

**DETEKSI PERTANYAAN *INSINCERE* PADA KOMUNITAS TANYA JAWAB  
ONLINE MENGGUNAKAN METODE *MULTINOMIAL*  
*NAÏVE BAYES***

**SKRIPSI**

Oleh:  
**RIZHA ALFIANITA**  
**NIM. 200605110177**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

**DETEKSI PERTANYAAN *INSINCERE* PADA KOMUNITAS TANYA  
JAWAB *ONLINE* MENGGUNAKAN METODE *MULTINOMIAL*  
*NAÏVE BAYES***

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:  
**RIZHA ALFIANITA**  
**NIM. 200605110177**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

**HALAMAN PERSETUJUAN**

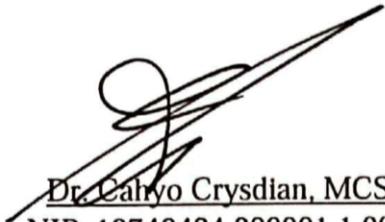
**DETEKSI PERTANYAAN *INSINCERE* PADA KOMUNITAS TANYA  
JAWAB *ONLINE* MENGGUNAKAN METODE *MULTINOMIAL  
NAÏVE BAYES***

**SKRIPSI**

Oleh:  
**RIZHA ALFIANITA**  
**NIM. 200605110177**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 31 Mei 2024

Pembimbing I,

  
Dr. Cahyo Crysdian, MCS  
NIP. 19740424 200901 1 008

Pembimbing II,

  
Dr. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Fachrul Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

**HALAMAN PENGESAHAN**

**DETEKSI PERTANYAAN *INSINCERE* PADA KOMUNITAS TANYA  
JAWAB *ONLINE* MENGGUNAKAN METODE *MULTINOMIAL  
NAÏVE BAYES***

**SKRIPSI**

Oleh:  
**RIZHA ALFIANITA**  
**NIM. 200605110177**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal: 7 Juni 2024

**Susunan Dewan Penguji**

Ketua Penguji : Dr. M. Amin Hariyadi, M.T  
NIP. 19670118 200501 1 001

Anggota Penguji I : A'la Syauqi, M.Kom  
N'IP. 19771201 200801 1 007

Anggota Penguji II : Dr. Cahyo Crysdian, MCS  
NIP. 19740424 200901 1 008

Anggota Penguji III : Dr. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

(  )  
(  )  
(  )  
(  )

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Fachrul Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : RIZHA ALFIANITA  
NIM : 200605110177  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : DETEKSI PERTANYAAN *INSINCERE* PADA  
KOMUNITAS TANYA JAWAB *ONLINE*  
MENGUNAKAN METODE *MULTINOMIAL*  
*NAÏVE BAYES*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 7 Juni 2024

Yang membuat pernyataan,



RIZHA ALFIANITA  
NIM. 200605110177

## **MOTTO**

*~ It always seems impossible until it's done ~*  
*~ Keliatannya emang susah, tapi kerjain aja dulu pasti juga selesai ~*

## **HALAMAN PERSEMBAHAN**

Dengan rasa syukur, saya dedikasikan karya ini kepada:

Diri saya sendiri, untuk usaha dan ketekunan yang telah saya curahkan.

Keluarga saya, yang selalu memberikan dukungan tanpa henti.

Para dosen, yang telah membimbing dan memberikan ilmu.

Rekan-rekan, yang telah berkontribusi melalui dukungan dan kebersamaan.

## KATA PENGANTAR

*Assalamualaikum wr wb.*

Puji syukur penulis panjatkan kehadirat Allah Swt. atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian ini yang berjudul “Deteksi Pertanyaan *Insincere* pada Komunitas Tanya *Online* Menggunakan Metode *Multinomial Naïve Bayes*”. Dalam penyusunan karya ini, penulis telah mendapatkan bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Prof. Dr. H.M. Zainuddin, MA., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Cahyo Crysdiyan, MCS selaku Dosen Pembimbing I yang telah membimbing dan memberikan arahan serta motivasi dalam proses penelitian dan penyusunan skripsi ini dari awal hingga selesai.
5. Dr. M. Imamudin Lc, MA selaku Dosen Pembimbing II yang telah memberikan banyak dukungan dan masukan selama proses penulisan sehingga karya ini menjadi lebih baik.
6. Dr. M. Amin Hariyadi, M.T selaku Ketua Penguji dan A’la Syauqi, M.Kom selaku Dosen Penguji II yang telah meluangkan waktunya untuk menguji serta memberikan arahan dalam penulisan skripsi ini.

7. Segenap dosen dan jajaran staf program studi Teknik Informatika yang telah memberikan banyak ilmu dan arahan dalam skripsi ini.
8. Kedua orang tua penulis yang penulis sangat sayangi. Ibu Sri Wahyuni yang senantiasa mendoakan dan memberikan dukungan baik moral, motivasi, ataupun materi serta kasih sayang sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Almarhum Bapak Mualik yang selalu menjadi inspirasi dan semangat dalam setiap langkah penulis menempuh pendidikan hingga saat ini.
9. Kakak-kakak Elisa Ekayanti dan Rezky Alfian Maliq yang selalu memberikan perhatian, saran, serta kesabaran dalam mendukung penulis.
10. Adik keponakan Aksara Janitra Maliq yang secara tidak langsung selalu menghibur penulis.
11. Teman-teman dekat yang sudah penulis anggap sebagai saudara sendiri, Maharini Nabela Ayuningsih, Alfania Dwi Safitri, Thoyibatul Aita, dan Maharani Aulia Ayuningrum.
12. Fahrendra Khoirul Ihtada selaku teman karib dan rekan seperjuangan penulis, beserta Hikmatul Maulidia, Mohammad Yoga Pratama, dan Zul Iflah Al Juhaeda yang selalu menjadi sobat penunggu Upajiwo dan Kidjang bersama penulis.
13. Para penghuni kos Joyosuko 61A, Tegar, Adam, Ilham, Aji, dan Daniyal yang selalu mengejutkan penulis dengan segala gebrakannya.

14. Teman-teman Angkatan 2020 Teknik Informatika "INTEGER" yang telah memberikan dukungan, semangat, dan kebersamaan selama masa studi dan penyusunan skripsi ini.
15. Penulis sendiri yang telah berusaha dengan keras dalam memahami semua materi yang sebelumnya belum familiar hingga akhirnya menjadi bisa dan mampu menyelesaikan skripsi ini.
16. Seluruh pihak dan teman-teman penulis lainnya yang tidak dapat penulis sebutkan satu persatu, yang secara tidak langsung juga turut serta memberi dukungan dan membantu penulis dalam pengerjaan skripsi ini,

Sebagai penutup, penulis mengakui bahwa skripsi ini masih terdapat banyak kekurangan. Penulis berharap bahwa skripsi ini dapat membawa manfaat bagi para pembaca dan tentunya penulis sendiri. *Aamiin yaa robbal 'alamin.*

*Wassalamualaikum Warahmatullahi Wabarakatuh.*

Malang, 7 Juni 2024

Penulis

## DAFTAR ISI

<b>HALAMAN PERSETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>ABSTRAK .....</b>	<b>xv</b>
<b>ABSTRACT .....</b>	<b>xvi</b>
<b>المخلص .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>18</b>
1.1 Latar Belakang .....	18
1.2 Pernyataan Masalah .....	23
1.3 Tujuan Penelitian .....	23
1.4 Batasan Masalah .....	23
1.5 Manfaat Penelitian .....	24
<b>BAB II STUDI PUSTAKA .....</b>	<b>25</b>
2.1 Permasalahan Komunitas Tanya Jawab <i>Online</i> .....	25
2.2 Klasifikasi Pertanyaan.....	27
<b>BAB III DESAIN DAN IMPLEMENTASI .....</b>	<b>31</b>
3.1 Akuisisi Data.....	31
3.2 Desain Sistem.....	31
3.3 <i>Text Preprocessing</i> .....	32
3.3.1 <i>Case Folding</i> .....	33
3.3.2 <i>Cleansing</i> .....	34
3.3.3 <i>Tokenization</i> .....	35
3.3.4 <i>Stopword Removal</i> .....	36
3.3.5 <i>Lemmatization</i> .....	37
3.4 Ekstraksi Fitur TF-IDF.....	38
3.5 <i>Multinomial Naïve Bayes</i> .....	41
<b>BAB IV UJI COBA DAN PEMBAHASAN .....</b>	<b>46</b>
4.1 Skenario Pengujian .....	46
4.1 Hasil Uji Coba.....	50
4.2.1 Hasil Skenario ke-1 .....	50
4.2.2 Hasil Skenario ke-2 .....	53
4.2.3 Hasil Skenario ke-3 .....	56
4.2.4 Hasil Skenario ke-4 .....	59
4.3 Pembahasan.....	62
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>81</b>

5.1 Kesimpulan .....	81
5.2 Saran.....	82
<b>DAFTAR PUSTAKA</b>	

## DAFTAR TABEL

Tabel 4. 1 Sampel Dataset.....	46
Tabel 4. 2 Skenario Pengujian .....	48
Tabel 4. 3 <i>Confusion Matrix</i> .....	49
Tabel 4. 4 Hasil Skenario ke-1 .....	52
Tabel 4. 5 Hasil Skenario ke-2.....	55
Tabel 4. 6 Hasil skenario ke-3.....	58
Tabel 4. 7 Hasil Skenario ke-4 .....	61
Tabel 4. 8 Hasil Uji Coba.....	62
Tabel 4. 9 Hasil <i>K-fold Cross Validation</i> .....	74
Tabel 4. 10 Sampel Prediksi <i>Error</i> .....	74
Tabel 4. 11 Ranking Skor TF-IDF pada Kelas <i>Sincere</i> .....	76
Tabel 4. 12 Ranking Skor TF-IDF pada Kelas <i>Insincere</i> .....	76

## DAFTAR GAMBAR

Gambar 3. 1 Desain Sistem.....	32
Gambar 3. 2 Diagram Alur <i>Text Preprocessing</i> .....	33
Gambar 3. 3 <i>Flowchart</i> Tahap <i>Case Folding</i> .....	34
Gambar 3. 4 <i>Flowchart</i> Tahap <i>Cleansing</i> .....	35
Gambar 3. 5 <i>Flowchart</i> Tahap <i>Tokenization</i> .....	36
Gambar 3. 6 <i>Flowchart</i> Tahap <i>Stopword Removal</i> .....	37
Gambar 3. 7 <i>Flowchart</i> Tahap <i>Lemmatization</i> .....	38
Gambar 3. 8 <i>Flowchart</i> Perhitungan TF-IDF .....	40
Gambar 4. 1 Distribusi Kelas Aktual pada Skenario ke-1 .....	51
Gambar 4. 2 Distribusi Kelas Aktual pada Skenario ke-1 .....	51
Gambar 4. 3 <i>Confusion Matrix</i> Skenario ke-1 .....	52
Gambar 4. 4 Distribusi Kelas Aktual pada Skenario ke-2 .....	54
Gambar 4. 5 Distribusi Kelas Prediksi pada Skenario ke-2 .....	54
Gambar 4. 6 <i>Confusion Matrix</i> Skenario ke-2 .....	55
Gambar 4. 7 Distribusi Kelas Prediksi pada Skenario ke-3 .....	57
Gambar 4. 8 Distribusi Kelas Prediksi pada Skenario ke-3 .....	57
Gambar 4. 9 <i>Confusion Matrix</i> Skenario ke-3 .....	58
Gambar 4. 10 Distribusi Kelas Aktual pada Skenario ke-4 .....	60
Gambar 4. 11 Distribusi Kelas Prediksi pada Skenario ke-4.....	60
Gambar 4. 12 <i>Confusion Matrix</i> Skenario ke-4 .....	61
Gambar 4. 13 Perbandingan Metrik <i>Accuracy</i> .....	65
Gambar 4. 14 Perbandingan Metrik <i>Precision</i> .....	66
Gambar 4. 15 Perbandingan Metrik <i>Recall</i> .....	67
Gambar 4. 16 Perbandingan Metrik <i>F1-score</i> .....	68
Gambar 4. 17 Nilai <i>Accuracy</i> pada <i>K-fold Cross Validation</i> Skenario ke-1 .....	70
Gambar 4. 18 Nilai <i>Accuracy</i> pada <i>K-fold Cross Validation</i> Skenario ke-2 .....	71
Gambar 4. 19 Nilai <i>Accuracy</i> pada <i>K-fold Cross Validation</i> Skenario 8:2 .....	72
Gambar 4. 20 Nilai <i>Accuracy</i> pada <i>K-fold Cross Validation</i> Skenario 9:1 .....	73
Gambar 4. 21 <i>Wordcloud</i> dari Kata-kata yang Dominan pada Kelas <i>Sincere</i> .....	76
Gambar 4. 22 <i>Wordcloud</i> dari Kata-kata yang Dominan pada Kelas <i>Insincere</i> ...	77

## ABSTRAK

Alfianita, Rizha. 2024. **Deteksi Pertanyaan *Insincere* pada Komunitas Tanya Jawab Online Menggunakan *Multinomial Naïve Bayes***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Cahyo Crys dian, MCS (II) Dr. M. Imamudin Lc, MA.

**Kata kunci:** *Deteksi, Pertanyaan Insincere, Multinomial Naïve Bayes, TF-IDF*

Terdapat beragam kemudahan yang ditawarkan oleh platform komunitas tanya jawab *online*. Platform ini memungkinkan pengguna mengajukan pertanyaan langsung daripada harus mencarinya melalui mesin pencari sehingga dapat menghemat waktu. Alih-alih menelusuri ribuan situs, pengguna dapat langsung mendapatkan jawaban yang relevan. Namun, di samping banyaknya kemudahan yang ditawarkan, platform ini juga menghadapi beberapa tantangan. Salah satunya adalah munculnya pertanyaan *insincere*, seperti yang saat ini dihadapi oleh Quora. Pertanyaan-pertanyaan *insincere* ini ditandai dengan beberapa karakteristik, diantaranya memiliki nada yang tidak netral, bersifat meremehkan atau menghasut, tidak berdasarkan kenyataan, dan menggunakan konten seksual. Penelitian ini menggunakan *Multinomial Naïve Bayes* untuk mendeteksi pertanyaan *insincere* tersebut. Performa sistem diukur dengan menggunakan *confusion matrix* meliputi metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil terbaik didapatkan pada rasio data 90% data latih dan 10% data uji dengan nilai *accuracy* 0.83, *precision* 0.76, *recall* 0.94, dan *F1-score* 0.84. Model divalidasi menggunakan *k-fold cross validation* pada  $k=10$  yang menghasilkan rata-rata *accuracy* 0.82. Analisis kata menggunakan TF-IDF dilakukan untuk menemukan kata-kata yang merepresentasikan masing-masing kelas. Hasil analisis menunjukkan bahwa kelas *sincere* didominasi oleh kata 'best', 'get', 'good', dan 'like', sementara kelas *insincere* ditandai dengan kata 'people', 'woman', 'trump', dan 'muslim'. Penelitian ini juga menunjukkan bahwa semakin banyak data latih yang digunakan maka performa model menjadi semakin meningkat. Diharapkan penelitian ini dapat membantu mengurangi konten *toxic* seperti pertanyaan *insincere* sebagai upaya peningkatan kualitas platform komunitas tanya jawab *online*.

## ABSTRACT

Alfianita, Rizha. 2024. **The Detection of Insincere Questions in Online Question-and-Answer Community Using Multinomial Naïve Bayes.** Undergraduate Thesis. Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisor: (I) Dr. Cahyo Crys dian, MCS (II) Dr. M. Imamudin Lc, MA.

Online question-and-answer community platform offers ease. It enables the users to directly ask questions rather than browsing them using search engines to save time. Instead of browsing thousands of sites, users can directly get relevant answers. However, besides its ease, the platform also faces challenges, one of which is insincere questions. They often come to Quora nowadays. The question is marked by some characteristics, such as using a one-sided tone, disdain and provocative, unrealistic, and sexual content. The research employs Multinomial Naïve Bayes to detect these insincere questions. The system performance is measured using confusion matrixes, including accuracy, precision, recall, and F1-score. The best result is achieved using a ratio of 90% of training data and 10% of testing data with accuracy, precision, recall, and F1 scores are 0.83, 0.76, 0.94, and 0.84, respectively. The model is verified using k-fold cross-validation at k=10, leading to an accuracy average of 0.82. Word analysis using TF-IDF aims to search for words representing each class. The analysis shows that the sincere class is dominated by the words 'best', 'get', 'good', and 'like'. Meanwhile, the insincere class is indicated with the word 'people', 'woman', 'trump', and 'muslim'. The research also shows that the more training data used, the more the model performance increases. It is expected to help reduce toxic content, like insincere questions, to improve the quality of online question-and-answer platforms.

**Keywords:** *Detection, Insincere Questions, Multinomial Naïve Bayes, TF-IDF*

## الملخص

ألفيانيتا، ريزا. 2024. الكشف عن الأسئلة غير الصادقة في مجتمعات الأسئلة والأجوبة عبر الإنترنت باستخدام المصنف البايزي الساذج متعدد الحدود. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د جاهيو كريسديان، الماجستير. المشرف الثاني: د. محمد إمام الدين، الماجستير.

**الكلمات الرئيسية:** كشف، أسئلة غير صادقة، مصنف بايزي ساذج متعدد الحدود، TF-IDF.

هناك عدة وسائل الراحة التي توفرها منصات الأسئلة والأجوبة المجتمعية عبر الإنترنت. تتيح المنصة للمستخدمين طرح الأسئلة مباشرة بدلا من الاضطرار إلى البحث عنها من خلال محركات البحث وبالتالي توفير الوقت. بدلا من تصفح آلاف المواقع، يمكن للمستخدمين الحصول على إجابات ذات صلة على الفور. ومع ذلك، بالإضافة إلى عدة وسائل الراحة المتوفرة، تواجه المنصة أيضا بعض التحديات. أحدها هو ظهور أسئلة جوهريّة، مثل تلك التي يواجهها Quora حاليا. تتميز هذه الأسئلة غير الصادقة بعدة خصائص، بما في ذلك وجود نغمة غير محايدة أو مهينة أو تحريضية، ولا تستند إلى الواقع، واستخدام المحتوى الجنسي. استخدم هذا البحث مصنف بايزي ساذج متعدد الحدود للكشف عن هذه الأسئلة غير الصادقة. تم قياس أداء النظام باستخدام مصفوفة الارتباك بما في ذلك الدقة والثبات والاستدعاء وقيمة ف1. تم الحصول على أفضل النتائج في نسبة البيانات 90٪ من بيانات التدريب و 10٪ من بيانات الاختبار بقيمة دقة 0.83، ثبات 0.76، استدعاء 0.94، وقيمة ف1 0.84. تم التحقق من صحة النموذج باستخدام التحقق المتقاطع k-fold عند  $k = 10$  مما أدى إلى متوسط دقة 0.82. تم إجراء تحليل الكلمات باستخدام TF-IDF للعثور على الكلمات التي تمثل كل فئة. أظهرت نتائج التحليل أن الطبقة (sincere) المخلصة تهيمن عليها كلمات "best" و "get" و "good" و "like"، بينما تتميز الطبقة (insincere) غير المخلصة بكلمات "people" و "woman" و "trump" و "muslim". أظهرت أيضا كلما تم استخدام المزيد من بيانات التدريب، أصبح أداء النموذج أفضل. من المأمول أن يساعد هذا البحث في تقليل المحتوى السام (toxic) مثل الأسئلة غير الصادقة كمحاولة لتحسين جودة منصات الأسئلة والأجوبة المجتمعية عبر الإنترنت.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Komunitas tanya jawab *online* merupakan suatu platform yang digunakan sebagai sarana tanya-jawab secara daring mengenai berbagai macam topik. Pada komunitas tanya jawab *online*, pengguna dibagi menjadi tiga jenis, yaitu pengguna yang menanyakan pertanyaan, pengguna yang menjawab pertanyaan, dan pengguna yang menanyakan pertanyaan melalui pertanyaan dan jawaban dari pengguna lain (Priyambowo & Adriani, 2019). Pengguna yang mengajukan pertanyaan adalah pengguna yang tidak memiliki pengetahuan terhadap topik tertentu sehingga mencari ahli yang dapat memberikan pengetahuannya. Sementara itu, para ahli menjadi sumber informasi yang menggantikan sumber data, seperti dokumen atau basis data, berdasarkan topik yang diajukan oleh pengguna (Baltadzhieva & Chrupala, 2015).

