Mata Kuliah - Machine Learning

Nama Kelompok : Sandwich Generation

Anggota :

[202110370311458 - Ardhika Yoga Pratama]

[202110370311304 - Anissa Yulidha Rodiyah]

[202110370311308 - Rahajeng Febri Shafiyah]

Berikut ini merupakan update template laporan Mini Project kuliah Machine Learning

Nilai Total: 120 poin

Tahap 0 (poin: 25): Business Objective

1. Melalui analisis data dari Twitter, bisa mengidentifikasi situasi atau peristiwa yang sering dikaitkan dengan peningkatan stress di kalangan generasi sandwich, seperti tanggung jawab ganda seperti mengurus orang tua dan anak, tekanan ekonomi
2. Perusahaan dapat merancang kebijakan kerja fleksibel, seperti waktu kerja yang lebih fleksibel atau opsi remote working untuk karyawan dari generasi sandwich. Ini memberi mereka lebih banyak waktu untuk mengelola tanggung jawab pribadi dan keluarga tanpa mengorbankan pekerjaan. Selanjutnya, program kesejahteraan dari perusahaan dapat mengembangkan program kesejahteraan yang lebih spesifik untuk karyawan generasi sandwich, seperti program konseling di tempat kerja atau akses ke layanan kesehatan mental dengan biaya terjangkau.
3. Perusahaan teknologi atau startup dapat menggunakan data ini untuk mengembangkan aplikasi atau layanan kesehatan mental yang terfokus pada generasi sandwich. Dengan memahami kapan dan mengapa stres mereka meningkat, perusahaan dapat menciptakan fitur-fitur yang relevan seperti sesi konseling, meditasi, latihan mindfulness, atau saran untuk mengelola tanggung jawab ganda.

Tahap 1 (poin: 25): Original Data

- Urgensi topik/kasus yang dipilih.

  Generasi sandwich adalah istilah yang digunakan untuk menggambarkan individu yang berada di tengah dua generasi, yaitu mereka yang merawat orang tua yang sudah lanjut usia sambil juga membesarkan anak-anak. Orang-orang dalam generasi ini sering merasa "terjepit" antara tanggung jawab mengurus kedua kelompok ini. Mereka menghadapi berbagai tantangan, terutama dalam hal keuangan, seperti biaya pendidikan anak dan perawatan kesehatan orang tua. Dengan banyaknya tanggung jawab yang dihadapi, mereka sering mengalami stres, kecemasan, dan masalah kesehatan mental lainnya. Beban yang berat ini juga dapat mempengaruhi hubungan mereka dengan pasangan, anak, dan orang tua, sehingga berpotensi menyebabkan ketegangan dalam keluarga. Selain itu, tidak semua anggota generasi sandwich memiliki akses yang memadai terhadap sumber daya, seperti dukungan dari keluarga lain, layanan kesehatan mental, atau bantuan finansial. Hal ini dapat membuat mereka merasa sendirian dalam menghadapi beban yang berat.

- Data yang digunakan.

  - Data yang digunakan dalam analisis ini adalah kumpulan tweet yang berasal dari platform media sosial Twitter. Kumpulan data ini berisi informasi yang mencerminkan pemikiran, perasaan, dan pengalaman pengguna Twitter terkait dengan stres yang dialami oleh generasi sandwich.

  - Sebutkan dan jelaskan atribut pada data tersebut.

Nama Attribute	**Keterangan**
conversation_id_str	ini adalah id unik untuk setiap percakapan di Twitter.
created_at	tanggal dan waktu saat tweet dibuat.
favorite_count	jumlah "likes" yang diterima tweet tersebut. Ini menunjukkan seberapa

	populer atau diterima tweet oleh pengguna lain.
full_text	isi berbagai opini dari pengguna twitter
id_str	ID unik untuk setiap tweet.
image_url	url yang mengarah ke gambar yang mungkin dilampirkan dalam tweet.
in_reply_to_screen_name	nama pengguna yang di-reply oleh tweet ini
lang	bahasa yang digunakan dalam tweet.
location	lokasi pengguna saat membuat tweet
quote_count	jumlah kali tweet ini dijadikan kutipan oleh pengguna lain. Ini menunjukkan seberapa sering konten tweet dianggap relevan atau menarik untuk dibagikan kembali.
reply_count	jumlah balasan yang diterima tweet. Ini bisa memberikan indikasi seberapa banyak diskusi yang dipicu oleh tweet tersebut.
retweet_count	jumlah kali tweet ini di-retweet oleh pengguna lain. Ini menunjukkan seberapa viral atau berpengaruhnya tweet tersebut.

tweet_url	url yang mengarah ke tweet tersebut di Twitter.
user_id_str	id unik untuk pengguna twitter yang membuat tweet
username	nama pengguna twitter yang membuat tweet.

