INFOTECH journal
ISSN : 2460-1861 (Print), 2615-4250 (Online)
Vol. 9 No. 2, pp. 383-392

KLASIFIKASI ULASAN PENGGUNA APLIKASI DISCORD MENGGUNAKAN METODE *INFORMATION GAIN* DAN *NAÏVE BAYES CLASSIFIER*

Syafrida Nabila Salsabila[1], Betha Nurina Sari[2], Rini Mayasari[3]
[123]Informatika, Ilmu Komputer, Universitas Singaperbangsa Karawang
Email: 1910631170235@student.unsika.ac.id[1], betha.nurina@staff.unsika.ac.id[2], rini.mayasari@staff.unsika.ac.id[3]

ABSTRACT
At the time of the Covid-19 outbreak, many people downloaded the Discord application because it could be used for online learning from home. Many reviews are given by users of the Discord application on the Google Play Store. Reviews from users can contain information for application development, therefore a discord application user review classification is carried out. This study used the 6CRISP-DM methodology with the initial stage of web scraping, then the data was labeled positive and negative using the InSet Lexicon and after that it was validated by a linguist. After being validated, the data will be cleaned and go through the preprocessing stage. Then the data will go through the feature selection stage using Information Gain and proceed with the Naïve Bayes Classifier algorithm. Classification results using the Naive Bayes Classifier algorithm use Information Gain compared to those that do not use Information Gain. After comparison, the highest accuracy values were obtained, namely 84%, 84% precision, 84% recall, 83% f1-score/f-measure obtained with 90% training data and 10% testing data using Information Gain.
Keywords: CRISP-DM, Discord, Information Gain, InSet Lexicon, *Naïve Bayes Classifier*

ABSTRAK
Pada saat terjadinya covid-19, aplikasi *discord* banyak di *download* oleh masyarakat karena dapat digunakan untuk pembelajaran yang dilakukan secara daring dari rumah. Banyak ulasan yang diberikan oleh pengguna aplikasi *discord* di *google play store*. Ulasan dari pengguna dapat berisi informasi bagi pengembangan aplikasi, oleh karena itu dilakukan klasifikasi ulasan pengguna aplikasi *discord*. Penelitian ini menggunakan metodologi CRISP-DM dengan tahap awal dilakukan *scraping web*, lalu data diberi label positif dan negatif menggunakan *InSet Lexicon* dan setelah itu divalidasi oleh ahli bahasa. Setelah divalidasi, data akan dibersihkan dan melalui tahap *preprocessing*. Lalu data akan melalui tahap seleksi fitur menggunakan *Information Gain* dan dilanjutkan dengan algoritma *Naïve Bayes Classifier*. Hasil klasifikasi dengan algoritma *Naive Bayes Classifier* menggunakan *Information Gain* dibandingkan dengan yang tidak menggunakan *Information Gain.* Setelah dibandingkan didapatlah nilai *accuracy* tertinggi yaitu 84%, *precision* 84%, *recall* 84%, *f1-score/f-measure* 83% yang didapatkan dengan 90% *data training* dan 10% *data testing* menggunakan *Information Gain.*

Kata Kunci: CRISP-DM, *Discord*, *Information Gain*, *InSet Lexicon*, *Naïve Bayes Classifier*

Riwayat Artikel :

Tanggal diterima : 26-07-2023

Tanggal revisi : 29-07-2023

Tanggal terbit : 30-07-2023

DOI :

https://doi.org/10.31949/infotech.v9i2.6277

1. PENDAHULUAN

1.1. Latar Belakang

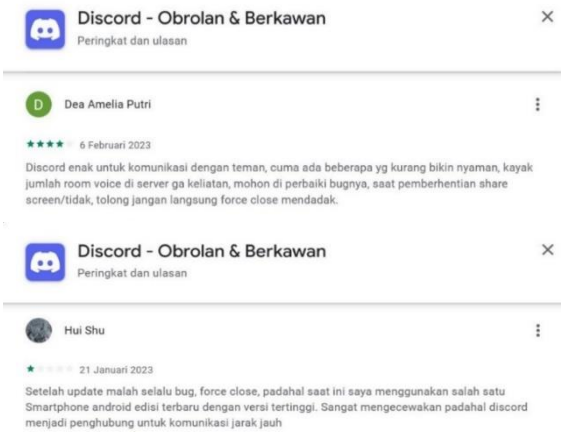
Pandemi COVID-19 mengakibatkan seluruh institusi pendidikan kesulitan untuk beroperasi (Tjahjadi, Paramita, & Salman, 2021). Hal tersebut menyebabkan semua kegiatan di seluruh institusi pendidikan harus dilakukan secara daring dari rumah (Rohanah, Dermawan, & Purnamasari, 2021). Pembelajaran daring adalah suatu sistem pembelajaran yang tidak dilakukan secara tatap muka tetapi menggunakan media yang juga mendukung pembelajaran jarak jauh (Panggabean, 2021). Tujuan pemilihan media yang tepat untuk pembelajaran daring adalah untuk mencapai hasil yang baik yang menyesuaikan dengan kondisi dan kebutuhan yang sudah terpenuhi (Ridho, Muhaimin, & Harjono, 2021). Salah satu aplikasi yang digunakan masyarakat untuk belajar daring adalah aplikasi *discord* (Rakhmawan et al., 2020).

