J-SIMTEK e-issn: 2987-1115
Jurnal Sistem Informasi, Manajemen dan Teknologi Informasi
Vol. 3, No. 1, Januari, 2025
Published by STMIK Palangkaraya

Perbandingan Algoritma Naïve Bayes Dan Support Vector Machine Dalam Menentukan Sentimen Publik Terhadap Copilot

Meilinda[1], Nevin Julian Masidin[2], Agustio Dwitama[3], Andri Wijaya[4*]

[1] Sistem Informasi, Universitas Katolik Musi Charitas, Palembang
[2] Sistem Informasi, Universitas Katolik Musi Charitas, Palembang
[3] Sistem Informasi, Universitas Katolik Musi Charitas, Palembang
[4] Sistem Informasi, Universitas Katolik Musi Charitas, Palembang
[1]meilindachen05@gmail.com, [2]nevinnjul2003@gmail.com , [3]agustiod298@gmail.com, [4]andri_wijaya@ukmc.ac.id*

INTISARI

Chatbot modern, termasuk Copilot dari Microsoft Edge, berkembang pesat berkat teknologi pemrosesan bahasa alami (NLP). Penelitian ini menganalisis sentimen publik terhadap Copilot menggunakan algoritma Naïve Bayes dan Support Vector Machine (SVM). Dengan mengumpulkan 20.000 ulasan dari Google Play Store melalui library Python "google_play_scraper", data diproses dengan langkah-langkah preprocessing, diikuti pelabelan menggunakan VADER untuk mengklasifikasikan ulasan menjadi positif, negatif, dan netral. Metode TF-IDF digunakan untuk mengindeks dan memberikan bobot pada istilah sebelum menerapkan model pencarian berbasis vektor. Hasil evaluasi menunjukkan SVM unggul dibandingkan Naïve Bayes dalam akurasi (96,54% vs 90,32%), precision, recall, dan F1-score, terutama dalam mendeteksi ulasan negatif. Word Cloud analisis menunjukkan kata kunci positif dominan seperti "app" dan "good", mencerminkan persepsi pengguna yang baik terhadap Copilot. Penelitian ini merekomendasikan pengoptimalan lebih lanjut untuk Naïve Bayes dan menegaskan bahwa SVM adalah pilihan lebih baik untuk analisis sentimen kompleks.

Kata kunci: analisis sentimen, copilot, support vector machine, naïve bayes

ABSTRACT

Modern chatbots, including Copilot from Microsoft Edge, are evolving rapidly due to natural language processing (NLP) technology. This research analyzes public sentiment towards Copilot using Naïve Bayes and Support Vector Machine (SVM) algorithms. By collecting 20,000 reviews from the Google Play Store using the Python library "google_play_scraper", the data undergoes preprocessing and labeling with VADER to classify reviews as positive, negative, or neutral. The TF-IDF method indexes and weights terms before applying a vector-based search model. Results show that SVM significantly outperforms Naïve Bayes in accuracy (96.54% vs 90.32%), precision, recall, and F1-score, especially in detecting negative reviews. Word Cloud analysis reveals dominant positive keywords like "app" and "good," reflecting users' favorable perception of Copilot. This research recommends further optimization for Naïve Bayes and emphasizes that SVM is a better choice for complex sentiment analysis.

Keywords: sentiment analysis, copilot, support vector machine, naïve bayes

1. PENDAHULUAN

Chatbot atau robot obrolan, merupakan program komputer yang dirancang untuk berinteraksi dengan pengguna layaknya manusia. Meskipun tidak semua chatbot menggunakan kecerdasan buatan, chatbot modern semakin canggih dengan

memanfaatkan teknologi seperti pemrosesan bahasa alami. Hal ini memungkinkan chatbot untuk memahami pertanyaan pengguna secara lebih baik dan memberikan respons yang lebih relevan dan otomatis (IBM, n.d.). Dalam beberapa tahun terakhir, aplikasi chatbot telah berkembang pesat dan menjadi bagian integral dari berbagai sektor. Beberapa contoh chatbot yang begitu populer, diantaranya: ChatGPT, Siri, Gemini, Perplexity, dan salah satunya yang akan dibahas pada penelitian ini adalah Copilot, milik Microsoft Edge. Bersaingnya para-AI Chatbot, menjadikan penulis memiliki ketertarikan tersendiri untuk meneliti lebih lanjut mengenai aplikasi Copilot. Copilot sendiri memang tidak terlalu dikenal oleh masyarakat namun memiliki ulasan yang terbilang cukup banyak, yakni 613 ribu ulasan di Google Play Store.