- Jelaskan data mining task yang akan digunakan

  Dalam analisis data ini, kita menggunakan task classification untuk mengelompokkan data ke dalam kategori yang telah ditentukan, yaitu stress, tidak stress, dan others. Proses ini bertujuan untuk memahami kondisi stress dan tidak stress yang dialami oleh individu, terutama dalam konteks generasi sandwich.

- Sumber data (sertakan link).

Data yang digunakan dalam analisis ini berasal dari Twitter, yang dikumpulkan melalui proses crawling menggunakan pemrograman Python. Proses ini secara otomatis mengumpulkan data dengan memanfaatkan API Twitter, menghasilkan total 3.629 baris dan 15 kolom.

Berikut adalah tautan hasil crawling yang dapat diakses :

Crawling Generasi Sandwich - Colab (google.com)

crawling_sandwich_😋.ipynb - Colab (google.com)

Crawling_GenerasiSandwich - Colab (google.com)

Untuk hasil datanya :

[combined_sandwich - Google Spreadsheet]

Contoh Sampel Dataset:

full_text	sentiment	username
ya Allah aku cape kapan ya aku ga jadi sandwich generation lagi	Stress	noonejess_
Smoga generasi sandwich di mudahkan dan dilancarkan financialnya dr banyak pintu aamiin	Tidak Stress	Diaz22Vika
Persembahan Visinema Pictures ini diadaptasi dari novel best-seller karya Almira Bastari yang membahas fenomena sandwich generation. #HomeSweetLoan https://t.co/3sMeESUksr	Others	film_bioskop

Tahap 2 (poin: 10): Target Data (Optional)

Pada tahap analisis ini, kita akan fokus hanya pada kolom full_text dari dataset yang telah dikumpulkan. Kolom full_text berisi teks yang mencerminkan pengalaman, pandangan, dan perasaan individu mengenai situasi yang dihadapi oleh generasi sandwich. Dengan memfokuskan analisis pada full_text, kita dapat melakukan analisis emosi dan klasifikasi tingkat stres yang dialami oleh pengguna. Klasifikasi ini akan membantu kita memahami apakah komentar tersebut bersifat netral, other, atau menggambarkan tingkat stres yang rendah, sedang, atau tinggi.

- **EDA**

1) Wordcloud

Gambar 1. Wordcloud

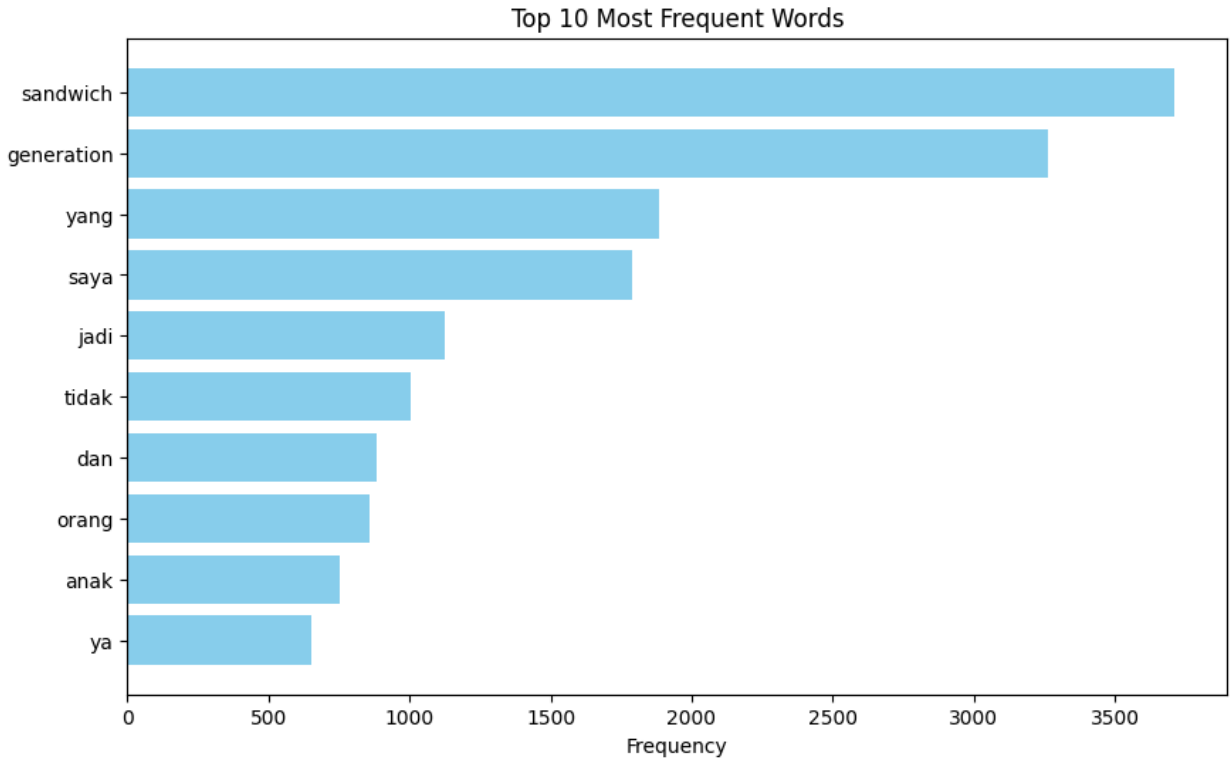
Hasil visualisasi word cloud menunjukkan kata-kata yang paling sering muncul dalam pembicaraan tentang generasi sandwich. Salah satu kata yang paling besar adalah "sandwich generation," yang menunjukkan bahwa banyak orang menyadari dan membahas istilah ini. Ini berarti banyak orang berbicara tentang tantangan yang dihadapi oleh mereka yang merawat orang tua dan anak sekaligus.