Discord adalah suatu aplikasi gratis untuk media komunikasi yang mirip dengan aplikasi *Slack* atau *Skype* dan memungkinkan pengguna aplikasi tersebut dapat berkomunikasi secara *real time* melalui teks, suara atau video (Ridho et al., 2021). Discord bisa digunakan untuk berbagai sistem yaitu Windows, IOS, Android, Linux, dan Mac (Rakhmawan et al., 2020). Pada awalnya, *discord* didesain untuk komunitas pemain *game online* agar mempermudah berkomunikasi antara sesama pemain di dalam satu tim atau satu komunitas (Ridho et al., 2021). Tidak hanya untuk pemain *game online*, kenyataannya berbagai komunitas dari programming, bisnis, dan belajar daring dapat dilakukan menggunakan aplikasi ini (Tjahjadi et al., 2021).

Keunggulan pada aplikasi *discord* adalah pengguna aplikasi bisa berkomunikasi dengan kualitas suara yang bersih dan dapat membuat kelas atau *channel* sendiri sehingga dapat membuat grup untuk menjalin komunikasi (Minarni, Prasetyaningrum, & Ismail, 2022). Fungsi ini memungkinkan untuk pengajar dalam menyampaikan materi secara leluasa pada para peserta didik (Ridho et al., 2021). Berdasarkan data yang ada pada Gambar 1, aplikasi *discord* adalah aplikasi yang aktif digunakan oleh pengguna sebagai wadah untuk saling bertukar informasi. Aplikasi *discord* sudah memiliki lebih dari 100 juta pengguna yang aktif bulanan, 13,5 juta server yang aktif mingguan, dan 4 miliar menit obrolan yang ada di dalam server setiap hari. Selain itu, angka statistik menunjukkan bahwa pengguna aplikasi *discord* telah mencapai lebih dari 300 juta pengguna pada Juni 2020, dengan jumlah yang meningkat pesat dibandingkan Mei 2017 yang saat itu berjumlah 45 juta pengguna (Tumewu & Kurniasari, 2022).

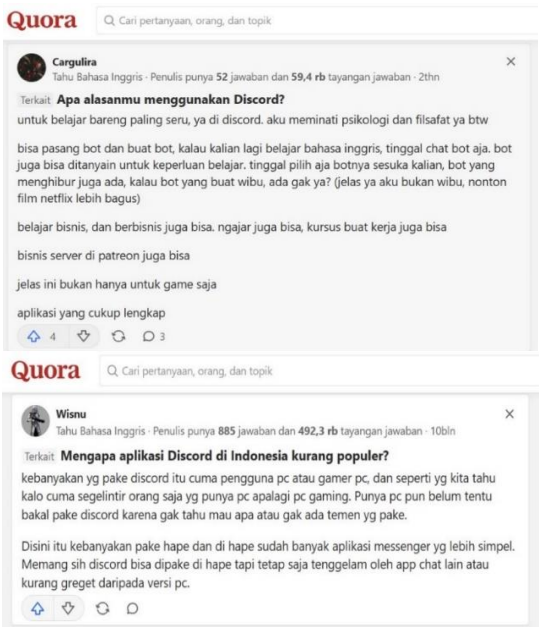
Gambar 1. Data pengguna aplikasi *discord*

Keunggulan dari aplikasi *discord* tidak lepas dari beberapa kekurangan saat menggunakan aplikasi tersebut. Beberapa kekurangannya yaitu pada fitur *voice call* yang mudah terpengaruh saat kondisi jaringan internet melemah. Selain itu, fasilitas *share screen* / berbagi layar pun menjadi sangat terkendala saat jaringan internet melemah (Rakhmawan et al., 2020). Kelebihan dan kekurangan dari aplikasi tersebut menimbulkan berbagai ulasan dari pengguna aplikasi tersebut (Erfina, Basryah, Saepulrohman, & Lestari, 2020). Ulasan dari pengguna aplikasi tersebut banyak dikeluhkan di *google play store*. Gambar 2 merupakan contoh ulasan aplikasi *discord* pada *google play store*.



Gambar 2 Ulasan pengguna pada aplikasi *discord* di *google play store*

Keluhan dari pengguna ini berisi informasi yang dapat membantu pengembang untuk mengembangkan aplikasi tersebut agar lebih baik dari sebelumnya. Untuk memudahkan pengolahan informasi ulasan, maka diperlukan klasifikasi terhadap ulasan pada aplikasi yang tersedia pada *google play store* (Subagja, Widiastiwi, & Chamidah, 2021). Analisis sentimen adalah cara mengumpulkan berbagai macam pendapat seseorang tentang topik tertentu menggunakan sosial media dengan tujuan memperoleh informasi sentimen ke arah positif atau negatif. Di dalam analisis sentimen terdapat teks-teks yang tidak terstruktur (Putra & Juanita, 2021). Analisis sentimen dapat mengubah informasi yang ada pada sosial media yang tidak terstruktur menjadi data yang terstruktur (Thaha & Aziz, 2020). Oleh karena itu, analisis sentimen bermanfaat agar mengetahui tingkat kepuasaan masyarakat terhadap aplikasi tersebut (Pintoko & L, 2018). Berdasarkan pada Gambar 3 merupakan pendapat masyarakat melalui *website* quora.com tentang adanya aplikasi *discord* salah satunya yaitu karena di *handphone* sudah banyak aplikasi *messenger* yang lebih *simple*, tetapi komentar lainnya mengatakan bahwa adanya aplikasi *discord* bisa berguna untuk berbagai macam kegiatan.