Menunjang para peneliti untuk melakukan penelitian mendalam terkait sentimen – sentimen pada suatu AI Chatbot, terutama Copilot, dapat diterapkannya analisis sentimen. Suatu metode dalam bidang pemrosesan bahasa alami (*Natural Language Processing* atau NLP), yaitu analisis sentimen yang dapat berguna dalam melakukan seleksi atau klasifikasi suatu pendapat berupa teks yang memiliki perasaan secara positif, negatif, ataupun netral. Dengan menganalisis sentimen, dapat terbantu dalam melacak persepsi konsumen terhadap suatu produk, mengukur efektivitas kampanye pemasaran, memprediksi tren pasar, dan bahkan memantau sentimen politik. (Purnamasari et al., 2023).

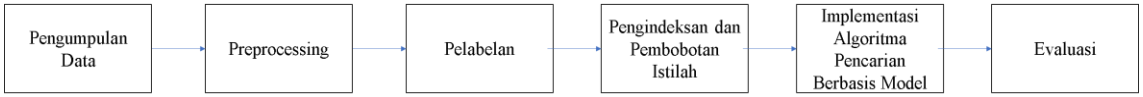
Berdasarkan pada hal-hal yang sudah disampaikan sebelumnya, menjadikan perhatian lebih bagi penulis untuk melakukan klasifikasi sentimen publik pada Copilot menggunakan dua algoritma, yakni Naïve Bayes dan Support Vector Machine (SVM) agar dapat dilakukannya juga perbandingan keakuratan analisis sentimen. Naïve Bayes merupakan algoritma yang sering digunakan untuk mengelompokkan data dan membuat ramalan. Algoritma ini memiliki keunggulan dalam implementasi yang sederhana dan waktu komputasi yang efisien untuk menghasilkan prediksi. Sebagai algoritma pembelajaran mesin berbasis probabilitas, Naïve Bayes dapat digunakan untuk mengklasifikasikan dokumen teks, melakukan diagnosis medis otomatis, dan mendeteksi spam. Algoritma ini membuat prediksi dengan menghitung kemungkinan suatu data termasuk dalam kategori tertentu (Purnamasari et al., 2023). Sedangkan, Support Vector Machine atau dikenal sebagai SVM merupakan metode yang sangat

baik (akurat) untuk mengelompokkan data dan memprediksi nilai (Wang & Springer, 2005). SVM menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam ruang fitur berdimensi tinggi. Dengan kata lain, SVM mencari sebuah persamaan matematis yang dapat memisahkan data dengan jelas (Cristianini & Shawe-Taylor, 2000).

Digunakannya dua algoritma secara bersamaan untuk perbandingan keakuratan masing-masing, terdapat beberapa penelitian pendukung dan relevan yang pernah menggunakan salah satu atau sekaligus algoritma ini. Penelitian oleh Widia Ningsih, Baginda Alfianda, Rahmaddeni, dan Denok Wulandari menyebutkan bahwa perbandingan akurasi dari kedua algoritma yang digunakan, yaitu Naïve Bayes dan SVM mereka terhadap analisis sentimen Twitter terhadap penggunaan mobil Listrik Indonesia yang diambil 1517 data, masing – masing sebesar 63,02% dan 70,83% (Ningsih et al., 2024). Ada juga penelitian Oleh John Friadi dan Dwi Ely Kurniawan, (Friadi & Kurniawan, 2024) menggunakan dua metode yang sama dengan penelitian ini untuk melakukan analisis sentimen ulasan wisatawan terhadap alun-alun kota Batam (Sarimole & Kudrat, 2024), dengan 1140 data diambil dan mendapatkan akurasi Naïve Bayes sebanyak 83% dan SVM sebanyak 94%. Tidak hanya itu, penelitian lain oleh Frencis Matheos Sarimole dan Kudrat, melakukan analisis sentiment pada Aplikasi Satu Sehat di Twitter, dengan 1046 data dan menghasilkan akurasi Naïve Bayes sebesar 81,65% dan SVM sebesar 87,95%. Penelitian-penelitian diatas tidak terlalu dijelaskan bagaimana cara ketika melakukan pelabelan. Maka dari itu, penelitian ini akan lebih memperjelas dengan menggunakan alat analisis sentimen berbasis leksikon dalam menentukan sentimen dalam teks (VaderSentiment, n.d.). Selain itu, dataset yang diambil akan lebih diperbanyak pada penelitian ini agar dapat memberikan representasi yang lebih komprehensif dan dapat diandalkan tentang sentimen publik, khususnya pada Copilot. Terakhir, penambahan keunikan dari penelitian ini adalah pendekatan inovatif dalam menggabungkan analisis sentimen dengan pencarian berbasis model. Ini bisa membantu dalam meningkatkan akurasi prediksi sentimen, terutama dalam dataset yang besar dan kompleks.

