Klasifikasi Emosi pada Komentar Youtube terkait Kesehatan Mental atau self-improvement

Nama Kelompok:
- Yunita Sangadji - 202210370311332 | sangadjiyunita@gmail.com | 082336531935
- Annisaa Salsabila SF -202210370311352| salsabilaannissaa@gmail.com | 081937539482

Urgensi Topik

Saat ini, YouTube menjadi salah satu platform yang banyak dimanfaatkan oleh masyarakat untuk mencari inspirasi, motivasi, maupun informasi terkait kesehatan mental dan pengembangan diri (*self-improvement*). Pada kolom komentar, sering dijumpai berbagai ekspresi emosional dari para pengguna, seperti perasaan bahagia, sedih, kecewa, maupun empati terhadap isi video yang ditonton. Berdasarkan fenomena tersebut, penerapan machine learning, khususnya melalui metode klasifikasi emosi, memiliki peran penting untuk mengelompokkan komentar sesuai dengan emosi yang terkandung di dalamnya. Melalui proses klasifikasi ini, data komentar dapat dianalisis secara terstruktur untuk mengidentifikasi pola emosi masyarakat terhadap isu kesehatan mental. Hasil analisis tersebut dapat memberikan gambaran tentang cara individu mengekspresikan emosinya di ruang digital, sekaligus menjadi landasan dalam pengembangan sistem yang mampu mengenali serta menyaring komentar negatif agar tercipta lingkungan diskusi yang lebih sehat dan suportif.

Deskripsi umum dataset

Dataset ini diperoleh dari komentar di YouTube melalui API, dan data yang dikumpulkan berjumlah 1500 entri. Setiap entri berisi komentar dengan beberapa atribut terkait, seperti waktu komentar dibuat, nama pengguna, isi komentar, dan jumlah suka (like count). Label yang digunakan untuk mengkategorikan komentar adalah sedih, kecewa, netral, empati, bahagia. Berikut adalah elemen-elemen yang ada dalam dataset:

1. Label:
    - Sedih: Komentar yang menunjukkan perasaan kesedihan mendalam terkait dengan pengalaman atau perasaan dalam konteks kesehatan mental.
    - Kecewa: Komentar yang mencerminkan rasa kecewa terhadap kondisi atau pengalaman yang berhubungan dengan kesehatan mental, atau merasa tidak didukung dalam perjuangan mereka
    - Netral: Komentar yang tidak menunjukkan perasaan kuat terkait dengan topik kesehatan mental, mungkin berupa opini atau informasi tanpa emosi yang jelas
    - Empati: Komentar yang menunjukkan rasa empati terhadap orang yang berjuang dengan masalah kesehatan mental
    - Bahagia: Komentar yang menggambarkan perasaan positif terkait dengan topik kesehatan mental, seperti kebahagiaan atau kesejahteraan setelah mendapatkan bantuan atau pengetahuan tentang cara mengelola masalah mental.

- Other: Komentar yang tidak relevan atau berada di luar konteks kesehatan mental, sehingga tidak dapat dimasukkan ke dalam kategori emosi utama lainnya.

2. Key (kunci) dan Tipe data:
    - Timestamp (object): Waktu kapan komentar tersebut diposting
    - Username (object): nama pengguna yang memberikan komentar
    - Comment (object): Isi komentar yang ditulis oleh pengguna
    - likeCount (int64): Jumlah suka yang diterima oleh komentar tersebut

EDA

EDA berfungsi untuk memahami dan mengecek kondisi data sebelum diproses, seperti melihat struktur data, distribusi label, serta memastikan tidak ada data hilang atau duplikat agar model bisa bekerja dengan baik.

- Info Dataset

  Dataset ini terdiri dari 1500 komentar dengan lima kolom: waktu komentar, nama pengguna, isi komentar, jumlah like, dan label emosi.

Tabel 1. Info Dataset

Index	Column	Non-Null Count	Dtype
0	Timestamp	1500 non-null	Object
1	Username	1500 non-null	Object
2	Comment	1500 non-null	Object
3	LikeCount	1500 non-null	int64
4	Label	1500 non-null	Object

- Describe

  Kolom LikeCount menunjukkan bahwa rata-rata like adalah sekitar 6,56, dengan nilai terkecil 0 dan terbesar 2506. Median-nya 0, artinya sebagian besar komentar tidak mendapat like, tetapi ada beberapa komentar yang sangat banyak like-nya sehingga memengaruhi rata-rata.

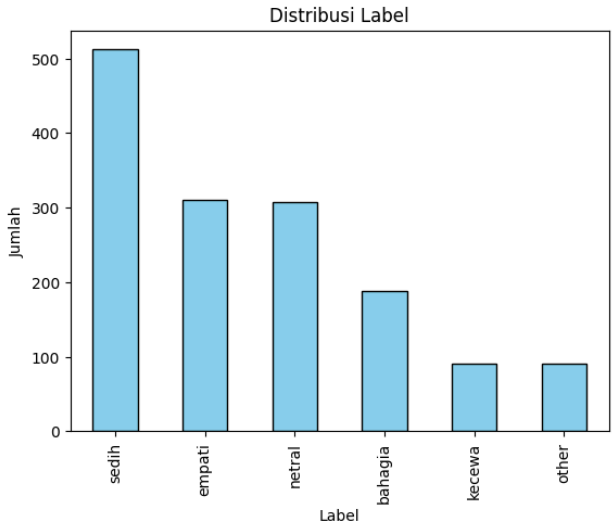
Tabel 2. Describe

	LikeCount
count	1500.000000
mean	6.568667
std	85.002421

min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	2506.000000

- Grafik Distribusi Label

  Grafik distribusi label menunjukkan bahwa kelas *sedih* memiliki jumlah data terbanyak, diikuti oleh *empati* dan *netral*. Sementara itu, kelas *bahagia*, *kecewa*, dan *other* memiliki jumlah data yang lebih sedikit. Secara umum, jumlah data cukup bervariasi, namun masih dapat digunakan untuk proses klasifikasi.



