Menggunakan Metode *Naive Bayes***”.

1.2. Rumusan Masalah

Dengan mempertimbangkan latar belakang yang telah dijelaskan, dapat dirumuskan suatu masalah sebagai berikut:

1. Bagaimana implementasi metode Naive Bayes untuk mengklasifikasikan sentimen data *Twitter* terhadap karir E-Sports di Indonesia?
2. Berapa tingkat akurasi dari implementasi metode Naive Bayes dalam mengklasifikasikan sentimen data *Twitter* terhadap karir E-Sports di Indonesia?
3. Bagaimana visualisasi data dari hasil klasifikasi pada sentimen karir E-Sports di Indonesia pada media sosial *Twitter* berdasarkan wordcloud?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang penulis buat, adapun tujuan dari penelitian ini ialah sebagai berikut:

1. Mengimplementasikan metode Naive Bayes untuk melakukan klasifikasi sentimen data *Twitter* yang berkaitan dengan karir E-Sports di Indonesia.
2. Melihat seberapa akurat implementasi metode Naive Bayes dalam mengklasifikasikan sentimen data *Twitter* terhadap karir E-Sports di Indonesia.

3. Mengetahui hasil visualisasi data dari hasil klasifikasi pada sentimen karir *E-Sports* di Indonesia pada media sosial *Twitter* berdasarkan *wordcloud*.

1.4. Manfaat Penelitian

Adapun hasil dari penelitian ini diharapkan mampu memberikan manfaat sebagai berikut:

1. Dengan hasil analisis sentimen karir *e-sports* di Indonesia pada media sosial *Twitter* ini, maka dapat dijadikan sebagai informasi serta sarana pertimbangan masyarakat untuk berkarir di bidang *e-sports*.
2. Hasil penelitian ini dapat dijadikan sebagai sumber referensi untuk penulisan selanjutnya tentang *e-sports* Indonesia, terutama dalam konteks karir *e-sports* di Indonesia.

1.5. Batasan Masalah

Untuk mencapai sasaran dan tujuan yang diharapkan, penelitian ini hanya membatasi analisis pada data *Twitter* dan tidak termasuk *emoticon*, gambar, foto, video, dan link.

BAB 2. TINJAUAN TEORI

2.1. E-Sport

Beberapa negara seperti Korea Selatan, Swedia, dan Indonesia telah mengakui e-sport sebagai olahraga dan menciptakan badan pengatur resmi di bawah kementerian olahraga. Langkah ini menunjukkan bahwa e-sport semakin diakui sebagai bentuk olahraga yang sah dan mendapatkan tempat di dunia olahraga internasional. Industri e-sport telah tumbuh pesat dalam beberapa tahun terakhir, menjadi salah satu industri dengan pertumbuhan tercepat di dunia. Meskipun sejarah e-sport dapat dilacak sejauh sejarah game itu sendiri, popularitasnya semakin meningkat sejak mode multiplayer diperkenalkan. Hal ini membuka kemungkinan bagi para gamer untuk merasakan pengalaman baru dalam dunia virtual, di mana mereka dapat bermain, bersosialisasi, dan berkompetisi dengan pemain lain dari seluruh dunia. Dengan semakin berkembangnya teknologi dan jaringan internet yang semakin luas, para gamer dapat dengan mudah terhubung dan berpartisipasi dalam kompetisi e-sport, membuktikan bahwa e-sport adalah olahraga yang berkembang dengan pesat dan semakin populer di dunia.

Karir esports di Indonesia semakin berkembang pesat dalam beberapa tahun terakhir. Industri ini menawarkan berbagai peluang bagi individu yang berbakat dan berkomitmen dalam permainan video kompetitif. Banyak turnamen dan kompetisi esports yang diadakan secara rutin di Indonesia, baik secara online maupun offline, dengan hadiah yang menggiurkan. Hal ini memberikan kesempatan bagi para pemain untuk membangun karir profesional dan mencapai pengakuan di dunia esports. Selain itu, semakin banyak sponsor dan investor yang tertarik untuk mendukung industri esports di Indonesia, menciptakan lebih banyak peluang kerja dan peluang bisnis di sektor ini.

Meskipun karir esports di Indonesia menawarkan peluang yang menarik, terdapat tantangan yang perlu dihadapi. Salah satu tantangan utama adalah kurangnya pemahaman dan apresiasi dari masyarakat luas terhadap esports sebagai profesi yang sah dan berpotensi menghasilkan pendapatan yang signifikan. Banyak orang masih melihat esports hanya sebagai hobi atau

kegiatan untuk mengisi waktu luang, tanpa memahami kompleksitas dan dedikasi yang dibutuhkan untuk menjadi seorang pemain profesional. Selain itu, infrastruktur dan dukungan yang memadai, seperti fasilitas pelatihan dan kebijakan yang mendukung, juga masih perlu diperkuat untuk memastikan perkembangan yang berkelanjutan bagi karir esports di Indonesia.

Namun, karir esports di Indonesia menjanjikan masa depan yang cerah. Pemerintah Indonesia dan berbagai pihak terkait semakin menyadari potensi ekonomi dan sosial dari industri esports, dan berupaya meningkatkan dukungan dan pembangunan infrastruktur untuk memajukan sektor ini. Banyak lembaga dan sekolah esports yang didirikan untuk melatih dan mengembangkan bakat-bakat muda dalam permainan kompetitif. Selain itu, para pemain esports Indonesia juga telah meraih prestasi internasional yang membanggakan, membuktikan bahwa Indonesia memiliki bakat yang mampu bersaing di tingkat global. Dengan perkembangan terus-menerus dan komitmen yang kuat dari berbagai pihak, karir esports di Indonesia memiliki potensi untuk terus berkembang dan memberikan peluang yang menjanjikan bagi para pemain yang berdedikasi dan berkompeten.

2.2. Sentimen Analisis

Bidang ilmu komputer yang dikenal sebagai analisis sentimen bertujuan untuk mengolah bahasa alami, komputasi linguistik, dan text mining. Analisis sentimen membahas mengenai emosi, penilaian, sikap, pendapat, sentimen, dan evaluasi seseorang terhadap pembicara atau penulis terkait suatu topik, kegiatan, atau acara. Tugas utama dalam melakukan analisis sentimen adalah memilah-milah teks dalam sebuah kalimat atau dokumen untuk menentukan apakah opini yang disampaikan berbentuk positif, negatif, atau netral. Analisis sentimen dapat digunakan untuk mengetahui opini publik terhadap topik, tokoh, atau institusi tertentu dan membantu menentukan pandangan positif atau negatif tentang hal tersebut (Wayan & Saraswati, 2013). Salah satu teknik dalam analisis sentimen adalah opinion mining yang berfokus pada identifikasi sikap terhadap suatu objek atau orang (Beineke et al., 2004).

Dalam praktiknya, analisis sentimen memungkinkan kita untuk mengukur persentase sentimen positif dan negatif terhadap orang, perusahaan, topik, atau situasi tertentu. Skor dari analisis sentimen dapat dikelompokkan ke dalam tiga kategori yaitu sentimen positif, sentimen negatif, dan sentimen netral. Dengan melakukan analisis sentimen, kita dapat memahami pandangan masyarakat secara lebih terperinci dan membuat keputusan yang lebih baik berdasarkan hasil analisis sentimen tersebut. Namun, analisis sentimen juga dapat lebih terperinci lagi untuk menemukan sumber dari sentimen positif atau negatif pada individu atau kelompok tertentu. Tujuan dari analisis sentimen adalah untuk mengklasifikasikan teks menjadi kalimat atau dokumen, kemudian menentukan apakah pendapat yang dikemukakan dalam teks tersebut bersifat positif, negatif, atau netral (Haaff, 2010).

Dalam analisis sentimen, penting untuk mempertimbangkan bahwa setiap sentimen yang disampaikan terkait dengan topik tertentu dapat memiliki makna yang berbeda tergantung pada konteksnya. Maka dari itu, beberapa penelitian terhadap topik tertentu, sering dimulai dengan mengidentifikasi permasalahan dari topik yang sedang dibahas sebelum memulai proses opinion mining (Manalu, 2014). Hal ini berfungsi untuk memastikan bahwa opini yang dianalisis sesuai dengan topik yang relevan dan memiliki nilai informasi yang lebih tinggi.

Analisis sentimen dilakukan untuk mengevaluasi hal yang diekspresikan oleh pembicara atau penulis sehubungan dengan topik atau tokoh publik meliputi emosi, sikap, pandangan, dan penilaian. Oleh karena itu, banyak penelitian yang berfokus pada topik yang dibahas dan mengidentifikasi permasalahan dari topik yang dibahas sebelum melakukan analisis sentimen. Misalnya, kalimat positif seperti "Jokowi sangat cerdas dan pantas menjadi Presiden" dan kalimat negatif seperti "Saya sangat tidak suka dengan Prabowo" dapat diidentifikasi menggunakan analisis sentimen. Namun, tidak semua kalimat memiliki sentimen yang jelas. Sebagai contoh, kalimat netral seperti "Saya selalu mendukung presiden mana pun untuk kemajuan Indonesia" tidak menunjukkan sentimen yang kuat. Dalam analisis sentimen, skor dapat dibagi menjadi tiga jenis yaitu sentimen positif,

sentimen negatif, dan sentimen netral. Dalam penelitian yang lebih rinci, peneliti dapat menentukan sumber dari sentimen positif atau negatif pada individu atau kelompok tertentu.

2.3. Text Mining

Text mining adalah proses ekstraksi informasi yang berguna dan memiliki arti dari teks yang tidak terstruktur. Ini melibatkan penggunaan metode dan teknik komputasional untuk mengenali dan mengekstrak pola, informasi, dan pengetahuan dari teks yang besar dan kompleks. Tujuan utama dari text mining adalah untuk mengubah teks menjadi bentuk yang dapat dianalisis secara komputasional, memahami makna di balik teks, dan mendapatkan wawasan yang berharga dari data teks tersebut (Hotho et al., 2005).