2. METODOLOGI

Proses penelitian akan dimulai dari beberapa tahapan – tahapan penting. Gambar berikut merupakan tahapan yang akan dilalui.



Gambar 1. Alur Penelitian

2.1. Pengumpulan Data

Pengumpulan data bertujuan untuk dikumpulkannya suatu informasi dan sekumpulan fakta yang diperlukan dalam penelitian atau kajian (Wajdi et al., n.d.). Penulis mengumpulkan data sebanyak 20.000 dan hanya ulasan yang masuk kategori "*most relevant*". Penulis memilih kategori tersebut dikarenakan ulasan yang tersedia biasanya lebih informatif dan bermanfaat, ulasan publiknya cenderung merupakan pengalaman nyata dengan aplikasi, sehingga memberikan wawasan yang lebih mendalam tentang kelebihan dan kekurangan aplikasi tersebut. Data ulasan diambil di Google Play Store dengan menggunakan *library* Python "google_play_scraper". *Library* ini menyediakan antarmuka yang user-friendly untuk mengakses data ulasan tanpa perlu melakukan *web scraping* secara manual dan berdasarkan informasi yang ada, meskipun proyek pengembangan *library* ini adalah *open-source*, namun nama JoMingyu adalah yang paling sering muncul dalam repositori GitHub untuk "google_play scraper" serta aktif berkontribusi dalam pengembangan *library* ini (JoMingyu, n.d.).

2.2. Preprocessing

Tahap ini dilakukan karena merupakan proses awal yang dilakukan pada data mentah sebelum digunakan dalam analisis data atau pengembangan model. Tahapan ini bertujuan untuk memperbaiki kualitas data, seperti menghilangkan data yang tidak lengkap atau tidak konsisten, sehingga hasil analisis yang didapatkan lebih akurat (Rahayu et al., n.d.).

2.3. Pelabelan

Pada proses pelabelan, penulis secara khusus menggunakan VADER (*Valence Aware Dictionary and Entiment Reasoner*) untuk analisis lebih lanjut mengenai ulasan terhadap Copilot, yang merupakan sebuah perangkat lunak yang dirancang untuk menganalisis sentimen dalam teks. Dengan menggunakan kamus khusus dan aturan

tertentu, VADER dapat mengukur seberapa positif, negatif, atau netral sebuah kalimat atau paragraf (VaderSentiment, n.d.).

2.4. Pengindeksan dan Pembobotan Istilah

Skema pembobotan istilah sangat penting dalam klasifikasi teks yang akurat. Dengan menggunakan pendekatan pembobotan istilah yang tepat, maka dapat mengidentifikasi kata-kata yang paling relevan dan memberikan nilai penting yang sesuai untuk membantu dalam mengklasifikasikan teks dengan lebih baik (Ren & Sohrab, 2013). Metode pembobotan istilah yang digunakan penulis adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). Bobot suatu kata dalam TF-IDF ditentukan oleh seberapa sering kata itu muncul dalam dokumen yang sedang dianalisis dan seberapa umum kata itu ditemukan di seluruh kumpulan dokumen (Rahayu et al., n.d.). Pengindeksan digunakan untuk mempercepat proses pencarian dokumen dengan memastikan bahwa dokumen yang paling relevan dengan *query* pengguna dapat ditemukan dengan cepat dan efisien (1Library, n.d.). Dalam pengambilan data dan informasi, penulis menggunakan Inverted Index, yang merupakan struktur data yang memetakan setiap kata unik dalam sebuah kumpulan dokumen ke daftar dokumen yang mengandung kata tersebut. Struktur ini dioptimalkan untuk pencarian kata-kata tertentu secara efisien (Binus University, n.d.).