Terdapat beragam kemudahan yang ditawarkan oleh platform komunitas tanya jawab *online*. Platform ini memfasilitasi individu untuk langsung mengajukan pertanyaan mereka daripada harus mencarinya melalui mesin pencari. Menggunakan platform komunitas tanya jawab *online* dapat menghemat waktu pengguna dalam melakukan pencarian jawaban mereka. Jika mesin pencari mengharuskan pengguna untuk menelusuri ribuan bahkan jutaan situs, platform komunitas tanya jawab *online* dapat langsung menemukan jawaban yang relevan

dengan pertanyaan yang pengguna ajukan (Menaha *et al.*, 2021). Di samping itu, platform ini memungkinkan pengguna untuk bertanya dalam berbagai macam topik, seperti teknologi, *life style* hingga pengalaman hidup. Pengguna juga dapat memposting pertanyaan kapanpun dan akan dijawab oleh pengguna lain yang ahli atau memiliki pengetahuan dalam topik yang diajukan.

Meskipun komunitas tanya jawab *online* memiliki banyak kemudahan yang diberikan, platform ini juga menghadapi beberapa masalah. Pada platform komunitas tanya jawab *online* seperti Quora, sebagian besar penggunanya memang bermaksud untuk mengajukan pertanyaan yang mereka benar-benar tertarik. Namun, dalam beberapa kasus, seseorang akan mengajukan pertanyaan yang provokatif dengan tujuan untuk membuat pernyataannya sendiri atau tidak dimaksudkan untuk mencari jawaban dimana pertanyaan tersebut mungkin saja bersifat menghasut, menjebak, atau memecah belah suatu kelompok. Pertanyaan semacam ini disebut sebagai pertanyaan *insincere* (Gaire *et al.*, 2019).

Pada komunitas tanya jawab *online*, pertanyaan *insincere* mengacu pada pertanyaan yang memiliki niat tidak tulus. Karakteristik dari pertanyaan *insincere* ini berupa pertanyaan yang memiliki maksud tersembunyi untuk menghasut atau merendahkan sekelompok orang. Pertanyaan-pertanyaan tersebut biasanya didasarkan pada asumsi-asumsi yang keliru atau memiliki tujuan untuk membuat konflik daripada mencari jawaban yang membantu. Pertanyaan *insincere* juga dapat mencakup pertanyaan yang mengandung konten seksual. Pertanyaan-pertanyaan *insincere* ini termasuk dalam konten *toxic* yang penting untuk diidentifikasi demi menjaga kualitas dan keberlanjutan platform komunitas tanya-jawab.

Karakteristik dari pertanyaan *insincere* mencerminkan sikap yang bertentangan dengan ajaran Islam, khususnya dalam prinsip etika dan ketulusan dalam berkomunikasi. Hal tersebut sejalan dengan ajaran Islam yang menekankan pentingnya etika dan bertutur kata yang baik dalam berbincang, tercantum dalam Surah Al-Hujurat ayat 11:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا يَسْخَرْ قَوْمٌ مِّنْ قَوْمٍ عَسَىٰ أَن يَكُونُوا خَيْرًا مِّنْهُمْ وَلَا نِسَاءٌ مِّنْ نِّسَاءٍ عَسَىٰ أَن يَكُنَّ خَيْرًا مِّنْهُنَّ وَلَا تَلْمِزُوا أَنفُسَكُمْ وَلَا تَنَابَزُوا بِالْألقَابِ بِئْسَ الإِسْمُ الفُسُوقُ بَعْدَ الإِيمَانِ وَمَن لَّمْ يَتُبْ فَأُولَئِكَ هُمُ الظَّالِمُونَ

*“Wahai orang-orang yang beriman! Janganlah suatu kaum mengolok-olok kaum yang lain, (karena) boleh jadi mereka (yang diperolok-olokkan) lebih baik dari mereka (yang mengolok-olok), dan jangan pula perempuan-perempuan (mengolok-olokkan) perempuan lain, (karena) boleh jadi perempuan (yang diperolok-olokkan) lebih baik dari perempuan (yang mengolok-olok). Janganlah kamu saling mencela satu sama lain, dan janganlah saling memanggil dengan gelar-gelar yang buruk. Seburuk-buruk panggilan adalah (panggilan) yang buruk (fasik) setelah beriman. Dan barangsiapa tidak bertobat, maka mereka itulah orang-orang yang zalim.” (QS. Al-Hujurat 49:11).*

Surah Al-Hujurat ayat 11 menjelaskan tentang tindakan yang harus dihindari ketika berinteraksi dengan sesama manusia. Ayat tersebut mengajarkan untuk tidak mencemooh atau menghina orang lain dan menghindari panggilan-panggilan buruk. Hal ini merupakan pengingat dalam Islam untuk menjaga martabat orang lain, menghindari perilaku yang merugikan, dan menjalani komunikasi dengan etika yang baik (Lutfi, 2020).

Deteksi pertanyaan *insincere* dapat dilakukan dengan menggunakan suatu teknik, yaitu klasifikasi teks. Klasifikasi teks merupakan suatu teknik untuk mengatur dokumen teks ke dalam berbagai kategori berdasarkan ke dalam satu atau lebih kategori yang telah ditentukan sebelumnya (Mutawalli *et al.*, 2019). Teknik

ini dapat digunakan untuk berbagai macam kasus, termasuk deteksi email spam ataupun klasifikasi berita. Jika jumlah dokumen teks yang ada sedikit, maka memungkinkan untuk melakukan klasifikasi secara manual dengan membaca dan mengidentifikasi jenis teks tersebut. Namun, akan menjadi semakin sulit jika jumlah teks yang akan diklasifikasikan mencapai ribuan bahkan jutaan (Sarkar, 2016). Penerapan ekstraksi fitur dan *machine learning* khususnya *supervised learning* sangat berguna untuk dapat mengklasifikasikan pertanyaan *insincere* secara cepat dan otomatis.

Diperlukannya sistem klasifikasi untuk mendeteksi pertanyaan *insincere* (tidak tulus) sesuai dengan firman Allah Swt. yang tertuang pada QS. Al-Baqarah ayat 42 yang berbunyi:

وَلَا تَلْبِسُوا الْحَقَّ بِالْبَاطِلِ وَتَكْتُمُوا الْحَقَّ وَأَنْتُمْ تَعْلَمُونَ

“Jangan kalian mencampur kebenaran dengan kebatilan. Jangan juga kalian menyembunyikan kebenaran. Padahal kalian menyadarinya,” (QS. Al-Baqarah 2:42).

Imam Al-Baidhawi berkata “Jangan kalian membuat sebuah kebenaran menjadi samar karena bercampur dengan kebatilan yang kalian tulis di sela-selanya atau kalian sebutkan dalam ta’wilnya”. Prinsip bahwa yang *haq* dan *bathil* tidak boleh dicampuradukkan harus dipegang teguh. Sejalan dengan konsep ini, pertanyaan-pertanyaan yang terdapat pada komunitas tanya jawab *online* perlu dipisahkan mana pertanyaan yang tulus dan tidak tulus.

Klasifikasi teks dapat dilakukan dengan berbagai metode, salah satunya adalah *Multinomial Naïve Bayes*. Algoritma *Multinomial Naïve Bayes* merupakan

salah satu varian dari metode *Naïve Bayes*, yang termasuk dalam ranah *machine learning*. Metode ini didasarkan pada konsep *Teorema Bayes* dengan asumsi bahwa semua fitur yang digunakan untuk klasifikasi adalah independent (Jiang *et al.*, 2011). *Multinomial Naïve Bayes* telah banyak digunakan dalam penelitian klasifikasi teks, contohnya penelitian oleh Yuyun *et al.*, (2021) yang menggunakannya untuk menganalisis sentimen *tweet* tentang penanganan Covid-19 oleh pemerintah dan mencapai akurasi klasifikasi sebesar 74%. Dataset yang digunakan berasal dari Twitter dimana terdapat sebanyak 2000 komentar dengan keluaran tiga label kelas, yaitu negatif, positif, dan netral. Ekstraksi fitur TF-IDF juga diimplementasikan pada model ini dan menghasilkan akurasi sebesar 74%. Penelitian menunjukkan bahwa metode *Multinomial Naïve Bayes* yang diusulkan memiliki tingkatan *fair classification* atau cukup baik.

Yusliani *et al.*, (2022) juga melakukan penelitian dengan mengimplementasikan metode *Multinomial Naïve Bayes* untuk klasifikasi pertanyaan berbahasa Indonesia. Dataset yang digunakan terdiri dari 1.195 pertanyaan dengan label *factoid*, *non-factoid*, dan *other*. Peneliti menerapkan metode *Chi-Square* untuk melakukan seleksi fitur dimana menghasilkan peningkatan pada *accuracy* sebesar 10% dari 77% menjadi 87%, serta *precision*, *recall*, dan *f-measure* mengalami juga mengalami peningkatan. Jumlah fitur yang digunakan berkurang dari 1374 menjadi 248 fitur. Penelitian ini menunjukkan bahwa kombinasi *Multinomial Naïve Bayes* dan *Chi-Square* meningkatkan performa pada sistem klasifikasi.

Berdasarkan beberapa penelitian yang telah ada, metode *Multinomial Naïve Bayes* memiliki performa yang baik dalam tugas analisis sentimen dan klasifikasi pertanyaan. Peneliti mengusulkan implementasi metode *Multinomial Naïve Bayes* ini untuk menyelesaikan permasalahan pertanyaan *insincere* pada komunitas tanya jawab *online*. Model yang dihasilkan dari penelitian ini diharapkan dapat membantu dalam mengidentifikasi pertanyaan *insincere* dengan cepat dan lebih akurat sehingga dapat menciptakan lingkungan tanya jawab yang aman dan bermanfaat bagi pengguna, serta meningkatkan integritas dari platform tanya jawab dengan mengurangi jumlah pertanyaan *insincere*.

## 1.2 Pernyataan Masalah

Seberapa baik performa metode *Multinomial Naïve Bayes* untuk membedakan pertanyaan *insincere* pada komunitas tanya jawab *online*?

## 1.3 Tujuan Penelitian

Mengukur performa sistem dengan menerapkan metode *Multinomial Naïve Bayes* untuk memisahkan pertanyaan *sincere* dan *insincere* dalam teks pada komunitas tanya jawab *online*.

## 1.4 Batasan Masalah

- a. Data penelitian ini merupakan data sekunder yang berasal dari Kaggle dan berisi kumpulan kalimat pertanyaan dari situs Quora.
- b. Bahasa yang digunakan pada penelitian ini terbatas pada Bahasa Inggris.

## 1.5 Manfaat Penelitian

- a. Bagi Quora, penelitian ini dapat membantu mendeteksi pertanyaan *insincere* yang berpotensi merusak keamanan dan integritas dari platform serta meningkatkan reputasi dengan mengurangi konten *insincere*.
- b. Bagi pengguna Quora, adanya sistem deteksi pertanyaan ini membuat pengguna dengan cepat membedakan mana yang termasuk pertanyaan yang *sincere* dan *insincere* sehingga dapat meningkatkan kualitas pengalaman mereka dalam menggunakan platform.

## **BAB II**

### **STUDI PUSTAKA**

#### **2.1 Permasalahan Komunitas Tanya Jawab *Online***

Platform komunitas tanya-jawab atau *Community-based Question Answering* (CQA) menyimpan berbagai macam pertanyaan dan jawaban serta menyediakan informasi yang tidak bisa dengan mudah diakses melalui mesin pencarian (Riahi *et al.*, 2012). CQA membawa banyak dampak positif terhadap penggunanya, seperti memudahkan pengguna dalam mencari jawaban pada pertanyaan-pertanyaan yang ada sehingga dapat menghemat waktu dan usaha mereka. Meskipun memiliki banyak dampak positif, platform CQA ini juga mengalami beberapa tantangan, seperti menyaring konten *toxic*, menghapus pertanyaan duplikat, hingga mengidentifikasi pengguna yang ahli (Roy, 2020).

Platform CQA memiliki masalah dalam rendahnya keikutsertaan pengguna dalam tanya-jawab. Kebanyakan pengguna tidak menjawab pertanyaan pengguna lain dan jumlah pengguna yang menjawab pertanyaan lebih rendah dari yang mengajukan pertanyaan. Beberapa pengguna juga menjadi kesulitan untuk mencari pertanyaan yang mereka ahli di dalamnya karena jumlah pertanyaan yang membludak (Ghasemi *et al.*, 2021). Oleh karena itu, terjadi perbedaan yang cukup besar antara pertanyaan yang tidak terjawab dan jawaban yang memungkinkan (Neshati *et al.*, 2017).

Banyaknya pertanyaan-pertanyaan yang ada membuat pengguna semakin sulit untuk menemukan pertanyaan yang dapat mereka jawab. Ditambah lagi jika

muncul konten-konten *toxic* yang dapat mengaburkan atau menyembunyikan pertanyaan-pertanyaan yang relevan dan bernilai. Konten *toxic* adalah ancaman serius yang dihadapi oleh 48% pengguna internet. Beberapa contohnya, media sosial seperti Facebook, Twitter, dan Youtube saat ini menghadapi permasalahan dari beberapa pengguna tidak bertanggung jawab yang mencoba merusak integritas forum tersebut (Roy *et al.*, 2018). Platform komunitas tanya jawab (CQA) seperti Quora tidak menjadi pengecualian dari masalah ini. Tidak sedikit pengguna membuat postingan *toxic* yang menyerang suatu pihak. Dalam konteks platform Quora, konten *toxic* salah satu jenisnya adalah pertanyaan *insincere*.

Quora telah mengkategorikan karakteristik dari pertanyaan *insincere*. Pertanyaan *insincere* merupakan pertanyaan yang didasarkan pada asumsi-asumsi yang salah atau yang dimaksudkan untuk membuat suatu pernyataan daripada mencari jawaban yang membantu. Lebih detail, pertanyaan dapat disebut *insincere* jika ditandai dengan beberapa karakteristik, diantaranya memiliki nada yang tidak netral, bersifat meremehkan atau menghasut, tidak berdasarkan kenyataan, dan menggunakan konten seksual. Nada yang tidak netral diartikan sebagai pertanyaan yang bernada berlebihan untuk menggarisbawahi suatu poin dengan tujuan menjatuhkan sekelompok orang dan dimaksudkan sebagai pernyataan untuk melawan kelompok tertentu. Bersifat meremehkan atau menghasut berarti pertanyaan yang mendorong tindakan diskriminatif terhadap suatu kelompok. Tidak didasarkan pada kenyataan artinya pertanyaan yang berdasarkan informasi yang salah atau mengandung asumsi yang tidak masuk akal. Mengandung konten seksual maksudnya adalah pertanyaan yang menggunakan konten seksual, seperti

pedofilia, inses, dan bestialitas (tindakan seksual yang menyimpang) dan tidak dimaksudkan untuk mencari jawaban yang benar (Kim *et al.*, 2019).

## 2.2 Klasifikasi Pertanyaan

Pada komunitas tanya-jawab (CQA), salah satu aspek krusial dalam meningkatkan keefektifan suatu platform adalah kemampuan untuk mengklasifikasikan pertanyaan secara otomatis sesuai dengan label yang ditetapkan (Bae & Ko, 2019). Tujuan klasifikasi pertanyaan adalah untuk menyarankan kategori yang paling tepat untuk pertanyaan akan yang diajukan pengguna, memudahkan dalam mendapatkan tanggapan dari pengguna lain, dan memberikan rekomendasi untuk pertanyaan yang relevan dalam kategori tersebut.

Klasifikasi pertanyaan juga memiliki peran penting untuk menyaring konten pertanyaan yang berkualitas rendah, seperti pertanyaan *insincere*. CQA memberikan fasilitas untuk pengguna menjawab pertanyaan dari pengguna lain dengan jawaban yang bervariasi sesuai masing-masing pengguna. Namun, pengguna mungkin saja tidak dapat secara langsung menemukan jawaban yang diinginkannya. Ditambah lagi, jika terdapat banyak pertanyaan yang bernada tidak netral, berbau konten seksual, ataupun mendiskriminasi suatu kelompok atau disebut dengan pertanyaan *insincere* tentunya dapat menambah upaya lebih besar dari pengguna untuk menemukan jawaban yang diinginkannya (Mishra & Kumar, 2023). Dalam konteks ini, teknik klasifikasi digunakan untuk mengidentifikasi apakah pertanyaan tersebut bersifat *insincere* atau tidak.

Sebuah penelitian terkait klasifikasi pertanyaan telah dilakukan oleh Gottipati *et al.*, (2020) untuk mengidentifikasi konten *toxic* pada komunitas tanya jawab

*online*. Peneliti menggunakan *fastText*, sebuah library berbasis *Neural Network*. Dataset yang digunakan berasal dari Quora, dengan total sebanyak 783 ribu data. Namun, jumlah data untuk masing-masing label, yaitu *sincere* dan *insincere* tidak seimbang sehingga peneliti perlu melakukan *up-sampling*. Pemisahan data dilakukan dengan rasio 60-20-20, yang merepresentasikan 60% untuk *training*, 20% untuk *testing*, dan 20% untuk validasi. Peneliti menggunakan *F1-score* untuk menguji performa model yang menghasilkan nilai sebesar 74% untuk model *fastText* ini. Peneliti juga mencoba 3 model lain sebagai perbandingan, yaitu *Naive Bayes*, *Logistic Regression*, dan *Stochastic Gradient Descent (SGD)* dengan nilai *F1-score* berturut-turut 35%, 59%, dan 46%.

Al Qodrin *et al.*, (2022) melakukan penelitian mengenai klasifikasi pertanyaan berbahasa Indonesia dengan menggunakan algoritma *Support Vector Machine (SVM)* dan seleksi fitur *mutual information*. Data yang digunakan adalah data sekunder berupa kumpulan kalimat pertanyaan berbahasa Indonesia. Kalimat pertanyaan tersebut nantinya dikategorikan menjadi *factoid*, *non-factoid*, dan *others*. Tahap *preprocessing* untuk data pada penelitian ini melalui proses *case folding*, *noise removal*, dan *tokenizing*. Data dipisahkan dengan metode *cross-validation*, lalu dilakukan seleksi fitur menggunakan *mutual information* dan ekstraksi fitur kata menggunakan TF-IDF. Hasil TF-IDF digunakan untuk klasifikasi menggunakan SVM. Pengujian dilakukan dengan 10 *k-fold cross validation*, dengan hasil terbaik adalah kombinasi seleksi fitur *mutual information* dan SVM kernel *linear*, mencapai *accuracy* 92%, *precision* 93%, dan nilai *f-measure* 90%.