Gambar 1. Grafik Distribusi Label

- Missing value dan duplikat

  Tidak ditemukan missing value maupun data duplikat pada semua kolom, sehingga dataset sudah bersih dan siap digunakan.

Tabel 3. Missing Value

timestamp	0
username	0
comment	0
likecount	0

label	0

Preprocessing

Tahap preprocessing dilakukan sebagai langkah awal untuk membersihkan serta menyiapkan data sebelum digunakan dalam proses pelatihan model machine learning. Langkah ini berperan penting untuk memastikan data yang digunakan memiliki kualitas tinggi, terbebas dari unsur yang tidak relevan (noise), dan dapat diproses secara optimal oleh algoritma. Adapun hasil serta penjelasan dari setiap tahap preprocessing yang telah dilakukan dijelaskan sebagai berikut:

1. Pada tahap awal pengolahan data, dilakukan proses import dataset komentar YouTube yang telah diberi label emosi menggunakan bahasa pemrograman Python dengan bantuan library Pandas. Dataset ini disimpan dalam format Excel dan berisi beberapa atribut seperti waktu komentar diunggah (timestamp), nama pengguna (username), isi komentar (comment), jumlah suka (likeCount), serta label emosi yang telah dikategorikan sebelumnya. Sebelum data dianalisis lebih lanjut, nama-nama kolom disesuaikan agar lebih mudah dipahami dan konsisten dengan kebutuhan analisis. Proses ini bertujuan untuk memastikan struktur dataset sudah sesuai serta memudahkan tahap preprocessing.

Tabel 4. Sample Data Mentah

Index	Timestamp	Username	Comment	LikeCount	Label
0	2019-10-11 T15:12:40Z	@Menjadi Manusia	Ingat, kamu juga manusia. @<a href=...	2506	netral
1	2019-10-11 T17:47:05Z	@sandiikin	Manungsa = Manunggaling Rasa	26	netral
2	2019-10-12 T01:58:29Z	@staypray8954	Tertata sekali... 💚💚💚	15	empati
3	2019-10-12 T04:15:43Z	@dinipraba7035	kalau adhd bukan gangguan mental yah?	6	netral
4	2019-10-12 T04:20:53Z	@staypray8954	@@dinipraba7035 lebih baik	8	netral

			konsul sama pakarny...		

2. Data Cleaning

Tahap ini dilakukan untuk menghapus elemen yang tidak relevan dan menghilangkan noise agar teks lebih terstruktur. Proses ini mencakup penghapusan tag HTML, URL, tanda baca, angka, serta mention pengguna karena tidak berpengaruh terhadap makna emosi dalam komentar. Selain itu, pada dataset ini juga terdapat beberapa emoji yang menggambarkan ekspresi perasaan pengguna. Untuk menjaga konteks emosional dalam teks, emoji tersebut diubah menjadi bentuk teks yang sesuai dengan maknanya (misalnya :) → "senang", 😢 → "sedih"). Langkah ini bertujuan agar model machine learning dapat memahami emosi yang disampaikan melalui emoji secara lebih akurat. Hasilnya, data menjadi lebih bersih dan tetap mempertahankan makna emosional sebelum diproses pada tahap analisis berikutnya.

Tabel 5. Data Cleaning

Comment	Comment Cleaning
Bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yNg brutalnya minta ampun dan selalu membuat cemas hati ini? Anak yg kelakuan bejat, suka bohong, maling, trus aku mesti gimana nih	Bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yNg brutalnya minta ampun dan selalu membuat cemas hati ini Anak yg kelakuan bejat suka bohong maling trus aku mesti gimana nih

3. Lower case

Mengubah seluruh teks menjadi huruf kecil (*lowercase*). Langkah ini bertujuan untuk menyeragamkan penulisan kata agar sistem tidak membedakan kata yang sama hanya karena perbedaan huruf besar dan kecil, seperti "Bahagia" dan "bahagia", sehingga makna tetap konsisten saat diproses oleh model.

Tabel 6. Lower case

Comment	Comment Lowercase
Bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yNg brutalnya minta ampun dan selalu membuat cemas hati ini? Anak yg kelakuan bejat, suka bohong, maling, trus aku mesti gimana nih	bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yng brutalnya minta ampun dan selalu membuat cemas hati ini anak yg kelakuan bejat suka bohong maling trus aku mesti gimana nih

4. Normalisasi

Untuk mengganti kata tidak baku atau singkatan menjadi bentuk baku agar lebih mudah dipahami oleh model. Contohnya, kata seperti “gk”, “nggak”, dan “tdk” diubah menjadi “tidak”, variasi penulisan dapat diseragamkan sehingga mengurangi potensi kesalahan dalam proses analisis.

Tabel 7. Normalisasi

Comment	Comment Normalisasi
bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yng brutalnya minta ampun dan selalu membuat cemas hati ini anak yg kelakuan bejat suka bohong maling trus aku mesti gimana nih	bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yang brutalnya minta ampun dan selalu membuat cemas hati ini anak yang kelakuan bejat suka bohong maling terus saya mesti gimana nih

5. Translate

Fungsi translate pada tahap ini untuk menerjemahkan komentar berbahasa Inggris ke Bahasa Indonesia sebelum proses analisis dilanjutkan. Tujuannya agar semua teks berada dalam bahasa yang sama, sehingga model bisa lebih mudah memahami dan mempelajari pola bahasa secara konsisten.

6. Tokenisasi

Tahap ini memecah kalimat menjadi potongan kata (token) agar teks lebih mudah dianalisis dan diolah oleh model machine learning.