Menurut Feldman dan Sanger (Feldman, R., & Sanger, 2007), beberapa proses yang dilakukan dalam text mining adalah sebagai berikut :

1. Pengumpulan Data: Tahap awal dalam Text Mining adalah mengumpulkan teks yang akan dianalisis. Data teks dapat berasal dari berbagai sumber, seperti dokumen, situs web, media sosial, atau basis data.
2. Pra-Pemrosesan Teks: Setelah data teks dikumpulkan, langkah selanjutnya adalah pra-pemrosesan teks. Ini melibatkan pembersihan dan normalisasi teks untuk menghilangkan karakter khusus, tanda baca, mengubah huruf besar ke huruf kecil, menghapus kata-kata pengisi (stop words), dan melakukan stemming atau lemmatisasi untuk mengubah kata-kata menjadi bentuk dasar.
3. Tokenisasi: Proses tokenisasi melibatkan pembagian teks menjadi unit-unit yang lebih kecil yang disebut token. Token bisa berupa kata, frasa, atau kata-kata terpisah tergantung pada tujuan analisis.
4. Analisis Frekuensi Kata: Pada tahap ini, frekuensi kemunculan kata dalam dokumen atau korpus dianalisis. Hal ini melibatkan menghitung berapa kali setiap kata muncul dalam teks dan menciptakan distribusi frekuensi kata

5. **Klasifikasi dan Pengelompokan:** Proses ini melibatkan pengelompokan atau klasifikasi teks berdasarkan pola atau karakteristik yang ada. Teknik seperti klasifikasi teks, pengelompokan teks, atau analisis sentimen dapat digunakan untuk mengelompokkan teks ke dalam kategori atau label yang relevan.
6. **Ekstraksi Informasi:** Tahap ini melibatkan pengekstrakan informasi penting atau pengetahuan dari teks. Ini bisa termasuk identifikasi entitas, ekstraksi relasi, ekstraksi kejadian, atau pengenalan pola tertentu dalam teks.
7. **Analisis Sentimen:** Analisis sentimen melibatkan penentuan sentimen atau pendapat yang terkandung dalam teks. Hal ini dapat dilakukan dengan menggunakan teknik seperti analisis sentimen berbasis aturan, pendekatan berbasis klasifikasi, atau metode berbasis pembelajaran mesin.
8. **Visualisasi dan Interpretasi:** Setelah tahap analisis, hasil dari Text Mining dapat divisualisasikan untuk membantu pemahaman dan interpretasi. Visualisasi seperti *wordcloud*, grafik, atau plot dapat digunakan untuk menampilkan hasil secara intuitif.

2.3.1 Preprocessing

Tahap awal dalam melakukan *Text Mining* adalah *Preprocessing*, proses ini memiliki tujuan untuk memproses seluruh dokumen teks yang awalnya tidak terstruktur menjadi terstruktur agar saat pengolahan data pada proses selanjutnya dapat digunakan dengan baik. *Preprocessing* merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur (Luhulima, 2013) . Berikut adalah beberapa tahapan *preprocessing* dan penjelasannya :

1. *Case Folding*

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil atau huruf besar. Tujuannya adalah untuk mencapai konsistensi dalam representasi teks. Misalnya, kata-kata "Data", "data", dan "DATA" akan diubah menjadi "data" untuk menghindari perbedaan yang

tidak relevan dalam analisis teks.

2. *Cleansing*

Cleansing adalah proses membersihkan teks dari elemen yang tidak relevan atau mengganggu, seperti tanda baca, nomor, karakter khusus, atau URL. Teknik ini melibatkan penggunaan ekspresi reguler atau aturan pemfilteran khusus untuk menghapus elemen yang tidak diinginkan dalam teks.

3. *Tokenizing*

Tokenizing adalah proses membagi teks menjadi unit-unit yang lebih kecil yang disebut token. Token dapat berupa kata, frasa, atau kalimat. Proses tokenizing memungkinkan kita untuk memperlakukan setiap unit sebagai entitas terpisah yang dapat diolah dan dianalisis lebih lanjut. Misalnya, kalimat "Saya sedang belajar text mining" akan dipecah menjadi token-token berikut: "Saya", "sedang", "belajar", "text", "mining".

4. *Stopword Removal*

Stopword removal melibatkan penghapusan kata-kata pengisi atau kata-kata umum yang sering muncul dalam teks, seperti "dan", "di", "untuk". Kata-kata ini biasanya tidak memberikan banyak informasi dan dapat mempengaruhi hasil analisis. Menghapus kata-kata pengisi membantu memfokuskan analisis pada kata-kata yang lebih informatif.

5. *Stemming*

Stemming adalah proses mengubah kata-kata menjadi bentuk dasarnya atau akar kata. Stemming menghapus akhiran kata untuk mendapatkan akar yang sama. Misalnya, kata-kata "bermain", "bermainan", dan "bermain-main" akan diubah menjadi "main". Tujuan stemming adalah untuk menggabungkan variasi kata yang memiliki akar yang sama agar tidak mempengaruhi analisis yang berkaitan dengan kata-kata tersebut.

2.4. **Algoritma Term Frequency-Inverse Document Frequency (TF-IDF)**

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode yang digunakan dalam analisis teks untuk mengukur seberapa penting suatu kata dalam suatu dokumen. Metode ini menggabungkan dua konsep

yaitu *term frequency* (frekuensi kata) dan *inverse document frequency* (kebalikan frekuensi dokumen).

1. Term Frequency (TF):

TF mengukur seberapa sering suatu kata muncul dalam suatu dokumen. Rumus umum untuk menghitung TF adalah:

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \dots\dots\dots(1)$$

TF memberikan bobot lebih tinggi pada kata-kata yang muncul lebih sering dalam dokumen tersebut. Dengan demikian, kata-kata yang muncul lebih sering cenderung memiliki nilai TF yang lebih tinggi.

2. Inverse Document Frequency (IDF):

IDF mengukur seberapa umum atau jarang suatu kata muncul di seluruh koleksi dokumen. Rumus umum untuk menghitung IDF adalah:

$$idf_t = \log_{10} \left(\frac{N}{df(t)} \right) \dots\dots\dots(2)$$

Keterangan :

N : total seluruh dokumen yang ada.

df(t) : total kemunculan frekuensi pada term.

IDF memberikan bobot lebih tinggi pada kata-kata yang jarang muncul di banyak dokumen dalam koleksi. Dengan demikian, kata-kata yang jarang muncul di banyak dokumen cenderung memiliki nilai IDF yang lebih tinggi.

3. TF-IDF:

TF-IDF adalah hasil perkalian antara TF dan IDF. Rumus umum untuk menghitung TF-IDF adalah:

$$W_{t,d} = W_{tf,t,d} \times Idf_t \dots\dots\dots(3)$$

Keterangan:

$W_{tf,t,d}$: nilai dari Term Frequency.

Idft : nilai Invers Document Frequency.

Dengan mengalikan TF dengan IDF, kita mendapatkan bobot yang mencerminkan seberapa penting suatu kata dalam dokumen tersebut dan dalam konteks keseluruhan koleksi dokumen.

2.5. Metode Naive Bayes

Metode klasifikasi Naïve Bayes menggunakan Teorema Bayes untuk melakukan klasifikasi data dengan efisien berdasarkan training set (Hadiyani, 2013). Secara spesifik, Naïve Bayes mengasumsikan bahwa nilai input atribut untuk kelas tertentu tidak bergantung pada nilai atribut lainnya. Teorema Bayes sendiri merupakan konsep yang dikemukakan oleh Thomas Bayes dari Inggris, yang bertujuan untuk memprediksi peluang di masa depan berdasarkan pengalaman di masa lalu, sehingga dikenal sebagai Teorema Bayes. Persamaan matematika (4) yang mendasari Teorema Bayes adalah:

$$P(c|x) = \frac{P(x|c) P(c)}{P(x)} \dots\dots\dots(4)$$

Keterangan:

- X : Data dengan kelas yang belum diketahui
- C : Hipotesis data X merupakan suatu kelas spesifik
- P(C|X): Probabilitas hipotesis C berdasar kondisi X (probabilitas posterior)
- P(C) : Probabilitas hipotesis C (probabilitas prior)
- P(X|C) : Probabilitas X berdasarkan kondisi pada hipotesis C
- P(X) : Probabilitas X

Untuk menjelaskan Teorema Naive Bayes, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, Teorema Bayes pada persamaan (4) disesuaikan menjadi persamaan (5):

$$P(C_k | X_1, \dots, X_n) = \frac{P(X_1, \dots, X_n | C_k) P(C_k)}{P(X_1, \dots, X_n)} \dots\dots\dots(5)$$

Jika Variabel C merepresentasikan kelas dan Variabel X1, Xn merepresentasikan karakteristik petunjuk untuk melakukan klasifikasi atau

kriteria, maka persamaan tersebut menjelaskan peluang masuknya sampel karakteristik tertentu ke dalam kelas C (Posterior). Persamaan tersebut dinyatakan dengan mengalikan peluang munculnya kelas C sebelum masuknya sampel tersebut (prior) dengan peluang kemunculan karakteristik-karakteristik sampel pada kelas C (likelihood), dan kemudian dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (evidence). rumus tersebut dapat ditulis secara sederhana sebagai berikut:

$$Posterior = \frac{Likelihood \times Prior}{Evidence} \dots\dots\dots(6)$$

Dalam metode Naïve Bayes, nilai Evidence pada suatu sampel tetap sama untuk setiap kelas. Kemudian nilai posterior tersebut dibandingkan dengan nilai posterior dari kelas-kelas lainnya untuk menentukan kelas mana yang akan diberikan pada sampel tersebut. Untuk menjelaskan persamaan Teorema Bayes secara lebih rinci, persamaan $(C|X_1, \dots, X_n)$ dinyatakan dengan rumus sebagai berikut.:

$$P(C|X_1, \dots, X_n) = P(C) P(X_1, \dots, X_n|C) \dots\dots\dots(7)$$

Terlihat bahwa hasil penguraian dapat menyebabkan faktor-faktor persyaratan yang mempengaruhi nilai probabilitas menjadi semakin banyak dan kompleks, yang sulit untuk dianalisis secara individu. Oleh karena itu, perhitungan menjadi sulit untuk dilakukan. Di sinilah asumsi independensi yang sangat tinggi (naïf) digunakan, yaitu bahwa setiap kriteria (X_1, X_2, \dots, X_n) tidak terkait satu sama lain secara signifikan dan saling bebas (independen).