Puspita *et al.*, (2021) juga mengerjakan penelitian terkait klasifikasi pertanyaan soal kuis matakuliah *data warehouse* dan *business intelligence* pada Universitas Telkom. Data diperoleh dari kumpulan data kuis dari 4 kelas mahasiswa S1 angkatan 2018 Universitas Telkom dengan total 3.135 soal kuis. Kemudian, soal kuis tersebut dilakukan pembersihan untuk menghilangkan soal yang duplikat ataupun data soal yang tidak relevan sehingga total data yang diperoleh adalah 160 soal kuis. Soal-soal kuis nantinya dikategorikan dengan 4 label, yaitu *business intelligence*, *data warehouse*, pengukuran kerja, dan data analitik. Tahap *pre-processing* melibatkan *case folding*, *tokenizing*, *stop word removal*, dan *stemming*. Selanjutnya, dilakukan ekstraksi fitur pada dokumen menggunakan TF-IDF. Tugas klasifikasi pada penelitian ini menggunakan algoritma *Naïve Bayes* dan SVM dengan metode SMOTE dan *cross validation* untuk mengukur performa kedua algoritma. SMOTE digunakan untuk menyeimbangkan data dengan membuat data tambahan untuk kelas minor. Hasil pengujian menunjukkan bahwa rata-rata akurasi meningkat setelah penerapan SMOTE. Setelah menggunakan SMOTE, akurasi rata-rata *Naïve Bayes* adalah 94.79%, sedangkan SVM mencapai 96.52%.

Putra & Rochmawati, (2021) mengerjakan penelitian terkait klasifikasi pertanyaan pada platform *Stack Overflow*. Penelitian ini menggunakan dataset sebanyak 40.000 dataset yang didapat dari hasil *crawling* situs <https://stackoverflow.com/>. Dataset tersebut kemudian dibagi menjadi 80% data *training* dan 20% *testing*. Data pertanyaan dibagi menjadi 20 kategori dengan total data yang seimbang, yaitu 2000 untuk masing-masing kategori. Setelah dataset diinputkan, sistem akan mengerjakan *preprocessing text*,

diantaranya HTML *decoding*, *lowercase text*, *delete symbol*, dan *filtering*. Data pertanyaan yang telah dibersihkan selanjutnya dilakukan ekstraksi fitur dengan menggunakan TF-IDF. Pada proses TF-IDF, bobot untuk setiap kata diambil dari hasil perkalian dari TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*). Kemudian, fitur yang dihasilkan dari TF-IDF digunakan dalam model klasifikasi *Naïve Bayes*. Model ini dievaluasi dalam empat percobaan. Hasil terbaik ditemukan pada percobaan ke-4 dengan *accuracy*, *precision*, *recall*, dan *F1-score* masing-masing 75%, 75%, 75%, dan 74%.

## **BAB III**

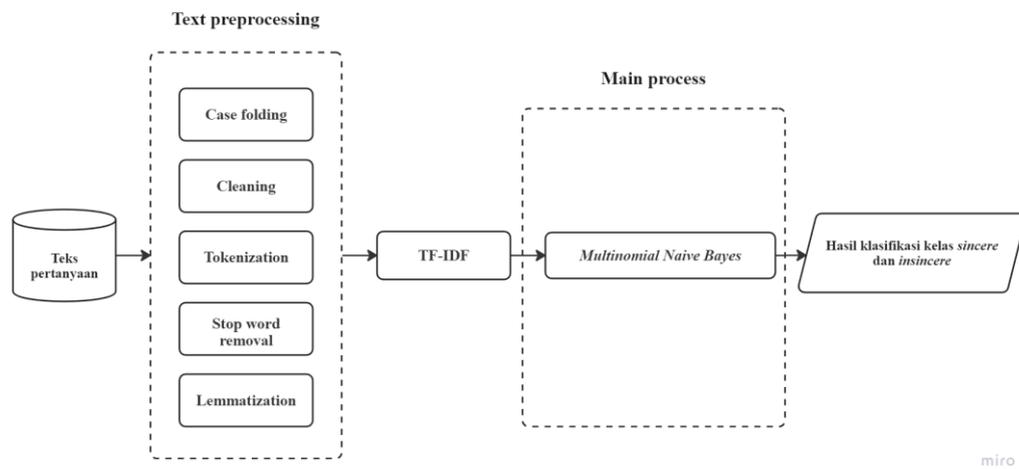
### **DESAIN DAN IMPLEMENTASI**

#### **3.1 Akuisisi Data**

Penelitian ini menggunakan data sekunder yang telah dikumpulkan dari situs kaggle.com, dengan dataset yang bernama *Quora Insincere Questions Classification*. Dataset ini merupakan kumpulan pertanyaan dalam bahasa Inggris yang berasal dari situs Quora. Contoh-contoh pertanyaannya adalah "How difficult is it to find a good instructor to take a class near you?" dan "Did Julius Caesar bring a tyrannosaurus rex on his campaigns to frighten the Celts into submission?". Setiap pertanyaan tersebut dikategorikan ke dalam salah satu dari dua kelas, yaitu *sincere* dan *insincere*. Penting untuk diingat bahwa pelabelan data ini telah dilakukan oleh Quora sendiri sehingga tidak ada pelabelan manual yang dilakukan oleh peneliti.

#### **3.2 Desain Sistem**

Desain sistem merepresentasikan alur dari sistem klasifikasi secara keseluruhan mulai dari input data hingga menghasilkan teks yang telah dikelompokkan ke dalam kategori yang sesuai. Desain sistem yang menggambarkan alur pada penelitian ini ditunjukkan pada Gambar 3.1:



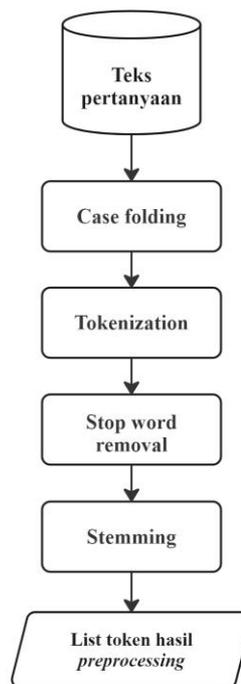
Gambar 3. 1 Desain Sistem

Penelitian ini memulai proses klasifikasi dengan memasukkan data teks pertanyaan sebagai input. Teks tersebut selanjutnya masuk ke dalam tahap *preprocessing* untuk membersihkan dan mempersiapkan teks sebelum diolah lebih lanjut. Setelah itu, setiap kata dalam teks diberi bobot menggunakan TF-IDF. Fitur yang dihasilkan dari TF-IDF kemudian diproses dengan menggunakan metode *Multinomial Naïve Bayes* untuk melakukan klasifikasi. Terdapat 2 kelas yang dihasilkan dari *klasifikasi* ini, yaitu *sincere* dan *insincere*.

### 3.3 Text Preprocessing

Dalam *natural language processing* (NLP), *text preprocessing* merupakan langkah yang penting untuk mempersiapkan permodelan *corpus* dan mempengaruhi hasil dari sistem (Chai, 2023). *Text preprocessing* mengacu pada serangkaian metode yang digunakan untuk mengelola dan mengubah *raw text* yang tidak terstruktur menjadi teks yang lebih terstruktur atau bermakna. Tujuan utamanya adalah untuk mendapatkan wawasan, informasi, atau pengetahuan dari teks. Pada penelitian ini, terdapat berbagai metode yang digunakan di dalam fase

*text preprocessing*, diantaranya *case folding*, *tokenization*, *stop word removal*, dan *stemming* atau *lemmatization* (Anandarajan *et al.*, 2019). Tahapan *text preprocessing* dapat dilihat pada Gambar 3.2.

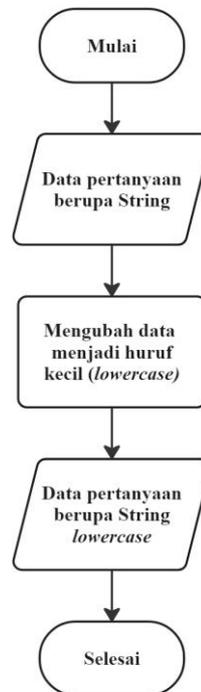


Gambar 3. 2 Diagram Alur *Text Preprocessing*

### 3.3.1 *Case Folding*

*Case folding* adalah suatu tahapan dalam *pre-processing* untuk mengubah teks menjadi huruf kecil (*lowercase*). *Case folding* dilakukan dengan tujuan untuk menyetarakan semua teks sehingga dapat menghindari terjadinya masalah variasi huruf besar yang dapat mempengaruhi akurasi teks. Sementara itu, karakter lain yang bukan termasuk huruf dan angka, seperti tanda baca akan dihilangkan. Sebagai contoh, kalimat “Should I leave my friends and find new one” menjadi “should i leave my friends and find new one?”. Proses *case folding* dimulai dengan menginputkan data berupa *string* pertanyaan. Kemudian, kata tersebut akan melalui

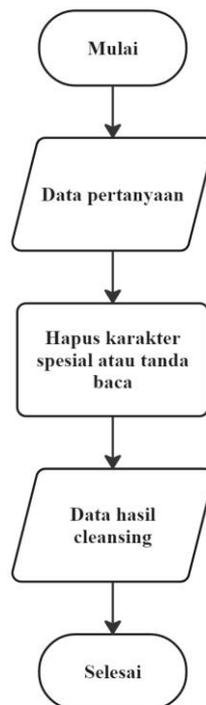
proses *case folding* atau penyetaraan kapitalisasi huruf. Gambar 3.3 menunjukkan tahapan-tahapan dari proses tersebut.



Gambar 3. 3 Flowchart Tahap Case Folding

### 3.3.2 Cleansing

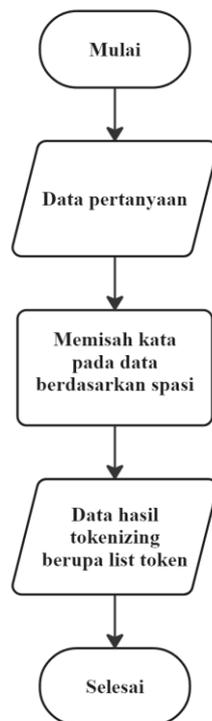
*Cleansing* merupakan tahapan untuk membersihkan karakter-karakter yang tidak perlu, seperti tanda baca, *link*, symbol-simbol (seperti #, @, \$), emoji, dan angka yang tidak memiliki hubungan dengan teks. Sebagai contoh, terdapat teks “should i leave my friends and find new one?” akan dihilangkan beberapa karakter yang tidak perlu, yaitu tanda tanya (?) sehingga menjadi “should i leave my friends and find new one”. Ini juga berlaku untuk karakter-karakter lain yang tidak bermakna. Proses dari tahapan *cleansing* yang ditunjukkan pada Gambar 3.4 dimulai dengan menginputkan data pertanyaan hasil *case folding*. Setelah itu, data teks akan dibersihkan dengan menghapus karakter-karakter yang tidak bermakna.



Gambar 3. 4 *Flowchart Tahap Cleansing*

### 3.3.3 *Tokenization*

*Tokenization* merupakan proses dimana teks yang panjang dipecah-pecah menjadi per kata. Dalam kasus paling sederhana, pemisahan kata dilakukan setiap ada spasi. Misalnya, kalimat “should i leave my friends and find new one” diubah menjadi unit-unit kecil seperti [“should”, “i”, “leave”, “my”, “friends”, “and”, “find”, “new”, “one”]. Proses dari tahapan tokenisasi yang ditunjukkan pada Gambar 3.5 dimulai dengan menginputkan data pertanyaan hasil *cleansing*. Setelah itu, data akan mengalami proses tokenisasi untuk membaginya menjadi token-token kata.



Gambar 3. 5 *Flowchart Tahap Tokenization*

### 3.3.4 *Stopword Removal*

*Stop word removal* berfungsi untuk menghilangkan kata-kata yang sering muncul dalam kalimat sehingga tidak mendefinisikan karakteristik kelas dari kalimat bertujuan untuk menghilangkan kata-kata yang tidak memiliki makna penting dalam sebuah teks. Contoh kata-kata yang termasuk ke dalam *stop words* dalam bahasa Inggris adalah kata hubung (*conjunction*) dan kata ganti (*pronoun*), seperti “and”, “on”, “at”, “to”, “I”, “you”, “he”, “she”, dan sejenisnya. Hal ini dilakukan agar model dapat lebih memusatkan perhatian pada kata-kata yang signifikan dan mengurangi kata-kata yang tidak memiliki arti khusus.. Misalnya, terdapat *list* token kata [“should”, “i”, “leave”, “my”, “friends”, “and”, “find”, “new”, “one”], setelah mengalami *stop word removal* berubah menjadi [“leave”, “friends”, “find”, “new”, “one”]. Proses dari tahapan *stop word removal* yang

ditunjukkan pada Gambar 3.6 dimulai dengan menginputkan data pertanyaan hasil tokenisasi. Selanjutnya, data akan menjalani tahap penghapusan kata-kata yang tidak penting melalui proses *stop word removal*.



Gambar 3. 6 Flowchart Tahap Stopword Removal

### 3.3.5 Lemmatization

*Lemmatization* adalah tahap *pre-processing* dimana kata-kata diubah menjadi bentuk dasar. Misalnya, kata seperti “playing”, “player”, dan “playful” diubah menjadi bentuk dasar “play”. Mirip seperti *stemming*, tetapi *lemmatization* menggunakan analisis makna dan tata bahasa untuk mengembalikan kata ke bentuk dasar sesuai dengan kamus. *Lemmatization* membantu dalam mencocokkan kata-kata yang memiliki arti sama, seperti “hot” dan “warm” sehingga hasil pencariannya menjadi lebih luas. Pada contoh lain, *lemmatization* juga mengubah kata “better” menjadi “good” karena keduanya memiliki makna yang sama. Proses

dari tahapan *lemmatization* yang ditunjukkan pada Gambar 3.7 dimulai dengan menginputkan data pertanyaan yang telah melalui tahap penghapusan kata-kata yang tidak penting. Selanjutnya, data tersebut akan diproses menggunakan *lemmatization* untuk mengubah kata-kata ke bentuk dasarnya..



Gambar 3. 7 Flowchart Tahap Lemmatization

### 3.4 Ekstraksi Fitur TF-IDF

TF-IDF atau *Term Frequency - Inverse Document Frequency* adalah teknik yang digunakan untuk mengekstrak atribut atau fitur yang dapat mendeskripsikan dokumen yang tidak dikategorikan dengan baik. Metode TF-IDF bekerja dengan memberikan bobot pada setiap istilah di dokumen yang direpresentasikan dalam bentuk matriks (Dalaorao *et al.*, 2019). Pada TF-IDF, jika istilah sering muncul pada suatu atau beberapa dokumen, istilah tersebut dianggap penting dan diberi nilai tinggi. Namun, jika istilah banyak muncul hampir di semua dokumen, istilah

tersebut akan dianggap umum dan diberi nilai rendah (Nur Khusna & Agustina, 2018).

Pada penelitian ini, algoritma TF-IDF diimplementasikan untuk melakukan ekstraksi fitur karena algoritma ini terkenal di bidang *text mining*, mudah diterapkan, dan memberikan hasil akurasi yang baik. TF-IDF juga mempertimbangkan bobot setiap kata dengan menggunakan dua pendekatan, frekuensi sebuah term dan berapa banyak sebuah istilah dapat ditemukan (Hakim *et al.*, 2014).

TF menunjukkan seberapa sering atau frekuensi kemunculan karakteristik dalam suatu dokumen. Kata yang sama kemungkinan akan sering muncul dalam dokumen yang panjang daripada dokumen yang pendek, dimana semakin jarang sebuah kata muncul, semakin tinggi nilai IDF-nya karena kata tersebut memiliki potensi untuk memberikan informasi yang lebih besar. Persamaan TF:

$$tf_{(t,d)} = \frac{N_{t,d}}{\sum_{k=0}^n N_{k,d}} \quad (3.1)$$

Keterangan:

$N_{t,d}$  = jumlah kemunculan kata  $t$  dalam kalimat  $d$   
 $\sum_{k=0}^n N_{k,d}$  = total seluruh kata dalam kalimat  $d$

Selanjutnya, IDF mengukur seberapa penting suatu kata dalam kalimat. Hal ini menjadi metrik penilaian umum atau tidaknya kata dalam semua dokumen.

Persamaan IDF:

$$idf_{(t)} = \log\left(\frac{|D|}{df_t}\right) \quad (3.2)$$

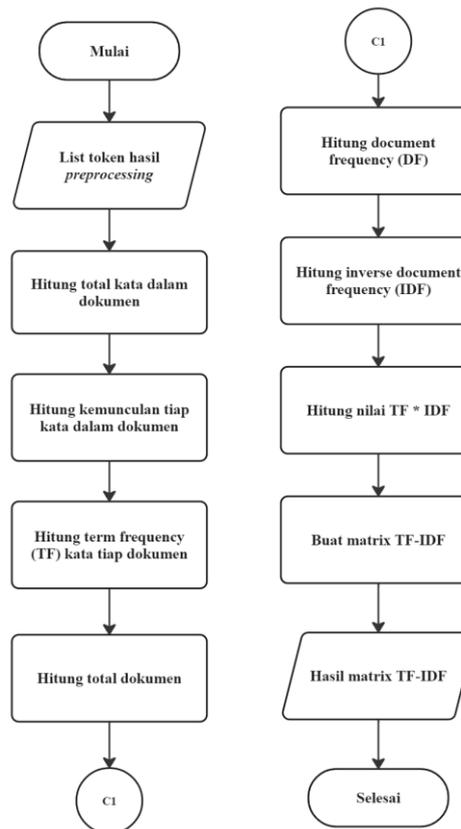
Keterangan:

$|D|$  = total kalimat yang ada di dataset  
 $df_t$  = total kalimat  $d$  dalam koleksi  $D$  yang mengandung kata  $t$

Bobot kata dalam TF-IDF dihitung dengan mengalikan frekuensi kemunculan kata (TF) dan seberapa jarang kata muncul di seluruh dokumen (IDF) yang dihitung pada Persamaan 3.3:

$$w = tf_{(t,d)} \times idf_{(t)} \quad (3.3)$$

Ilustrasi proses perhitungan TF-IDF disajikan dalam Gambar 3.8.



Gambar 3. 8 *Flowchart* Perhitungan TF-IDF

Pada penelitian ini, TF-IDF diperlukan karena dapat membantu dalam mengukur seberapa penting suatu kata. Kata-kata yang sering muncul hampir di setiap kalimat dalam semua kelas mendapatkan bobot rendah sehingga model klasifikasi dapat lebih fokus pada kata-kata yang lebih spesifik pada setiap kelas.

Jika tidak menggunakan TF-IDF dan langsung menggunakan frekuensi kata dalam dokumen dapat mengakibatkan hasil klasifikasi tampak bias karena kata dengan frekuensi tinggi bisa saja mendominasi. Dengan menggunakan TF-IDF, dampak dari kata-kata tersebut dapat diminimalkan karena mempertimbangkan TF-IDF yang bekerja tidak hanya menghitung frekuensi tetapi juga seberapa unik dan penting suatu kata.