Tabel 8. Tokenisasi

Comment	Comment Token
bagaimana kalau hal itu yang dialami diri sendiri yang selalu cemas melihat kelakuan anak sendiri yang brutalnya minta ampun dan selalu membuat cemas hati ini anak yang kelakuan bejat suka bohong maling terus saya mesti gimana nih	bagaimana,kalau,hal,itu,yang,dialami,diri,sendiri,yang,selalu,cemas,melihat,kelakuan,anak,sendiri,yang,brutalnya,minta,ampun,dan,selalu,membuat,cemas,hati,ini,anak,yang,kelakuan,bejat,suka,bohong,maling,terus,saya,mesti,gimana,nih

7. Label encoder

Label yang digunakan dalam tahap ini adalah Label Encoding, yaitu metode untuk mengubah label emosi yang awalnya berbentuk teks seperti bahagia, sedih, kecewa, empati, netral, dan others menjadi representasi numerik tunggal. Setiap kategori emosi diberikan angka unik, misalnya 0 untuk bahagia, 1 untuk sedih, dan seterusnya. Tahapan ini penting agar model machine learning dapat mengenali dan memproses label emosi secara akurat, meskipun metode ini menganggap adanya urutan nilai antar kategori.

Tabel 9. Encoder label

Sebelum Encoding	Setelah Encoding
bahagia	0
empati	1
kecewa	2
netral	3
other	4
sedih	5

Ekstraksi fitur

Ekstraksi Fitur adalah Proses mengubah data mentah menjadi informasi penting (fitur) agar model lebih mudah memahami dan mengolahnya. Pada proses ini digunakan beberapa ekstraksi fitur, seperti TF-IDF, TF-IDF with N-gram, Bag of Words (BoW), dan BoW with N-gram.

1. Tf - idf

   Proses TF-IDF (Term Frequency–Inverse Document Frequency) mengubah teks pada kolom "*comment_tokens*" menjadi representasi numerik yang menunjukkan kepentingan tiap kata dalam dokumen. Nilai TF-IDF mencerminkan frekuensi kata dalam satu komentar dibanding seluruh dataset, sehingga setiap teks dapat direpresentasikan sebagai vektor angka. Metode ini juga menurunkan pengaruh kata umum, memungkinkan model fokus pada kata yang penting untuk membedakan isi komentar.

   Tabel 10. Tf - idf

aaaa	aamiin	aamiinsenang	aamiinterima	abaikan
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

2. TF-IDF with n-grams

   Selain unigram, TF-IDF juga dapat diterapkan pada **n-gram**, yaitu kombinasi berurutan dari n kata. Misalnya, bigram menggabungkan dua kata, trigram tiga kata. Dengan n-gram, model dapat menangkap konteks atau frasa penting yang tidak terlihat

pada kata tunggal, sehingga lebih baik membedakan komentar atau dokumen dengan kata sama tetapi makna berbeda.

3. Bag Of Word

   Bag-of-Words adalah metode representasi teks menjadi vektor berdasarkan frekuensi kata dalam dokumen, tanpa mempertimbangkan urutan kata. Setiap kata unik menjadi fitur, dan setiap dokumen direpresentasikan oleh seberapa sering kata tersebut muncul, sehingga sederhana tetapi efektif menangkap kata kunci penting.

Tabel 11. Bag Of Word

aaaa	aamiin	aamiinsenang	aamiinterima	abaikan
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

4. Bag Of Word with n-grams

   Selain unigram, Bag-of-Words juga dapat diterapkan pada n-gram, yaitu kombinasi berurutan dari n kata (misal bigram 2 kata, trigram 3 kata). Dengan n-gram, model bisa menangkap konteks atau frasa penting yang tidak terlihat pada kata tunggal, sehingga lebih baik membedakan dokumen atau komentar dengan kata sama tetapi makna berbeda.

Modeling

Pada tahap modeling, penelitian ini membangun beberapa model klasifikasi untuk memprediksi emosi, yaitu SVM, Logistic Regression, dan Random Forest sebagai model machine learning klasik, serta LSTM dan IndoBERT sebagai model deep learning untuk melihat potensi peningkatan performa dari pendekatan berbasis jaringan saraf. Berikut adalah model yang digunakan pada setiap metode ekstraksi fitur:

1. Machine Learning
   - Tf-idf

     Ekstraksi fitur TF-IDF digunakan untuk merepresentasikan teks berdasarkan tingkat kepentingan setiap kata dalam dokumen. Metode ini membantu model fokus pada kata yang relevan untuk membedakan emosi.

     a. Svm

        Hasil evaluasi menunjukkan bahwa model SVM dengan ekstraksi fitur TF-IDF menghasilkan akurasi sebesar 58%. Model cukup baik dalam mengenali kelas sedih dan empati, ditunjukkan oleh nilai recall yang tinggi pada kelas

tersebut. Namun, model kurang efektif mendeteksi emosi kecewa, terlihat dari nilai recall yang sangat rendah pada kelas tersebut. Hal ini menunjukkan bahwa beberapa jenis emosi yang memiliki makna berdekatan, seperti sedih dan kecewa, masih sulit dibedakan oleh model.

Tabel 12. Tf-idf dan SVM

	Precision	Recall	F1-score	Support
0	0.57	0.41	0.48	49
1	0.78	0.47	0.50	110
2	1.00	0.03	0.06	34
3	0.43	0.54	0.48	80
4	0.67	0.41	0.51	29
5	0.58	0.90	0.71	148
Accuracy			0.58	450
Macro avg	0.67	0.46	0.47	450
Weighted avg	0.64	0.58	0.55	450

b. Logistic Regression

Model Logistic Regression dengan TF-IDF menunjukkan bahwa performa model belum stabil di seluruh kelas. Kelas 0, 1, dan 5 memiliki precision dan recall yang cukup seimbang, sedangkan kelas 2 dan 4 memiliki nilai recall dan F1-score sangat rendah, menandakan model gagal mengenali kelas tersebut. Hal ini menunjukkan model masih kesulitan membedakan beberapa kelas, terutama yang jumlah datanya sedikit.