Gambar 3. Komentar masyarakat tentang aplikasi *discord*

Pada penelitian sebelumnya yang dilakukan oleh Surohman, Aji, Rousyati, & Wati (2020) mengenai perbandingan algoritma *Naïve Bayes Classifier* dan *K-Nearest Neighbor* (KNN) untuk analisis terhadap *review fintech* pada aplikasi dana. Pada penelitian ini dibahas tentang kinerja dari *Naïve Bayes Classifier* dan *K-Nearest Neighbor* (KNN) yang menjadi seleksi fitur untuk mengolah data ulasan. Pengujian pertama menggunakan algoritma *naïve bayes* dan menghasilkan nilai akurasi sebesar 84,76% +/- 3,93% dengan rata-rata mikro 84,85%. Lalu selanjutnya pengujian dengan menggunakan algoritma KNN yang memberikan akurasi sebesar 82,92% +/-4,87% dengan rata-rata mikro 82,96%. Berdasarkan skor akurasi uji analisis sentimen review pada aplikasi dana, lebih baik menggunakan algoritma *naïve bayes* daripada menggunakan KNN.

Lu & Liang (2017) juga melakukan penelitian tentang klasifikasi ulasan pengguna menggunakan algoritma *naïve bayes*, *mesin bagging*, dan J48. Algoritma *naïve bayes* menghasilkan evaluasi terbaik dengan nilai *F-measure* 0,720. Meskipun algoritme *naïve bayes* merupakan algoritme terbaik dalam penelitian ini, algoritme *naïve bayes* tidak berpengaruh terhadap atribut pada data sehingga dapat mengakibatkan besarnya dimensi fitur saat proses klasifikasi. Berdasarkan penelitian dari Sari, Widowati, & Lhaksmana (2019) untuk menangani tingginya dimensi pada data saat menerapkan algoritma *naïve bayes*, diterapkan pemilihan fitur yang berpengaruh untuk setiap label kelas dengan menggunakan seleksi fitur yaitu metode *information gain*. Meskipun demikian, dalam praktiknya algoritma *naïve bayes* bekerja cukup baik dan memiliki akurasi dan kecepatan yang tinggi saat dimasukkan ke dalam *database* (Subagja et al., 2021).

Berdasarkan latar belakang penelitian yang telah dijelaskan sebelumnya, maka penelitian ini akan menggunakan *Naïve Bayes Classifier* dan *Information Gain* dalam klasifikasi ulasan pengguna aplikasi *discord* dengan menerapkan standar perangkat lunak yaitu ISO/IEC 9126 dengan judul "Klasifikasi Ulasan Pengguna Aplikasi Discord Menggunakan Metode *Information Gain* dan *Naïve Bayes Classifier*".

1.2. Rumusan Masalah

Berdasarkan permasalahan yang mendasari, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara melakukan klasifikasi ulasan pengguna aplikasi *Discord* pada *Google Play Store*?
2. Bagaimana mengetahui performa yang dihasilkan dari algoritma *Naïve Bayes Classifier* dalam melakukan klasifikasi ulasan pengguna aplikasi *Discord* pada *Google Play Store*?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan di atas, tujuan dilaksanakannya penelitian ini yaitu sebagai berikut:

1. Menerapkan seleksi fitur *Information Gain* dan algoritma *Naïve Bayes Classifier* dalam melakukan klasifikasi ulasan pengguna aplikasi *Discord* pada *Google Play Store*.
2. Mengukur performa metode Naïve Bayes Classifier dalam melakukan klasifikasi ulasan pengguna aplikasi *Discord* pada *Google Play Store*., dengan menggunakan *Confusion Matrix*.

1.4. Metodologi Penelitian

Metodologi yang akan digunakan pada penelitian ini yaitu *Cross Industry Standard Process for Data Mining* (CRISP-DM) dengan tahapan sebagai berikut:

1. *Business Understanding*
2. *Data Understanding*
3. *Data Preparation*
4. *Modeling*
5. *Evaluation*
6. *Deployment*

2. TINJAUAN PUSTAKA

2.1 Analisis Sentimen

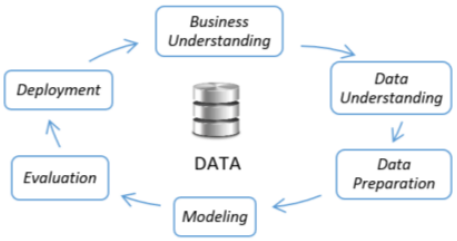
Analisis sentimen merupakan bidang ilmu yang mempelajari bagaimana opini, keluhan, pandangan, dan penilaian lainnya dianalisis. Analisis sentimen digunakan untuk menilai suka ataupun tidak suka pada suatu barang dan jasa (Indriati & Ridok, 2016). Tujuan dari analisis sentimen tidak hanya untuk mendapatkan sentimen, tetapi juga untuk melihat opini tentang topik atau objek dalam teks apakah memiliki sentimen positif atau negatif (Watrianthos, Suryadi, Irmayani, Nasution, & Simanjorang, 2019).

2.2 *Data Mining*

Data mining berkembang pesat dan terkait erat dengan perkembangan teknologi yang memungkinkan pengumpulan data dalam jumlah besar (Marlina & Bakri, 2021). Data itu berjumlah besar dan bertambah setiap hari. Analisis data merupakan kebutuhan yang penting, sehingga peran *data mining* dapat memenuhi kebutuhan dalam menyediakan alat dan menemukan suatu pengetahuan dari data. Data mining adalah gabungan dari beberapa cabang ilmu komputer, yaitu proses pencarian pola dalam dataset yang besar (Subagja et al., 2021).