Kata "saya" juga muncul sering, yang menunjukkan bahwa banyak orang berbagi pengalaman pribadi mereka. Mereka menceritakan bagaimana rasanya berada dalam situasi ini dan perasaan yang mereka alami. Selain itu, kata "keluarga" juga banyak disebutkan, yang menunjukkan bahwa hubungan dengan anggota keluarga sangat penting dalam konteks ini. Ini berarti bahwa orang-orang saling membantu satu sama lain dalam menghadapi tanggung jawab yang berat.

Di sisi lain, ada juga kata-kata seperti "tidak," "masalah," "bantu," "berat," dan "susah." Kata-kata ini menunjukkan bahwa banyak orang merasa stres dan menghadapi

kesulitan dalam menjalani peran mereka. Mereka mungkin merasa perlu dukungan karena beban yang mereka tanggung.

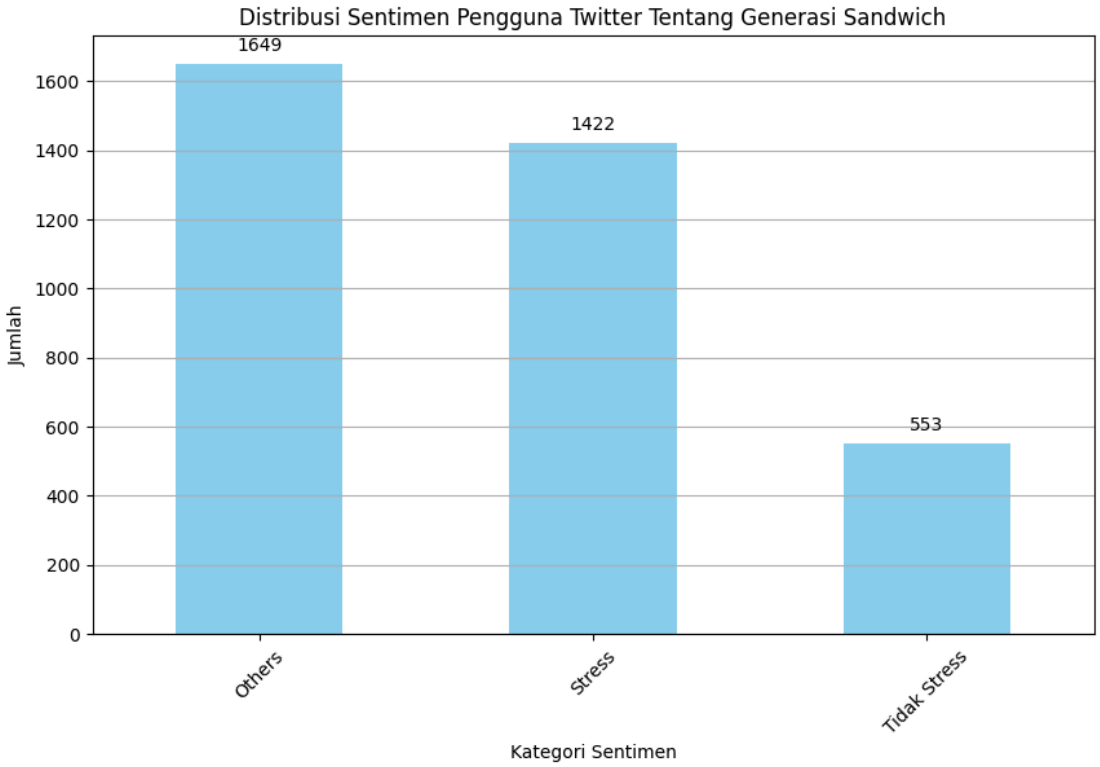
2) Barchart



Gambar 2. 10 kata yang paling sering muncul

Hasil visualisasi dalam bentuk grafik batang ini menggambarkan sepuluh kata yang paling sering digunakan dalam diskusi tentang generasi sandwich di Twitter. Kata "sandwich" dan "generation" muncul paling dominan, menunjukkan bahwa istilah ini menjadi fokus utama pembicaraan, yang menggambarkan orang-orang yang berada di tengah antara merawat orang tua dan anak-anak. Selain itu, kata "yang," "saya," dan "tidak" mencerminkan banyaknya pengalaman pribadi yang dibagikan oleh pengguna Twitter, dengan "tidak" menandakan adanya tantangan atau kesulitan yang dihadapi dalam peran ini.

3) Distribusi label sentiment



Gambar 3. Distribusi Sentimen

Grafik di atas menunjukkan distribusi sentimen pengguna Twitter tentang generasi sandwich, dengan kategori sentimen yang dilabeli sebagai Others, Stress, dan Tidak Stress. Kategori Others dan Stress mendominasi dengan jumlah tweet yang kurang lebih sama, menunjukkan bahwa sebagian besar pengguna memiliki sentimen yang termasuk dalam kategori ini. Kategori yang paling sedikit adalah Tidak Stress dengan 553 tweet. Distribusi ini menunjukkan variasi sentimen yang cukup signifikan, di mana kategori Others paling banyak muncul.