Tabel 13. TF-IDF dan Logistic Regression

	Precision	Recall	F1-score	Support
0	0.67	0.53	0.59	49
1	0.71	0.51	0.59	110
2	1.00	0.00	0.00	34
3	0.40	0.59	0.48	80
4	0.00	0.00	0.00	29

5	0.61	0.89	0.72	148
Accuracy			0.58	450
Macro avg	0.40	0.42	0.40	450
Weighted avg	0.52	0.58	0.53	450

c. RandomForest

Dengan model Random Forest, akurasi yang dihasilkan adalah 57%. Model menunjukkan performa yang belum stabil di seluruh kelas. Beberapa kelas seperti 0, 3, dan 5 memiliki precision dan recall yang cukup baik, sedangkan kelas 2 sangat rendah, menandakan model sulit mengenali kelas tersebut. Dengan akurasi 57%, SVM dengan TF-IDF masih kurang efektif karena belum mampu menangani ketidakseimbangan data dan kemiripan antar label.

Tabel 14. Tf-idf dan RandomForest

	Precision	Recall	F1-score	Support
0	0.60	0.55	0.57	49
1	0.76	0.44	0.55	110
2	1.00	0.03	0.06	34
3	0.49	0.50	0.50	80
4	0.38	0.66	0.48	29
5	0.59	0.83	0.69	148
Accuracy			0.57	450
Macro avg	0.40	0.42	0.48	450
Weighted avg	0.52	0.58	0.55	450

- Tf-idf with n-gram

Pada metode ini, TF-IDF dikombinasikan dengan n-gram agar model dapat memahami konteks dari kombinasi kata berurutan, bukan hanya kata tunggal.

a. Svm

Model dengan TF-IDF dan n-gram menunjukkan performa yang belum stabil, dengan nilai precision, recall, dan F1-score yang bervariasi antar kelas.

Akurasinya mencapai 55%, menunjukkan bahwa penambahan n-gram membantu menangkap konteks kata, namun belum memberikan peningkatan signifikan terhadap performa keseluruhan model.

Tabel 15. Tf-idf with n-gram dan SVM

	Precision	Recall	F1-score	Support
0	0.70	0.43	0.53	49
1	0.86	0.39	0.54	110
2	1.00	0.03	0.06	34
3	0.53	0.40	0.46	80
4	0.67	0.48	0.56	29
5	0.48	0.93	0.63	148
Accuracy			0.55	450
Macro avg	0.71	0.44	0.46	450
Weighted avg	0.66	0.55	0.52	450

b. Logistic Regression

Model dengan TF-IDF dan n-gram menunjukkan akurasi 55%, menandakan kemampuan model dalam memahami konteks sudah sedikit lebih baik, tetapi belum optimal. Nilai evaluasi yang tidak seimbang mengindikasikan bahwa model masih sensitif terhadap distribusi data dan belum mampu memanfaatkan kombinasi kata secara konsisten untuk membedakan emosi dengan tepat.

Tabel 16. Tf-idf with n-gram dan Logistic Regression

	Precision	Recall	F1-score	Support
0	0.67	067	0.56	49
1	0.74	0.74	0.53	110
2	0.00	0.00	0.00	34
3	0.42	0.54	0.47	80
4	0.00	0.00	0.00	29
5	0.54	0.91	0.68	148

Accuracy			0.55	450
Macro avg	0.40	0.39	0.38	450
Weighted avg	0.51	0.55	0.50	450

c. RandomForest

Model dengan TF-IDF dan n-gram memiliki akurasi 54%. Beberapa kelas menunjukkan keseimbangan precision dan recall yang cukup baik, namun ada juga kelas yang masih sulit dikenali dengan benar, menandakan performa model belum merata di seluruh kelas.

Tabel 17. Tf-idf with n-gram dan RandomForest

	Precision	Recall	F1-score	Support
0	0.62	0.53	0.57	49
1	0.81	0.39	0.53	110
2	1.00	0.03	0.06	34
3	0.49	0.46	0.47	80
4	0.29	0.69	0.41	29
5	0.56	0.80	0.66	148
Accuracy			0.54	450
Macro avg	0.63	0.48	0.45	450
Weighted avg	0.63	0.54	0.52	450

- BOW

BoW digunakan untuk merepresentasikan teks berdasarkan frekuensi kata dalam komentar. Metode ini menjadi dasar untuk membangun model klasifikasi menggunakan SVM, Logistic Regression, dan Random Forest, guna memprediksi emosi berdasarkan pola kata yang muncul pada teks.

a. Svm

Berdasarkan hasil evaluasi, model SVM memperoleh akurasi sebesar 0.47, yang merupakan salah satu hasil terendah dibandingkan metode ekstraksi fitur lainnya. SVM hanya mampu mengenali beberapa kelas secara moderat, namun gagal mendeteksi kelas minoritas seperti kelas kecewa dan kelas emosi lainnya yang memiliki nilai F1-score 0.00, menandakan model tidak dapat mengklasifikasikan data pada kelas tersebut sama sekali.

Tabel 18. Bag Of Words dan SVM

	Precision	Recall	F1-score	Support
0	0.45	0.37	0.40	49
1	0.79	0.30	0.43	110
2	0.00	0.00	0.00	34
3	0.30	0.74	0.43	80
4	0.00	0.00	0.00	29
5	0.60	0.70	0.64	148
Accuracy			0.47	450
Macro avg	0.36	0.35	0.32	450
Weighted avg	0.49	0.47	0.44	450

b. LogisticRegression

Model Logistic Regression menunjukkan performa yang lebih baik dibandingkan SVM dengan akurasi 0.55. Model ini mampu mengenali sebagian besar kelas dengan lebih stabil, walaupun kelas dengan jumlah sampel sedikit tetap sulit diprediksi dengan baik. Terlihat bahwa beberapa kelas, terutama emosi tertentu, menghasilkan nilai recall yang rendah sehingga model masih rentan salah dalam mengidentifikasi emosi yang jarang muncul dalam data.