2.5. Implementasi Algoritma Pencarian Berbasis Model

Algoritma pencarian berbasis model, terdiri dari beberapa varian, seperti Boolean Retrieval Model, Probabilistic Model, termasuk Vector Space Model yang akan digunakan oleh penulis. Algoritma ini merupakan teknik dasar dalam pengolahan informasi yang digunakan untuk mengukur kemiripan antara dokumen dengan *query* pencarian. Model ini juga bisa dipakai untuk mengelompokkan dokumen yang memiliki topik serupa atau mengklasifikasikan dokumen ke dalam kategori tertentu (Binus University, n.d.).

2.6. Evaluasi

Sebelum hasil klasifikasi, ada tahap – tahap lain yang perlu dilakukan yaitu uji data latih dan data uji. Kemudian melakukan inisialisasi model Naïve Bayes dan SVM.

Setelah dilatih, maka dapat digunakannya model tersebut untuk melakukan prediksi data.

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan lebih memperjelas tahapan hasil yang didapati selama dilakukannya penelitian. Sesuai dengan metodologi penelitian, dimulai dari pengumpulan data, *preprocessing*, pelabelan, pengindeksan dan pembobotan istilah, implementasi algoritma pencarian berbasis model, dan terakhir evaluasi.

3.1. Pengumpulan Data

Data yang diambil oleh penulis berasal dari Google Play Store yang tertuju pada ulasan berbahasa inggris di aplikasi Copilot. Gambar dibawah merupakan 20.000 ulasan yang sudah dikumpulkan.

19977	so helpful!
19978	"Alhamdulillah, Splendid"
19979	رائع جدا و انصح به ولاكن الرجاء إضافه اللغه العربية للتحدث
19980	better choice 😊👍
19981	ye app se aap bahut kuchh nahi sab kuchh Sikh sakte hai
19982	good 😊 bhot acha
19983	"excellent work, 💯💪"
19984	just good 😊
19985	عالی بود برنامه خوبیه پیشنهاد میکنم دانلود کنید.
19986	I've never had a question I couldn't get help with
19987	really fast accurate and reliable
19988	good 👍🏻
19989	Keep crashing !! 😡
19990	Good pal!
19991	great AI!
19992	Very impressive.
19993	perfect 😍 Lovett
19994	love copilot.
19995	very useful 😊
19996	so wonderful ✨
19997	"helpful, quick"
19998	very helpful 😊
19999	Great Experience.
20000	very impressive 👏
20001	very useful.

Gambar 2. Hasil Scraping

3.2. Preprocessing

Langkah – langkah *preprocessing* yang dilakukan pada teks penelitian ini adalah sebagai berikut: menghapus URL, menghapus karakter non-alfabet, mengubah teks ke huruf kecil, menghapus spasi berlebih, mengubah kontraksi, tokenisasi, menghapus stopwords, lemmatization, menghapus token dengan panjang <=2, menggabungkan kembali token menjadi string agar dapat menghasilkan teks akhir yang bersih dan siap untuk analisis lebih lanjut. Gambar dibawah merupakan salah satu contoh kalimat yang sebelum dan sesudah dilakukan *preprocessing*.

```
my_df['content'][1]
```

```
"While this app has significantly improved, it still has one bizarre bug in
it. After using this app for several minutes chatting, it mysteriously stops
generating text in mid-sentence and disables the submit button. What is the
purpose, if any, of my this, I can't discern. Improvements are gone. It lose
s your previous chats. Time to uninstall this cr-app. Goodbye Copilot."
```

```
my_df['CleanReview'][1]
```

```
'app significantly improve still one bizarre bug use app several minutes cha
t mysteriously stop generate text mid sentence disable submit button purpose
discern improvements lose previous chat time uninstall app goodbye copilot'
```

Gambar 3. Hasil *Preprocessing*

3.3. Pelabelan

Pelabelan dimulai dengan memberikan label "*positive*", "*negative*", dan "*neutral*" serta mendapatkan skor polaritas untuk setiap ulasan. Terakhir, penghapusan data yang terdeteksi "*neutral*". Karena ulasan yang bersifat netral tidak memberikan informasi yang cukup kuat atau jelas mengenai opini pengguna. Gambar dibawah merupakan sebagian contoh kalimat yang hanya dikumpulkan 10 data teratas, dengan dihitung skor polaritasnya dari ulasan yang sudah dilakukan *preprocessing* dan penghapusan data "*neutral*".