Tahap 3-4 (poin: 25): Data Pre-processing & Transformation

Beberapa teknik yang bisa digunakan yaitu (tentu sesuai kondisi dan kebutuhan):

- data integration

  Pada proses ini, tiga file CSV yang berisi hasil crawling data dari Twitter digabungkan menjadi satu dataset utama. Penggabungan file CSV dilakukan agar seluruh data yang diperoleh bisa dianalisis secara bersamaan dalam satu kerangka data (dataframe) tanpa perlu memisahkannya berdasarkan sumber file.

Table 1. Sample Dataset CSV 1.

full_text	sentiment	username
Mau nonton home sweet loan Tapi takut ketrigger Dasar makhluk-makhluk sandwich generation	Others	0x_annelish14
Hari ini gaes! Buat sandwich generation pejuang rupiah pencari cuan pejuang KPR kaum perintis bukan pewaris bakalan relate banget sama film ini. PS: jangan lupa bawa tissue ya	Others	ardibhironx
@ponakan_sirius Imo kalo km nanggung ortu/keluarga berarti ya sandwich generation. Kamu bukan sandwich generation kalo ortumu ada penghasilan sendiri dan km gak wajib ngasih (ngasih ataupun ngga ya suka suka).	Tidak stress	spadameow

Table 2. Sample Dataset CSV 2.

full_text	sentiment	username
@tanyarlfes Sorry oot nderr barang kali ada tmn2 yg mau.. silahkan..	Others	Fajaragungmu
@tanyarlfes Untuk gua pribadi bantu orang tua itu menyenangkan. Karena emang ortu gua gabutuh dibantu.. Malah gua yg sering dibantu sampe sekarang. Tapi bukan berarti semua orang berpengalaman sama kyk gua. Gua tau ada orang diluar sana yang ortunya gak sebaik itu nuntut hal yg-	Tidak Stress	Lostranauts_300

@tanyarlfes Jadi sandwich gen itu gapapa tp orang2 yg nganggep mereka ga bersyukur krn ngeluh berat buat biayain keluarga yg bikin masalah manusiawi aja kl emg mereka blm terbiasa dan merasa cape atau belm ikhlas. Stage of life tiap orang itu beda2.	Stress	kurakuraninjas

Table 3. Sample Dataset CSV 3.