Algoritma Naive Bayes Classifier merupakan suatu konsep probabilitas yang digunakan untuk mengelompokkan dokumen ke dalam kelas-kelas tertentu. Dalam pengolahan data, algoritma Naive Bayes Classifier dapat mengatasi volume data yang besar dengan tingkat akurasi yang tinggi. Proses algoritma ini melibatkan dua tahap, yaitu tahap pelatihan yang melibatkan kumpulan contoh dokumen sebagai data latih, dan tahap klasifikasi dokumen yang belum memiliki label kelas.

Algoritma ini didasarkan pada teori probabilitas yang dikembangkan

oleh Thomas Bayes, seorang ilmuwan asal Inggris, yang memungkinkan perkiraan probabilitas di masa depan berdasarkan pengalaman di masa lalu. Algoritma Naive Bayes Classifier mengasumsikan bahwa setiap data dalam dokumen tidak memiliki hubungan satu sama lain, sehingga disebut "naive".

2.6. Gaussian Naïve Bayes

Ketika berhadapan dengan data kontinu, asumsi yang khas adalah bahwa nilai kontinu yang berkorelasi dengan setiap kelas didistribusikan sesuai dengan distribusi Gaussian. Data pelatihan dibagi berdasarkan kelas dan mean dan standar deviasi dari setiap kelas dihitung. Oleh karena itu untuk memperkirakan probabilitas kumpulan data kontinu, persamaan berikut dapat digunakan

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \dots\dots\dots(8)$$

2.7. Confusion Matrix

Confusion matrix adalah sebuah teknik yang sering digunakan dalam konsep data mining untuk menghitung akurasi model. Matriks ini memberikan gambaran tentang sejauh mana model berhasil dalam mengklasifikasikan data ke dalam kelas yang benar atau salah (Powers, 2020). Confusion matrix terdiri dari empat istilah utama: true positive (TP), true negative (TN), false positive (FP), dan false negative (FN).

Tabel 2.1 Tabel *Confusion Matrix*

Aktual	Classified as	
	+	-
+	TP (<i>True positives</i>)	FN (<i>False negative</i>)
-	FP (<i>False positives</i>)	TN (<i>True negative</i>)

Perhitungan akurasi dengan tabel *Confusion Matrix* adalah sebagai berikut:

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN}$$

Presisi dapat dijelaskan sebagai rasio antara item yang relevan yang terpilih dengan jumlah total item yang terpilih. Presisi mencerminkan sejauh mana jawaban dari sistem atau model yang diberikan cocok dengan permintaan informasi. Rumus presisi adalah:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Presisi dapat dinyatakan dalam bentuk angka sebagai persentase (1-100%) atau dalam bilangan antara 0-1. Sistem rekomendasi yang memiliki nilai presisi yang tinggi dianggap sebagai sistem yang baik. Rumus Recall adalah :

$$\text{Recall} = \frac{TP}{TP+FN}$$

Recall adalah gambaran tolak ukur keberhasilan suatu model dalam menemukan sebuah informasi dengan menentukan rasio prediksi yang memiliki nilai *true positif* dibandingkan dengan seluruh data yang memiliki nilai *true positif*. Rumus F-measure adalah:

$$\text{F-measure} = \frac{2 * (\text{Recall} + \text{presisi})}{\text{Recall} + \text{Presisi}} \times 100 \%$$

F-measure, juga dikenal sebagai F1 score, adalah metrik evaluasi yang menggabungkan presisi (precision) dan recall. F-measure mengukur keseimbangan antara presisi dan recall dengan menghitung nilai harmonik dari keduanya.

2.8. *Wordcloud*

Wordcloud adalah gambaran grafis dari sebuah dokumen yang dibuat dengan cara menempatkan kata-kata yang sering muncul dalam dokumen ke dalam sebuah ruang dimensi. Ukuran kata dalam gambaran tersebut merepresentasikan frekuensi kemunculan kata tersebut dalam dokumen. Jika ukuran kata semakin besar, maka menunjukkan bahwa frekuensi kemunculan kata tersebut semakin tinggi dalam dokumen.

2.9. Penelitian Terdahulu

Sebelum dilakukannya penelitian ini, telah ada beberapa penelitian sebelumnya yang relevan dengan topik yang akan dibahas dalam penelitian ini. Salah satunya adalah penelitian yang dilakukan oleh Mujib Ridwan, Hadi Suyono, dan M. Sarosa pada tahun 2013 (Ridwan et al., 2013). Penelitian tersebut dilakukan dengan beberapa tahap, seperti membuat proposal,

mengumpulkan data, mengolah data, melakukan implementasi NBC, melakukan pengujian, dan menganalisis hasilnya. Pengujian ini menghasilkan nilai precision, recall, dan accuracy masing-masing 83%, 50%, dan 70%..

Penelitian kedua dilakukan oleh Ling dkk (LING et al., 2014) yang mengembangkan aplikasi untuk menganalisis sentimen dengan menggunakan metode Naïve Bayes dan fitur Chi Square. Penelitian tersebut mencapai tingkat akurasi sebesar 83% dengan F1-scorenya yaitu 90.713%.

Penelitian ketiga dilakukan oleh Falahah dan Nur (Falahah & Nur, 2015) dengan judul "Pengembangan Aplikasi Sentimen Analisis Menggunakan Metode Naïve Bayes (Studi Kasus Sentimen Analisis dari Media Twitter)". Penelitian ini mengklasifikasikan opini publik di Twitter terkait layanan pemerintah terhadap masyarakat dengan menggunakan metode Naïve Bayes Classifier untuk melakukan analisis sentimen dan menghasilkan klasifikasi sentimen positif, negatif, atau netral. Aplikasi Sentimen Analisis yang dikembangkan dianggap cukup memadai, dikarenakan hasil uji akurasi klasifikasi yang dilakukan oleh aplikasi, pada sekumpulan data testing dan dilakukan dengan 3 variasi scenario, menghasilkan nilai akurasi tertinggi sebesar 73% dengan jumlah kata training 100 negatif, 100 positif, dan 100 netral.

Sebuah studi keempat dilakukan oleh Wicaksono (WICAKSONO, 2016) dengan tujuan untuk menganalisis sentimen pada pemilihan Presiden Amerika Serikat melalui penggunaan media sosial dengan menggunakan metode Naive Bayes, khususnya Multinomial Naïve Bayes. Studi ini menunjukkan bahwa Naive Bayes Classifier menghasilkan tingkat akurasi sebesar 83%.