	CleanReview	Polarity	Predicted_Label
0	overall rate four star honestly good program u...	0.8977	positive
1	app significantly improve still one bizarre bu...	-0.1779	negative
2	need work app frequently lose microphone acces...	-0.9042	negative
3	love one others test except need add bookmark ...	0.9712	positive
4	hello copilot decent chatbox get help things i...	0.8020	positive
5	one intuitive creative humorous versions ever ...	0.9349	positive
6	start impress days flaw show conversation part...	0.8271	positive
7	edit find previous review identify problem app...	-0.2023	negative
8	wow read review shock leave shake head well ev...	0.6249	positive
9	really love interpersonal skills pilot also gi...	0.9769	positive

Gambar 4. Hasil Pelabelan

```
total_rows_after_removal = my_df.shape[0]
print(f"Jumlah total baris setelah penghapusan label netral: {total_rows_after_removal}")
```

```
Jumlah total baris setelah penghapusan label netral: 17468
```

Gambar 5. Total Setelah Penghapusan Label Netral

3.4. Pengindeksan dan Pembobotan Istilah

Pada tahap ini, menggunakan algoritma TF-IDF untuk pembobotan istilah dan membangun struktur pengindeksan menggunakan teknik Inverted Index. Gambar dibawah merupakan hasil dari pembobotan istilah dari pemrosesan TF-IDF dengan hanya menampilkan 10 fitur pertama dan pengindeksan dari Inverted Index.

```
['aaaa' 'aai' 'aap' 'aapl' 'aback' 'abandon' 'abc' 'abd' 'abdoloute'
 'abdullah']
  (0, 6979)     0.21258030690593552
  (0, 2513)     0.17850710596868516
  (0, 7499)     0.11538471912255148
  (0, 6966)     0.15239443341435915
  (0, 5772)     0.11775431060230065
  (0, 4454)     0.13424559228124167
  (0, 4967)     0.09318452680649467
  (0, 2556)     0.10959684588968203
  (0, 3162)     0.0889235806792898
  (0, 7635)     0.17728227319185505
  (0, 7065)     0.3256092547018875
  (0, 5264)     0.14443390503143477
  (0, 2264)     0.16409948765813098
  (0, 4412)     0.13187661169010959
  (0, 1611)     0.0696361247431998
  (0, 4702)     0.09327204998567143
  (0, 6905)     0.11147531551673787
```

Gambar 6. Hasil TF-IDF

```
Term: overall, Document IDs: [0, 0, 43, 70, 80, 92, 107, 115, 144, 168, 181, 196, 200, 204, 230, 236, 252, 334, 360, 36
5, 410, 496, 532, 543, 673, 687, 697, 839, 864, 866, 869, 913, 915, 939, 1011, 1063, 1092, 1105, 1113, 1214, 1361, 1379,
1381, 1419, 1468, 1529, 1548, 1570, 1581, 1620, 1626, 1671, 1714, 1725, 1819, 1922, 1981, 2018, 2020, 2023, 2058, 2066,
2076, 2161, 2209, 2244, 2297, 2512, 2627, 2662, 2662, 2674, 2721, 3028, 3055, 3059, 3067, 3085, 3153, 3153, 3210, 3222,
3228, 3239, 3276, 3652, 3665, 3675, 3677, 3693, 3699, 3702, 3781, 3866, 3870, 3872, 3873, 3883, 3911, 4068, 4123, 4239,
4324, 4355, 4371, 4441, 4445, 4515, 4794, 4808, 4857, 4872, 4943, 4955, 5060, 5101, 5125, 5157, 5215, 5309, 5323, 5337,
5419, 5464, 5488, 5523, 5674, 5705, 5862, 6429, 6484, 6484, 6503, 6581, 6617, 6628, 6637, 6757, 6758, 6820, 6936, 7016,
7141, 7428, 7918, 8039, 8134, 8141, 8452, 8522, 8578, 8622, 8711, 9320, 9396, 9405, 9414, 9495, 9511, 9518, 9574, 9584,
9589, 9593, 9688, 9756, 9813, 10121, 10367, 10483, 10826, 11432, 11527, 11594, 11720, 11805, 12424, 12631, 13648, 14216,
17152]
Term: rate, Document IDs: [0, 26, 158, 419, 508, 750, 928, 1432, 1432, 1459, 1597, 1864, 1905, 2533, 2644, 2724, 3153, 3
213, 3333, 3372, 3475, 3676, 4428, 5098, 5639, 6249, 7235, 8309, 8535, 9243, 9301, 9458, 10039, 10242, 11085, 12481, 129
90, 13045, 15307, 16175, 17143]
Term: four, Document IDs: [0, 0, 13, 23, 72, 93, 95, 116, 121, 176, 679, 794, 1870, 3213, 3748, 4107, 4927, 5344, 6451,
9964, 13010]
Term: star, Document IDs: [0, 0, 20, 23, 30, 37, 42, 48, 48, 52, 81, 84, 90, 90, 107, 110, 114, 120, 157, 172, 176, 198,
199, 204, 204, 211, 215, 235, 241, 252, 261, 291, 325, 330, 353, 365, 398, 422, 435, 441, 441, 452, 464, 489, 501, 511,
511, 555, 572, 590, 626, 636, 667, 724, 776, 776, 798, 798, 821, 832, 832, 836, 839, 861, 892, 897, 897, 909, 917, 940,
1033, 1056, 1095, 1095, 1124, 1133, 1135, 1179, 1179, 1184, 1184, 1211, 1211, 1226, 1247, 1276, 1300, 1348, 1374, 1400,
```