2.3 CRISP-DM

CRISP-DM merupakan metode yang memberikan standar untuk *data mining* dan dapat diterapkan pada strategi pemecahan masalah umum dalam bisnis atau penelitian (Putra & Juanita, 2021). CRISP-DM merupakan hasil dari penggabungan kerjasama mereka yang dapat diterapkan pada jenis data tertentu. Tahapan CRISP-DM adalah sebagai berikut:



Gambar 4. Tahapan CRISP-DM

2.4 Seleksi Fitur *Information Gain*

Seleksi fitur adalah cara untuk mengurangi fitur dengan memilih kata-kata yang informatif. *Information gain* adalah teknik pemilihan fitur yang mengukur kata dengan menghitung jumlah informasi dengan mengamati kemunculan kata dalam sebuah dokumen (Sari, Widowati, & Lhaksmana, 2019). Dalam penelitian (Sari et al., 2019) disebutkan bahwa perolehan informasi digunakan dalam fase fungsi pemilihan untuk mengurangi fitur yang tidak relevan. Rumus untuk menentukan *entropy* adalah sebagai berikut:

$$Info(D) = -\sum_{i=1}^{c} P(i) log_2 P(i) \quad (1)$$

Keterangan:

c = Jumlah nilai pada atribut klasifikasi

P(i) = Proporsi *sample* pada kelas i

Lalu, mencari nilai *entropy* setelah pembobotan dalam setiap fiturnya yaitu dengan rumus:

$$Info_{\mathrm{A}}(D) = -\sum_{j=1}^{c} \frac{|Dj|}{|D|} InfoD_{\mathrm{j}} \quad (2)$$

Keterangan:

A = Atribut

|Dj| = Jumlah *sample* untuk nilai partisi j

|D| = Jumlah seluruh *sample* data

$InfoD_{\mathrm{j}}$ = *Entropy* untuk setiap partisi j

Nilai *information gain* diperoleh dengan cara mengurangkan nilai dari persamaan (1) dengan nilai persamaan (2), seperti rumus berikut:

$$InfoGain(A) = Info(D) - Info_{\mathrm{A}}(D) \quad (3)$$

2.5 *Naïve Bayes Classifier*

Naïve Bayes Classifier adalah algoritma klasifikasi data mining yang dapat digunakan untuk memprediksi probabilitas kelas dan statistik keanggotaan. Ciri-ciri algoritma *naïve bayes* adalah independensi yang sangat kuat (*naïve*) yang sangat kuat dari setiap kondisi atau peristiwa (Permana, Widiastuti, & Saepudin, 2023). *Naïve Bayes Classifier* juga merupakan metode klasifikasi sederhana dan efektif yang menerapkan teorema Bayes dengan akurasi dan kecepatan tinggi saat diimplementasikan pada *database* dengan kumpulan data besar (Subagja et al., 2021). *Naïve Bayes* memiliki beberapa kekurangan yaitu sangat sensitif terhadap pemilihan fitur. Terlalu banyak fitur dapat meningkatkan waktu komputasi tetapi juga mengurangi akurasi klasifikasi (Negara, Muhardi, & Putri, 2020). Dalam penelitian (Sari et al., 2019) menyebutkan bahwa perhitungan *Naïve Bayes* dirumuskan dengan menggunakan persamaan sebagai berikut:

$$\boldsymbol{C}_{map} = \frac{argmax}{c \in C} P(c|d) = \frac{argmax}{c \in C} P(c)\, \pi_i^n = {}_1P(wi|c) \quad (4)$$

Keterangan:

$P(c|d)$ = *Posterior probability* kelas d terhadap kelas c

$P(c)$ = *Prior probability* kelas c

$P(wi|c)$ = *Conditional probability* kemunculan kata pada kelas c

$$P(c) = \frac{Nc}{N} \quad (5)$$

Keterangan:

Nc = Jumlah dokumen kelas c

N = Jumlah dokumen data latih

Untuk perhitungan *conditional probability* dilakukan dengan menggunakan persamaan sebagai berikut:

$$P(w, c) = \frac{count\ (w,c)+1}{count\ (c)+|v|} \quad (6)$$

Keterangan:

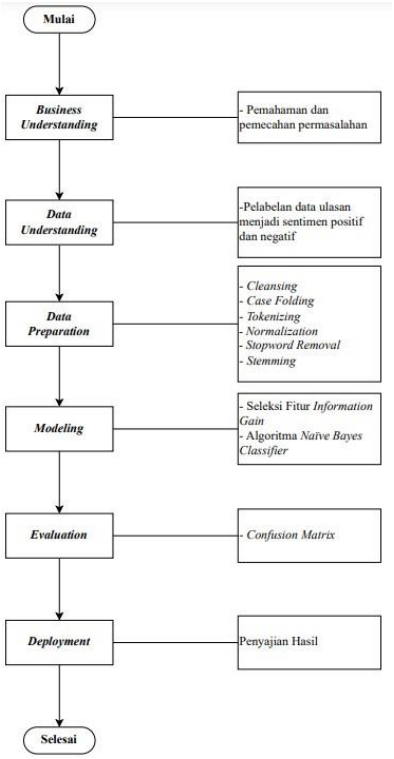
count (w, c) = Jumlah kata yang dikenali di kelas c

count (c) = Jumlah seluruh kata yang ada di kelas c

|v| = Kata unik yang terdapat pada data latih.

3. PEMBAHASAN

Penelitian ini menggunakan CRISP-DM untuk analisis dan memproses data. Diagram rancangan penelitian pada Gambar 5.



Gambar 5. Rancangan Penelitian

3.1 *Business Understanding*

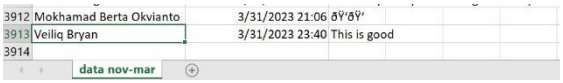
Pada tahap ini dilakukan pemecahan permasalahan pada penelitian. Adapun permasalahan pada aplikasi *discord* adalah banyaknya pengguna aplikasi tersebut yang memberikan ulasan di *google play store*. Karena banyaknya ulasan pada aplikasi *discord*, maka untuk memudahkan dalam mendapatkan informasi akan dilakukan klasifikasi ulasan pengguna. Keluhan pengguna pada aplikasi *discord* ini dianalisis dan dilakukan penambangan data dengan menggunakan teknik *scraping web* yang nantinya akan diklasifikasi berdasarkan sentimen positif dan negatif. Algoritma klasifikasi yang digunakan pada penelitian ini adalah *Naïve Bayes Classifier*.