Tabel 19. Bag Of Words dan LogisticRegression

	Precision	Recall	F1-score	Support
0	0.58	0.57	0.58	49
1	0.64	0.51	0.57	110
2	0.31	0.12	0.17	34
3	0.36	0.65	0.46	80
4	0.25	0.03	0.06	29
5	0.68	0.71	0.70	148
Accuracy			0.55	450
Macro avg	0.47	0.43	0.42	450

Weighted avg	0.55	0.55	0.53	450

c. RandomForest

Model Random Forest dengan BoW memperoleh akurasi 0.55, mirip dengan Logistic Regression. Meskipun precision tinggi pada beberapa kelas, recall rendah pada kelas minoritas menunjukkan model lebih fokus pada kelas mayoritas. Secara keseluruhan, hasil BoW menunjukkan ketiga model masih terbatas dalam mengenali emosi dengan distribusi data yang tidak seimbang.

Tabel 20. Bag Of Words danRandomForest

	Precision	Recall	F1-score	Support
0	0.57	0.51	0.54	49
1	0.75	0.45	0.56	110
2	1.00	0.03	0.06	34
3	0.50	0.45	0.47	80
4	0.30	0.66	0.41	29
5	0.57	0.78	0.66	148
Accuracy			0.55	450
Macro avg	0.61	0.48	0.45	450
Weighted avg	0.62	0.55	0.53	450

- BOW with n-gram

BoW dengan n-gram digunakan untuk menangkap kombinasi kata berurutan sehingga model dapat memahami konteks sederhana antar kata, bukan hanya frekuensi kata tunggal.

a. Support Vector Machine (SVM)

Model SVM dengan BoW n-gram menunjukkan hasil yang belum stabil. Beberapa kelas terdeteksi cukup baik, namun kelas dengan data sedikit memiliki recall sangat rendah, menandakan model sulit mengenali emosi minoritas. Dengan akurasi 46%, SVM belum efektif untuk klasifikasi emosi pada data ini.'

Tabel 21. BOW with n-gram dan SVM

	Precision	Recall	F1-score	Support
0	0.49	0.39	0.43	49

1	0.83	0.26	0.40	110
2	1.00	0.03	0.06	34
3	0.28	0.75	0.41	80
4	0.00	0.00	0.00	29
5	0.60	0.65	0.62	148
Accuracy			0.46	450
Macro avg	0.53	0.35	0.32	450
Weighted avg	0.58	0.46	0.43	450

b. LogisticRegression

Model Logistic Regression dengan BoW n-gram menunjukkan performa paling stabil di antara ketiga model. Nilai precision, recall, dan F1-score lebih seimbang, terutama pada kelas mayoritas. Meski masih lemah pada kelas minoritas, model ini mencapai akurasi 54% dan menjadi yang paling konsisten dalam memprediksi berbagai emosi.

Tabel 22. BOW with n-gram dan LogisticRegression

	Precision	Recall	F1-score	Support
0	0.56	0.55	0.56	49
1	0.69	0.47	0.56	110
2	0.22	0.06	0.09	34
3	0.36	0.68	0.47	80
4	0.17	0.03	0.06	29
5	0.68	0.74	0.71	148
Accuracy			0.54	450
Macro avg	0.45	0.42	0.41	450
Weighted avg	0.55	0.54	0.52	450

c. RandomForest

Model Random Forest dengan BoW n-gram memiliki akurasi 54%, namun performanya kurang stabil. Meski cukup baik pada beberapa kelas, model gagal

mengenali kelas dengan data sedikit. Hal ini karena Random Forest kurang optimal untuk data teks berbasis sparse vector seperti BoW atau TF-IDF.

Tabel 23. BOW with n-gram dan RandomForest

	Precision	Recall	F1-score	Support
0	0.57	0.51	0.54	49
1	0.78	0.42	0.54	110
2	1.00	0.03	0.06	34
3	0.49	0.45	0.47	80
4	0.28	0.69	0.40	29
5	0.56	0.76	0.65	148
Accuracy			0.54	450
Macro avg	0.61	0.48	0.44	450
Weighted avg	0.62	0.54	0.52	450

2. Deep Learning
   - Deep Learning

     Penelitian ini menggunakan dua pendekatan deep learning, yaitu LSTM dan IndoBERT. LSTM digunakan untuk memproses data teks secara berurutan dan menangkap konteks emosi melalui mekanisme *memory cell*, sedangkan IndoBERT digunakan sebagai model *pretrained* berbasis transformer yang telah dilatih pada korpus bahasa Indonesia sehingga mampu menghasilkan representasi teks yang lebih kaya dan kontekstual. Kombinasi kedua model ini memungkinkan sistem memahami pola emosi dalam kalimat dengan lebih akurat.

     a. Long Short-Term Memory (LSTM)

        LSTM digunakan dalam kasus klasifikasi emosi karena mampu memproses teks secara berurutan dan mengingat konteks kata sebelumnya melalui *memory cell*. Dengan kemampuan ini, LSTM dapat menangkap pola-pola emosi dalam kalimat—seperti nada, intensitas, dan hubungan antar kata—sehingga membantu model membedakan emosi seperti marah, sedih, atau senang dengan lebih akurat.

Tabel 24. Model LSTM

	Precision	Recall	F1-score	Support
0	0.00	0.00	0.00	49

1	1.00	0.01	0.02	110
2	0.00	0.00	0.00	34
3	0.20	0.78	0.32	80
4	0.00	0.00	0.00	29
5	0.52	0.50	0.51	148
Accuracy			0.30	450
Macro Avg	0.29	0.21	0.14	450
Weighted Avg	0.45	0.30	0.23	450

b. IndoBERT

IndoBERT merupakan model pretrained yang sudah memahami struktur dan konteks bahasa Indonesia. Dengan menghasilkan representasi teks yang lebih kaya dan kontekstual, IndoBERT mampu menangkap nuansa emosi dalam suatu kalimat dengan lebih baik, sehingga meningkatkan akurasi dalam membedakan emosi seperti marah, senang, sedih, atau takut.