Gambar 7. Hasil Inverted Index

3.5. Implementasi Algoritma Pencarian Berbasis Model

Vector Space Model akan diterapkan sebagai algortima pencarian berbasis model. Algoritma Pencarian diimplementasikan agar dapat memperkaya fungsionalitas sistem pencarian dan memberikan manfaat signifikan dalam hal relevansi hasil pencarian berdasarkan kebutuhan spesifik pengguna. Gambar dibawah merupakan hasil dari implementasi Vector Space Model dengan *query* “data analysis”.

```
# Contoh kueri pencarian
query = "data analysis"
ranked_indices, cosine_similarities = vector_space_search(query, tfidf_vectorizer, tfidf_matrix)
```

Gambar 8. Contoh Queri Pencarian

```
Document ID: 7431, Similarity: 0.586057548550476
need feature analyse stock market like fundamental analysis technical analysis candle stick pattern analysis stock latest
data

Document ID: 14506, Similarity: 0.5691729770235163
perfect gpt image analysis

Document ID: 10128, Similarity: 0.4465996287898555
good real time data interaction analysis restrictions like limit follow question limit responses

Document ID: 2699, Similarity: 0.4440046180657932
copilot excellent tool reduce hours manual research second leave time analysis data

Document ID: 8111, Similarity: 0.39463421276314303
give answer gemini give good quality answer analysis

Document ID: 1369, Similarity: 0.3820216029361576
good experience copilot give real understand theme idea analysis

Document ID: 674, Similarity: 0.3793812830724078
great simple analysis still need better search engines data improve accuracy information provide sometimes felt like mimic
give insight

Document ID: 13422, Similarity: 0.34730872723597506
save time give data

Document ID: 11382, Similarity: 0.3213282210037308
great access latest data

Document ID: 10598, Similarity: 0.3211281457855926
kindly request implement data save mode feature application currently consume excessive data deliver result data save mode
feature would enable application provide better result reduce data consumption
```