3.2 *Data Understanding*

Data Understanding adalah tahapan untuk mengumpulkan data awal dan mempelajari data yang telah diambil. Adapun proses dari pemahaman data tersebut dilakukan agar mengetahui struktur data penanganan terhadap data. Adapun tahapan pengumpulan data yang dilakukan diantaranya adalah sebagai berikut:

3.2.1 *Collect Data*

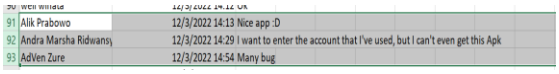
Setelah membuka web *google colaboratory*, memasukkan kode yang diperlukan untuk web scraping. Lalu, membuka halaman situs web *google play store* dan menuju pada halaman aplikasi *discord* untuk mengambil *link discord* dan dimasukkan ke dalam kode perintah yang telah dibuat sebelumnya. Jumlah data yang diambil pada proses *web scraping* ini adalah sebanyak 5000 data ulasan dari ulasan yang terbaru. Karena pada penelitian ini data yang akan diambil yaitu dari November 2022 sampai Maret 2023, maka ada beberapa ulasan yang dihapus dan tersisa sebanyak 3912 ulasan yang ditunjukkan pada Gambar 6.



Gambar 6. Hasil *Scraping Web*

3.2.2 *Data Selection*

Penyeleksian data dilakukan dengan menghapus data yang tidak diperlukan. Data awal yang diperoleh yaitu berjumlah 5000 data ulasan, data yang diperlukan yaitu dari November 2022 sampai Maret 2023. Oleh karena itu, 1088 data ulasan pengguna discord dihapus dan tersisa sebanyak 3.912 data ulasan. 3912 data ulasan tersebut masih memiliki ulasan selain bahasa Indonesia, karena batasan masalah pada penelitian ini adalah hanya mengambil ulasan berbahasa Indonesia, maka ulasan selain bahasa Indonesia harus dihapus dan tersisa sebanyak 2766 ulasan. Gambar 7 merupakan ulasan selain bahasa Indonesia yang harus dihapus.



Gambar 7. Contoh ulasan selain bahasa Indonesia

3.2.3 Pelabelan Data

Dalam mengolah data ulasan yang akan diproses untuk pengklasifikasian, data tersebut akan ditentukan atribut baru yaitu label. Pada data ulasan ini akan diberikan label sentimen positif dan negatif berdasarkan nilai skor akhir yang dilakukan secara otomatis menggunakan bantuan *InSet Lexicon*. Jika suatu ulasan pada kolom sentimen memiliki skor akhir bernilai >0, maka label yang diberikan adalah positif begitu pula sebaliknya jika suatu ulasan pada kolom sentimen memiliki skor akhir <0, maka label yang diberikan adalah negatif. Dan jika ulasan pada kolom sentimen memiliki skor akhir 0 akan dihapus, karena merujuk ke batasan masalah pada penelitian ini yaitu hanya memakai sentimen positif dan negatif. Gambar 8 merupakan hasil dari pelabelan menggunakan *InSet Lexicon*.

	Ulasan Pengguna	Skor Sentiment	Sentiment
0	Sangat asik dn membantu	2	positif
1	Di update malah semakin buruk, lebih bagus ver...	2	positif
2	Makin update makin aneh masa cuma bisa tag beb...	1	positif
3	Setelah up discord jadi semakin lag gak seperti...	1	positif
4	Saya tidak bisa masuk ke akun yang saya buat d...	-5	negatif
...	...	...	...
2755	Mayan lah ya	1	positif
2758	Saya lagi masukin sebuah kode tapi di coba kat...	-4	negatif
2759	Love bangetttttttt ga bisa berkata kata	-1	negatif
2760	Banyak bug ganggu banget semoga cepat di benar...	1	positif
2762	Hehe....sorry waktu itu lagi emosi	-1	negatif

1584 rows × 3 columns

Gambar 8. Hasil dari pelabelan

Hasil dari pelabelan menggunakan *InSet Lexcion* kemudian disimpan dalam bentuk file .CSV untuk divalidasi secara manual oleh ahli bahasa. Tabel 1 merupakan jumlah data dan presentase data sentimen positif dan negatif yang telah diklasifikasikan otomatis dengan *InSet Lexicon* dan ahli bahasa.

Tabel 1. Jumlah data dan presentase sentimen positif & negatif

Kelas	***InSet Lexicon***	**Ahli Bahasa**
Positif	929 (58,65%)	929 (58,65%)
Negatif	655 (41,35%)	655 1,35%)

3.3 *Data Preparation*

Pada tahap *data preparation* ini, data yang sudah dilabeli dilakukan *cleaning* data agar mempermudah dalam proses klasifikasi. Adapun hasil dari *text preprocessing* data sebagai berikut:

a). *Cleansing*

Proses *cleansing* ini bertujuan untuk membersihkan data dari karakter yang dinilai tidak akan berpengaruh pada hasil klasifikasi. Proses *cleansing* ini diantaranya yaitu menghapus *punctuation, number, emoticon,* dan *space*. Berikut ini merupakan proses *cleansing* pada dataset penelitian ini.

- *Cleansing Punctuation*

  Tabel 2 menunjukkan karakter tanda baca yang dihapus dari dataset.