Tabel 25. Model IndoBERT Lite Base P1

	Precision	Recall	F1-score	Support
0	0.99	0.98	0.99	124
1	0.99	0.98	0.98	199
2	1.00	0.90	0.95	52
3	0.99	0.97	0.98	193
4	0.93	0.96	0.95	55
5	0.97	1.00	0.98	336
Accuracy			0.98	959
Macro Avg	0.98	0.97	0.97	959
Weighted Avg	0.98	0.98	0.98	959

c. Fine-tuning Improvement: LoRA

Dalam penelitian ini, teknik Fine-Tuning berbasis Low-Rank Adaptation (LoRA) diterapkan untuk meningkatkan kinerja IndoBERT sekaligus mengurangi kebutuhan komputasi saat proses pelatihan. LoRA merupakan

metode *parameter-efficient fine-tuning* (PEFT) yang memungkinkan model besar seperti IndoBERT untuk di-*fine-tune* dengan lebih ringan tanpa perlu memperbarui seluruh parameter model.

Tabel 26. Fine-tuning LoRa dan IndoBERT

	Precision	Recall	F1-score	Support
0	0.65	0.45	0.53	38
1	0.64	0.56	0.60	63
2	0.00	0.00	0.00	18
3	0.56	0.66	0.61	61
4	1.00	0.22	0.36	18
5	0.63	0.88	0.74	103
Accuracy			0.62	300
Macro Avg	0.58	0.46	0.47	300
Weighted Avg	0.61	0.62	0.59	300

Dari hasil pengujian model, Logistic Regression dan SVM dengan ekstraksi fitur TF-IDF menunjukkan performa terbaik dengan akurasi 58%, meskipun keduanya masih belum stabil di seluruh kelas karena ketidakseimbangan data dan kemiripan pola antar label. Model LSTM memperoleh akurasi jauh lebih rendah, yaitu 30%, terutama karena kesulitannya memahami konteks pada komentar pendek dan variatif.

Pada model IndoBERT, penerapan teknik LoRA (Low-Rank Adaptation) menghasilkan akurasi 0.62. Meskipun tidak setinggi fine-tuning penuh, LoRA tetap memberikan peningkatan adaptasi model dengan kebutuhan komputasi yang jauh lebih efisien, sehingga cocok untuk kondisi pelatihan dengan sumber daya terbatas.

IndoBERT Lite Base P1 tetap menjadi model dengan performa paling unggul, mencapai akurasi 0.98 dengan precision, recall, dan F1-score di atas 0.93 pada seluruh kelas. Hasil ini menunjukkan bahwa IndoBERT mampu mengenali emosi secara konsisten dan tetap menjadi model terbaik dalam penelitian ini.

Error Analysis

Error analysis adalah proses meninjau kesalahan prediksi untuk memahami penyebabnya dan meningkatkan performa model. Pada penelitian ini, error analysis dilakukan pada model machine learning (SVM, Logistic Regression, dan Random Forest) dengan ekstraksi fitur TF-IDF karena kombinasi tersebut memberikan hasil terbaik, serta diterapkan juga pada model

LSTM untuk melihat pola kesalahan pada pendekatan deep learning. Analisis difokuskan pada ketidakseimbangan kelas dengan melakukan oversampling dan undersampling, serta mencoba penambahan n-gram dan peningkatan max_features pada TF-IDF. Perbaikan ini meningkatkan metrik evaluasi meskipun belum sepenuhnya optimal. Berikut hasil setelah perbaikan error analysis:

a. SVM

SVM menunjukkan beberapa kelas dengan peningkatan presisi dan recall, namun ada juga beberapa kelas yang menurun. Akurasi keseluruhan pun menurun. Berdasarkan hasil error analysis, upaya perbaikan ini kurang efektif untuk SVM. SVM kurang efektif karena cenderung memprioritaskan kelas dengan jumlah data besar, kesulitan membedakan kelas yang mirip, dan perubahan preprocessing atau parameter tidak cukup meningkatkan performanya.

Tabel 27. Classification Report SVM Setelah Error Analysis

	Precision	Recall	F1-score	Support
0	0.72	0.43	0.54	49
1	0.83	0.45	0.58	110
2	1.00	0.03	0.06	34
3	0.55	0.46	0.50	80
4	0.56	0.52	0.54	29
5	0.51	0.91	0.65	148
Accuracy			0.57	450
Macro avg	0.69	0.47	0.48	450
Weighted avg	0.66	0.57	0.54	450

b. LogisticRegression

Logistic Regression menunjukkan peningkatan performa dibanding hasil sebelumnya. Berdasarkan error analysis, perbaikan pada model ini terbukti efektif. Hasil performanya paling stabil dan seimbang di antara model lainnya, mampu menangani kelas mayoritas maupun minoritas dengan lebih baik, dan akurasinya juga meningkat dari percobaan sebelumnya. Walaupun terdapat satu kelas yang nilai presisi dan recall-nya masih rendah, model ini masih tergolong stabil karena perbedaan presisi dan recall pada kelas tersebut tidak terlalu besar.

Tabel 28. Classification Report LogisticRegression Setelah Error Analysis

	Precision	Recall	F1-score	Support
0	0.62	0.65	0.63	49
1	0.69	0.55	0.61	110
2	0.42	0.24	0.30	34
3	0.52	0.53	0.52	80
4	0.36	0.59	0.45	29
5	0.68	0.76	0.72	148
Accuracy			0.60	450
Macro avg	0.55	0.55	0.54	450
Weighted avg	0.61	0.60	0.60	450

c. RandomForest

Random Forest mengalami penurunan performa sehingga kurang efektif, dan error analysis kurang membantu memperbaikinya karena model kesulitan menangani distribusi kelas yang tidak seimbang pada data teks berbasis TF-IDF dan n-gram. Terdapat beberapa kelas yang memiliki presisi tinggi tetapi recall rendah, begitu pun sebaliknya, menunjukkan bahwa model menghasilkan prediksi yang tidak seimbang dan kurang stabil antar kelas.