### 3.5 *Multinomial Naïve Bayes*

*Multinomial Naïve Bayes* merupakan algoritma machine learning yang termasuk dalam keluarga *Naïve Bayes*. Algoritma ini mengklasifikasikan teks dengan menghitung probabilitas kemunculan kata-kata dan menggunakan *Teorema Bayes*, dengan asumsi bahwa setiap kata dalam dokumen tidak saling terkait. (Sabrani *et al.*, 2020). *Multinomial Naïve Bayes* menjelaskan apakah suatu kata muncul dalam sebuah dokumen atau tidak dan bekerja dengan menggunakan konsep *term frequency* yang artinya menghitung kemunculan kata tersebut (Yuyun *et al.*, 2021). Karena metode ini adalah metode *supervised learning*, maka setiap data yang akan digunakan perlu diberi label terlebih dahulu. Konsep *Teorema Bayes* pada tugas klasifikasi teks tertuang dalam Persamaan 3.4:

$$P(C|d) = \frac{P(C) P(t|C)}{P(t)} \quad (3.4)$$

Keterangan:

- $P(C|t)$  : Probabilitas *posterior* kelas  $C$  terhadap seluruh fitur kata pada kalimat pertanyaan.
- $P(C)$  : Probabilitas awal (*prior*) kelas  $C$
- $P(t|C)$  : Probabilitas (*likelihood*) tiap-tiap fitur kata dari kalimat pertanyaan pada kelas  $C$
- $P(d)$  : Probabilitas total fitur kata pada semua kemungkinan (kelas)

Variabel  $C$  merepresentasikan kelas dari teks dan variabel  $t$  adalah kalimat pertanyaan yang berisi kata-kata atau fitur-fitur yang menjadi petunjuk dalam mengindikasikan suatu kelas. Persamaan 3.4 bertujuan untuk mencari peluang munculnya kelas  $C$  jika diberikan kalimat pertanyaan  $t$  yang disebut sebagai probabilitas *posterior*. Probabilitas *posterior* diperoleh dari peluang kemunculan kelas  $C$  sebelum masuknya fitur kata dari kalimat (probabilitas *prior*) dikalikan dengan peluang kemunculan fitur kata pada kelas  $C$  (*likelihood*), kemudian dibagi dengan peluang kemunculan karakteristik atau fitur kata secara umum (*evidence*).

Sesuai dengan konsep *Teorema Bayes*, *Multinomial Naïve Bayes* mengkombinasi probabilitas *prior* dengan kontribusi dari masing-masing fitur (*term*) atau *likelihood* yang hasilnya adalah probabilitas *posterior*. Nilai maksimum dari probabilitas *posterior* kemudian digunakan untuk menentukan kelas. Konsep ini dikenal sebagai *maximum a posteriori* (MAP). Untuk menghitung MAP, perlu dicari terlebih dahulu nilai dari *prior*, *likelihood*, *evidence*, dan *posterior*.

Rumus perhitungan dari probabilitas *prior* dapat dilihat pada Persamaan 3.5:

$$P(C) = \frac{N_C}{N} \quad (3.5)$$

Keterangan:

$P(C)$  : Probabilitas *prior* (awal) dari kelas  $C$   
 $N_C$  : Jumlah kalimat pertanyaan yang termasuk dalam kelas  $C$   
 $N$  : Jumlah seluruh kalimat pertanyaan

Pada persamaan 3.5,  $P(C)$  merepresentasikan probabilitas *prior* dari kelas *sincere* dan *insincere* yang menandakan peluang munculnya masing-masing kelas atau peluang banyaknya kalimat yang termasuk dalam kelas  $C$  (*sincere* atau *insincere*) sebelum melihat fitur-fitur kata nya. Probabilitas *prior* dihitung dengan

membagi jumlah kalimat yang termasuk ke dalam kelas  $C$  dengan total kalimat dari seluruh data yang ada pada data *training*.

Selanjutnya, terdapat perhitungan dari *likelihood*. *Likelihood* mengukur seberapa bagus fitur kata menjelaskan kelas dari suatu kalimat. Perhitungan dari *likelihood* terdapat pada Persamaan 3.6:

$$P(t_k|C) = \frac{N_{t_k,C} + 1}{\sum N_{t,C} + |V|} \quad (3.6)$$

Keterangan:

$P(t C)$	: Peluang ( <i>likelihood</i> ) kemunculan setiap fitur kata ( $t_k$ ) terhadap kelas $C$
$N_{t_k,C}$	: Jumlah fitur kata ke- $k$ muncul di kelas $C$
$\sum N_{t,C}$	: Total fitur kata yang ada di kelas $C$
$ V $	: Total kata unik yang ada di dataset

Pertama, akan dihitung berapa kali suatu *term* ( $t_k$ ) muncul di kelas  $C$ , dalam konteks ini kelas *sincere* atau *insincere*. Selanjutnya, dihitung juga total *term* ( $N_{t,C}$ ) yang ada di dalam kelas  $C$  dan dijumlahkan dengan total *vocabulary* yang ada di kelas tersebut. Hasil dari keduanya dibagi dan menjadi nilai dari *likelihood*. Karena *Multinomial Naïve Bayes* mengalikan semua fitur *likelihood*, jika terdapat 1 *likelihood* yang memiliki nilai 0 maka *posterior* atau outputnya kemungkinan menghasilkan 0 tidak peduli berapa nilai *likelihood* yang lain. Oleh karena itu, untuk mengatasinya dilakukan *add-one smoothing*.

Karena penelitian ini menerapkan TF-IDF untuk ekstraksi fitur, maka input yang digunakan pada model *Multinomial Naïve Bayes* adalah vektor TF-IDF dari masing-masing kata. Dengan demikian, rumus perhitungan dari *likelihood* menjadi seperti Persamaan 3.7:

$$P(t_k|C) = \frac{W_{ct} + 1}{(\sum_{w' \in V} W_{ct'}) + B'} \quad (3.7)$$

Keterangan:

$W_{ct}$  : Nilai bobot TF-IDF kata  $t$  pada kelas  $C$

$\sum_{w' \in V} W_{ct'}$  : Total bobot TF-IDF dari seluruh kata yang terdapat dalam kelas  $C$

$B'$  : Total IDF dari keseluruhan kata

Langkah berikutnya adalah menghitung *evidence*. Dalam konsep *Teorema Bayes*, *evidence* merujuk pada probabilitas total dari semua kemungkinan observasi. Perhitungan *evidence* ditunjukkan pada Persamaan 3.8:

$$P(t) = \sum P(C) P(t_k|C) \quad (3.8)$$

Keterangan:

$P(t)$  : Probabilitas (*evidence*) fitur kata

$\sum P(C) P(t_k|C)$  : Total probabilitas kemunculan fitur kata  $t_k$  pada semua kemungkinan atau seluruh kelas

Pada *evidence*, nilai *likelihood* dari masing-masing fitur kata yang ada pada *training* dikalikan dengan probabilitas *prior*. Hal ini dilakukan untuk semua kemungkinan kelas dari fitur kata. Lalu, hasil dari tiap-tiap perkalian tersebut dijumlahkan seluruhnya dan inilah yang menjadi nilai probabilitas total (*evidence*). Perhitungan terakhir metode *Multinomial Naïve Bayes* sebelum menentukan kelas adalah probabilitas *posterior*. Probabilitas *posterior* dihitung dengan menggunakan Persamaan 3.9:

$$P(C | t_1, t_2, \dots, t_n) = \frac{P(C) P(t_1 | C) P(t_2 | C) \dots P(t_n | C)}{P(t)} \quad (3.9)$$

Karna fitur-fitur kata yang ada tidak berubah (konstan) untuk semua input (kalimat pertanyaan), maka probabilitas *osterior* dapat diekspresikan seperti pada Persamaan 3.10:

$$P(C | t_1, t_2, \dots, t_n) = P(C) P(t_1 | C) P(t_2 | C) \dots P(t_n | C) \quad (3.10)$$

Keterangan:

$P(C | t_1, t_2, \dots, t_n)$  : Probabilitas (*posterior*) kelas  $C$  terhadap fitur  $t_1$  hingga  $t_n$   
 $P(t_n | C)$  : Probabilitas (*likelihood*) fitur ke- $n$  terhadap kelas  $C$

Kelas dari suatu kalimat ditentukan berdasarkan hasil *posterior* yang nilainya paling besar atau *maximum a posteriori* (MAP). Dengan demikian, didapatkan persamaan untuk model klasifikasi seperti pada Persamaan 3.11:

$$y = \operatorname{argmax} P(C) \prod_{k=1}^n P(t_k | C) \quad (3.11)$$

Keterangan:

$y$  : output / kelas terbesar dari perhitungan MAP  
 $\operatorname{argmax} P(C) \prod_{k=1}^n P(t_k | C)$  : perhitungan untuk menemukan kelas dengan posterior tertinggi

Dimana  $y$  menunjukkan hasil klasifikasi yang diperoleh dari MAP. Nilai *posterior* yang mana perkalian dari *prior* dan semua *likelihood* diambil nilai terbesarnya menggunakan *argmax*.

## BAB IV

### UJI COBA DAN PEMBAHASAN

#### 4.1 Skenario Pengujian

Setiap langkah dalam proses pengujian dijalankan secara berurutan dengan tujuan untuk memastikan bahwa proses uji coba dapat dilakukan dengan teratur dan dapat dipahami dengan mudah. Tahapan dari pengujian pada penelitian ini dijabarkan sebagai berikut:

1. Input dataset

Dataset yang akan digunakan pada uji coba penelitian ini terdiri dari 4.000 kalimat pertanyaan berbahasa Inggris yang tersedia di situs *Kaggle*. Dataset ini dipublikasikan oleh akun resmi Quora pada November 2018. Dalam dataset tersebut, sumber data menyediakan data pertanyaan yang telah dilabeli sebagai *sincere* dan *insincere*. Penelitian ini menggunakan 2.000 baris kalimat yang berlabel *sincere* dan 2.000 kalimat yang berlabel *insincere*. Beberapa sampel dataset penelitian disajikan pada Tabel 4.1.

Tabel 4. 1 Sampel Dataset

No	Question Text	Label
1	What is the qualification for an animator or an animation director?	Sincere
2	Do different flavoured protein shakes have different effects?	Sincere
3	How much time required to learn Bollywood dance? If I have 1 month to learn	Sincere
4	What skills can I have to fit in formulae one?	Sincere
5	What are some things you wish you had thought to bring with you to college?	Sincere
6	How many Saudis are working in the oil field?	Sincere
7	What is the best tourist destination in Rajasthan?	Sincere
8	What has Commercial Radio Hong Kong accomplished in the radio broadcasting industry?	Sincere
9	How do logarithms turn multiplication into addition?	Sincere

No	Question Text	Label
10	What would happen if we all stopped using money (including virtual currencies)?	Sincere
11	Why are the undereducated people that overwhelmingly voted for President Obama held in higher regard by the left than the undereducated Trump voters?	Insincere
12	Why is Quora becoming so Anti-Islam and Islamophobic?	Insincere
13	Innocent children die of cancer every day, but the barbaric evil Quora moderator live. Is this the ultimate irrefutable evidence against the existence of a diety?	Insincere
14	Why are dishonest/untrustworthy persons or companies more successful in their fields than ethical ones? For example, US President Trump and Facebook concludes that honesty is not a good policy after all.	Insincere
15	Is it fair to say that all Muslims are not terrorists but most of the terrorists are Muslims?	Insincere
16	What stops an atheist from lying?	Insincere
17	Has Prince Harry betrayed the European royalty with marriage to a black divorcee?	Insincere
18	Why are Hindus so violent against Christians and Muslims in India?	Insincere
19	Are scorpio people lazy specially men?	Insincere
20	Why do Democrat women like to pretend that Melania Trump is not inspiring?	Insincere

Selanjutnya, dataset mengalami tahap *preprocessing* yang bertujuan untuk membersihkan dan menormalisasi kata pada setiap kalimat. Proses *preprocessing* ini mencakup tahapan *cleansing*, *case folding*, *tokenizing*, *lemmatization*, dan *stop word removal*. Hasil dari tahap *preprocessing* ini adalah *list* token kata yang telah dibersihkan.

## 2. Pembagian dataset menjadi 2 subset

Dataset yang telah diolah dan dibersihkan akan dipisahkan menjadi dua subset, diantaranya data untuk membangun model dan menemukan pola dalam data (disebut data *train*) dan data untuk mengevaluasi performa model pada data yang belum pernah dilihat sebelumnya (disebut data *test*). Dataset akan dibagi menjadi 4 rasio, yaitu 6:4, 7:3, 8:2, dan 9:1 secara acak (*random*) oleh sistem. Pembagian dataset ini dapat membantu dalam menghindari *overfitting* karena memungkinkan

untuk menguji model pada data yang tidak terlibat dalam proses pelatihan. Tabel 4.2 menunjukkan 4 skenario pengujian berdasarkan rasio pembagian data tersebut.

Tabel 4. 2 Skenario Pengujian

Skenario pengujian	Data Latih	Data Uji
1	60%	40%
2	70%	30%
3	80%	20%
4	90%	10%

### 3. Pelatihan model menggunakan subset data latih

Tahap pelatihan model dikerjakan dengan menggunakan data latih yang telah ditentukan sebelumnya. Sesuai dengan skenario pengujian, pelatihan model dilakukan dengan mengambil beberapa persentase data dari dataset, diantaranya 60%, 70%, 80% dan 90%. Pada tahap ini, nilai-nilai TF-IDF akan dihitung berdasarkan frekuensi *term* dari setiap kalimat pada subset data latih. Panjang per-*array* dari TF-IDF ditentukan disini berdasarkan total *vocabulary* dari data latih, begitu juga dengan nilai IDF masing-masing *term* sehingga pengujian model hanya perlu mengacu dari hasil perhitungan pada proses ini. Kemudian, nilai dari TF-IDF digunakan untuk melatih model *Multinomial Naïve Bayes*, yaitu untuk menghitung nilai dari *likelihood*. Probabilitas *prior* juga dihitung pada tahap ini dengan menggunakan jumlah baris kalimat pertanyaan pada setiap kelas.

### 4. Pengujian model menggunakan subset data uji

Pada tahap pengujian model, dilakukan perhitungan probabilitas *posterior* dari model *Multinomial Naïve Bayes* untuk menentukan hasil klasifikasi. Perhitungan ini dikerjakan dengan menggunakan kalimat pertanyaan dari data uji. Sesuai dengan rumus probabilitas *posterior* pada Persamaan 3.11, perhitungannya melibatkan nilai dari probabilitas *prior* dan *likelihood* yang telah dihitung

sebelumnya pada proses pelatihan model. Hasil dari probabilitas *posterior* yang tertinggi menjadi hasil klasifikasi dari kalimat yang diujikan.

5. Evaluasi performa sistem menggunakan *confusion matrix*

Hasil klasifikasi yang didapat dari pengujian model dievaluasi performanya menggunakan *confusion matrix*. *Confusion matrix* digunakan untuk mengukur seberapa baik suatu model klasifikasi dalam memprediksi kelas data. *Confusion matrix* ini merupakan sebuah matriks dua dimensi, yang diindeks dimana satu dimensi berdasarkan kelas sebenarnya (*ground-truth*) dan dimensi lainnya berdasarkan kelas yang diprediksi (Ting, 2016). Penelitian ini melakukan tugas klasifikasi 2 kelas atau *binary classification task* sehingga *confusion matrix* yang digunakan adalah matriks 2x2 yang menyajikan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Representasi dari *confusion matrix* dapat dilihat pada Tabel 4.3.

Tabel 4.3 *Confusion Matrix*

Kelas Aktual	Kelas Prediksi	
	Insincere	Sincere
Insincere	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Sincere	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Indikator yang digunakan untuk mengukur performa pada penelitian ini adalah *accuracy*, *precision*, *recall*, dan *F1-score*. Persamaan 4.1, 4.2, 4.3, dan 4.4 secara berturut-turut digunakan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4.4)$$

Keterangan:

<i>Accuracy</i>	mengukur berapa persentase kalimat pertanyaan yang berhasil diklasifikasikan dengan benar oleh model dari seluruh kalimat pertanyaan yang ada
<i>Precision</i>	mengukur jumlah kalimat pertanyaan positif ( <i>insincere</i> ) yang diprediksi sebagai kelas positif ( <i>insincere</i> ) pada data aktual.
<i>Recall</i>	mengukur seberapa banyak dari kalimat pertanyaan positif ( <i>insincere</i> ) aktual yang berhasil diprediksi sebagai kelas positif ( <i>insincere</i> ) oleh model.
<i>F1-score</i>	mengukur keseimbangan antara <i>precision</i> dan <i>recall</i> sehingga skornya adalah hasil gabungan dari keduanya.

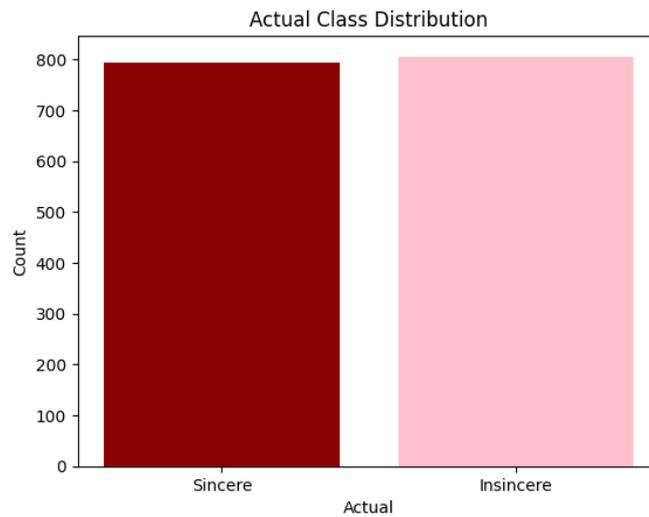
#### 4.1 Hasil Uji Coba

Sesuai dengan skenario pengujian, penelitian ini dilakukan 4 kali uji coba dengan membagi dataset menjadi beberapa rasio. Tujuan dari skenario ini adalah untuk menemukan rasio data latih dan data uji yang memberikan hasil paling optimal untuk model klasifikasi. Masing-masing model klasifikasi yang dihasilkan dievaluasi performanya dengan menggunakan parameter *accuracy*, *precision*, *recall*, dan *F1-score*. Untuk mendapatkan nilai-nilai tersebut, kelas aktual dibandingkan dengan kelas prediksi sehingga dapat terlihat berapa data pertanyaan yang diprediksi dengan benar dan tidak.

##### 4.2.1 Hasil Skenario ke-1

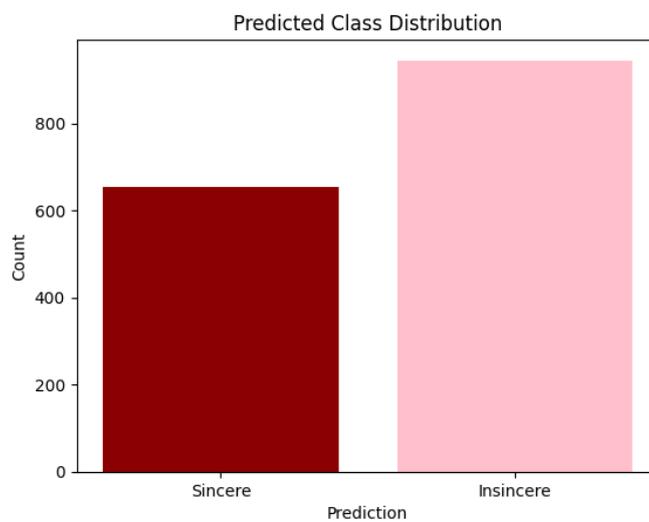
Uji coba dengan rasio skenario data 60:40 terdiri dari 2.400 data *train* (latih) dan 1.600 data *test* (uji) yang telah ditentukan secara *random* oleh sistem. Pada data latih, jumlah data dengan label *sincere* terdapat sebanyak 1.206, sedangkan label

*insincere* sebanyak 1.194. Sementara itu, data uji yang digunakan sebagai data kelas aktual terdiri dari 794 data dengan label *sincere* dan 806 data berlabel *insincere* yang diilustrasikan pada Gambar 4.1.