BAB 3. METODOLOGI PENELITIAN

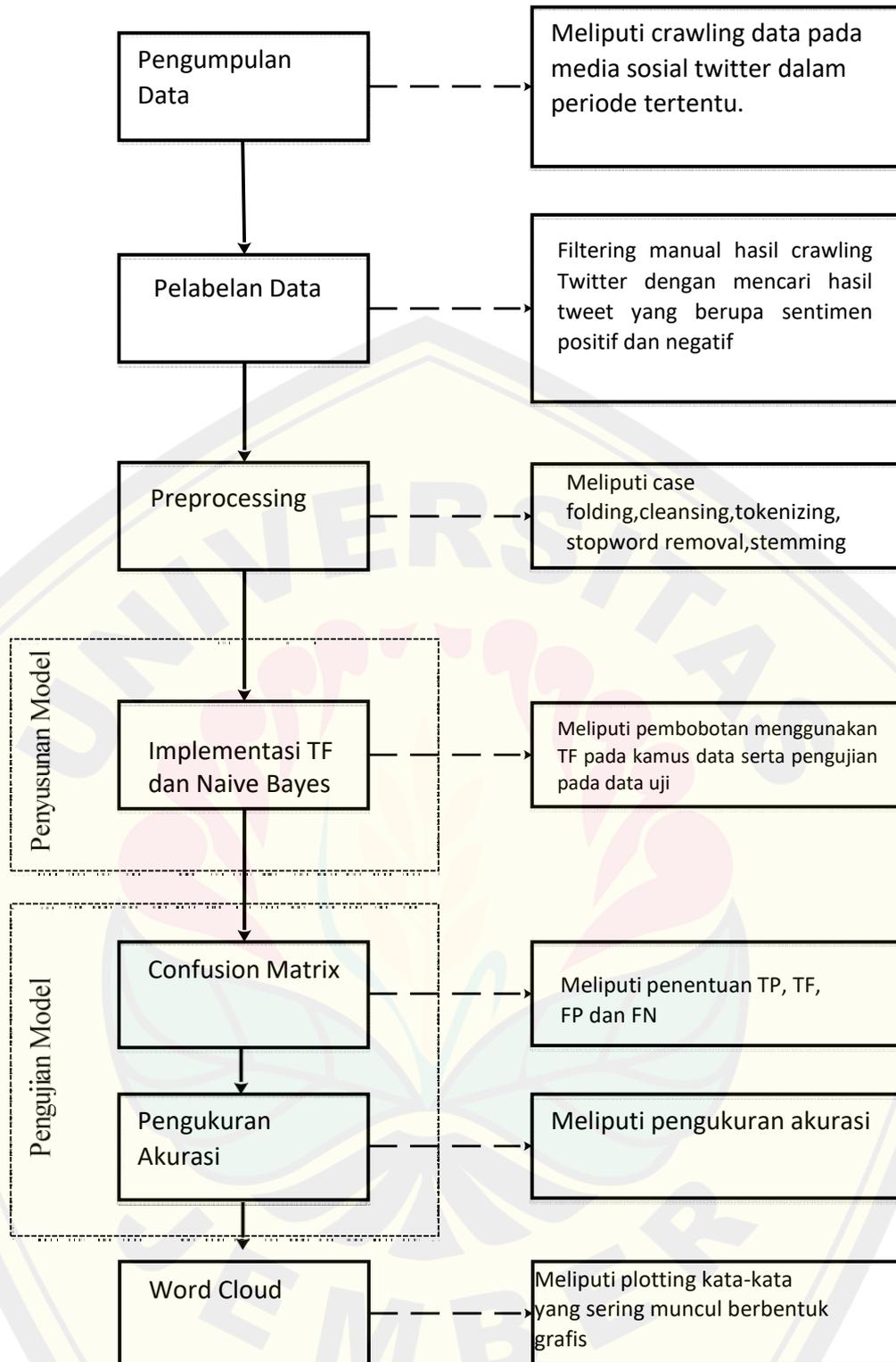
3.1. Jenis Penelitian

Penelitian ini berfokus pada analisis sentimen masyarakat mengenai karir e-sport di Twitter. Jenis penelitian yang digunakan adalah penelitian kuantitatif karena penelitian ini bertujuan untuk mengumpulkan dan menganalisis data numerik secara sistematis. Dalam penelitian ini, data sentimen dari tweet-tweet yang terkait dengan karir e-sport di Twitter akan dikumpulkan dan dianalisis menggunakan metode kuantitatif.

Penelitian kuantitatif akan memungkinkan peneliti untuk mengukur sejauh mana sentimen masyarakat terhadap karir e-sport, apakah cenderung positif atau negatif. Metode pengumpulan data yang digunakan dapat berupa pengambilan sampel tweet secara acak atau dengan menggunakan kata kunci tertentu. Selanjutnya, data sentimen dari tweet-tweet tersebut dapat dihitung dan dianalisis menggunakan teknik-teknik statistik untuk mendapatkan pemahaman yang lebih komprehensif tentang pandangan masyarakat terhadap karir e-sport.

3.2. Tahapan Penelitian

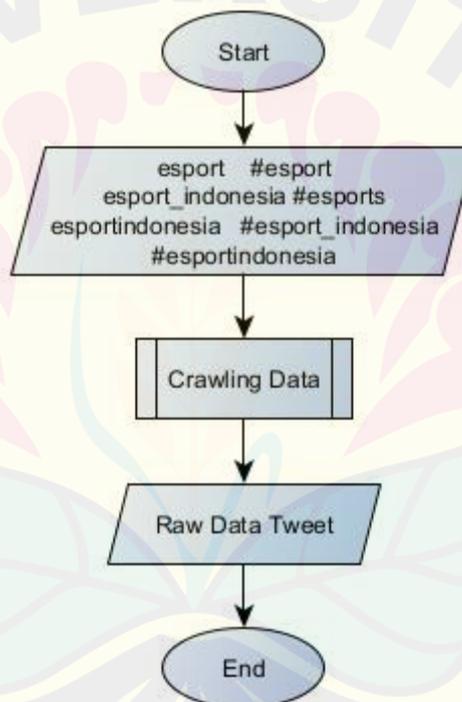
Diagram tahapan penelitian untuk mengklasifikasikan sentimen masyarakat di Twitter terkait karir e-sport dilakukan melalui beberapa tahap. Struktur visual dari tahapan penelitian tersebut dapat dilihat diagram pada Gambar 3.1



Gambar 3.1. Tahapan Penelitian

3.3. Metode Pengumpulan Data

Pengumpulan data didapat dari hasil crawling opini masyarakat terhadap karir e-sport dengan kata kunci dan tagar `esport`, `esport_indonesia`, `esportindonesia`, `#esport`, `#esports`, `#esport_indonesia` dan `#esportindonesia` di Twitter. Dijelaskan bahwa langkah awal untuk mendapatkan data tweet adalah dengan melakukan coding menggunakan bahasa pemrograman python. Peneliti menggunakan library `snsrape` sebab menggunakan library tersebut penggunaannya cukup mudah dan tidak memerlukan API Key dalam proses crawling. Berikut adalah flowchart dari proses crawling :



Gambar 3.2 Alur Pengumpulan Data

Proses pengambilan data menggunakan kata kunci dan tagar yang sudah ditentukan sesuai topik untuk pengambilan data tweet. Data yang diperoleh dari crawling masih dalam bentuk data mentah berupa file csv. Selanjutnya, data tersebut akan dipilah dan dipecah berdasarkan kebutuhan penelitian, dalam hal ini untuk tujuan analisis sentimen, akan dibuat dataset latih dan dataset uji.

a. Partisi data berdasar calon

Setelah data hasil crawling dari Python diperoleh dalam bentuk csv, tahap selanjutnya adalah menghapus atribut yang tidak diperlukan. Dalam hal ini, beberapa atribut yang dihapus adalah *favorited*, *favoriteCount*, *replyToSN*, *created*, *truncated*, *replyToSID*, *id*, *replyToUID*, *statusSource*, *retweetCount*, *isRetweet*, *retweeted*, *longitude*, dan *latitude*. Selanjutnya, noise akan dihilangkan, seperti Retweet yang dianggap tidak memberikan tambahan informasi dan sama dengan tweet asli. Proses penghapusan Retweet dilakukan secara manual pada excel dengan mencari kata kunci "RT" dan menghapusnya. Contoh penghapusan data Retweet dapat dilihat pada tabel 3.1.

Semua tweet yang memiliki awalan "RT" yang berarti retweet akan dihapus secara keseluruhan. Setelah itu, data tweet dari hasil *crawling* akan dipisahkan menjadi dua, yaitu tweet yang digunakan dan yang tidak digunakan. Pada penelitian ini, tweet yang berisi retweet akan dihapus karena dianggap memiliki perasaan yang sama seperti tweet asli tanpa menambahkan konteks atau informasi yang berbeda.

b. Penyusunan data latih dan uji

Setelah tweet dikelompokkan ke dalam beberapa kategori, langkah selanjutnya adalah mengorganisasikannya menjadi dataset latih dan uji. Ada empat kali pengujian yang dilakukan pada dataset bersih yang sudah dibuat, dengan membandingkan skema perbandingan antara data training dan data testing sebesar 60%:40%, 70%:30%, 80%:20%, dan 90%:10% dari total data.

3.4. Pelabelan Data

Penulis melakukan pelabelan dataset secara manual dengan menghilangkan kata-kata yang disingkat oleh pengguna twitter dan merujuk pada kamus stopword dari pihak ketiga yaitu Sastrawi yang merujuk pada KBBI. Dalam penelitian ini, terdapat 2000 data tweet sentimen yang diperoleh dari beberapa hashtag yang ditentukan, kemudian dilakukan pemilihan dataset yang mewakili sentimen positif dan negatif. Kelas positif ditentukan dari tweet yang mengandung kalimat positif, pernyataan setuju dan mendukung, sedangkan kelas negatif dituliskan sebagai tweet yang

mengandung kata-kata negatif, penolakan, dan kontra.

3.5. Preprocessing

Proses *preprocessing* bertujuan untuk mempersiapkan data agar siap diproses pada tahap pemodelan. Pada tahap ini, data tweet yang digunakan untuk training dan testing dibersihkan dari berbagai noise seperti tautan, "RT", "@", *stopword*, simbol, angka, konversi gambar dan video, dan hashtag. Tahap *preprocessing* terdiri dari beberapa tahapan, yaitu *case folding*, pembersihan, *stopword*, konversi emoticon, konversi negasi, tokenizer, dan *stemming*. Setiap tahapan memiliki penjelasan berikut ini:

a. Case Folding

Proses *Case Folding* dilakukan untuk mengubah huruf besar menjadi huruf kecil agar semua huruf memiliki format yang seragam. Sebagai contoh, dapat dilihat sebagai:

b. Cleansing

Dalam tweet yang terkait dengan Pilpres, terdapat beberapa komponen seperti karakter #, link dan RT, angka, dan simbol yang tidak memberikan pengaruh pada sentimen. Oleh karena itu, komponen-komponen tersebut akan dihapus melalui proses *cleansing*. Berikut ini adalah contoh dari proses *cleansing* tersebut.

c. Convert Negation

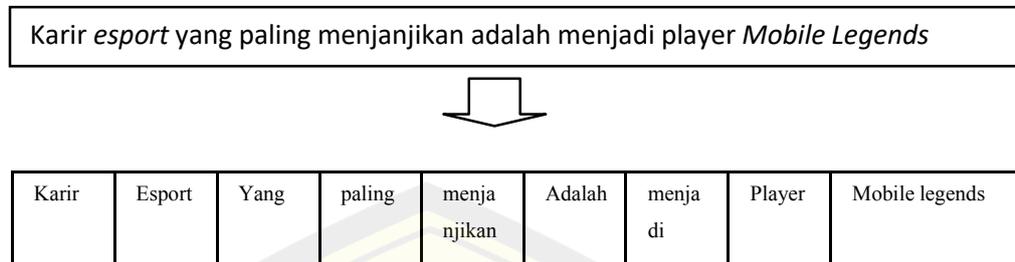
Proses *Convert Negation* bertujuan untuk mengonversi kata-kata negasi dalam suatu tweet, karena kata-kata tersebut dapat mempengaruhi sentimen yang terkandung dalam tweet tersebut. Dalam proses ini, kata-kata negasi yang terdapat dalam tweet akan digabungkan dengan kata yang mengikutinya. Contohnya antara lain "bukan", "bkn", "tidak", "enggak", "g", "ga", "jangan", "nggak", "tak", "tdk", dan "gak".

d. Tokenizing

Pada tahap *tokenizing*, setiap kata pada tweet akan dipisahkan. Hal ini dilakukan dengan memisahkan setiap kata yang dipisahkan oleh spasi.