Tabel 29. Classification Report Random Forest Setelah Error Analysis

	Precision	Recall	F1-score	Support
0	0.52	0.55	0.53	49
1	0.72	0.35	0.47	110
2	1.67	0.06	0.11	34
3	0.41	0.45	0.43	80
4	0.25	0.66	0.36	29
5	0.62	0.75	0.68	148
Accuracy			0.52	450
Macro avg	0.53	0.47	0.43	450

Weighted avg	0.58	0.52	0.50	450

d. LSTM

Pada model LSTM, error analysis juga menunjukkan peningkatan performa setelah dilakukan beberapa perbaikan, seperti penyesuaian jumlah epoch, penerapan sampling untuk menangani ketidakseimbangan kelas, serta penambahan dropout untuk mengurangi overfitting. Perbaikan tersebut membuat metrik evaluasi LSTM naik, meskipun hasil akhirnya masih belum sepenuhnya optimal.

Tabel 30. Classification Report LSTM Setelah Error Analysis

	Precision	Recall	F1-score	Support
0	0.48	0.51	0.50	49
1	0.56	0.53	0.54	110
2	0.32	0.24	0.27	34
3	0.51	0.44	0.47	80
4	0.40	0.55	0.46	29
5	0.65	0.70	0.67	148
Accuracy			0.55	450
Macro avg	0.49	0.49	0.49	450
Weighted avg	0.54	0.55	0.54	450

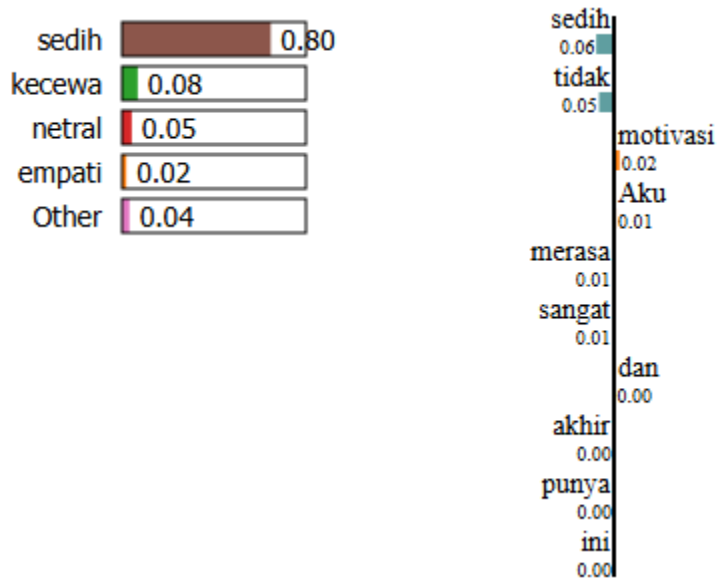
Dari hasil *error analysis*, dapat disimpulkan bahwa Logistic Regression adalah model yang paling stabil dan akurat untuk klasifikasi emosi pada dataset ini, dengan kinerja terbaik pada sebagian besar kelas. SVM dan Random Forest cenderung kurang konsisten, terutama pada kelas dengan jumlah data sedikit. Secara umum, tantangan utama dalam penelitian ini adalah ketidakseimbangan data, yang membuat model cenderung lebih mudah memprediksi kelas dengan jumlah data besar. Selain itu, beberapa label memiliki kemiripan pola sehingga sulit dibedakan, sehingga model juga mengalami kesulitan dalam memprediksi kelas tersebut. Pada model *deep learning* LSTM, upaya perbaikan melalui penyeimbangan data dan penyesuaian parameter pelatihan meningkatkan akurasi dari 30% menjadi 55%, namun performanya masih belum melampaui Logistic Regression berbasis TF-IDF sehingga Logistic Regression tetap dipilih sebagai model terbaik dalam penelitian ini.

Explainable AI

Explainable AI (XAI) digunakan dalam penelitian ini untuk menjelaskan alasan model menghasilkan suatu prediksi. Metode yang digunakan adalah LIME dan SHAP. LIME menjelaskan prediksi dengan menyoroti kata-kata yang paling berpengaruh melalui perubahan kecil pada input, sementara SHAP menghitung kontribusi setiap kata menggunakan nilai Shapley sehingga memberikan penjelasan yang lebih konsisten. Kedua metode ini tidak menghasilkan akurasi, tetapi memberikan visualisasi dan interpretasi pengaruh kata dalam teks agar keputusan model lebih transparan dan mudah dipahami.

a. LIME

LIME (Local Interpretable Model-Agnostic Explanations) adalah metode XAI yang menjelaskan prediksi model dengan menyoroti kata-kata yang paling berpengaruh. LIME bekerja dengan memodifikasi teks secara kecil-kecilan untuk melihat perubahan prediksi, sehingga dapat menunjukkan alasan model memilih suatu label secara lokal pada satu sampel input.



Text with highlighted words

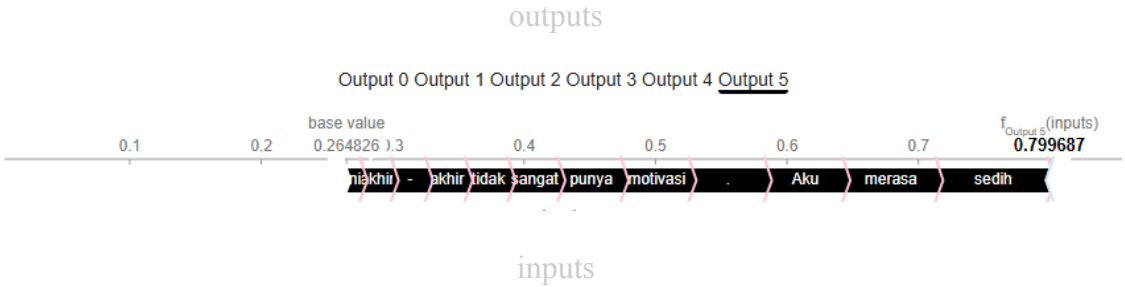
Aku merasa sangat sedih akhir-akhir ini dan tidak punya motivasi.