Tabel 2. *Cleansing Punctuation*

Ulasan dengan dengan karakter tanda baca	**Ulasan setelah *cleansing punctuation***
Jelek banget volume nya cuman bisa sampe 200, temen aku (evee) suaranya kecil banget, tolong bikinin ampe volume 1000	Jelek banget volume nya cuman bisa sampe 200 temen aku evee suaranya kecil banget tolong bikinin ampe volume 1000

- *Cleansing Number*

  Pada proses ini, karakter nomor akan dihapus dan akan ditampilkan di Tabel 3.

Tabel 3. *Cleansing Number*

Ulasan dengan dengan karakter nomor	**Ulasan setelah *cleansing number***
Bagus sekali Ã°ÂŸÂ‘Â• Ã°ÂŸ Â‘Â• Ã°ÂŸÂ‘Â• bisa ga aku kasih bintang 10	Bagus sekali Ã°ÂŸÂ‘Â• Ã°ÂŸ Â‘Â• Ã°ÂŸÂ‘Â• bisa ga aku kasih bintang

- *Cleansing Emoticon*

Pada proses ini, merupakan tahapan untuk menghapus emoticon pada dataset seperti pada Tabel 4.

Tabel 4. *Cleansing Emoticon*

Ulasan dengan dengan karakter *emoticon*	**Ulasan setelah *cleansing emoticon***
Bagus sekali Ã°ÂŸÂ‘Â• Ã°ÂŸ Â‘Â• Ã°ÂŸÂ‘Â• bisa ga aku kasih bintang	Bagus sekalibisa ga aku kasih bintang

- *Cleansing Space*

Pada proses ini, menghapus data yang kelebihan spasi yang ada di dalam dataset seperti yang ada pada Tabel 5.

Tabel 5. *Cleansing Space*

Ulasan dengan dengan karakter spasi	**Ulasan setelah *cleansing space***
Bagus sekalibisa ga aku kasih bintang	Bagus sekali bisa ga aku kasih bintang

b). *Case Folding*

Tahap *case folding* ini adalah tahap untuk mengubah semua huruf besar yang terdapat pada tahap sebelumnya menjadi huruf kecil (*lowercase*). Hasil dari tahap ini dapat dilihat pada Tabel 6.

Tabel 6. *Case Folding*

Ulasan dari *cleansing*	**Ulasan setelah *case folding***
Udah bagus tapi lambat loading nya	udah bagus tapi lambat loading nya

c). *Tokenization*

Tahap *tokenization* yaitu ulasan pada *dataset* akan menjadi penggalan kata sehingga menghasilkan token. Hasil dari tahap *tokenizing* terdapat pada Tabel 7.

Tabel 7. *Tokenization*

Ulasan dari *case folding*	**Ulasan setelah *tokenization***
udah bagus tapi lambat loading nya	'udah', 'bagus', 'tapi', 'lambat', 'loading', 'nya'

d). *Normalize*

Pada tahap *normalize*, data yang tidak sesuai ejaan akan diperbaiki dan menyesuaikannya dengan menggunakan kamus normalisasi kata. Tabel 5 merupakan hasil dari *normalize*.

Tabel 8. *Normalize*

Ulasan dari *tokenization*	**Ulasan setelah *normalize***
'udah', 'bagus', 'tapi', 'lambat', 'loading', 'nya'	sudah bagus tapi lambat loading nya

e). *Stopword Removal*

Pada tahap *stopword* akan dilakukan untuk menghapus kata-kata yang dinilai kurang berpengaruh untuk hasil klasifikasi. Hasil dari *stopword removal* pada Tabel 6.

Tabel 9. *Stopword Removal*

Ulasan dari *normalize*	**Ulasan setelah *stopword removal***
sudah bagus tapi lambat loading nya	bagus lambat loading nya

f). *Stemming*

Tahap *stemming* merupakan proses mengubah kata-kata menjadi bentuk dasar menggunakan *library* di python yaitu *library* NLTK dan Sastrawi Nazief-Andriani. Hasil dari proses *stemming* ada pada Tabel 10.

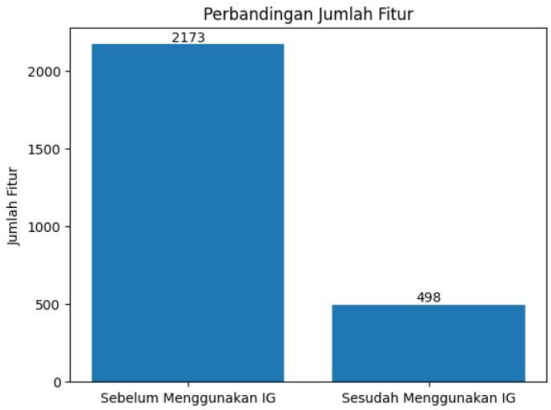
Tabel 10. *Stemming*

Ulasan dari *stopword removal*	**Ulasan setelah *stemming***
beri bintang setiap chat chat tidak terkirim diperbaiki	beri bintang tiap chat chat tidak kirim baik

3.4 *Modeling*

Tahap *modeling* merupakan tahapan dilakukannya penerapan *Information Gain*. Data *categoric* ke *numeric* dengan melakukan pembobotan kata dan mendeteksi fitur yang paling berpengaruh dari kelas yang telah ditentukan dengan perhitungan nilai *entropy*. Untuk mencari *entropy*, data akan diproses dengan ketentuan *entropy set* dan dihitung berdasarkan nilai kemungkinan kemunculan kata yang akan dikumpulkan menjadi nilai pembobotan. Hasilnya akan menjadi nilai *entropy word* yang nantinya akan diolah untuk mencari nilai *gain*.