full_text	sentiment	username
@VisinemaID Mematahkan opini novel dijadikan film tidak menjamin ekspektasi pembaca itu SALAH meski kemiripannya sekitar 70% justru sisa 30% di tempat benar semakin menyoroti sosok Kaluna si generasi sandwich tokoh utama yg dekat di sekitar kita atau justru kita adalah Kaluna.	Others	anaazwan
gamau jadi generasi sandwich tp gua doang yg bisa diandelin keluarga yaudadah mau gimana lagi	Stress	siraditn
@gustime_ kata - kata anda sangat betul dan tepat sasaran. saya dari kaum keluarga yg tidak harmonis dan anak tunggal + generasi sandwich sangat terenyuh mental dan bo'ol nya membaca kalimat anda.	Stress	RRA_P

Table 4. Sample Dataset CSV Gabungan.

full_text	sentiment	username
Mau nonton home sweet loan Tapi takut ketrigger Dasar makhluk-makhluk sandwich generation	Others	0x_annelish14
Hari ini gaes! Buat sandwich generation pejuang rupiah pencari cuan pejuang KPR kaum perintis bukan pewaris bakalan relate banget sama film ini. PS: jangan lupa bawa tissue ya	Others	ardibhironx
@ponakan_sirius Imo kalo km nanggung ortu/keluarga	Tidak stress	spadameow

- **Data Cleaning (and/or data correcting)**
  - ➢ Pemeriksaan *Missing Value* (Data yang Hilang)

    Penanganan *missing value* harus dilakukan dengan hati-hati, karena dapat mempengaruhi kualitas dan keakuratan analisis data. Berdasarkan hasil pengecekan, bahwa dataset yang digunakan tidak memiliki missing value. Sehingga kita tidak perlu melakukan langkah-langkah penanganan seperti mengisi data yang hilang (imputation), menghapus baris/kolom yang memiliki missing value, atau metode lainnya

  - ➢ Pemeriksaan *Duplicate data*

    Data duplikat terjadi ketika ada baris-baris dalam dataset yang berisi informasi yang sama persis. Keberadaan data duplikat dapat mengganggu hasil analisis karena dapat memberikan hasil yang bias atau tidak akurat. Pada dataset yang digunakan, ditemukan 4 baris data yang duplikat, dan data tersebut telah dihapus untuk memastikan analisis yang lebih valid dan akurat.

Table 5. Dataset Duplikat

full_text	sentiment	username
Cita2ku adalah jadi sandwich generation (generasi yg berada di tengah2 mereka seperti sandwich)	Tidak Stress	greatestsinr
Panggilan kepada para sandwich generation (termasuk diri sendiri)!! Mau nonton duluan kisah yang relate sama perjuangan para generasi rotis lapis seperti kita ini? Cek tanggal dan lokasi Special Screening Film Home Sweet Loan di sini ⬇️ https://t.co/Y9Tk1lF09W	Others	cinema21
@matthew_the_kid @satpamkuburan_ @kegblgnunfaedh apa sih blok dikit2 genz. kek bener aje generasi lu yang bnyak bikin gen z jadi sandwich generation dari generasi lu kocak!!	Stress	aboutplatypus
Panggilan kepada para sandwich generation (termasuk diri sendiri)!! Mau nonton duluan kisah yang relate sama perjuangan para generasi rotis lapis seperti kita ini? Cek tanggal dan lokasi Special Screening Film	Others	cinema21

Home Sweet Loan di sini ⬇️ https://t.co/Y9Tk1lF09W		
Tips Mengelola Keuangan bagi Generasi Sandwich https://t.co/z19ORiR8h4	Others	Fortune_IDN
Cita2ku adalah jadi sandwich generation (generasi yg berada di tengah2 mereka seperti sandwich)	Tidak Stress	greatestsinr
Generasi roti lapis atau sandwich generation sering dihadapkan pada tanggung jawab ganda: merawat orang tua sambil memenuhi kebutuhan	Stress	MandiriTaspen
Tips Mengelola Keuangan bagi Generasi Sandwich https://t.co/z19ORiR8h4	Others	Fortune_IDN
@matthew_the_kid @satpamkuburan_ @kegblgnunfaedh apa sih blok dikit2 genz. kek bener aje generasi lu yang bnyak bikin gen z jadi sandwich generation dari generasi lu kocak!!	Stress	aboutplatypus

➢ Remove Punctuation (pembersihan simbol-simbol)

Simbol-simbol seperti tanda baca, emotikon, dan karakter khusus tidak memiliki makna penting dalam konteks pemahaman teks atau klasifikasi sentimen, sehingga dihapus agar model atau analisis yang digunakan dapat fokus pada kata-kata yang lebih relevan.