Gambar 9. Implementasi Vector Space Model

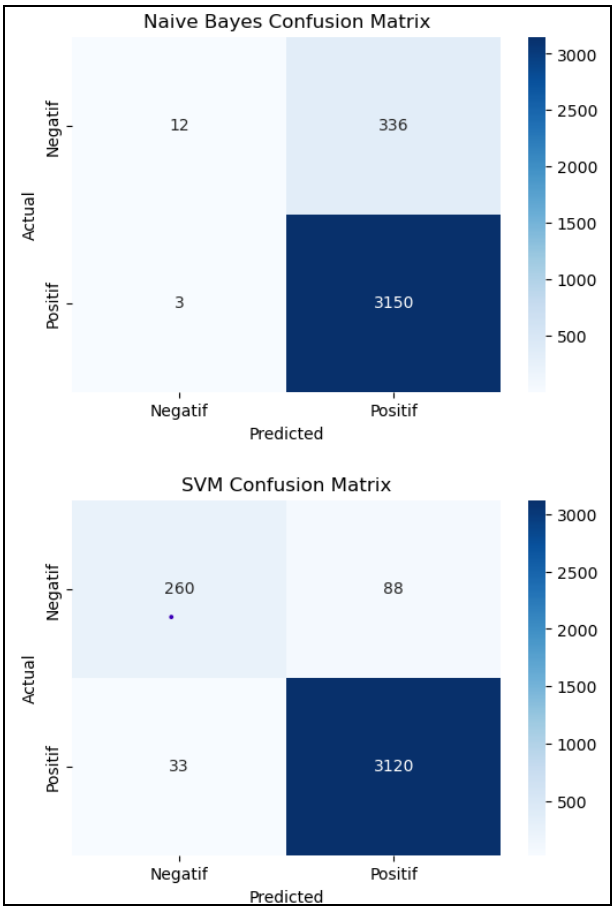
3.6. Evaluasi

Tahap ini dilakukannya proses penting seperti, persiapan data dengan membagi menjadi dua set: data pelatihan (80% dari total) dan data pengujian (20% dari total), serta melakukan prediksi dan evaluasi model klasifikasi menggunakan algoritma Naïve Bayes dan SVM dengan data yang telah diproses menggunakan TF-IDF. Kemudian menghitung akurasi serta menghasilkan matriks kebingungan dan laporan klasifikasi.

```
Naive Bayes Accuracy: 0.9031705227077977
Naive Bayes Confusion Matrix:
 [[  12  336]
 [   3 3150]]
Naive Bayes Classification Report:
              precision    recall  f1-score   support

    negative       0.80      0.03      0.07       348
    positive       0.90      1.00      0.95      3153

    accuracy                           0.90      3501
   macro avg       0.85      0.52      0.51      3501
weighted avg       0.89      0.90      0.86      3501

SVM Accuracy: 0.9654384461582405
SVM Confusion Matrix:
 [[ 260   88]
 [  33 3120]]
SVM Classification Report:
              precision    recall  f1-score   support

    negative       0.89      0.75      0.81       348
    positive       0.97      0.99      0.98      3153

    accuracy                           0.97      3501
   macro avg       0.93      0.87      0.90      3501
weighted avg       0.96      0.97      0.96      3501
```

Gambar 10. Laporan Klasifikasi Naïve Bayes dan SVM

Naive Bayes memiliki *precision* tinggi untuk kelas positif (90%) tetapi rendah untuk kelas negatif (80%), dengan *recall* yang sangat rendah untuk kelas negatif (3%) dan tinggi untuk kelas positif (100%), menghasilkan *F1-score* rendah (7%) untuk kelas negatif. Sebaliknya, SVM menunjukkan *precision* yang lebih baik di kedua kelas, dengan *recall* yang lebih seimbang (75% untuk negatif) dan *F1-score* jauh lebih baik di kedua kelas, terutama pada kelas positif.



Gambar 11. *Confusion Matrix* Naive Bayes dan SVM

Pada model Naive Bayes, terlihat bahwa meskipun jumlah prediksi *true positives* sangat tinggi (3150), model ini gagal untuk mengidentifikasi banyak contoh negatif dengan benar, seperti terlihat dari tingginya jumlah *false positives* (336). Sedangkan, pada model SVM, meskipun masih ada kesalahan, jumlah *false positives* dan *false negatives* jauh lebih rendah dibandingkan Naive Bayes, menunjukkan bahwa SVM lebih baik dalam membedakan antara kelas negatif dan positif.

Gambar 12. Word Cloud

Terlihat pada visualisasi Word Cloud bahwa kata kata “app”, “good”, dan “Copilot” tampil dalam ukuran yang lebih besar dibanding yang lain. Ini mengindikasikan bahwa kata-kata tersebut adalah tema sentral atau istilah kata kunci yang sering digunakan, juga memberikan wawasan penting tentang persepsi dan topik yang paling relevan bagi pengguna, serta dapat membantu dalam analisis lebih lanjut mengenai sentimen dan tren dalam ulasan atau *feedback* yang diberikan.