Nilai *gain* yang didapatkan akan diseleksi dengan menerapkan *threshold* 1, untuk kata yang memiliki nilai *gain* dibawah *threshold* yang telah ditentukan maka kata tersebut akan terhapus dan dianggap tidak relevan. Pada Gambar 9 adalah perbandingan jumlah fitur sebelum dan sesudah menerapkan *gain* dengan *threshold* 1.



Gambar 9. Perbandingan Jumlah Fitur Penggunaan *Information Gain*

Pada proses ini telah didapatkan 498 kata yang dianggap berpengaruh dalam proses klasifikasi. Setelah didapatkan fitur berpengaruh, data akan dibagi ke dalam data *training* dan data *testing* serta melakukan penerapan algoritma *Naïve Bayes Classifier*.

Tabel 11. Split Data Training dan Testing

Persentase Pengujian (%)	**Data *Training***	**Data *Testing***
90% dan 10%	1425	159
80% dan 20%	1267	317
70% dan 30%	1108	476
60% dan 40%	950	634
50% dan 50%	792	792

Setelah dilakukan pengujian melakukan *Information Gain* dan *Naïve Bayes Classifier*, selanjutnya

melakukan pengujian dengan menerapkan algoritma *Naïve Bayes Classifier* tanpa seleksi fitur *Information Gain.*

3.5 *Evaluation*

Pada tahap evaluasi, peneliti melakukan perbandingan dari klasifikasi algoritma *naïve bayes* yang menerapkan seleksi fitur *information gain* dan yang tidak menerapkan seleksi fitur. Evaluasi dilakukan untuk mengukur kinerja klasifikasi dengan menggunakan *confusion matrix* menggunakan nilai *accuracy, precision, recall,* dan *f-measure* dengan menerapkan 5 *scenario* pengujian. Tabel 12 merupakan perbandingan dari hasil pengujian *Naïve Bayes Classifier* dengan seleksi fitur *Information Gain* dan *Naïve Bayes Classifier* tanpa seleksi fitur.

Tabel 12. Perbandingan Algoritma

Nilai	*Information Gain* dan *Naïve Bayes Classifier*				
	90:10	80:20	70:30	60:40	50:50
Accuracy	84%	82%	80%	80%	80%
Precision	84%	82%	80%	80%	80%
Recall	84%	82%	80%	80%	80%
F-Measure	83%	82%	80%	79%	80%
Nilai	*Naïve Bayes Classifier*				
	90:10	80:20	70:30	60:40	50:50
Accuracy	77%	78%	76%	76%	77%
Precision	78%	79%	76%	77%	77%
Recall	77%	78%	76%	76%	77%
F-Measure	76%	77%	75%	76%	76%

Dapat dilihat bahwa *accuracy* tertinggi dengan nilai 84% yang didapatkan dari *scenario* 90:10 menggunakan *information gain* dan *naïve bayes*.

Selain itu, hasil dari klasifikasi ulasan pengguna aplikasi *discord* divisualisasikan dengan menggunakan *wordcloud* untuk mengetahui gambaran atau informasi umum tentang hasil dari penelitian ini.

Gambar 10. Visualisasi *wordcloud*

Gambar 10 menunjukkan *wordcloud* yaitu visualisasi dari kata-kata yang sering muncul pada penelitian ini. Hasil frekuensi kata kemunculan paling banyak yaitu kata "enggak", "aplikasi", "discord" menjadi kata yang sering digunakan oleh pengguna untuk memberikan ulasan pada aplikasi *discord* di *google play store*.

3.6 *Deployment*

Pada tahap ini dilakukan pembuatan laporan hasil kegiatan yang sudah dilakukan. Laporan akhir mengenai pengetahuan yang didapat tentang hasil analisis yang telah dievaluasi.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan sebagai berikut:

1. Penelitian ini melakukan klasifikasi ulasan pengguna aplikasi discord menggunakan *Naïve Bayes Classifier* dengan menerapkan seleksi fitur *Information Gain*. Metodologi yang digunakan pada penelitian ini yaitu CRISP-DM terdiri dari *business understanding* sebagai tahapan pemecahan permasalahan pada penelitian, *data understanding* sebagai tahap pengambilan data dan persiapan data untuk penelitian, *data preparation* sebagai tahap *text preprocessing* atau pembersihan data yang digunakan pada penelitian, *modeling* merupakan tahap penerapan seleksi fitur *Information Gain* dan algoritma *Naïve Bayes Classifier*, *evaluation* sebagai tahap untuk mengukur hasil dari pengujian menggunakan *confusion matrix*, dan *deployment* dilakukan pembuatan laporan akhir dari hasil penelitian yang sudah dilakukan. Gambaran atau informasi umum mengenai data ulasan pengguna aplikasi discord diperoleh dari visualisasi dengan *wordcloud*. Kata yang sering muncul adalah kata "enggak", "aplikasi", "discord".
2. Hasil *accuracy* tertinggi yang diperoleh yaitu pada pengujian menggunakan *Information Gain* dan *Naïve Bayes Classifier* dengan 90% *data training* dan 10% *data testing* dan memperoleh *accuracy* sebesar 84%. Penambahan seleksi fitur *Information Gain* berguna untuk mengoptimalkan hasil

klasifikasi dari *Naive Bayes Classifier* dengan cara mengurangi fitur-fitur yang kurang relevan.

Beberapa hal yang disarankan peneliti adalah sebagai berikut:

1. Analisis ulasan pengguna aplikasi *discord* dapat dilakukan dengan menggunakan metode klasifikasi lain seperti *K-Nearest Neighbor* (KNN) atau *Support Vector Machine*.
2. Data yang digunakan dalam penelitian ini data ulasan berbahasa Indonesia. Diharapkan pada penelitian selanjutnya dapat menggunakan data dari berbagai bahasa.
3. Untuk penelitian selanjutnya dapat ditambahkan seleksi fitur *N-Gram, Particle Swarm Optimization, Chi Square, Stable Mutation Jump Strategy* atau seleksi fitur lainnya untuk membandingkan kinerja metode *Naïve Bayes Classifier* dengan adanya penambahan seleksi fitur tersebut.