Proses ini bertujuan agar tahap *preprocessing* selanjutnya dapat dilakukan

dengan baik. Contoh dari proses tokenizing dapat dilihat seperti berikut:



Gambar 3.3. Gambaran *Tokenizing*

3.6. Penyusunan Model

Setelah tahapan preprocessing selesai dilakukan, langkah selanjutnya adalah membangun model dengan menggunakan algoritma TF-IDF. Tahap ini dimulai dengan menghitung nilai Term Frequency (TF) untuk setiap kata dalam setiap tweet. Selanjutnya, dilakukan penghitungan Inverse Document Frequency (IDF) dengan menggunakan nilai log dari Document Frequency (DF) yang telah dihitung sebelumnya. Kemudian, bobot setiap kata dalam sebuah tweet dihitung dengan mengalikan nilai TF dengan nilai IDF. Tahap selanjutnya adalah penyusunan model menggunakan algoritma Naive Bayes Classifier untuk mengklasifikasikan tweet ke dalam dua kelas yaitu positif dan negatif.

Penyusunan model bertujuan untuk mengevaluasi keakuratan prediksi sistem berdasarkan dataset yang telah dibersihkan dan diolah dengan model diuji dengan menggunakan skema pengujian empat kali dan rasio antara data training dan data testing sebesar 60%:40%, 70%:30%, 80%:20%, dan 90%:10% untuk mengevaluasi keakuratan sistem. Tahap training melibatkan pembangunan model probabilitas dengan menggunakan data training, yang meliputi menentukan nilai prior probability, menentukan nilai conditional probability, dan menghitung hasil prediksi dengan menggunakan data testing.

3.7. Pengujian Model

Confusion matrix digunakan untuk menguji model yang dihasilkan dan menilai kriteria serta mengukur tingkat akurasi.

3.7.1. Confusion Matrix

Dalam tahap pengukuran, akurasi dan presisi dari klasifikasi menggunakan Naïve Bayes classifier akan dievaluasi berdasarkan tabel Confusion Matrix. Hasil perhitungan model yang sudah dilakukan pada proses sebelumnya dimasukkan kedalam kriteria. Untuk menentukan kriteria tersebut, dapat merujuk pada tabel confusion matrix yang terdapat pada bab 2 ialah sebagai berikut:

- a. Apabila nilai aktual positif dan diklasifikasikan sebagai positif, maka dikategorikan sebagai *true positive* (TP).
- b. Apabila nilai aktual positif tetapi diklasifikasikan sebagai negatif, maka dikategorikan sebagai *false negative* (FN).
- c. Apabila nilai aktual negatif dan diklasifikasikan sebagai negatif, maka dikategorikan sebagai *true negative* (TN).
- d. Apabila nilai aktual negatif tetapi diklasifikasikan sebagai positif, maka dikategorikan sebagai *false positive* (FP).

3.7.2. Pengukuran Akurasi

Dalam penelitian ini, para peneliti menggunakan pengukuran tingkat akurasi untuk setiap profesi dalam industri esport dengan tujuan untuk mengevaluasi kualitas model klasifikasi yang telah dibuat. Hal ini dilakukan dengan merujuk pada persamaan yang telah ditentukan. :

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN}$$

Persamaan di atas digunakan untuk mengukur tingkat akurasi semua calon.

3.8 *Wordcloud*

Langkah berikutnya adalah melakukan plot terhadap kata-kata yang muncul pada dokumen sentimen tentang karir e-sport. Proses visualisasi *wordcloud* ini akan menghasilkan grafik yang memuat kata-kata yang terdapat pada dokumen tersebut. Ukuran font pada *wordcloud* akan menunjukkan seberapa sering kata tersebut muncul dalam dokumen. Ukuran font yang semakin besar akan menandakan bahwa frekuensi kemunculan kata tersebut semakin besar pula.



BAB 4. HASIL DAN PEMBAHASAN

4.1. Hasil Dataset

Proses pengumpulan *dataset* menggunakan library *snsrapper*. Peneliti menggunakan *library snsrapper* sebab penggunaannya yang mudah dan tidak memerlukan *API Key* dalam proses *crawling data* yang nantinya digunakan peneliti untuk pembentukan *dataset*. *Dataset* diperlukan untuk pembentukan data latih dan data uji kemudian akan dilabeli dengan kelas klasifikasi positif dan negatif. Penulis melakukan pelabelan dataset secara manual dengan menghapus kata-kata yang disingkat oleh pengguna Twitter dan menggunakan kamus stopword dari Sastrawi yang merujuk pada KBBI.

Total dataset dalam penelitian ini terdiri dari 2000 data tweet yang dikumpulkan dari beberapa hashtag yang telah ditentukan. Selanjutnya, dilakukan pemilihan dataset yang mewakili sentimen positif dan negatif. Contoh *dataset* dapat dilihat pada tabel 4.1 sebagai berikut:

Tabel 4.1. Contoh *Dataset*

No	Data Training	Sentimen
1	@molangta @nandayogie_ @Mistaaaw @mlbbfess Esport itu udah paling enak	positif
2	kakak gw nyoba daftar susah saingan banyak	Negatif
3	Kerja main doang, gaji gede anjay	Positif
4	player mentok ML juga jd streamer, percuma	Negatif
5	Pas pensiun plonga plongo ga jelas mo ngapain mending pns	Negatif

4.2. Hasil Implementasi Sistem

Pada penelitian ini, penulis menggunakan bahasa pemrograman *Python 3.11.1*. Hasil implementasi sistem dijelaskan sebagai berikut:

4.2.1 Implementasi Crawling Data

Proses *Crawling Data* menggunakan *library sncraper*. Kode program proses *Crawling Data* dapat dilihat pada gambar 4.1 sebagai berikut:

```
import sncraper.modules.twitter as nstwitter
import pandas as pd

data_container = []

for i, tweet in enumerate(nstwitter.TwitterSearchScrapper(query="esport OR esport_indonesia OR esportindonesia OR #esport_indonesia OR #esportindonesia")):
    if i > 2000 :
        break
    data_container.append([tweet.rawContent, tweet.hashtags, tweet.lang])

data_pd = pd.DataFrame(data_container, columns=["Tweet", "hashtags", "Lang"])
data_pd.to_excel('data\data_tweet.xlsx', index=False, header=True)
```

Gambar 4.1. Kode program proses *Crawling Data*

Pada kode program tersebut, sistem diberi perintah untuk mengambil data berdasarkan hastag : *esport*, *esport_indonesia*, *esportindonesia*, *#esport*, *#esports*, *#esport_indonesia*, *#esportindonesia* sebanyak 2000 data *tweet* dengan semua bahasa dan disimpan pada variabel sementara bernama *data_container*.

4.2.2 Implementasi Preprocessing

Proses *Preprocessing* terdiri dari beberapa tahapan yaitu:

a. Case Folding

Case Folding adalah proses mengubah semua huruf pada *dataset* menjadi huruf kecil. Kode program proses *Case Folding* dapat dilihat pada gambar 4.2 sebagai berikut:

```
# CASE FOLDING
def case_folding(text) :
    return text.lower()
df['Case Folding'] = df['Tweet'].apply(case_folding)
df['Tokenizing'] = df['Tweet'].apply(case_folding)
```

Gambar 4.2. Kode program proses *Case Folding*

b. Tokenizing

Tokenizing adalah proses menghapus *emoticon*, simbol, *hashtag*, *digit* atau angka, dan menghapus *username* atau *url*. Kode program proses *Tokenizing* dapat dilihat pada gambar 4.3 sebagai berikut

```
# TOKENIZING
def remove_tweet_special(text):
    text = text.replace('\t', " ").replace('\n', " ").replace('\u', " ").replace('\l', "")
    text = text.encode('ascii', 'replace').decode('ascii')
    text = ' '.join(re.sub("([@#][A-Za-z0-9+])|(\\w+:\\w+\\/\\w+)", " ", text).split())
    return text.replace("http://", " ").replace("https://", " ")
df['Tokenizing'] = df['Tokenizing'].apply(remove_tweet_special)

def remove_number(text):
    return re.sub(r"d+", "", text)
df['Tokenizing'] = df['Tokenizing'].apply(remove_number)

def remove_punctuation(text):
    return text.translate(str.maketrans("", "", string.punctuation))
df['Tokenizing'] = df['Tokenizing'].apply(remove_punctuation)

def remove_whitespace_LT(text):
    return text.strip()
df['Tokenizing'] = df['Tokenizing'].apply(remove_whitespace_LT)

def remove_whitespace_multiple(text):
    return re.sub('\s+', ' ', text)
df['Tokenizing'] = df['Tokenizing'].apply(remove_whitespace_multiple)

def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)
df['Tokenizing'] = df['Tokenizing'].apply(remove_singl_char)

def word_tokenize_wrapper(text):
    return word_tokenize(text)
df['Tokenizing'] = df['Tokenizing'].apply(word_tokenize_wrapper)
```

Gambar 4.3. Kode program proses *Tokenizing*

c. *Filtering*

Filtering adalah proses mengidentifikasi tiap kata untuk kemudian diklasifikasi apakah termasuk kedalam kamus *stopword*. Kode program proses *Filtering* dapat dilihat pada gambar 4.4 sebagai berikut:

```
# FILTERING (STOPWORD REMOVAL)
list_stopwords = stopwords.words('indonesian')

list_stopwords = set(list_stopwords)

def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

df['Stopword Removal'] = df['Tokenizing'].apply(stopwords_removal)
```