4. KESIMPULAN

Model SVM menunjukkan kinerja superior dibandingkan Naive Bayes dalam hal akurasi, *precision*, *recall*, dan *F1-score*, dengan kemampuan lebih baik dalam mendeteksi kedua kelas dan mengurangi jumlah false positives. SVM menunjukkan akurasi 96,54% dan Naïve Bayes 90,32%. Meskipun Naïve Bayes memiliki akurasi yang cukup baik, model ini kesulitan dalam mengidentifikasi kelas negatif secara efektif.

Analisis Word Cloud dari 20.000 ulasan terhadap "Copilot" mengungkapkan kata-kata kunci seperti "*app*", "*good*", dan "Copilot", yang menunjukkan fokus positif pengguna terhadap aplikasi tersebut. Untuk meningkatkan kinerja Naive Bayes, disarankan untuk menerapkan teknik *preprocessing* tambahan atau eksplorasi parameter *tuning*. Jika tujuan utama adalah deteksi kelas negatif yang lebih baik, SVM adalah pilihan yang lebih tepat berdasarkan hasil ini.

5. DAFTAR PUSTAKA

1Library. (n.d.). *Indexing and Weighting*. Retrieved December 17, 2024, from https://1library.net/article/indexing-and-weighting-text-preprocessing.qvv62n1q

Binus University. (n.d.). *Metode-Metode Information Retrieval*. Retrieved December

17, 2024, from https://online.binus.ac.id/computer-science/2020/06/15/metode-metode-information-retrieval/

Cristianini, Nello., & Shawe-Taylor, John. (2000). *An Introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press. https://doi.org/https://doi.org/10.1017/CBO9780511801389

Friadi, J., & Kurniawan, D. E. (2024). Analisis Sentimen Ulasan Wisatawan Terhadap Alun-Alun Kota Batam: Perbandingan Kinerja Metode Naive Bayes dan Support Vector Machine. *Jurnal Sistem Informasi Bisnis*, *14*(4), 403–407. https://doi.org/10.21456/vol14iss4pp403-407

IBM. (n.d.). *What is a chatbot?* Retrieved December 16, 2024, from https://www.ibm.com/think/topics/chatbots

JoMingyu. (n.d.). *JoMingyu/google-play-scraper*. Retrieved December 20, 2024, from https://github.com/JoMingyu/google-play-scraper

Ningsih, W., Alfianda, B., Rahmaddeni, & Wulandari, D. (2024). Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 556–562. https://doi.org/10.57152/malcom.v4i2.1253

Purnamasari, D., Aji, A. B., A.P., D. W., Reza, F. A., O., M. S., Yanda, N., & Hidayati, U. (2023). *Pengantar Metode Analisis Sentimen*. https://penerbit.gunadarma.ac.id/wp-content/uploads/2023/09/Pengantar-Metode-Analisis-Sentimen_Detty-cs_Watermark.pdf

Rahayu, P. W., Sudipa, I. G. I. S., Suryani, Surachman, A., Ridwan, A., Darmawiguna I Gede Mahendra, Sutoyo, Muh. N., Slamet, I., Harlina, S., & Maysanjaya, I. M. D. (n.d.). *BUKU AJAR DATA MINING*. https://www.researchgate.net/publication/377415198

Ren, F., & Sohrab, M. G. (2013). Class-indexing-based term weighting for automatic text classification. *Information Sciences*, *236*, 109–125. https://doi.org/10.1016/j.ins.2013.02.029

Sarimole, F. M., & Kudrat. (2024). Analisis Sentimen Terhadap Aplikasi Satu Sehat Pada Twitter Menggunakan Algoritma Naive Bayes Dan Support Vector Machine. *Jurnal Sains Dan Teknologi*, *5*(3), 783–790. https://doi.org/10.55338/saintek.v5i1.2702

VaderSentiment. (n.d.). *Welcome to VaderSentiment's documentation!* Retrieved December 16, 2024, from https://vadersentiment.readthedocs.io/en/latest/

Wajdi, F., Atiningsih, S., Sinurat, J., Agustina, E. B., Ridhasyah, R., Lidyawati, Hozairi, Amane Ade Putra Ode, Hantono, Jumiati, E., Suprapto, F. M., Rijal, K., Ginting, R., & The, H. Y. (n.d.). *METODOLOGI PENELITIAN & ANALISIS DATA KOMPREHENSIF*.

Wang, L., & Springer, B. (2005). *Support Vector Machines: Theory and Applications*. https://personal.ntu.edu.sg/elpwang/PDF_web/05_SVM_basic.pdf