PUSTAKA

Erfina, A., Basryah, E. S., Saepulrohman, A., & Lestari, D. (2020). Analisis Sentimen Aplikasi Pembelajaran Online Di Play Store Pada Masa Pandemi Covid-19 Menggunakan Algoritma Support Vector Machine (Svm). In *Seminar Nasional Informatika (SEMNASIF)*, 1(1), 145-152.

Indriati, I., & Ridok, A. (2016). Sentiment Analysis For Review Mobile Applications Using Neighbor Method Weighted K-Nearest Neighbor (Nwknn). *Journal of Environmental Engineering and Sustainable Technology*, 3(1), 23-32.

Lu, M., & Liang, P. (2017). Automatic classification of non-functional requirements from augmented app user reviews. *In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (pp. 344-353).

Marlina, D., & Bakri, M. (2021). Penerapan Data Mining Untuk Memprediksi Transaksi Nasabah Dengan Algoritma C4. 5. *Jurnal Teknologi Dan Sistem Informasi*, 2(1), 23-28.

Minarni, M., Prasetyaningrum, E., & Ismail, A. (2022). Pelatihan Pemanfaatan Aplikasi Discord Sebagai Kelas Virtual Bagi Guru Se-Kotawaringin Timur. Dinamisia: Jurnal Pengabdian Kepada Masyarakat, 6(4), 1068-1078.

Negara, A. B. P., Muhardi, H., & Putri, I. M. (2020). Analisis Sentimen Maskapai Penerbangan Menggunakan Metode Naive Bayes Dan Seleksi Fitur Information Gain. *J. Teknol. Inf. dan Ilmu Komput*, 7(3), 599-606.

Panggabean, F. (2021). Penerapan Media Pembelajaran Daring dengan Memanfaatkan Aplikasi Discord pada Mata Pelajaran IPA Terpadu Selama Pandemi Covid-19 di Kelas VIII-2 SMP Negeri 2 Tebing Tinggi. *SCHOOL EDUCATION JOURNAL PGSD FIP UNIMED*, 11(1), 35-41.

Permana, M. A., & Widiastuti, S. (2023). Analisis Sentimen Penggunaan Aplikasi Video Conference Pada Ulasan Google Play Store Menggunakan Metode NBC (Naive Bayes Classifier). *Jurnal Riset Sistem Informasi dan Teknologi Informasi (JURSISTEKNI)*, 5(1), 178-191.

Pintoko, B. M., & Lhaksmana, K. M. (2018). Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier. *eProceedings of Engineering*, 5(3), 8121–8130.

Putra, A. D. A., & Juanita, S. (2021). Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(2), 636-646.

Rakhmawan, A., Juansah, D. E., Nulhakim, L., Biru, L. T., Rohimah, R. B., Suryani, D. I., ... & Resti, V. D. A. (2020, November). Analisis Pemanfaatan Aplikasi Discord dalam Pembelajaran Daring Di Era Pandemi Covid-19. In *Prosiding Seminar Nasional Pendidikan FKIP*, 3(1), 55-59.

Ridho, M. R., Muhaimin, M., & Harjono, H. S. (2021). Pengaruh Aplikasi Discord dalam Pembelajaran Daring Terhadap Hasil Belajar pada Matakuliah Komputer. *Jurnal Ilmiah Bina Edukasi*, 14(1), 22-35.

Rohanah, A., Dermawan, B. A., & Purnamasari, I. Klasifikasi Ulasan Pengguna Zoom Cloud Meetings Menggunakan Metode Information Gain dan Naïve Bayes Classifier. *Jurnal Informatika Universitas Pamulang*, 6(2), 348-357.

Sari, A. E., Widowati, S., & Lhaksmana, K. M. (2019). Klasifikasi Ulasan Pengguna Aplikasi Mandiri Online di Google Play Store dengan Menggunakan Metode Information Gain dan Naive Bayes Classifier. *eProceedings of Engineering*, 6(2), 9143-9157.

Subagja, R. A., Widiastiwi, Y., & Chamidah, N. (2021). Klasifikasi Ulasan Aplikasi Jenius pada Google Play Store Menggunakan Algoritma Naive Bayes. Informatik: Jurnal Ilmu Komputer, 17(3), 197-208.

Thaha, A. R., & Aziz, F. (2020). Penambangan Teks Pada Tujuan Wisata di Bandung Raya (Studi Kasus: Tangkuban Perahu dan Kawah Putih). *Jurnal Sekretaris dan Administrasi Bisnis*, 4(2), 146-156.

Tjahjadi, E., Paramita, S., & Salman, D. (2021). Pembelajaran Era Pandemi Covid-19 di Indonesia (Studi terhadap Aplikasi Discord). *Koneksi*, 5(1), 83-89.

Tumewu, M. C. I. S., & Kurniasari, N. (2022). Motif Dan Kepuasan Komunitas Acid Pada Media Sosial Discord Sebagai Sarana Pemenuhan Kebutuhan Informasi. *Jurnal Pustaka Komunikasi*, 5(1), 25-37.

Watrianthos, R., Suryadi, S., Irmayani, D., Nasution, M., & Simanjorang, E. F. (2019). Sentiment Analysis Of Traveloka App Using Naïve Bayes Classifier Method. *Int. J. Sci. Technol. Res*, 8(7), 786-788.