Table 6. Dataset Sebelum Dilakukan Pembersihan Simbol-simbol

full_text	sentiment	username
@matthew_the_kid @satpamkuburan_ @kegblgnunfaedh apa sih blok dikit2 genz. kek bener aje generasi lu yang bnyak bikin gen z jadi sandwich generation dari generasi lu kocak!!	Stress	aboutplatypus
Tips Mengelola Keuangan bagi Generasi Sandwich https://t.co/z19ORiR8h4	Others	Fortune_IDN
@pratama_____ berguna banget kK itu buat saya .. apalagi sandwich generation yang gajian numoang lewat . stress parah	Others	lisalebe

Table 7. Dataset Sesudah Dilakukan Pembersihan Simbol-simbol

full_text	sentiment	username
apa sih blok dikit2 genz. kek bener aje generasi lu yang bnyak bikin gen z jadi sandwich generation dari generasi lu kocak	Stress	aboutplatypus
Tips Mengelola Keuangan bagi Generasi Sandwich	Others	Fortune_IDN
berguna banget kK itu buat saya apalagi sandwich generation yang gajian numoang lewat. stress parah	Others	lisalebe

➢ Case Folding

Pada proses ini merubah huruf kapital pada semua data menjadi huruf kecil. Tujuan dilakukannya hal tersebut supaya kata-kata yang identik tidak dikira berbeda karena variasi pada penerapan huruf kapital

Table 8. Sebelum Dataset Dilakukan Case Folding

full_text	sentiment	username
apa sih blok dikit2 genz. kek bener aje generasi lu yang bnyak bikin gen z jadi sandwich generation dari generasi lu kocak	Stress	aboutplatypus
Tips Mengelola Keuangan bagi Generasi Sandwich	Others	Fortune_IDN
berguna banget kK itu buat saya apalagi sandwich generation yang gajian numoang lewat. stress parah	Others	lisalebe

Table 9. Sesudah Dataset Dilakukan Case Folding

full_text	sentiment	username
apa sih blok dikit2 genz. kek bener aje generasi lu yang bnyak bikin gen z jadi sandwich generation dari generasi lu kocak	Stress	aboutplatypus
tips mengelola keuangan bagi generasi sandwich	Others	Fortune_IDN

berguna banget kk itu buat saya apalagi sandwich generation yang gajian numoang lewat. stress parah	Others	lisalebe

- data normalization
  - ➢ Normalisasi teks

    Mengubah kata-kata yang disingkat atau tidak baku menjadi kata-kata yang lebih formal atau standar. Normalisasi menciptakan konsistensi dalam teks, yang sangat penting untuk pemrosesan skala besar. Data yang tidak konsisten akan menyebabkan banyak variasi yang tidak perlu dan sulit ditangani oleh mesin, terutama jika ada perbedaan penulisan akibat singkatan atau penggunaan kata informal.

Table 10. Dataset sebelum dinormalisasi

full_text	sentiment	username
imo kalo km nanggung ortu keluarga berarti ya sandwich generation kamu bukan sandwich generation kalo ortumu ada penghasilan sendiri dan km gak wajib ngasih ngasih ataupun ngga ya suka suka	Tidak Stress	spadameow
berguna banget kk itu buat saya apalagi sandwich generation yang gajian numoang lewat. stress parah	Others	lisalebe
apa lagi yg ga sandwich generation	Tidak Stress	ieyainaja

Table 11. Dataset yang telah dilakukan teks normalisasi

full_text	sentiment	username
menurut pendapat saya jika anda nanggung orang tua keluarga berarti ya sandwich generation kamu bukan sandwich generation jika orang tua anda ada penghasilan sendiri dan anda tidak wajib ngasih ngasih ataupun tidak ya suka suka	Tidak Stress	spadameow
kemarin habis ngobrol sama orang yang ternyata dia sandwich generation baru sadar saya kurang bersyukur	Others	julaasple

apa lagi yang tidak sandwich generation	Tidak Stress	ieyainaja

➢ Label Encoding

Salah satu teknik yang digunakan dalam pengolahan data untuk mengonversi variabel kategorikal menjadi variabel numerik

Table 12. Keterangan Sentimen

sentiment (sebelum encode)	sentiment (setelah encode)
Others	0
Stress	1
Tidak Stress	2

- data transformation
  - ➢ Tokenisasi

    Memisahkan kalimat atau teks menjadi frasa atau kata. Tujuannya agar mudah dalam tahapan selanjutnya seperti perhitungan jumlah frasa, pembobotan pada frasa