Gambar 4. 1 Distribusi Kelas Aktual pada Skenario ke-1

Sementara itu, distribusi pada kelas prediksi menunjukkan terdapat sebanyak 655 kalimat termasuk ke dalam kelas *sincere* dan 945 kalimat termasuk ke dalam kelas *insincere*, seperti yang ditampilkan pada Gambar 4.2.



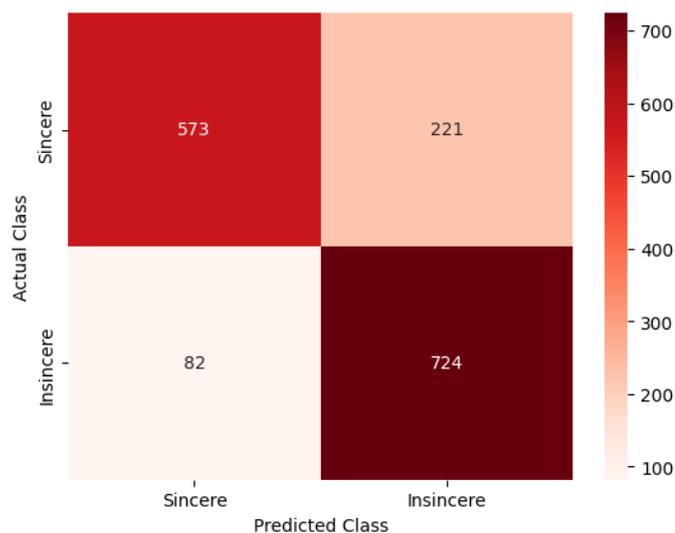
Gambar 4. 2 Distribusi Kelas Aktual pada Skenario ke-1

Hasil prediksi yang dilakukan oleh model pada skenario ke-1 disajikan dalam Tabel 4.4. Dalam tabel, setiap kalimat memiliki kelas prediksinya sendiri, serta kelas aktualnya. Terdapat juga kolom-kolom yang merepresentasikan parameter-parameter dari *confusion matrix*, di mana nilai 1 menunjukkan bahwa kalimat tersebut masuk ke dalam kategori parameter (TP/TN/FP/FN), sementara nilai 0 menunjukkan bahwa kalimat tersebut tidak termasuk dalam kategori parameter (TP/TN/FP/FN).

Tabel 4. 4 Hasil Skenario ke-1

No	Aktual	Prediksi	TP	TN	FP	FN
0	Sincere	Insincere	0	0	1	0
1	Sincere	Sincere	0	1	0	0
2	Sincere	Sincere	0	1	0	0
3	Sincere	Sincere	0	1	0	0
4	Insincere	Insincere	1	0	0	0
5	Sincere	Sincere	0	1	0	0
...	...	...	...	...	...	...
1.599	Insincere	Insincere	1	0	0	0

Berdasarkan hasil klasifikasi yang didapatkan, maka *confusion matrix* untuk model pada skenario ke-1 ini seperti yang terdapat dalam Gambar 4.3.



Gambar 4. 3 *Confusion Matrix* Skenario ke-1

Hasil *confusion matrix* pada Gambar 4.3 menunjukkan bahwa model berhasil mengklasifikasikan kelas *insincere* dengan tepat (*True Positive*) sebanyak 724 data. Terdapat juga data yang keliru diklasifikasikan (*False Positive*) sebagai kelas *insincere* sebanyak 221 data. Sementara itu, kelas *sincere* berhasil diklasifikasikan dengan tepat (*True Negative*) sebanyak 573 data, tetapi terdapat 82 data yang keliru diklasifikasikan sebagai kelas *sincere* (*False Negative*). Melalui Persamaan 4.1, 4.2, 4.3, dan 4.4, performa model dapat dihitung, sebagai berikut:

$$\text{Accuracy} = \frac{724 + 573}{724 + 221 + 573 + 82} = 0.81$$

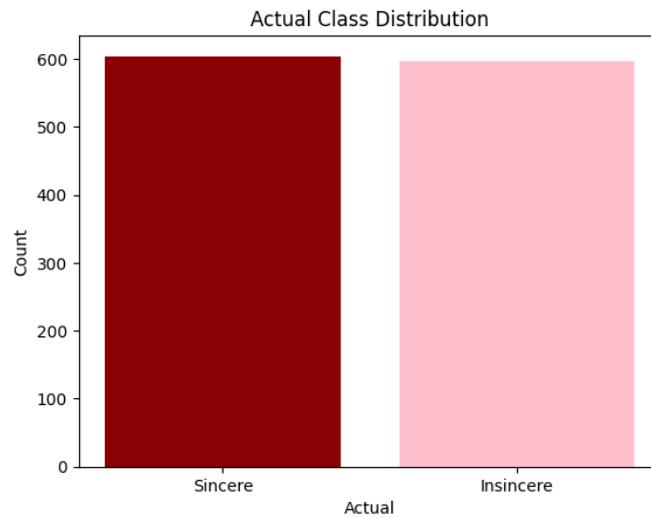
$$\text{Precision} = \frac{724}{724 + 221} = 0.77$$

$$\text{Recall} = \frac{724}{724 + 82} = 0.9$$

$$\text{F1-score} = 2 \times \frac{0.77 \times 0.9}{0.77 + 0.9} = 0.83$$

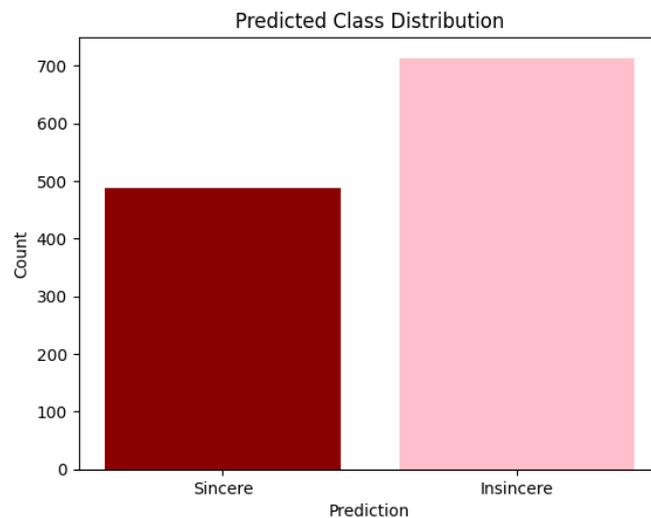
#### 4.2.2 Hasil Skenario ke-2

Uji coba dengan rasio skenario data 70:30 terdiri dari 2.800 data *train* (latih) dan 1.200 data *test* (uji) yang telah ditentukan secara *random* oleh sistem. Pada data latih, jumlah data dengan label *sincere* terdapat sebanyak 1.396, sedangkan label *insincere* sebanyak 1.404. Sementara itu, data uji yang digunakan sebagai data kelas aktual terdiri dari 604 data dengan label *sincere* dan 596 data berlabel *insincere* yang ditunjukkan pada Gambar 4.4.



Gambar 4. 4 Distribusi Kelas Aktual pada Skenario ke-2

Sementara itu, distribusi pada kelas prediksi menunjukkan terdapat sebanyak 487 kalimat termasuk ke dalam kelas *sincere* dan 713 kalimat termasuk ke dalam kelas *insincere*, seperti yang diilustrasikan pada Gambar 4.5.



Gambar 4. 5 Distribusi Kelas Prediksi pada Skenario ke-2

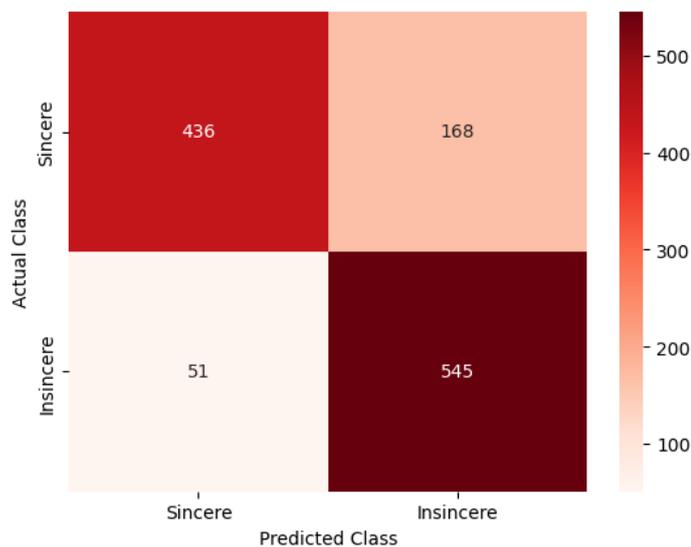
Hasil prediksi yang dilakukan oleh model pada skenario ke-2 disajikan dalam Tabel 4.5. Dalam tabel, setiap kalimat memiliki kelas prediksinya sendiri, serta kelas aktualnya. Terdapat juga kolom-kolom yang merepresentasikan parameter-

parameter dari *confusion matrix*, di mana nilai 1 menunjukkan bahwa kalimat tersebut masuk ke dalam kategori parameter (TP/TN/FP/FN), sementara nilai 0 menunjukkan bahwa kalimat tersebut tidak termasuk dalam kategori parameter (TP/TN/FP/FN).

Tabel 4. 5 Hasil Skenario ke-2

No	Aktual	Prediksi	TP	TN	FP	FN
0	Sincere	Insincere	0	0	1	0
1	Sincere	Sincere	0	1	0	0
2	Sincere	Sincere	0	1	0	0
3	Sincere	Sincere	0	1	0	0
4	Insincere	Insincere	1	0	0	0
5	Sincere	Sincere	0	1	0	0
...	...	...	...	...	...	...
1199	Sincere	Sincere	0	1	0	0

Berdasarkan hasil klasifikasi yang didapatkan, maka *confusion matrix* untuk model pada skenario ke-2 ini seperti yang terdapat dalam Gambar 4.6.



Gambar 4. 6 *Confusion Matrix* Skenario ke-2

Hasil *confusion matrix* pada Gambar 4.6 menunjukkan bahwa model berhasil mengklasifikasikan kelas *insincere* dengan tepat (*True Positive*) sebanyak 545 data. Terdapat juga data yang keliru diklasifikasikan (*False Positive*) sebagai kelas

*insincere* sebanyak 168 data. Sementara itu, kelas *sincere* berhasil diklasifikasikan dengan tepat (*True Negative*) sebanyak 436 data, tetapi terdapat 51 data yang keliru diklasifikasikan sebagai kelas *sincere* (*False Negative*). Melalui Persamaan 4.1, 4.2, 4.3, dan 4.4, performa model dapat dihitung, sebagai berikut:

$$\text{Accuracy} = \frac{545 + 436}{545 + 168 + 436 + 51} = 0.82$$

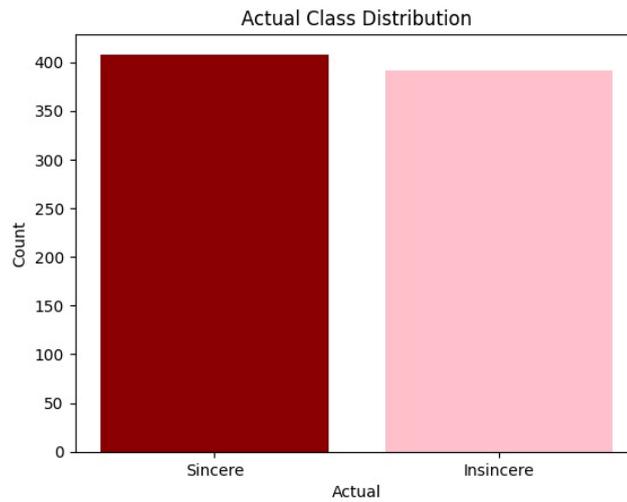
$$\text{Precision} = \frac{545}{545 + 168} = 0.76$$

$$\text{Recall} = \frac{545}{545 + 51} = 0.91$$

$$\text{F1-score} = 2 \times \frac{0.76 \times 0.91}{0.76 + 0.91} = 0.83$$

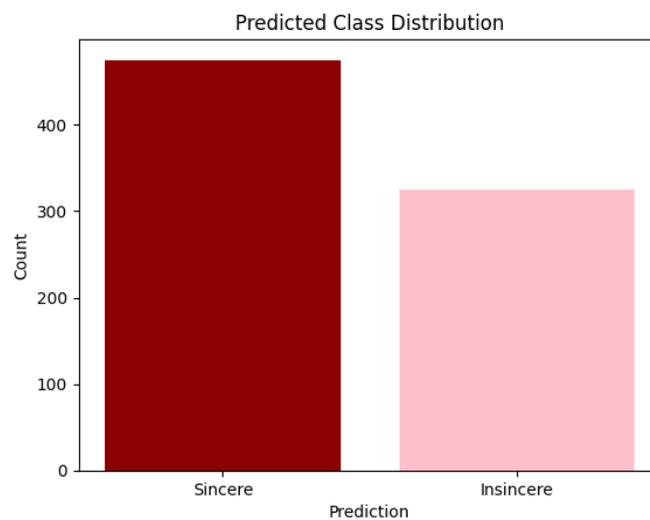
### 4.2.3 Hasil Skenario ke-3

Uji coba dengan rasio skenario data 80:20 terdiri dari 3.200 data latih (*train*) dan 800 data uji (*test*) yang telah ditentukan secara *random* oleh sistem. Pada data latih, jumlah data dengan label *sincere* terdapat sebanyak 1.592, sedangkan label *insincere* sebanyak 1.608. Sementara itu, data uji yang digunakan sebagai data kelas aktual terdiri dari 408 data dengan label *sincere* dan 392 data berlabel *insincere* yang ditunjukkan pada Gambar 4.7.



Gambar 4. 7 Distribusi Kelas Prediksi pada Skenario ke-3

Sementara itu, distribusi pada kelas prediksi menunjukkan terdapat sebanyak 325 kalimat termasuk ke dalam kelas *sincere* dan 475 kalimat termasuk ke dalam kelas *insincere*, seperti yang ditampilkan pada Gambar 4.8.



Gambar 4. 8 Distribusi Kelas Prediksi pada Skenario ke-3

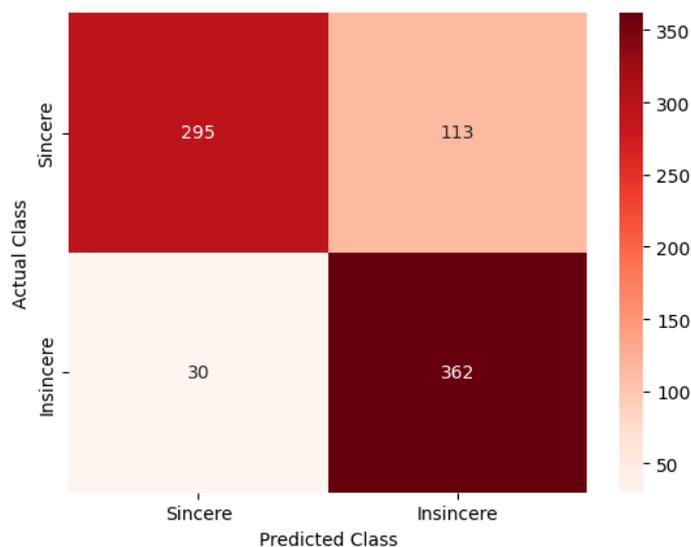
Hasil prediksi yang dilakukan oleh model pada skenario ke-3 disajikan dalam Tabel 4.6. Dalam tabel, setiap kalimat memiliki kelas prediksinya sendiri, serta kelas aktualnya. Terdapat juga kolom-kolom yang merepresentasikan parameter-

parameter dari *confusion matrix*, di mana nilai 1 menunjukkan bahwa kalimat tersebut masuk ke dalam kategori parameter (TP/TN/FP/FN), sementara nilai 0 menunjukkan bahwa kalimat tersebut tidak termasuk dalam kategori parameter (TP/TN/FP/FN).

Tabel 4. 6 Hasil skenario ke-3

No	Aktual	Prediksi	TP	TN	FP	FN
0	Sincere	Insincere	0	0	1	0
1	Sincere	Sincere	0	1	0	0
2	Sincere	Sincere	0	1	0	0
3	Sincere	Sincere	0	1	0	0
4	Insincere	Insincere	1	0	0	0
5	Sincere	Sincere	0	1	0	0
...	...	...	...	...	...	...
799	Sincere	Sincere	0	1	0	0

Berdasarkan hasil klasifikasi yang didapatkan, maka *confusion matrix* untuk model pada skenario ke-3 ini seperti yang terdapat dalam Gambar 4.9.



Gambar 4. 9 *Confusion Matrix* Skenario ke-3

Hasil *confusion matrix* pada Gambar 4.9 menunjukkan bahwa model berhasil mengklasifikasikan kelas *insincere* dengan tepat (*True Positive*) sebanyak 295 data. Terdapat juga data yang keliru diklasifikasikan (*False Positive*) sebagai kelas

*insincere* sebanyak 30 data. Sementara itu, kelas *sincere* berhasil diklasifikasikan dengan tepat (*True Negative*) sebanyak 362 data, tetapi terdapat 113 data yang keliru diklasifikasikan sebagai kelas *sincere* (*False Negative*). Melalui Persamaan 4.1, 4.2, 4.3, dan 4.4, performa model dapat dihitung, sebagai berikut:

$$\textit{Accuracy} = \frac{362 + 295}{362 + 113 + 295 + 30} = 0.82$$

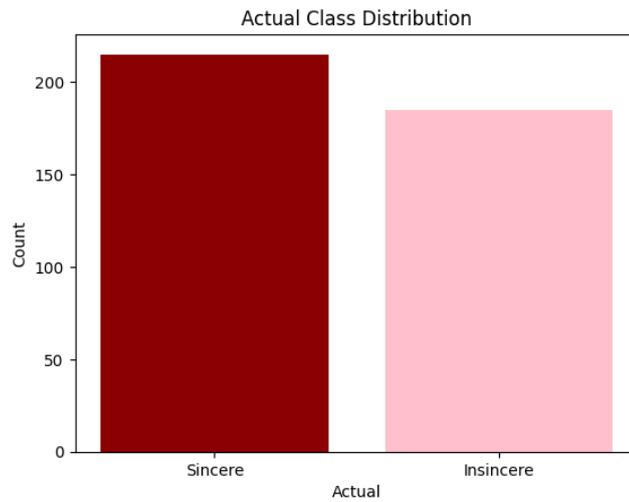
$$\textit{Precision} = \frac{362}{362 + 113} = 0.76$$

$$\textit{Recall} = \frac{362}{362 + 30} = 0.92$$

$$\textit{F1-score} = 2 \times \frac{0.76 \times 0.92}{0.76 + 0.92} = 0.84$$

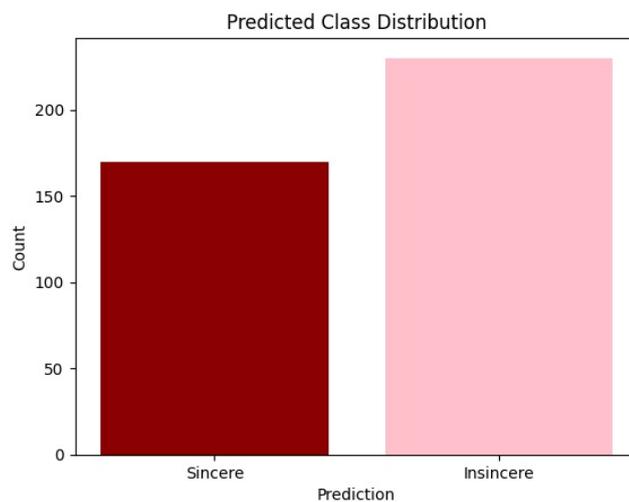
#### 4.2.4 Hasil Skenario ke-4

Uji coba dengan rasio skenario data 90:10 terdiri dari 3.600 data latih (*train*) dan 400 data uji (*test*) yang telah ditentukan secara *random* oleh sistem. Pada data latih, jumlah data dengan label *sincere* terdapat sebanyak 1.785, sedangkan label *insincere* sebanyak 1.815. Sementara itu, data uji yang digunakan sebagai data kelas actual terdiri dari 215 data dengan label *sincere* dan 185 data berlabel *insincere* yang ditunjukkan pada Gambar 4.10.