Gambar 4.4. Kode program proses *Filtering*

d. *Stemming*

Stemming adalah proses adalah proses mengubah kata berimbuhan menjadi kata dasar menggunakan *library sastrawi*. Kode program proses *Stemming* dapat dilihat pada gambar 4.5 sebagai berikut:

```
# PERSIAPAN DATA
data = pd.read_excel('data/data_preprocessing.xlsx', usecols=["Tweet", "Stopword Removal"])
df = pd.DataFrame(data)
df['Stopword Removal'] = df['Stopword Removal'].apply(literal_eval)

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

def stemmed_wrapper(text):
    return stemmer.stem(text)

term_dict = {}
for document in df['Stopword Removal']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ''

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)

def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['Stemmer'] = df['Stopword Removal'].swifter.apply(get_stemmed_term)

df.to_excel('data/data_preprocessing2.xlsx', index=False, header=True)
```

Gambar 4.5. Kode program proses *Stemming*

4.2.3 Implementasi Penyusunan Model

Langkah pertama adalah proses pemberian bobot data tweet berdasarkan klasifikasi positif dan negatif yang sudah diberikan oleh peneliti. Klasifikasi positif akan diberi bobot satu dan negatif akan diberi bobot minus satu. Kode program proses *pembobotan* dapat dilihat pada gambar 4.6 sebagai berikut:

```
data = pd.read_excel('data/data_analisis2.xlsx')
df = pd.DataFrame(data)

tweet_positif = 0
tweet_netral = 0
tweet_negatif = 0

for label in df['Labeling']:
    if label == "positif":
        tweet_positif += 1
    elif label == "netral":
        tweet_netral += 1
    else:
        tweet_negatif += 1

df['Tweet_clean'] = df['Preprocessing']

def set_sentiment(text):
    if text == "positif":
        return 1
    elif text == "netral":
        return 0
    else:
        return -1

df['Sentiment'] = df['Labeling'].apply(set_sentiment)
df.to_excel('data/data_analisis3.xlsx', index=False, header=True, columns=["Tweet", "Tweet_clean", "Sentiment"])
```

Gambar 4.6. Kode program proses *pembobotan*

Proses perhitungan TF-IDF menggunakan *Library TfidfVectorizer*. Langkah selanjutnya adalah proses pembentukan *confusion matrix* dengan menghitung nilai akurasi, presisi, recall dan *F-measure* menggunakan *library sklearn*. Kode program proses pembentukan *confusion matrix* dapat dilihat pada gambar 4.7 sebagai berikut:

```
data = pd.read_excel('data/data_analisis3.xlsx')
df = pd.DataFrame(data)

x = df['Tweet_clean']
y = df['Sentiment']

vactorizer = TfidfVectorizer(max_features=2500)
model_g = GaussianNB()

v_data = vactorizer.fit_transform(x).toarray()

X_train, X_test, y_train, y_test = train_test_split(v_data, y, test_size=0.1, random_state=0)
model_g.fit(X_train, y_train)
```

Gambar 4.7. Kode program proses pembentukan *confusion matrix*

Langkah selanjutnya adalah perhitungan akurasi metode *Naive Bayes Classifier* menggunakan *library sklearn GaussianNB*. Kode program proses perhitungan akurasi metode *Naive Bayes Classifier* dapat dilihat pada gambar 4.8 sebagai berikut:

```
v_data = vectorizer.fit_transform(x).toarray()

X_train, X_test, y_train, y_test = train_test_split(v_data, y, test_size=0.1, random_state=0)
model_g.fit(X_train, y_train)

y_preds = model_g.predict(X_test)

print(confusion_matrix(y_test, y_preds))
print(classification_report(y_test, y_preds))
print('nilai akurasinya adalah ', accuracy_score(y_test, y_preds))
```

Gambar 4.8. Kode program proses perhitungan akurasi metode *Naive Bayes Classifier*

4.2.4 Implementasi Penyusunan *Wordcloud*

Proses penyusunan *Wordcloud* menggunakan *library wordcloud* dan untuk menampilkan hasilnya digunakan *library matplotlib.pyplot*. Kode program proses penyusunan *wordcloud* dapat dilihat pada gambar 4.9 sebagai berikut:

```
# Load data dari file Excel
data = pd.read_excel('data/data_analisis2.xlsx')

# Filter data dengan sentimen positif
positive_words = data.loc[data['Labeling'] == 'POSITIF']['Preprocessing'].str.cat(sep=' ')

# Filter data dengan sentimen negatif
negative_words = data.loc[data['Labeling'] == 'NEGATIF']['Preprocessing'].str.cat(sep=' ')

# Membuat object WordCloud untuk sentimen positif
wordcloud_positive = WordCloud(width=800, height=800,
                                background_color='white',
                                stopwords=[],
                                min_font_size=10).generate(positive_words)

# Membuat object WordCloud untuk sentimen negatif
wordcloud_negative = WordCloud(width=800, height=800,
                                background_color='white',
                                stopwords=[],
                                min_font_size=10).generate(negative_words)
```

Gambar 4.9. Kode program proses penyusunan *Wordcloud*

4.3. Hasil Pengujian

4.3.1 Hasil Pengujian Implementasi *Preprocessing*

a. *Case Folding*

Hasil pengujian Implementasi *Preprocessing* tahap *CaseFolding* dapat dilihat pada tabel 4.1 sebagai berikut:

Tabel 4.1. Hasil pengujian tahap *Case Folding*

Tweet	Case Folding
@molangta @nandayogie_ @Mistaaaw @mlbbfess Esportitu udah paling enak	@molangta @nandayogie_ @mistaaaw @mlbbfess esport itu udah paling enak
kakak gw nyoba daftar susah saingan banyak di Jakarta	kakak gw nyoba daftar susah saingan banyak di Jakarta
Kerja main doang, gaji gede anjay	kerja main doang, gaji gede anjay
player mentok ML juga jdStreamer, percuma	player mentok ml juga jd streamer,percuma
Pas pensiun plonga plongo gajelas mo ngapain mending pns	pas pensiun plonga plongo ga jelas mo ngapain mending pns

b. *Tokenizing*

Hasil pengujian Implementasi *Preprocessing* tahap *Tokenizing* dapat dilihat pada tabel 4.2 sebagai berikut:

Tabel 4.2. Hasil pengujian tahap *Tokenizing*

Tweet	Tokenizing
@molangta @nandayogie_ @Mistaaaw @mlbbfess Esport itu udah paling enak	['esport', 'itu', 'udah', 'paling', 'enak']
kakak gw nyoba daftar susah saingan banyak	['kakak', 'gw', 'nyoba', 'daftar', 'susah', 'saingan', 'banyak']
Kerja main doang, gaji gede anjay	['kerja', 'main', 'doang', 'gaji', 'gede', 'anjay']
player mentok ML juga jdstreamer, percuma	['player', 'mentok', 'ml', 'juga', 'jd', 'streamer', 'percuma']
Pas pensiun plonga plongo ga jelas mo ngapain mending pns	['pas', 'pensiun', 'plonga', 'plongo', 'ga', 'jelas', 'mo', 'ngapain', 'mending', 'pns']

c. *Filtering*

Hasil pengujian Implementasi *Preprocessing* tahap *Filtering* dapat dilihat pada tabel 4.3 sebagai berikut:

Tabel 4.3. Hasil pengujian tahap *Filtering*

Tweet	Stopword Removal
@molangta @nandayogie_ @Mistaaaw @mlbbfess Esport itu udah paling enak	['esport', 'udah', 'enak']
kakak gw nyoba daftar susah saingan banyak	['kakak', 'gw', 'nyoba', 'daftar', 'susah', 'saingan']
Kerja main doang, gaji gede anjay	['kerja', 'main', 'doang', 'gaji', 'gede', 'anjay']
player mentok ML juga jd streamer, percuma	['player', 'mentok', 'ml', 'jd', 'streamer']
Pas pensiun plonga plongo ga jelas mo ngapain mending pns	['pas', 'pensiun', 'plonga', 'plongo', 'ga', 'mo', 'ngapain', 'mending', 'pns']

d. *Stemming*

Hasil pengujian Implementasi *Preprocessing* tahap *Stemming* dapat dilihat pada tabel 4.4 sebagai berikut:

Tabel 4.4. Hasil pengujian tahap *Stemming*

Tweet	Stemming
@molangta @nandayogie_ @Mistaaaw @mlbbfess Esport itu udah paling enak	['esport', 'udah', 'enak']
kakak gw nyoba daftar susah saingan banyak	['kakak', 'gw', 'nyoba', 'daftar', 'susah', 'saing']
Kerja main doang, gaji gede anjay	['kerja', 'main', 'doang', 'gaji', 'gede', 'anjay']
player mentok ML juga jd streamer, percuma	['player', 'mentok', 'ml', 'jd', 'streamer']
Pas pensiun plonga plongo ga jelas mo ngapain mending pns	['pas', 'pensiun', 'plonga', 'plongo', 'ga', 'mo', 'ngapain', 'mending', 'pns']

4.3.2 Hasil Implementasi Penyusunan Model

- a. Hasil pengujian implementasi evaluasi hasil proses *pembobotan* dapat dilihat pada tabel 4.5 sebagai berikut:

Tabel 4.5. Hasil pengujian tahap *pembobotan*

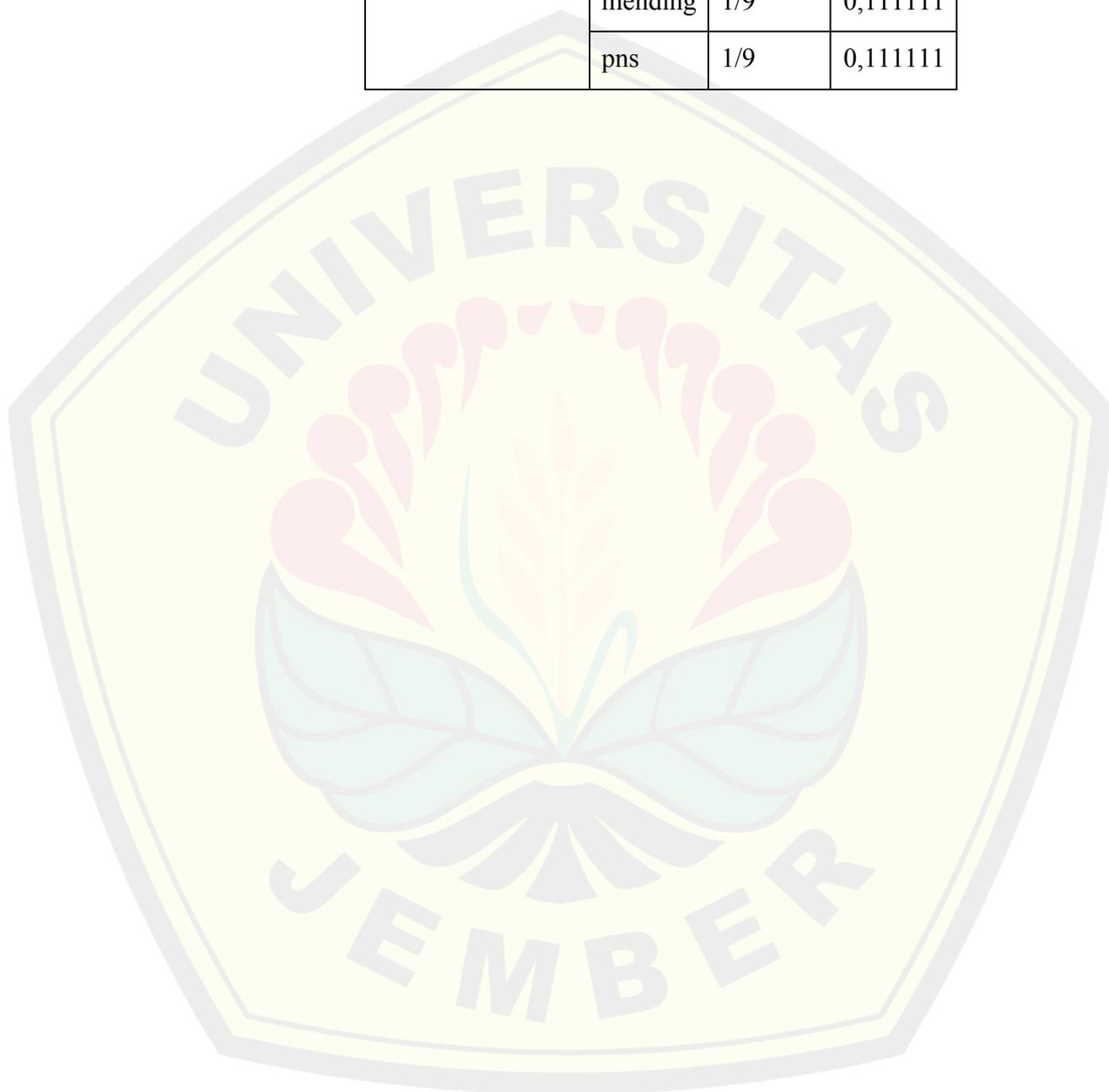
Tweet	Tweet_clean	Sentiment
@molangta @nandayogie_ @Mistaaaw @mlbbfess Esport itu udah paling enak	esport udah enak	1
kakak gw nyoba daftar susah saingan banyak	kakak gw nyoba daftar susah saing	-1
Kerja main doang, gaji gede anjay	kerja main doang gaji gede anjay	1
player mentok ML juga jd streamer, percuma	player mentok ml jd streamer	-1
Pas pensiun plonga plongo ga jelas mo ngapain mending pns	pas pensiun plonga plongo ga mo ngapain mending pns	-1

- b. Hasil pengujian Proses perhitungan TF-IDF menggunakan *Library TfidfVectorizer* dapat dilihat sebagai berikut:

Tabel 4.6. Contoh Hasil Perhitungan TF

Clean	Kata	Nilai TF	TF
['esport', 'udah', 'enak']	esport	1/3	0,333333
	udah	1/3	0,333333
	enak	1/3	0,333333
['kagak', 'gw', 'nyoba', 'daftar', 'susah', 'saing']	kagak	1/6	0,166667
	gw	1/6	0,166667
	nyoba	1/6	0,166667
	daftar	1/6	0,166667
	susah	1/6	0,166667
	saing	1/6	0,166667
['kerja', 'main', 'doang', 'gaji', 'gede', 'anjay']	kerja	1/6	0,166667
	main	1/6	0,166667
	doang	1/6	0,166667
	gaji	1/6	0,166667
	gede	1/6	0,166667
	anjay	1/6	0,166667
['player', 'mentok', 'ml', 'jd', 'streamer']	player	1/5	0,2
	mentok	1/5	0,2
	ml	1/5	0,2
	jd	1/5	0,2
	streamer	1/5	0,2
	pas	1/9	0,111111
	pensiun	1/9	0,111111

['pas', 'pensiun', 'plonga', 'plongo', 'ga', 'mo', 'ngapain', 'mending', 'pns']	plonga	1/9	0,111111
	plongo	1/9	0,111111
	ga	1/9	0,111111
	mo	1/9	0,111111
	ngapain	1/9	0,111111
	mending	1/9	0,111111
	pns	1/9	0,111111



Tabel 4.7. Contoh Hasil Perhitungan IDF

Clean	Kata	DF	$idf_t = \log_{10} \left(\frac{N}{df(t)} \right)$
['esport', 'udah', 'enak']	esport	1879	0,06577967719990968
	udah	129	2,0137537894901523
	enak	6	4,74493212836325
['kakak', 'gw', 'nyoba', 'daftar', 'susah', 'saing']	kakak	3	5,745131728261518
	gw	57	1,519638692232834
	nyoba	5	4,874922528678766
	daftar	23	2,650586873348685
	susah	14	3,1107253803250666
	saing	7	4,422647296720595
['kerja', 'main', 'doang', 'gaji', 'gede', 'anjay']	kerja	25	2,5238089736286145
	main	161	0,5474360588323952
	doang	34	1,8858399685725424
	gaji	14	3,1107253803250666
	gede	52	1,6862442060925418
	anjay	4	5,031188554656343
['player', 'mentok', 'ml', 'jd', 'streamer']	player	110	0,8694418154331266
	mentok	6	4,74493212836325
	ml	35	1,8272432537322436
	jd	47	1,560678321879297
	streamer	88	0,9501500452438576
['pas', 'pensiun', 'plonga', 'plongo', 'ga', 'mo']	pas	30	2,288721415184052
	pensiun	5	4,874922528678766
	plonga	2	6,291224932502474
	plongo	3	5,745131728261518
	ga	210	0,4461801227066382
	mo	7	4,422647296720595

'ngapain', 'mending', 'pns']	ngapain	6	4,74493212836325
	mending	40	1,7556942369969035
	pns	5	4,874922528678766

Tabel 4.8. Contoh Hasil Perhitungan TF-IDF

Kata	$W_{t,d} = W_{tf_{t,d}} \times Idf_t$
esport	0,02192655906663656
udah	0,6712512631633841
enak	1,58164404278775
kakak	0,9575219547102528
gw	0,253273115372139
nyoba	0,8124870881131277
daftar	0,44176447889144746
susah	0,5184542300541778
saing	0,7371078827867659
kerja	0,4206348289381025
main	0,09123934313873255
doang	0,3143066614287571
gaji	0,5184542300541778
gede	0,2810407010154236
anjay	0,8385314257760572
player	0,1738883630866253
mentok	0,94898642567265
ml	0,3654486507464487
jd	0,3121356643758594
streamer	0,19003000904877152
pas	0,25430237946589574

pensiun	0,5416580598531963
plonga	0,6990249925002749
plongo	0,6383479698068353
ga	0,04957556918962646
mo	0,4914052551911772
ngapain	0,5272157920403611
mending	0,19507713633298927
pns	0,5416580598531963

- c. Hasil perhitungan *prior probability* dapat dilihat pada tabel 4.9 sebagai berikut:

Tabel 4.9. Hasil perhitungan *prior probability*

Sentimen	$P(V_j) = \frac{D_j}{D}$
Positif	0,716328567
Negatif	0,283671433

- d. Contoh hasil perhitungan *conditional probability* dapat dilihat pada tabel 4.10 sebagai berikut:

Tabel 4.10. Hasil perhitungan *conditional probability*

Sentimen	Kata	$P(X_1 positif) = \frac{W_{ij} + 1}{N + N_j}$	$P(X_1 negatif) = \frac{W_{ij} + 1}{N + N_j}$
Positif	esport	0,733901011	0,266098989
	udah	0,80620155	0,19379845
	enak	0,166666667	0,833333333
	kakak	0	1
	gw	0,859649123	0,140350877

Negatif	nyoba	1	0
	daftar	0,913043478	0,086956522
	susah	0,285714286	0,714285714
	saing	0,428571429	0,571428571
Positif	kerja	1	0
	main	0,925465839	0,074534161
	doang	0,117647059	0,882352941
	gaji	0,714285714	0,285714286
	gede	0,519230769	0,480769231
	anjay	1	0
Negatif	player	0,9	0,1
	mentok	0	1
	ml	0,857142857	0,142857143

	jd	0,14893617	0,85106383
	streamer	0,409090909	0,590909091
Negatif	pas	0	1
	pensiun	0	1
	plonga	0	1
	plongo	0	1
	ga	0,847619048	0,152380952
	mo	0,428571429	0,571428571
	ngapain	0	1
	mending	0,4	0,6
	pns	0	1

- e. Pada tahap ini *Naive Bayes Classifier* menentukan nilai probabilitas tertinggi dari kedua kelas target. Contoh hasil perhitungan *conditional probability* data testing dapat dilihat pada tabel 4.11 sebagai berikut:

Tabel 4.11. Contoh hasil perhitungan *conditional probability* data testing

Sentimen Prediksi	Dokumen	Positif	Negatif	Actual Sentimen
positif	@molangta @nandayogie_ @Mistaaaw @mlbbfess Esport itu udah paling enak	1,706769228	1,293230772	positif
positif	kakak gw nyoba daftar susah saingan banyak	3,486978315	2,513021685	negatif
positif	Kerja main doang, gaji gede anjay	4,276629381	1,723370619	positif
negatif	player mentok ML juga jd streamer, percuma	2,315169936	2,684830064	negatif
negatif	Pas pensiun plonga plonga ga jelas mo ngapain mending pns	1,676190476	7,323809524	negatif

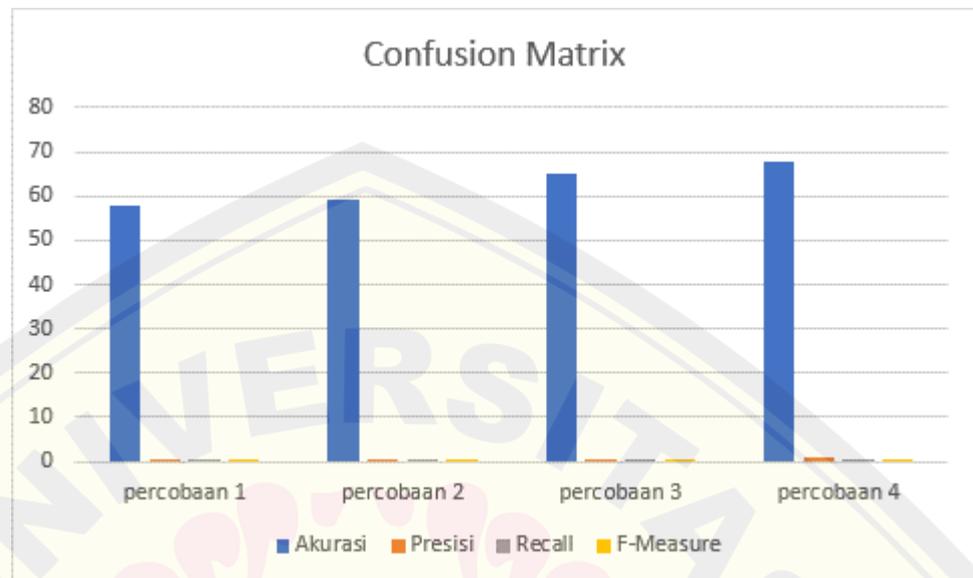
4.3.3 Hasil Uji Evaluasi

Proses uji evaluasi menggunakan *confusion matrix* guna mengetahui tingkat akurasi proses klasifikasi yang sudah dikerjakan oleh sistem dengan acuan pada nilai akurasi. Pada penelitian ini, penulis melakukan empat kali percobaan dengan perbandingan antara *data training* dengan *data testing* sebesar 60%:40%, 70%:30%, 80%:20%, dan 90%:10%.

Pada implementasinya, penulis menggunakan perhitungan akurasi, presisi, *recall* dan *f-measure*. Hasil uji evaluasi dapat dilihat pada tabel 4.12 dan gambar 4.10 sebagai berikut:

Tabel 4.12. Hasil Uji Evaluasi

Keterangan	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4
Data Training	60%	70%	80%	90%
Data Testing	40%	30%	20%	10%
True Positif (Tp)	227	172	127	61
True Negatif (Tn)	240	185	133	75
False Positif (Fp)	114	82	35	17
False Negatif (Fn)	219	161	105	47
Akurasi	58	59	65	68
Presisi	0,61	0,62	0,69	0,72
Recall	0,58	0,59	0,65	0,68
F-Measure	0,58	0,60	0,65	0,68



Gambar 4.10. Grafik Hasil Uji Evaluasi

Dari keempat percobaan yang dilakukan pada proses uji evaluasi menggunakan *confusion matrix* didapatkan hasil terbaik pada percobaan 4 dengan *data training* yang dipakai sebesar 90% , *data testing* sebesar 10% dan hasil akurasi sebesar 68%.

4.3.4 Hasil Uji *Wordcloud*

Hasil uji *wordcloud* dibedakan menjadi 2 yaitu hasil pada sentimen positif dan sentimen negatif. Kata yang sering muncul pada hasil uji *wordcloud* ditandai oleh besar kecilnya ukuran huruf pada gambar, semakin besar gambar pada suatu kata maka kata tersebut sering muncul pada dokumen yang telah diuji. Pada setiap hasil juga dimunculkan grafik kemunculan setiap kata, hasilnya dapat dilihat sebagai berikut :

BAB 5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat diberikan oleh penulis dan dapat diambil dari penelitian ini adalah sebagai berikut:

1. Analisis Sentimen Terhadap Karir E-Sports Di Indonesia Pada Media Twitter Menggunakan Metode Naive Bayes menggunakan Bahasa pemrograman python dapat memproses tahapan preprocessing dan mengimplementasikan metode Naive Bayes Classifier dengan baik.
2. Metode *Naive Bayes Classifier* pada Analisis Sentimen Terhadap Karir E-Sports Di Indonesia Pada Media Twitter Menggunakan Metode Naive Bayes memberikan akurasi sebesar 58% dengan skenario 60:40 dari 2000 data *tweet*, 59% dengan skenario 70:30 dari 2000 data *tweet*, 65% dengan skenario 80:20 dari 2000 data *tweet*, 68% dengan skenario 90:10 dari e data *tweet*. Hasil perhitungan *performance* dengan skenario 60:40 memberikan hasil : nilai presisi 0.61, nilai recall 0.58, dan F1-score 0.58, skenario 70:30 memberikan hasil : nilai presisi 0.62, nilai recall 0.59, dan F1-score 0.60, skenario 80:20 memberikan hasil : nilai presisi 0.69, nilai recall 0.65, dan F1-score 0.65, skenario 90:10 memberikan hasil : nilai presisi 0.72, nilai recall 0.68, dan F1-score 0.68.
3. Hasil visualisasi *Wordcloud* pada sentimen karir *e-sports* di Indonesia memberikan informasi frekuensi kemunculan setiap kata pada sentimen positif dan negatif yang berada di seluruh dokumen.

5.2. Saran

Saran dari penulis mengenai hasil penelitian yang sudah dilaksanakan dan bisa dipakai pada pengembangan penelitian yang akan mendatang adalah sebagai berikut

1. Pada penelitian selanjutnya dapat digunakan lebih banyak rasio uji dalam proses pengujian.
2. Pada penelitian selanjutnya dapat digunakan metode lain sehingga dapat melakukan perbandingan hasil dengan beberapa metode
3. Pada penelitian selanjutnya dapat menggunakan media sosial selain *twitter* seperti *Youtube*, *Tiktok*, *Instagram*, *Facebook*, dan juga sosial media lainnya agar bisa mendapatkan jangkauan *dataset* yang lebih banyak.

DAFTAR PUSTAKA

- Beineke, P., Hastie, T., & Vaithyanathan, S. (2004). The Sentimental Factor: Improving Review Classification Via Human-Provided Information. *Proceedings Of The Annual Meeting Of The Association For Computational Linguistics, 2002*, 263–270.
- Falahah, & Nur, D. D. A. (2015). Pengembangan Aplikasi Sentiment Analysis Menggunakan Metode Naïve Bayes (Studi Kasus Sentiment Analysis Dari Media Twitter). *Seminar Nasional Sistem Informasi Indonesia, November*, 335–339.
- Feldman, R., & Sanger, J. (2007). The Text Mining Handbook: Advanced Approaches In Analyzing Unstructured Data. *Critical Sociology, 37*(4), 493–497.
- Haaff, M. De. (2010). *Sentiment Analysis, Hard But Worth It!, Customerthink*.
- Hadiyani, E. P. (2013). *Rancang Bangun Sistem Pendukung Keputusan Untuk Pemilihan Anggota Terbaik Aiesec Surabaya Dengan Menggunakan Metode Naive Bayes*. Universitas Airlangga.
- Hotho, A., Nürnberger, A., & Paaß, G. (2005). A Brief Survey Of Text Mining. *Journal For Language Technology And Computational Linguistics, 20*(1), 19–62.
- Ling, J., N. Kencana, I. P. E., & Oka, T. B. (2014). Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square. *E-Jurnal Matematika, 3*(3), 92.
- Luhulima, Y. (2013). *Sentiment Analysis Pada Review Barang Berbahasa Indonesia Dengan Metode K-Nearest Neighbor (K-NN)*. Universitas Brawijaya.
- Manalu, B. U. (2014). Analisis Sentimen Pada Twitter Menggunakan Text Mining Skripsi Boy Utomo Manalu. *Teknologi Informasi Fak. Ilkom Unsut*, 1–111.
- Newzoo.Com. (2019). <https://Newzoo.Com/Insights/Trend-Reports/Newzoo->

Global-Games-Market-Report-2019-Light-Version/

Powers, D. M. W. (2020). *Evaluation: From Precision, Recall And F-Measure To ROC, Informedness, Markedness And Correlation*. May.

Ridwan, M., Suyono, H., & Sarosa, M. (2013). Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *Eeccis*, 7(1), 59–64.

Simorangkir, H., & Lhaksana, K. M. (2018). Analisis Sentimen Pada Twitter Untuk Games Online Mobile Legends Dan Arena Of Valor Dengan Metode Naïve Bayes Classifier. *E-Proceeding Of Englineering*, 5(3), 8131–8140.

Wayan, N., & Saraswati, S. (2013). *Naïve Bayes Classifier Dan Support Vector Machines Untuk Sentiment Analysis*. 2–4.

Wicaksono, A. J. (2016). Prediksi Pemilihan Presiden Amerika Serikat 2016 Dengan Menganalisis Sentimen Pada Sosial Media. *Jurnal Buana Informatika*, 1–56.

LAMPIRAN

Lampiran 1. [Hasil Crawling Data](#)

Lampiran 2. [Hasil Koding](#)