Table 13. Dataset sebelum dilakukan tokenisasi

full_text	sentiment	username
menurut pendapat saya jika anda nanggung orang tua keluarga berarti ya sandwich generation kamu bukan sandwich generation jika orang tua anda ada penghasilan sendiri dan anda tidak wajib ngasih ngasih ataupun tidak ya suka suka	Tidak Stress	spadameow
kemarin habis ngobrol sama orang yang ternyata dia sandwich generation baru sadar saya kurang bersyukur	Tidak Stress	julaasple
apa lagi yang tidak sandwich generation	Tidak Stress	ieyainaja

Table 14. Dataset setelah dilakukan tokenisasi

full_text	sentiment	username
menurut,pendapat,saya,jika,anda,nanggung,orang,tua,keluarga,berarti,ya,sandwich,generation,kamu,bukan,sandwich,generation,jika,orang,tua,anda,ada,penghasilan,sendiri,dan,anda,tidak,wajib,ngasih,ngasih,ataupun,tidak,ya,suka,suka	Tidak Stress	spadameow
kemarin,habis,ngobrol,sama,orang,yang,ternyata,dia,sandwich,generation,baru,sadar,saya,kurang,bersyukur	Others	julaasple
apa,lagi,yang,tidak,sandwich,generation	Tidak Stress	ieyainaja

➢ Stopword

Penggunaan stopword pada preprocessing untuk menghapus kata-kata yang tidak bermakna pada suatu kalimat atau teks seperti 'lhoo', 'awokwokwok', 'kakk', 'kak', 'nih', 'hehe', 'mah', 'ya', 'pas', 'tuh', 'nder', 'pun', 'si', 'wkwk', 'gaes'

Table 15. Dataset sebelum dilakukan stopword

full_text	sentiment	username
memang,sedih,banget,jujur,jadi,anak,kost,yang,sandwich,generation,tuh	Stress	ggukiesee
bisakah,saya,keluar,dari,zona,sandwich,generation,jika,nih,gaji,buat,sendiri,mah,setiap,bulan,bisa,liburan	Stress	workaholicniki
sebagai,sandwich,generation,saya,tuh,takut,banget,jika,nanti,duit,saya,banyak,kegocek,di,pertengahan,jalan,sumpah	Stress	bunny_chuuu

Table 16. Dataset setelah dilakukan stopword :

full_text	sentiment	username
memang,sedih,banget,jujur,jadi,anak,kost,yang,sandwich,generation	Stress	ggukiesee
bisakah,saya,keluar,dari,zona,sandwich,generation,jika,gaji,buat,sendiri,setiap,bulan,bisa,liburan	Stress	workaholicniki
sebagai,sandwich,generation,saya,takut,banget,jika,nanti,duit,saya,banyak,kegocek,di,pertengahan,jalan,sumpah	Stress	bunny_chuuu

- Feature Extraction

  Pada penelitian ini feature selection dilakukan menggunakan

  a. Bag of Word (BOW)

  Dalam klasifikasi dokumen, BoW adalah vektor jumlah kemunculan kata, yang disebut juga histogram dokumen tersebut. Misalkan memiliki kosakata, yaitu daftar semua kata unik yang ada di seluruh kumpulan teks yang dimiliki. Sebut jumlah kata unik dalam kosakata ini sebagai n. Jika mengambil kata ke-i dari kosakata, kita sebut kata itu sebagai V. Frekuensi kata V dalam dokumen D kita sebut f. Bag of Words (BoW) dari dokumen D adalah representasi yang berbentuk vektor dengan panjang n. Dalam vektor ini, setiap elemen menunjukkan seberapa sering kata tertentu muncul dalam dokumen tersebut.

  $$BOW = (f_{d1}, f_{d2}, \ldots\ldots\ldots, f_{dn}) \quad (1)$$

  Sampel kalimat sebelum BoW:

  - text 1 : apa lagi yang tidak sandwich generation
  - text 2 : semangat sandwich generation kamu pasti bisa
  - text 3 : semangat ya sandwich generation

  Table 17. Sampel BoW

apa	lagi	pasti	ya	yang	tidak	sandwich	generation	semangat	kamu	bisa
1	1	0	0	1	1	1	1	0	0	0