Gambar 4. 10 Distribusi Kelas Aktual pada Skenario ke-4

Sementara itu, distribusi pada kelas prediksi menunjukkan terdapat sebanyak 169 kalimat termasuk ke dalam kelas *sincere* dan 231 kalimat termasuk ke dalam kelas *insincere*, seperti yang ditampilkan pada Gambar 4.11.



Gambar 4. 11 Distribusi Kelas Prediksi pada Skenario ke-4

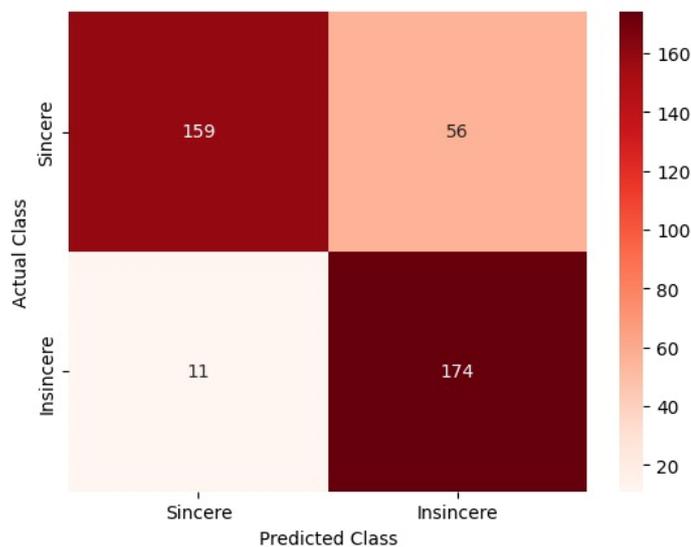
Hasil prediksi yang dilakukan oleh model pada skenario ke-4 disajikan dalam Tabel 4.7. Dalam tabel, setiap kalimat memiliki kelas prediksinya sendiri, serta kelas aktualnya. Terdapat juga kolom-kolom yang merepresentasikan parameter-

parameter dari *confusion matrix*, di mana nilai 1 menunjukkan bahwa kalimat tersebut masuk ke dalam kategori parameter (TP/TN/FP/FN), sementara nilai 0 menunjukkan bahwa kalimat tersebut tidak termasuk dalam kategori parameter (TP/TN/FP/FN).

Tabel 4. 7 Hasil Skenario ke-4

No	Aktual	Prediksi	TP	TN	FP	FN
0	Sincere	Insincere	0	0	1	0
1	Sincere	Sincere	0	1	0	0
2	Sincere	Sincere	0	1	0	0
3	Sincere	Sincere	0	1	0	0
4	Insincere	Insincere	1	0	0	0
5	Sincere	Sincere	0	1	0	0
...	...	...	...	...	....	...
399	Sincere	Sincere	0	1	0	0

Berdasarkan hasil klasifikasi yang didapatkan, maka *confusion matrix* untuk model pada skenario ke-4 ini seperti yang terdapat dalam Gambar 4.12.



Gambar 4. 12 *Confusion Matrix* Skenario ke-4

Hasil *confusion matrix* pada Gambar 4.12 menunjukkan bahwa model berhasil mengklasifikasikan kelas *insincere* dengan tepat (*True Positive*) sebanyak 174 data. Terdapat juga data yang keliru diklasifikasikan (*False Positive*) sebagai

kelas *insincere* sebanyak 56 data. Sementara itu, kelas *sincere* berhasil diklasifikasikan dengan tepat (*True Negative*) sebanyak 159 data, tetapi terdapat 11 data yang keliru diklasifikasikan sebagai kelas *sincere* (*False Negative*). Melalui Persamaan 4.1, 4.2, 4.3, dan 4.4, performa model dapat dihitung, sebagai berikut:

$$Accuracy = \frac{174 + 159}{174 + 56 + 159 + 11} = 0.83$$

$$Precision = \frac{174}{174 + 56} = 0.76$$

$$Recall = \frac{174}{174 + 11} = 0.94$$

$$F1-score = 2 \times \frac{0.76 \times 0.94}{0.76 + 0.94} = 0.84$$

### 4.3 Pembahasan

Berdasarkan uji coba yang dilakukan, diperoleh hasil performa dari empat model klasifikasi yang tercantum dalam Tabel 4.8. Tabel ini berisi nilai rata-rata dari seluruh uji coba yang telah dievaluasi menggunakan indikator, diantaranya *accuracy*, *precision*, *recall*, dan *F1-score* yang dihitung dengan *confusion matrix*. Evaluasi dengan *confusion matrix* ini bertujuan untuk mengukur seberapa akurat model mengklasifikasikan data pertanyaan pada setiap uji coba.

Tabel 4. 8 Hasil Uji Coba

Uji Coba	Pembagian Data					
	Data Latih	Data Uji	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
1	60%	40%	0.81	0.77	0.90	0.83
2	70%	30%	0.82	0.76	0.91	0.83
3	80%	20%	0.82	0.76	0.92	0.84
<b>4</b>	<b>90%</b>	<b>10%</b>	<b>0.83</b>	<b>0.76</b>	<b>0.94</b>	<b>0.84</b>

Terdapat empat skenario pembagian data yang digunakan dalam uji coba ini, yaitu 60:40, 70:30, 80:20, dan 90:10. Masing-masing data dibagi secara acak oleh

sistem sesuai dengan persentase jumlah data latih dan data uji yang ditentukan dalam skenario. Dataset yang telah dipisahkan selanjutnya digunakan pada proses pelatihan model untuk data latih dan pengujian model untuk data uji.

Pada uji coba pertama, dimana pembagian data adalah 60% data latih dan 40% data uji, model mencapai akurasi 0.81, menunjukkan tingkat kinerja yang baik. Dengan nilai *recall* sebesar 0.90 yang lebih tinggi daripada *precision* yang sebesar 0.77 menunjukkan bahwa model berhasil menemukan sebagian besar dari kelas *sincere* tetapi dengan tingkat presisi yang lebih rendah dalam memastikan bahwa sampel yang diklasifikasikan sebagai kelas *insincere* adalah kelas sebenarnya. Kombinasi dari nilai *precision* dan *recall* ini menghasilkan F1-score pada angka 0.83 dimana menunjukkan keseimbangan yang baik antara *precision* dan *recall*.

Beralih ke uji coba kedua dengan pembagian data 70% untuk pelatihan dan 30% untuk pengujian, model ini menunjukkan sedikit tren kenaikan pada hampir seluruh metrik performa. Model ini menghasilkan nilai *accuracy* sebesar 0.82 dimana lebih tinggi 1% daripada *accuracy* skenario sebelumnya. Nilai *precision* yang lebih rendah daripada nilai *recall*, yaitu secara berturut-turut 0.76 dan 0.91 kembali menunjukkan bahwa hampir keseluruhan data diprediksi sebagai kelas *insincere* tetapi terdapat sedikit keraguan pada model dalam memastikan kelas *insincere* sebagai kelas sebenarnya atau menyatakan adanya *False Positive* dari prediksi yang dilakukan. Sementara itu, nilai F1-score masih menunjukkan nilai yang konsisten yaitu pada angka 0.83.

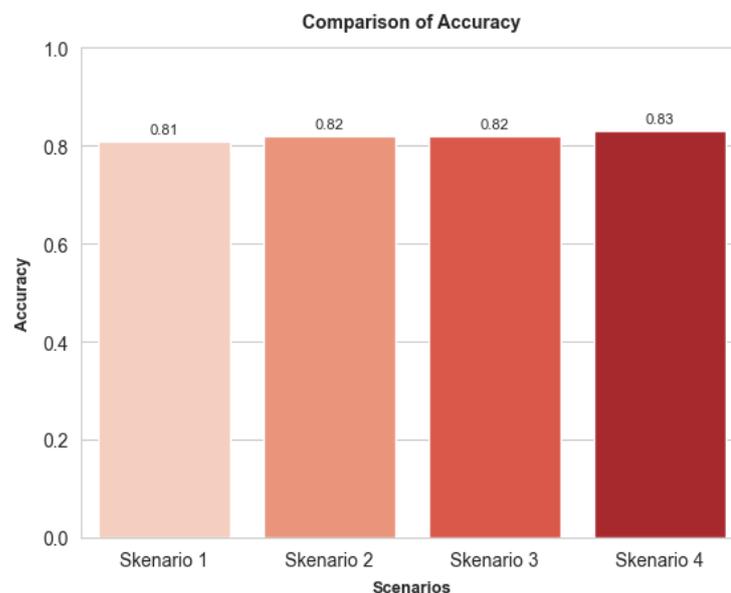
Uji coba ketiga yang menggunakan pembagian data 80% untuk latih dan 20% untuk uji, model menunjukkan kinerja yang sedikit lebih baik dibandingkan

dengan uji coba sebelumnya. Nilai *accuracy* model mencapai angka 0.82 menunjukkan performa yang konstan dengan skenario sebelumnya. Nilai *recall* pada skenario masih stabil pada angka 0.92. Hal ini menunjukkan bahwa model terus konsisten dalam memprediksi sebagian besar sampel sebagai kelas *insincere*. Begitu juga dengan nilai *precision* masih menunjukkan performa yang stabil pada angka 0.76. Dengan nilai *precision* dan *recall* yang cukup tinggi, nilai F1-score pada skenario berhasil mencapai angka 0.84.

Pada percobaan keempat, data dibagi dengan rasio 90% untuk latih dan 10% untuk uji, model menghasilkan nilai *accuracy* sebesar 0.83 yang menunjukkan performa yang lebih baik daripada seluruh skenario sebelumnya. Nilai *recall* model juga mengalami kenaikan sebesar 0.2 sehingga mencapai angka 0.94. Namun, nilai *precision* tidak mengalami peningkatan pada angka 0.76. Meskipun demikian, nilai *recall* yang semakin meningkat dan nilai *precision* yang terbilang cukup stabil mampu menjaga kestabilan dari nilai F1-score pada angka 0.84. Dengan nilai F1-score yang tinggi, model masih konsisten dalam keberhasilannya mengidentifikasi kelas *insincere* dan menjaga kinerja yang baik.

Perbandingan hasil metrik dari tiap skenario uji coba pada model *Multinomial Naïve Bayes* disajikan dalam Gambar 4.13 hingga Gambar 4.16. Terdapat empat bar pada gambar tersebut, masing-masing bar merepresentasikan nilai metrik performa tiap skenario. Pada skenario ke-4, model memberikan nilai *accuracy* yang paling tinggi dibandingkan dengan uji coba yang lain. *Accuracy* mengukur seberapa tepat model dalam mengklasifikasikan data dengan benar. Model *Multinomial Naïve Bayes* berhasil memprediksi kelas yang benar lebih banyak daripada model

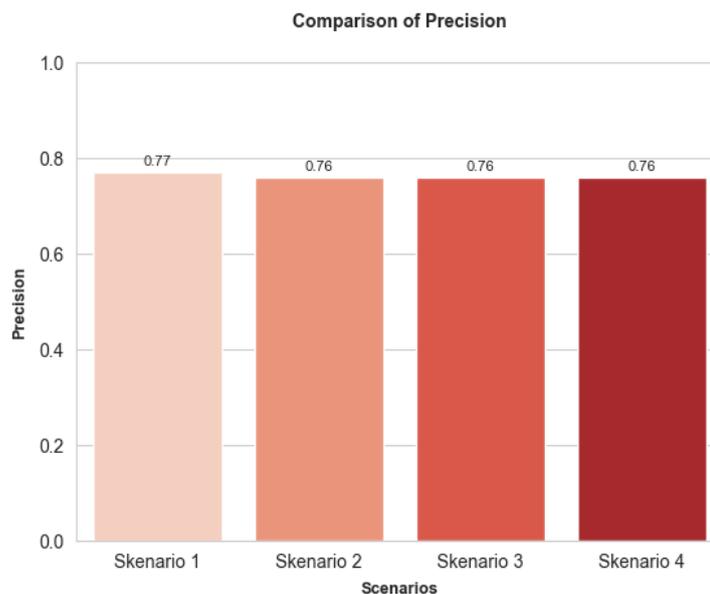
skenario lain. Skenario ke-4 dengan nilai *accuracy* 0.83 ini tergolong cukup tinggi menunjukkan kemampuan model yang handal dalam memprediksi kelas dengan akurat. Hasil juga menunjukkan bahwa terdapat tren peningkatan performa setiap bertambahnya rasio data. Hal ini menandakan bahwa model semakin optimal dalam memprediksi sampel dengan benar jika diberi lebih banyak data untuk proses latih. Informasi mengenai nilai metrik *accuracy* untuk tiap skenario dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Perbandingan Metrik *Accuracy*

Dalam hal metrik *precision*, performa model dalam skenario ke-1 unggul dibandingkan dengan yang lain, dengan nilai rata-rata *precision* mencapai 0.77. Nilai rata-rata *precision* yang tinggi ini menandakan bahwa model memiliki kemampuan yang baik dalam mengidentifikasi kalimat pertanyaan dari kelas *insincere* dengan akurat. Dengan kata lain, sekitar 0.77 dari prediksi kelas *insincere* yang dilakukan oleh model sesuai dengan kelas yang sebenarnya. Namun, terdapat tren penurunan nilai rata-rata *precision* dalam setiap skenario, mengindikasikan

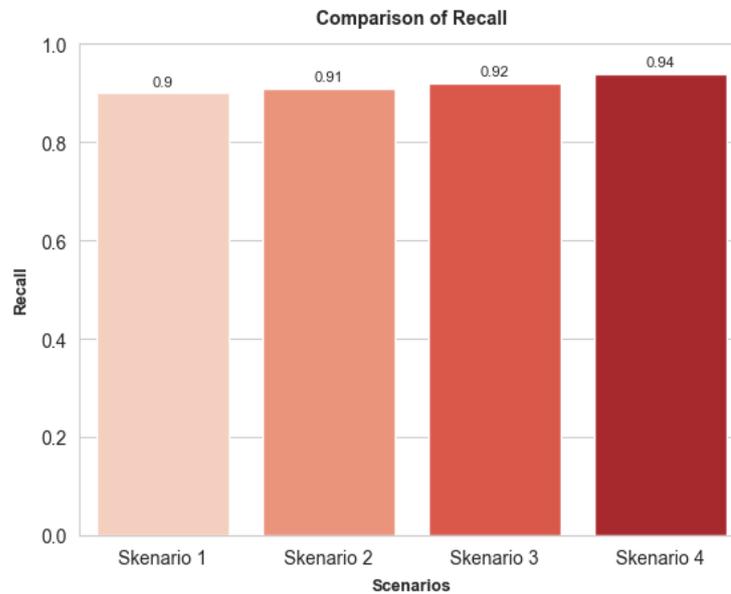
peningkatan *False Positive* (FP) yang dihasilkan oleh model. Meskipun demikian, dengan nilai rata-rata *precision* di atas 0.76 untuk semua skenario, model menunjukkan performa yang memuaskan dalam memprediksi kelas yang sesuai dengan kenyataan. Perbandingan nilai *precision* rata-rata untuk setiap skenario terdapat dalam Gambar 4.14.



Gambar 4. 14 Perbandingan Metrik *Precision*

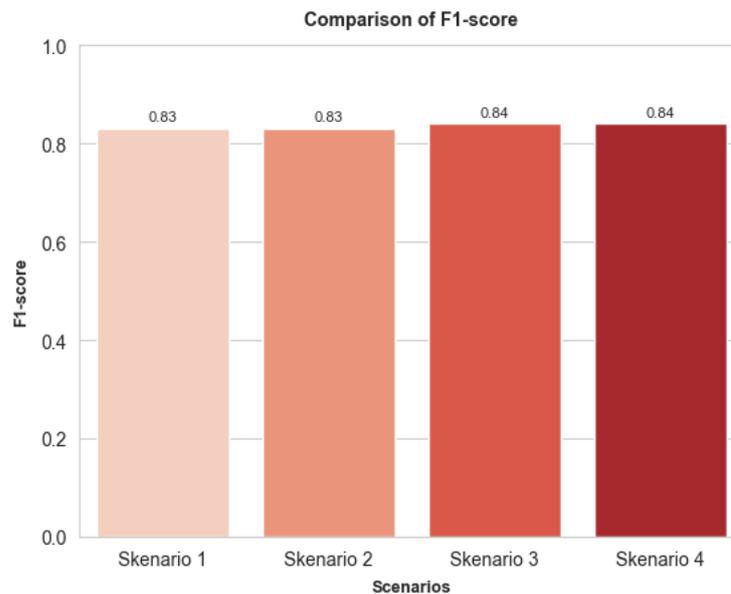
Jika *precision* membahas mengenai kualitas dari prediksi yang dibuat oleh model maka *recall* menunjukkan berapa banyak model berhasil menemukan kasus kelas *insincere*. Nilai *recall* tertinggi kembali didapatkan oleh model pada skenario ke-4, sebesar 0.94, hasil ini dapat dilihat pada Gambar 4.15. Dapat terlihat bahwa nilai *recall* meningkat cukup signifikan dari skenario 1 ke skenario 4, dengan total peningkatan sebesar 0.04. Hal ini menunjukkan bahwa model semakin baik dalam mengidentifikasi semua sampel *insincere* seiring dengan meningkatnya proporsi data latih. Dengan nilai *recall* masing-masing skenario yang mencapai lebih dari 0.9 juga menandakan bahwa model memiliki sensitivitas yang tinggi terhadap kasus

kelas *insincere*, sehingga kemungkinan besar sedikit sedikit sekali kasus *insincere* yang terlewatkan oleh model.



Gambar 4. 15 Perbandingan Metrik *Recall*

Pada metrik *F1-score*, seluruh skenario memberikan hasil yang tinggi, yaitu di atas 0.82. Hal ini menandakan bahwa seluruh model menghasilkan keseimbangan yang baik antara *precision* dan *recall*. Semua model memiliki kualitas yang baik dalam menemukan kasus kelas *insincere* (*recall*) serta memastikan bahwa prediksi *insincere* yang dibuat adalah relevan atau sesuai dengan kelas sebenarnya. Dengan demikian, meskipun terdapat penurunan performa yang dari metrik *precision*, model secara keseluruhan masih memberikan keseimbangan yang baik dari kedua metrik tersebut. Perbandingan dari metrik performa *F1-score* tercantum dalam Gambar 4.16.