Berdasarkan hasil LIME tersebut, ditampilkan visualisasi kontribusi setiap kata terhadap prediksi model. Kata “sedih” memiliki pengaruh paling besar dalam mendorong model memilih label *sedih*, diikuti oleh kata “tidak” dan “motivasi”. XAI juga menampilkan highlight pada kata-kata tersebut dalam teks sehingga memudahkan interpretasi keputusan model. Visualisasi ini menunjukkan alasan model memberikan prediksi, bukan berupa akurasi, melainkan berupa analisis kontribusi fitur dan penjelasan konteks yang mempengaruhi keputusan.

b. SHAP

SHAP (SHapley Additive Explanations) adalah metode XAI yang menjelaskan prediksi model dengan menghitung kontribusi setiap kata menggunakan konsep nilai Shapley dari teori permainan. SHAP menunjukkan seberapa besar sebuah kata meningkatkan atau menurunkan kemungkinan model memilih suatu label, sehingga memberikan penjelasan yang lebih konsisten dan terukur terhadap keputusan model.

PartitionExplainer explainer: 2it [00:33, 33.88s/it]



Aku merasa sangat sedih akhir-akhir ini dan tidak punya motivasi.

Visualisasi tersebut menunjukkan SHAP force plot untuk kelas *sedih* (Output 5), yaitu label yang diprediksi model dengan probabilitas sekitar 0.80. Nilai ini diperoleh dari base value sekitar 0.26 yang kemudian meningkat setelah mempertimbangkan kontribusi setiap kata dalam kalimat. Kata seperti “sedih”, “tidak”, dan “motivasi” memberi pengaruh paling besar sehingga mendorong model memilih kelas sedih. Visualisasi ini menjelaskan alasan model yakin bahwa teks tersebut termasuk kategori *sedih*.

Sebagai rangkuman dari seluruh hasil evaluasi, berikut disajikan tabel perbandingan kinerja semua model dan metode ekstraksi fitur yang digunakan dalam penelitian ini.

Tabel 31. Perbandingan model ML dan DL

Metode	Model	Ektrasi Fitur	Akurasi	Penjelasan
Machine Learning	SVM	Tf-idf	0.57	Pendekatan Machine Learning tidak mampu menangkap konteks emosi secara mendalam karena hanya mengandalkan representasi kata berbasis frekuensi. Walaupun stabil, performanya terbatas karena fitur manual tidak dapat memahami makna kalimat.
		Tf-idf+n-gram	0.55	
		BOW	0.47	
		BOW+n-gram	0.46	
	Logistic Regression	Tf-idf	0.60	
		Tf-idf +n-gram	0.55	
		BOW	0.55	
		BOW with+n-gram	0.54	
	Random Forest	Tf-idf	0.52	
		Tf-idf+n-gram	0.54	
		BOW	0.55	
		BOW+with n-gram	0.54	
Deep Learning Pre-Trained	LSTM	Tidak perlu manual, fitur dipelajari otomatis	0.55	LSTM kurang efektif untuk komentar YouTube yang pendek, variatif, dan banyak bahasa informal. Model kesulitan mengenali pola emosi karena konteksnya tidak panjang dan dataset relatif terbatas. Hal ini membuat LSTM underfitting sehingga performanya tidak optimal.

	IndoBERT Lite Base P1	Tidak perlu ekstraksi fitur, model sudah punya embedding sendiri	0.98	IndoBERT paling cocok untuk project ini dikarenakan Model ini memiliki kemampuan memahami makna kata dalam konteks, termasuk kalimat pendek, ambigu, atau informal.Karena sudah pretrained, IndoBERT membutuhkan sedikit data untuk mencapai performa tinggi. Hasilnya, model memberikan akurasi yang sangat unggul dalam klasifikasi emosi.
	Fine-tuning LoRA with IndoBERT Lite Base P1	Tidak perlu, karena LoRA dan IndoBERT langsung memproses teks menggunakan representasi contextual embedding dari model transformer	0.62	Metode LoRA memungkinkan fine-tuning IndoBERT secara lebih ringan dan efisien karena hanya melatih sebagian kecil parameter. Meskipun akurasinya (0.62) lebih rendah dibanding fine-tuning penuh IndoBERT, LoRA tetap menjadi pilihan efektif ketika sumber daya terbatas.

Berdasarkan tabel perbandingan, dapat disimpulkan bahwa metode Machine Learning memiliki kinerja yang cukup stabil namun terbatas, dengan akurasi berkisar antara 0.46–0.58 tergantung pada teknik ekstraksi fitur yang digunakan. Model Deep Learning berbasis LSTM tidak memberikan peningkatan signifikan dan justru menghasilkan akurasi rendah, yaitu 0.30. Sebaliknya, model Pretrained Transformer seperti IndoBERT Lite Base P1 menunjukkan performa yang sangat unggul dengan akurasi mencapai 0.98. Hal ini membuktikan bahwa model berbasis Transformer yang telah dilatih secara mendalam pada data bahasa Indonesia mampu menangani tugas klasifikasi emosi secara jauh lebih efektif dibandingkan pendekatan Machine Learning maupun Deep Learning tradisional.

Pada metode LoRA yang diterapkan pada IndoBERT, model dapat dilatih lebih efisien karena hanya sebagian kecil parameternya yang disesuaikan. Meskipun akurasinya lebih rendah, yaitu 0.62, LoRA tetap memberikan hasil yang cukup baik dan menjadi alternatif yang efisien ketika sumber daya komputasi terbatas.