Gambar 4. 16 Perbandingan Metrik F1-score

Berdasarkan analisis terhadap setiap metrik performa dari masing-masing uji coba, didapatkan hasil baik pada skenario ke-4 secara keseluruhan. Skenario ke-4 dengan proporsi data latih sebesar 90% dan data uji sebesar 10% menjadi opsi pembagian data terbaik pada kasus penelitian ini. Proporsi data latih yang besar membuat model dapat mempelajari lebih banyak data dan mengenali berbagai variasi pola kata pada kalimat pertanyaan. Hal ini membuat model menjadi lebih tergeneralisasi dan ketika diberikan kalimat pertanyaan baru untuk diprediksi, model menjadi lebih mudah dalam menemukan kelas yang tepat. Sementara itu, skenario dengan proporsi data latih yang lebih sedikit memberikan performa yang lebih rendah. Nilai *accuracy* pada skenario ke-1 data uji menghasilkan nilai yang paling rendah daripada skenario lainnya, seperti yang ditunjukkan pada Gambar 4.13. Pada skenario ke-1, hanya 60% data yang digunakan untuk pelatihan, sedangkan 40% sisanya digunakan untuk pengujian. Dengan data pelatihan yang lebih sedikit, model mungkin tidak mendapatkan cukup informasi untuk belajar

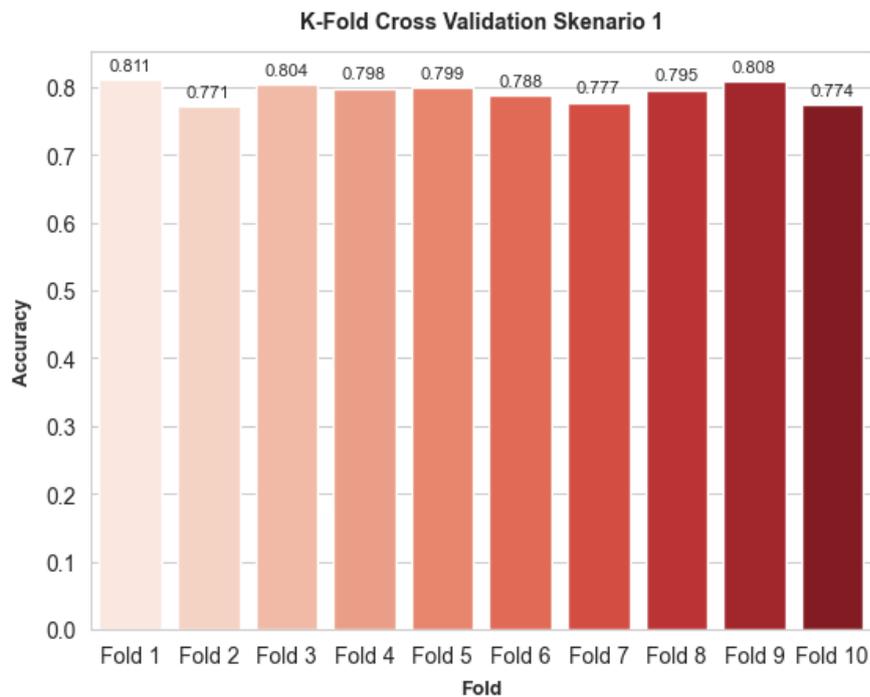
pola yang kompleks dalam data. Hal ini dapat menyebabkan performa model yang kurang optimal dibandingkan dengan skenario lainnya yang menggunakan lebih banyak data pelatihan.

Proporsi dari data latih dan data uji cukup berpengaruh terhadap performa dari model. Namun, tidak hanya itu, variasi dari data juga memberikan kontribusi. Jika dataset cenderung memiliki pola-pola yang mirip atau terfokus pada karakteristik tertentu, model akan menjadi kesulitan dalam mengenai data baru. Sebaliknya, jika variasi yang diperoleh luas, semakin mudah juga bagi model untuk melakukan prediksi dengan benar. Dapat dikatakan bahwa variasi data yang bagus tergantung pada faktor keberuntungan. Karna data dibagi atau diambil secara acak sehingga tidak dapat diprediksi data seperti apa yang akan diperoleh. Oleh karena itu, penelitian ini menerapkan *k-fold cross validation* untuk memastikan bahwa model diuji pada berbagai kombinasi data uji dan latih.

Pada penelitian ini, *k-fold cross validation* digunakan untuk mengevaluasi kinerja dari model. Pengujian *k-fold cross validation* ini menggunakan jumlah  $k = 10$ , dimana data dipecah menjadi 10 *fold* dengan ukuran yang sama. Salah satu subset dipilih sebagai data validasi untuk pengujian, sedangkan subset lainnya digunakan sebagai data pelatihan. Data akan dibagi menjadi sebanyak  $k$  subset yang sama besar atau disebut dengan *fold*. Setiap iterasi memilih satu subset sebagai data validasi dan yang lainnya sebagai data pelatihan. Proses ini berulang hingga setiap subset telah digunakan sebagai data validasi satu kali. Setelah semua iterasi selesai, rata-rata akurasi model dihitung. Proses *k-fold cross validation* diterapkan pada

seluruh skenario pembagian data 6:4, 7:3, 8:2, dan 9:1 untuk memastikan performa model di seluruh kombinasi data dan menentukan skenario terbaik.

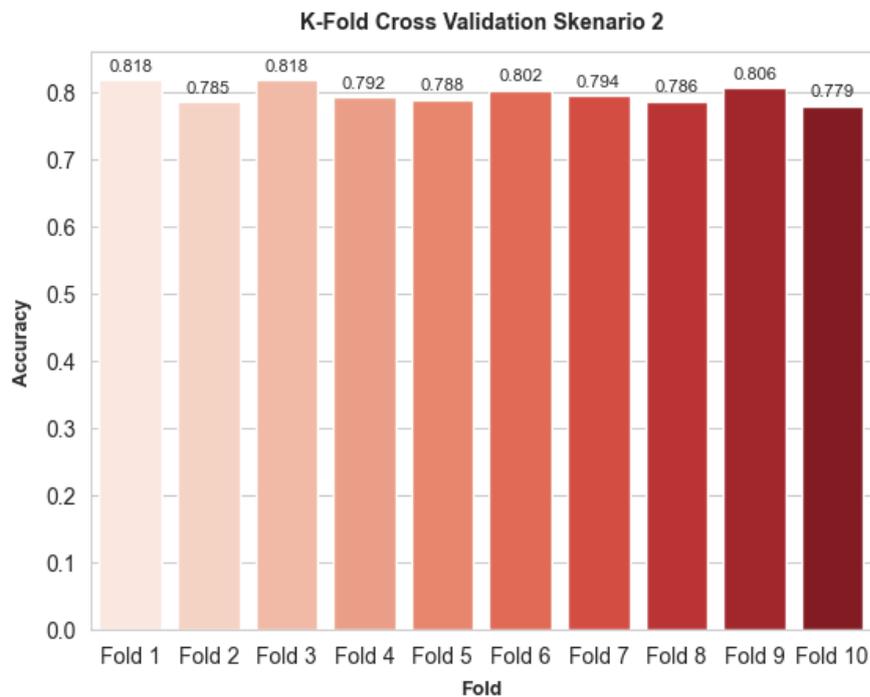
Hasil *k-fold cross validation* menunjukkan pada skenario ke-1 dengan rasio data latih 60% dan data uji 40% dengan nilai  $k=10$  mencapai rata-rata *accuracy* sebesar 0.792. Nilai *accuracy* tertinggi didapatkan pada fold ke-1 sebesar 0.811, sementara *accuracy* terendah diperoleh pada *fold* ke-2 dengan nilai 0.771. Perbandingan nilai *accuracy* untuk masing-masing *fold* pada skenario 6:4 ini dapat dilihat pada Gambar 4.17.



Gambar 4. 17 Nilai *Accuracy* pada *K-fold Cross Validation* Skenario ke-1

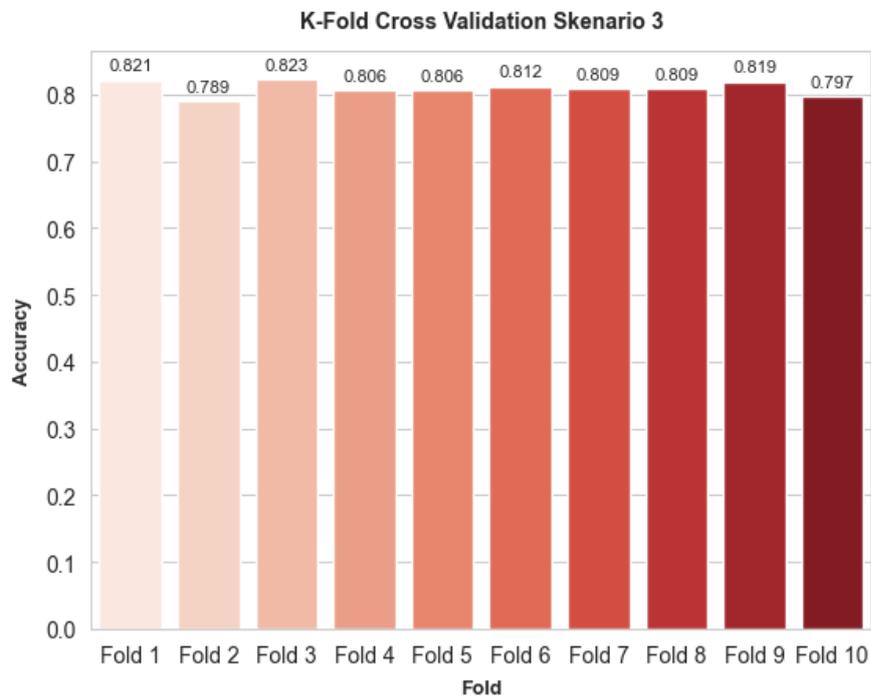
Sementara itu, hasil *k-fold cross validation* pada skenario ke-2 dengan rasio data latih 70% dan data uji 30% dengan nilai  $k=10$  mencapai rata-rata *accuracy* sebesar 0.797. Nilai *accuracy* tertinggi didapatkan pada fold ke-1 dan fold ke-3 sebesar 0.818, sementara *accuracy* terendah diperoleh pada fold ke-10 dengan nilai

0.779. Perbandingan nilai *accuracy* untuk masing-masing *fold* pada skenario 7:3 ini dapat dilihat pada Gambar 4.18.



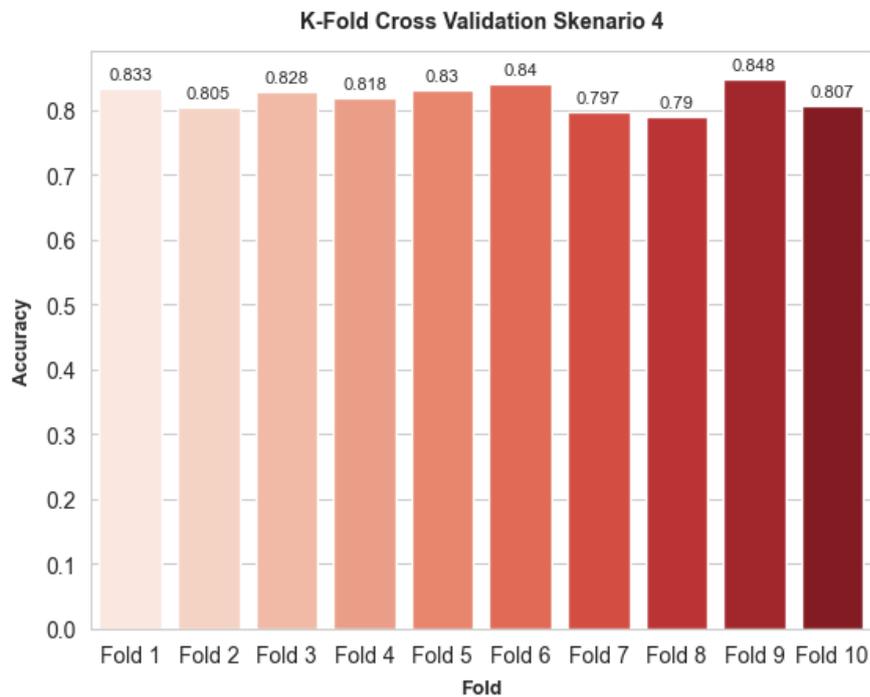
Gambar 4. 18 Nilai *Accuracy* pada *K-fold Cross Validation* Skenario ke-2

Pada skenario ke-3 dengan rasio data latih 80% dan data uji 20%, hasil *k-fold cross validation* dengan nilai  $k=10$  mencapai rata-rata *accuracy* sebesar 0.809. Nilai *accuracy* tertinggi didapatkan pada *fold* ke-3 sebesar 0.823, sementara *accuracy* terendah diperoleh pada *fold* ke-2 dengan nilai 0.789. Dapat terlihat bahwa rata-rata *accuracy* pada skenario ini menjadi lebih baik dari pada beberapa skenario sebelumnya dengan kenaikan mencapai lebih dari 0.01. Perbandingan nilai *accuracy* untuk masing-masing *fold* pada skenario 8:2 ini dapat dilihat pada Gambar 4.19.



Gambar 4. 19 Nilai *Accuracy* pada *K-fold Cross Validation* Skenario 8:2

Hasil *k-fold cross validation* pada skenario ke-4 dengan rasio data latih 90% dan data uji 10% dengan nilai  $k=10$  mencapai rata-rata *accuracy* sebesar 0.82. Nilai *accuracy* tertinggi didapatkan pada fold ke-9 sebesar 0.848, sementara *accuracy* terendah diperoleh pada fold ke-8 dengan nilai 0.79. Perbandingan nilai *accuracy* untuk masing-masing *fold* pada skenario 9:1 ini dapat dilihat pada Gambar 4.20.



Gambar 4. 20 Nilai *Accuracy* pada *K-fold Cross Validation* Skenario 9:1

Berdasarkan hasil keseluruhan *k-fold cross validation*, skenario ke-4 memberikan rata-rata *accuracy* terbaik dibandingkan skenario lainnya. Hal ini menunjukkan bahwa dengan semakin banyak proporsi data latih, model dapat memperoleh performa *accuracy* yang lebih baik. Oleh karena itu, skenario pembagian data 9:1 dianggap sebagai yang paling optimal untuk model ini, karena memberikan hasil evaluasi yang paling akurat dan andal. Ini mengindikasikan bahwa model memanfaatkan data latih yang lebih banyak untuk belajar lebih efektif, sehingga meningkatkan kemampuannya dalam membuat prediksi yang lebih tepat. Informasi mengenai rata-rata *accuracy* untuk setiap skenario disajikan dalam Tabel 4.9.

Tabel 4. 9 Hasil *K-fold Cross Validation*

No	Skenario	Rata-rata Accuracy
1	6:4	0.792
2	7:3	0.797
3	8:2	0.809
4	9:1	0.82

Mekipun begitu, hasil *accuracy* tertinggi tersebut masih berkisar pada angka di atas 0.80 baik pada *splitting* data random maupun yang menggunakan *k-fold cross validation*. Hal ini menunjukkan bahwa terdapat kesalahan atau *error* setidaknya sekitar lebih dari 15% dari prediksi yang dilakukan oleh model. Beberapa sampel pertanyaan yang gagal diprediksi dengan benar oleh model dapat dilihat pada Tabel 4.10.

Tabel 4. 10 Sampel Prediksi *Error*

No	Pertanyaan	Aktual	Prediksi
1	make donald trump greatest president history	Sincere	Insincere
2	remainder divided	Sincere	Insincere
3	feel like replaced	Sincere	Insincere
4	india save pakistan china	Sincere	Insincere
5	fuck get quora account never signed also delete put password never created	Insincere	Sincere

Pada sampel pertanyaan pertama, seharusnya sampel termasuk ke dalam kelas *sincere*, tetapi model memprediksinya sebagai kelas *insincere*. Hal ini mungkin terjadi karena model menentukan suatu kelas tidak berdasarkan konteks secara keseluruhan, tetapi seberapa besar frekuensi dari suatu kata. Model *Multinomial Naïve Bayes* bekerja dengan menghitung nilai posterior, yang menjadi acuan dalam menentukan kelas, berdasarkan *likelihood* yang melibatkan penggunaan nilai TF-IDF. Dengan demikian, jika terdapat beberapa kata yang memiliki nilai TF-IDF tinggi di suatu kelas maka nilai posterior dari kelas tersebut juga akan menjadi tinggi sehingga model akan menganggap kelas tersebut adalah

kelas yang benar. Meskipun mungkin di dalam sampel juga terdapat kata-kata dari kelas yang seharusnya, tetapi tidak cukup berguna jika nilainya tidak cukup tinggi. Selain itu, model juga akan menganggap suatu kata tidak ada jika kata tersebut sebelumnya tidak pernah muncul pada data latih yang mana hal ini berakibat pada berkurangnya informasi untuk mengklasifikasi sampel dengan benar.

Masalah ini menggarisbawahi keterbatasan model dalam memahami konteks semantik penuh dari pertanyaan. Model lebih fokus pada frekuensi dan bobot kata-kata tertentu daripada konteks keseluruhan. Oleh karena itu, untuk meningkatkan akurasi prediksi, pendekatan lain yang lebih canggih dalam menangkap konteks, seperti penggunaan model berbasis pembelajaran mendalam atau teknik *embedding* kata yang lebih maju, mungkin diperlukan.

Pada deteksi pertanyaan *insincere*, nilai TF-IDF memainkan peran penting dalam menentukan kelas. TF-IDF membantu dalam mengidentifikasi kata-kata kunci yang mungkin menunjukkan kecenderungan pada pertanyaan *insincere*. Pada penelitian ini, kata yang ada pada masing-masing kelas dianalisis dengan melihat nilai TF-IDF nya untuk menentukan kata apa saja yang mendominasi dalam kelas tersebut. Nilai TF-IDF dari masing-masing term dihitung *mean* atau rata-ratanya pada tiap kelas. Kemudian, hasil *mean* tersebut diurutkan dari yang terbesar hingga paling kecil.

Pada kelas *sincere*, diperoleh hasil bahwa kata-kata yang mendefinisikan pertanyaan *sincere* diantaranya ‘best’, ‘get’, ‘good’, ‘like’, ‘make’, ‘use’, dan ‘life’, ‘way’, ‘difference’, dan ‘thing’. Ranking skor TF-IDF untuk kata pada kelas *sincere*

ini dapat dilihat pada Tabel 4.11. Visualisasi dari kata-kata yang dominan pada kelas *sincere* dapat dilihat pada Gambar 4.21.

Tabel 4. 11 Ranking Skor TF-IDF pada Kelas *Sincere*

Ranking	Vocab	Skor TF-IDF
1	best	0.03508
2	get	0.02514
3	good	0.01893
4	like	0.01888
5	make	0.01593
6	use	0.01535
7	life	0.01431
8	way	0.01378
9	difference	0.01378
10	thing	0.01335



Gambar 4. 21 *Wordcloud* dari Kata-kata yang Dominan pada Kelas *Sincere*

Pada pertanyaan *insincere* didominasi dengan kata-kata, seperti ‘people’, ‘woman’, ‘trump’, ‘muslim’, ‘indian’, ‘girl’, ‘black’, ‘like’, ‘american’, dan ‘men’. Ranking skor TF-IDF untuk kata pada kelas *insincere* ini dapat dilihat pada Tabel 4.12. Visualisasi dari kata-kata yang dominan pada kelas *insincere* dapat dilihat pada Gambar 4.22.

Tabel 4. 12 Ranking skor TF-IDF pada Kelas *Insincere*

Ranking	Vocab	Skor TF-IDF
1	people	0.05025
2	woman	0.03984
3	trump	0.03251

Ranking	Vocab	Skor TF-IDF
4	muslim	0.03176
5	indian	0.02936
6	girl	0.02538
7	black	0.02533
8	like	0.02511
9	american	0.02491
10	men	0.02434



Gambar 4. 22 *Wordcloud* dari Kata-kata yang Dominan pada Kelas *Insincere*

Analisis kata-kata yang mendefinisikan pertanyaan *sincere* dan *insincere* mengungkapkan perbedaan dalam pola komunikasi dan inti pesan antara kedua jenis pertanyaan tersebut. Kata-kata yang mendominasi pertanyaan *sincere* menunjukkan pertanyaan yang berorientasi pada aspek positif, netral dan sopan, serta menunjukkan niat untuk bersungguh-sungguh memahami sesuatu. Sementara itu, kata-kata yang umum dalam pertanyaan *insincere* menyoroiti topik yang sensitif atau kontroversial. Dapat dilihat dari adanya kecenderungan menyebut orang/kelompok, gender, hingga nama pejabat politik. Analisis ini memberikan wawasan tentang karakteristik bahasa yang berbeda antara pertanyaan *sincere* dan *insincere*, yang dapat digunakan untuk pengembangan model deteksi pertanyaan *insincere* dan pemahaman yang lebih baik tentang perilaku komunikasi *online*.

Adanya deteksi pertanyaan *insincere* sangat berguna bagi pengguna platform tanya jawab *online*. Pengguna dapat mem-*filter* dan mengenali mana pertanyaan-pertanyaan yang memiliki tujuan buruk dan bersifat menghasut hingga mengandung informasi yang salah demi menjatuhkan suatu pihak. Dalam menetapkan kelas dari pertanyaan ini, terdapat ukuran atau standar yang telah ditentukan. Sebagaimana dengan firman Allah Swt. dalam QS. Al-Qamar ayat 49:

إِنَّا كُلَّ شَيْءٍ خَلَقْنَاهُ بِقَدَرٍ

“*Sesungguhnya Kami menciptakan segala sesuatu sesuai dengan ukuran.*” (QS. Al-Qamar 54:49)

Menurut Tafsir Ibnu Katsir, Allah Swt. telah menentukan kadar atau ukuran dari masing-masing makhluk-Nya dan memberi petunjuk kepada semua makhluk-Nya. Hal ini mencakup makhluk dan seluruh alam semesta. Sejalan dengan ini, masing-masing kelas yang diobservasi pada penelitian ini juga memiliki sesuatu yang menjadi tolar ukur, seperti karakteristik atau ciri khas yang mendefinisikan kelas tersebut.

Terkait dengan pertanyaan *insincere*, Allah Swt. berfirman dalam QS. Al-Anfal ayat 46:

وَاطِيعُوا اللَّهَ وَرَسُولَهُ ۗ وَلَا تَنَازَعُوا فَتَفْشَلُوا وَتَذْهَبَ رِيحُكُمْ وَاصْبِرُوا ۗ إِنَّ اللَّهَ مَعَ الصَّابِرِينَ

“*Taatilah Allah dan Rasul-Nya, janganlah kamu berbantah-bantahan yang menyebabkan kamu menjadi gentar dan kekuatanmu hilang, serta bersabarlah. Sesungguhnya Allah bersama orang-orang yang sabar.*” (QS. Al-Anfal 8:2)

Ibnu Jarir Ath Thabary berkata “Janganlah kalian berselisih karena kalian akan berpecah-belah, dan hati kalian akan saling bertikai. Dan kalian akan gagal,

kalian akan melemah dan menjadi gentar. Dan kemudian kekuatan kalian akan hilang”. Ayat ini telah mendaji kaidah umum dan pasti bahwa perselisihan dan permusuhan akan menyebabkan kelemahan dan kegagalan. Oleh karna itu, Allah Swt. melarang permusuhan dan bantah-bantahan di antara umat. Berkaitan dengan ini, pertanyaan *insincere* mungkin menciptakan ketegangan dan konflik yang tidak perlu di antara individu atau kelompok, yang pada gilirannya dapat mengganggu kedamaian dan kesejahteraan tidak hanya pengguna tetapi juga masyarakat luas yang membacanya. Oleh karena itu, penting untuk memastikan bahwa pertanyaan diajukan dengan niat yang baik dan tujuan yang jelas.

Deteksi pertanyaan-pertanyaan *insincere* dapat membantu platform-platform komunitas tanya jawab *online*, seperti Quora untuk meningkatkan integritas dan kualitasnya. Platform komunitas tanya jawab harus memiliki integritas sebagai bentuk tanggung jawab platform tersebut terhadap seluruh konten yang ada di dalamnya, meningkatnya integritas suatu platform akan meningkat juga kepercayaan pengguna terhadap platform tersebut. Pentingnya kejujuran dan integritas tertuang dalam sabda Rasulullah SAW dalam hadis riwayat Muslim:

قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ عَلَيْكُمْ بِالصِّدْقِ فَإِنَّ الصِّدْقَ يَهْدِي إِلَى الْبِرِّ وَإِنَّ الْبِرَّ يَهْدِي إِلَى الْجَنَّةِ وَمَا يَزَالُ الرَّجُلُ يَصْدُقُ وَيَتَحَرَّى الصِّدْقَ حَتَّى يُكْتَبَ عِنْدَ اللَّهِ صِدْقًا وَإِيَّاكُمْ وَالْكَذِبَ فَإِنَّ الْكَذِبَ يَهْدِي إِلَى الْفُجُورِ وَإِنَّ الْفُجُورَ يَهْدِي إِلَى النَّارِ وَمَا يَزَالُ الرَّجُلُ يَكْذِبُ وَيَتَحَرَّى الْكَذِبَ حَتَّى يُكْتَبَ عِنْدَ اللَّهِ كَذَابًا

*Rasulullah shallallahu 'alaihi wa sallam bersabda: “Kalian harus berlaku jujur, karena kejujuran itu akan membimbing kepada kebaikan. Dan kebaikan itu akan membimbing ke surga. Seseorang yang senantiasa berlaku jujur dan memelihara kejujuran, maka ia akan dicatat sebagai orang yang jujur di sisi Allah. Dan hindarilah dusta, karena kedustaan itu akan menggiring kepada kejahatan dan kejahatan itu akan menjerumuskan ke neraka. Seseorang yang senantiasa berdusta dan memelihara kedustaan, maka ia akan dicatat sebagai pendusta di sisi Allah” (HR. Muslim)*

Hadis ini menegaskan bahwa kejujuran adalah sifat yang membimbing kepada kebaikan, sedangkan kebohongan membawa kepada kefasikan. Rasulullah ﷺ juga memperingatkan bahwa seseorang yang terus menerus berbohong akan dicatat sebagai pendusta di sisi Allah. Oleh karena itu, menjaga integritas dan kejujuran dalam segala hal adalah hal yang sangat penting dalam Islam.

Setelah berhasil mendeteksi pertanyaan *insincere*, langkah selanjutnya yang dapat dilakukan adalah platform komunitas tanya jawab *online* khususnya Quora dapat memberikan penanganan terhadap pertanyaan *insincere* yang tidak sesuai dengan pedoman platform dan memberikan edukasi kepada pengguna tentang standar perilaku yang diharapkan. Dari sudut pandang pengguna Quora, penanganan pertanyaan *insincere* bisa dilakukan dengan melaporkan (*report*) pertanyaan tersebut sehingga tindakan yang sesuai dapat diambil. Pengguna juga dapat memberikan tanggapan yang jelas dan informatif untuk menjelaskan mengapa pertanyaan tersebut tidak sesuai dengan aturan dan memberikan pandangan yang lebih baik atau membimbing pengguna tersebut untuk bertanya dengan cara yang lebih sesuai. Dampaknya terhadap masyarakat khususnya pembaca bisa menjadi positif karena melalui tindakan ini, norma dan etika dalam berkomunikasi *online* ditegakkan sehingga menciptakan lingkungan yang lebih positif dan bermutu di platform tersebut. Hal ini juga dapat membantu memperkuat kepercayaan dan keamanan pengguna dalam berbagi informasi dan berdiskusi di platform tersebut.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Deteksi pertanyaan *insincere* pada komunitas tanya jawab *online* menggunakan algoritma *Multinomial Naïve Bayes* melakukan 4 uji coba variasi data. Dalam setiap uji coba, dataset dibagi menjadi bagian latih dan uji, dan model dievaluasi menggunakan *confusion matrix* untuk mengukur keberhasilannya dalam mengklasifikasikan pertanyaan-pertanyaan sebagai *sincere* atau *insincere*. Dengan mempertimbangkan berbagai skenario pembagian data dan teknik evaluasi seperti *K-fold Cross Validation*, penelitian ini memberikan wawasan yang mendalam tentang kemampuan model dalam mengenali dan membedakan pertanyaan-pertanyaan *insincere* dalam konteks tanya jawab *online*.

Uji coba ketiga, dengan proporsi data latih 90% dan uji 10%, memberikan performa terbaik dengan *accuracy* 0.83. Dengan menggunakan 3.600 data latih dan 400 data uji, model mencapai *precision* yang cukup tinggi di angka 0.76 dan *recall* sebesar 0.94 sehingga menghasilkan nilai *F1-score* yang cukup besar juga di angka 0.84. Model juga dilakukan pengujian dengan menggunakan *K-fold Cross Validation* dimana menghasilkan rata-rata akurasi tertinggi pada skenario ke-4 sebesar 0.82, dengan nilai *accuracy* tertinggi pada fold ke-9 sebesar 0.848.

Berdasarkan hasil dari pengujian yang dilakukan, dapat disimpulkan bahwa model yang telah dibangun mampu mengklasifikasikan sampel kalimat pertanyaan

yang ada dengan optimal. Dapat diamati dari nilai *accuracy* yang mencapai hasil cukup tinggi, meskipun masih terdapat beberapa kesalahan prediksi yang dilakukan oleh model. Adanya *error* atau kesalahan ini dapat terjadi karena beberapa faktor, salah satunya adalah model tidak dapat menangkap fitur dengan sempurna dan penggunaan ekstraksi fitur TF-IDF yang kurang efektif pada kasus deteksi kalimat pertanyaan *insincere* dengan *Multinomial Naïve Bayes* ini.

Proporsi rasio data latih dan data uji memberikan pengaruh terhadap performa dari model. Persebaran data juga penting untuk memperkaya kemampuan model menggeneralisasi pola dari kata. Selain itu, analisis mengungkapkan perbedaan dalam pola kata antara pertanyaan *sincere* dan *insincere*. Pertanyaan *sincere* cenderung menggunakan kata-kata yang menggambarkan pertanyaan yang berorientasi pada aspek positif, netral dan sopan, serta menunjukkan niat untuk bersungguh-sungguh memahami sesuatu. Sedangkan pertanyaan *insincere* sering kali menyoroti topik yang sensitif atau kontroversial.

## 5.2 Saran

Penelitian mengenai deteksi pertanyaan *sincere* pada komunitas tanya jawab *online* menggunakan *Multinomial Naïve Bayes* masih memiliki potensi untuk peningkatan lebih lanjut. Terdapat beberapa saran yang dapat dipertimbangkan untuk diterapkan pada penelitian yang akan mendatang:

1. Mengumpulkan dan menambahkan data pertanyaan baru dari berbagai sumber komunitas tanya jawab selain Quora untuk memperkaya dataset.

2. Menggunakan teknik ekstraksi fitur yang lebih canggih seperti Word2Vec, GloVe, atau BERT untuk menangkap konteks kata yang lebih baik daripada TF-IDF.
3. Mengeksplorasi metode klasifikasi lain yang mungkin memberikan performa lebih optimal untuk studi kasus deteksi pertanyaan *insincere* pada komunitas tanya jawab *online* ini.

Dengan menerapkan saran-saran tersebut, diharapkan penelitian selanjutnya dapat mengatasi keterbatasan-keterbatasan yang ditemukan dalam penelitian ini dan menghasilkan model yang lebih handal dalam melakukan deteksi pertanyaan *insincere*.

## DAFTAR PUSTAKA

- Afrillia, Y., Rosnita, L., & Siska, D. (2022). Analisis Sentimen Ciutan Twitter Terkait Penerapan Permendikbudristek Nomor 30 Tahun 2021 Menggunakan TextBlob dan Support Vector Machine. *G-Tech: Jurnal Teknologi Terapan*, 6(2), 387–394. <https://doi.org/10.33379/gtech.v6i2.1778>
- Al Qodrin, S., Yusliani, N., & Syahrini, A. (2022). Klasifikasi Pertanyaan Berbahasa Indonesia Menggunakan Algoritma Support Vector Machine dan Seleksi Fitur Mutual Information. *Jurnal JUPITER*, 14(2), 44.
- Anandarajan, M., Hill, C., & Nolan, T. (2019). Text Preprocessing. In *Practical Text Analytics, Advances in Analytics and Data Science* (Vol. 2, pp. 45–59). [https://doi.org/10.1007/978-3-319-95663-3\\_4](https://doi.org/10.1007/978-3-319-95663-3_4)
- Bae, K., & Ko, Y. (2019). Efficient question classification and retrieval using category information and word embedding on cQA services. *Journal of Intelligent Information Systems*, 53(1), 27–49. <https://doi.org/10.1007/s10844-019-00556-x>
- Baltadzhieva, A., & Chrupała, G. (2015). Question Quality in Community Question Answering Forums: a survey. *SIGKDD Explorations*, 17(1), 8–13.
- Chai, C. P. (2023). Comparison of text preprocessing methods. *Natural Language Engineering*, 29(3), 509–553. <https://doi.org/10.1017/S1351324922000213>
- Dalaorao, G. A., Sison, A. M., Aguinaldo, E., & Medina, R. P. (2019). Integrating Collocation as TF-IDF Enhancement to Improve Classification Accuracy. *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 282–285. <https://doi.org/10.1109/TSSA48701.2019.8985458>
- Gaire, B., Rijal, B., Gautam, D., Lamichhane, N., & Sharma, S. (2019). *Insincere* Question Classification Using Deep Learning. *International Journal of Scientific & Engineering*, 10(7). <http://www.ijser.org>
- Ghasemi, N., Fatourechi, R., & Momtazi, S. (2021). User Embedding for Expert Finding in Community Question Answering. *ACM Transactions on Knowledge Discovery from Data*, 15(4). <https://doi.org/10.1145/3441302>
- Gottipati, S., Tan, A., Chow, D., Shan, J., Lim, J., & Kiat, W. (2020). Leveraging Profanity for *Insincere* Content Detection-A Neural Network Approach. *11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020*, 41–47. <https://doi.org/10.1109/IEMCON51383.2020.9284844>
- Hakim, A. A., Erwin, A., Eng, I. K., Galinium, M., & Muliady, W. (2014). Automated Document Classification for News Article in Bahasa Indonesia based on Term

Frequency Inverse Document Frequency (TF-IDF) Approach. *2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*. <https://doi.org/10.1109/ICITEED.2014.7007894>

Jiang, Y., Lin, H., Wang, X., & Lu, D. (2011). A Technique for Improving the Performance of Naive Bayes Text Classification. *Web Information Systems and Mining*, 196–203. [https://doi.org/10.1007/978-3-642-23982-3\\_25](https://doi.org/10.1007/978-3-642-23982-3_25)

Kim, D. Y., Li, X., Wang, S., Zhuo, Y., & Lee, R. K. W. (2019). Topic enhanced word embedding for toxic content detection in Q&A sites. *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2019*, 1064–1071. <https://doi.org/10.1145/3341161.3345332>

Lutfi, S. (2020). Materi Pendidikan Akhlak menurut Al-Qur'an: Analisis Surah Al-Hujarat Ayat 11-12. *Al-Mudarris: Jurnal Ilmiah Pendidikan Islam*, 3(2), 159–168.

Menaha, R., Jayanthi, V. E., Krishnaraj, N., & Praveen Sundra Kumar, N. (2021). A Cluster-based Approach for Finding Domain wise Experts in Community Question Answering System. *Journal of Physics: Conference Series*, 1767(1). <https://doi.org/10.1088/1742-6596/1767/1/012035>

Mishra, S., & Kumar, N. (2023). *Insincere* Questions Classification Using CNN with Increased Vocabulary Coverage of GloVe Embedding. *Journal of The Institution of Engineers (India): Series B*, 104(2), 387–394. <https://doi.org/10.1007/s40031-023-00858-3>

Mutawalli, L., Taufan, M., Zaen, A., & Bagye, W. (2019). Klasifikasi Teks Sosial Media Twitter Menggunakan Support Vector Machine (Studi Kasus Penusukan Wiranto). *Jurnal Informatika & Rekayasa Elektronika*, 2(2), 43–51. <http://ejournal.stmiklombok.ac.id/index.php/jire>

Neshati, M., Fallahnejad, Z., & Beigy, H. (2017). On dynamicity of expert finding in community question answering. *Information Processing and Management*, 53(5), 1026–1042. <https://doi.org/10.1016/j.ipm.2017.04.002>

Nur Khusna, A., & Agustina, I. (2018). Implementation of Information Retrieval Using Tf-Idf Weighting Method On Detik.Com's Website. *2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. <https://doi.org/10.1109/TSSA.2018.8708744>

Priyambowo, H., & Adriani, M. (2019). *Insincere* Question Classification on Question Answering Forum. *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, 390–394. <https://doi.org/10.1109/ICEEI47359.2019.8988798>

Puspita, C. E., Pratiwi, O. N., & Sutoyo, E. (2021). Perbandingan Algoritma Klasifikasi Support Vector Machine dan Naive Bayes pada Imbalance Data. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 8(1), 11–18. <https://doi.org/10.33330/jurteksi.v8i1.1185>

- Putra, B. G., & Rochmawati, N. (2021). Klasifikasi Berdasarkan Question Dalam Stack Overflow Menggunakan Algoritma Naïve Bayes. *Journal of Informatics and Computer Science*, 02(4), 259–267.
- Riahi, F., Zolaktaf, Z., & Shafiei, M. (2012). Finding Expert Users in Community Question Answering. *WWW '12 Companion: Proceedings of the 21st International Conference on World Wide Web*, 791–798. <https://doi.org/https://doi.org/10.1145/2187980.2188202>
- Roy, P. K. (2020). Multilayer Convolutional Neural Network to Filter Low Quality Content from Quora. *Neural Processing Letters*, 52(1), 805–821. <https://doi.org/10.1007/s11063-020-10284-x>
- Roy, P. K., Singh, J. P., Baabdullah, A. M., Kizgin, H., & Rana, N. P. (2018). Identifying reputation collectors in community question answering (CQA) sites: Exploring the dark side of social media. *International Journal of Information Management*, 42, 25–35. <https://doi.org/10.1016/j.ijinfomgt.2018.05.003>
- Sabrani, A., Gede Putu Wirarama Wedashwara, I. W., & Bimantoro, F. (2020). Metode *Multinomial Naïve Bayes* untuk Klasifikasi Artikel *Online* tentang Gempa di Indonesia. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya*, 2(1), 89–100. <https://doi.org/https://doi.org/10.29303/jtika.v2i1.87>
- Sarkar, D. (2016). Text Classification. In *Text Analytics with Python* (pp. 167–215). Apress. [https://doi.org/10.1007/978-1-4842-2388-8\\_4](https://doi.org/10.1007/978-1-4842-2388-8_4)
- Ting, K. M. (2016). Confusion Matrix. In: Sammut, C., Webb, G.I. (eds) *Encyclopedia of Machine Learning*. In *Encyclopedia of Machine Learning and Data Science*. Springer US. <https://doi.org/10.1007/978-1-4899-7502-7>
- Yusliani, N., Aruda, S. A. Q., Marieska, M. D., Saputra, D. M., & Abdiansah, A. (2022). The effect of Chi-Square Feature Selection on Question Classification using *Multinomial Naïve Bayes*. *Sinkron*, 7(4), 2430–2436. <https://doi.org/10.33395/sinkron.v7i4.11788>
- Yuyun, Nurul Hidayah, & Supriadi Sahibu. (2021). Algoritma *Multinomial Naïve Bayes* Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(4), 820–826. <https://doi.org/10.29207/resti.v5i4.3146>