0	0	1	0	0	0	1	1	1	1	1
0	0	0	1	0	0	1	1	1	0	0

b. **TF-IDF**

TF-IDF (Term Frequency-Inverse Document Frequency) adalah metode yang paling sering digunakan dalam pengolahan bahasa alami (NLP) untuk mengubah dokumen teks menjadi bentuk representasi matriks vektor. Representasi TF-IDF menunjukkan sejauh mana pentingnya suatu kata dalam konteks kumpulan dokumen dibandingkan dengan dokumen secara individual. TF-IDF digunakan sebagai pembobotan fitur metode.

$$Tf.IDF = TF_{ij} \times IDF_{ij} = TF_{ij} \times log\frac{N}{DFJ} \quad (2)$$

Keterangan :

- N : jumlah dokumen dalam koleksi
- TF : frekuensi istilah
- IDF : frekuensi dokumen terbalik

Sampel kalimat sebelum TF-IDF:

- text 1 : apa lagi yang tidak sandwich generation
- text 2 : semangat sandwich generation kamu pasti bisa
- text 3 : semangat ya sandwich generation

Table 18. Sampel TF-IDF

apa	lagi	pasti	ya	yang	tidak	sandwich	generation	semangat	kamu	bisa
0.461381	0.461381	0	0	0.461381	0.461381	0.272499	0.272499	0	0	0
0	0	0.483591	0	0	0	0.285617	0.285617	0.367784	0.483591	0.483591

0	0	0	0.66284	0	0	0.391484	0.391484	0.504107	0	0

c. N-gram

Konsep N-Gram berkaitan dengan sekumpulan kata yang berasal dari kalimat tertentu. Memanfaatkan teknik N gram memungkinkan untuk menghasilkan kata atau karakter. N-Gram melibatkan pertimbangan kata-kata sebelumnya dan berikutnya. Klasifikasi N-Gram dapat didasarkan pada jumlah segmen kata atau substring yang dihasilkan, seperti Unigram, Bigram, Trigram, dan sebagainya, sesuai dengan nilai 'n' dalam N-Gram.

$$ngram_k = x - (n - 1) \quad (3)$$

x dilambangkan frekuensi kata pada kalimat k

- Table 19. Sample 2-Grams

apa lagi	**generation kamu**	**kamu pasti**	**lagi yang**	**pasti bisa**	**sandwich generation**
1	0	0	1	0	1
0	1	1	0	1	1
0	0	0	0	0	1

semangat sandwich	semangat ya	tidak sandwich	ya sandwich	yang tidak
0	0	1	0	1
1	0	0	0	0
0	1	0	1	0

- Table 20. Sample 3-Grams

apa lagi yang	generation kamu pasti	kamu pasti bisa	lagi yang tidak	sandwich generation kamu
1	0	0	1	1
0	1	1	0	0
0	0	0	0	0

semangat sandwich generation	semangat ya sandwich	tidak sandwich generation	ya sandwich generation	yang tidak sandwich
0	0	1	0	1

1	0	0	0	0
0	1	0	1	0

Tahap 5 (poin: 25): Data Mining

- Algoritma data mining yang digunakan (sesuai data mining task).
    - **Naive Bayes**

        Klasifikasi Naive Bayes didasarkan pada Teorema Bayes dengan asumsi independensi antar prediktor. Model Naive Bayes mudah dibangun karena tidak memerlukan estimasi parameter yang rumit secara iteratif, sehingga sangat berguna untuk dataset berukuran besar. Meskipun sederhana, model ini sering memberikan hasil yang baik dan banyak digunakan karena sering kali mampu mengungguli metode klasifikasi yang lebih kompleks.

        Berikut ini adalah rumus Teorema Bayes yang menjadi dasar dari algoritma Naive Bayes:

$$P(c \mid x) = \frac{P(x \mid c) \times P(c)}{P(x)} \quad (4)$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times P(x_n \mid c) \times P(c) \quad (5)$$

        Keterangan:

        - $P(c \mid X)$: probabilitas posterior dari kelas (target) diberikan prediktor (atribut).
        - $P(c)$: probabilitas prior dari kelas.
        - $P(x \mid c)$: kemungkinan, yaitu probabilitas prediktor diberikan kelas.
        - $P(X)$: probabilitas prior dari prediktor.

    - **Logistic Regression**

        Regresi Logistik adalah jenis model klasifikasi parametrik yang digunakan ketika variabel respons bersifat kategorikal. Ide dasar dari Regresi Logistik adalah menemukan hubungan antara fitur (prediktor) dan probabilitas suatu hasil tertentu. Dalam Regresi Logistik, fungsi sigmoid digunakan untuk memetakan prediksi ke dalam bentuk probabilitas. Berikut ini adalah rumus Regresi Logistik:

$$Y_i = \beta_0 + \beta_1 + \varepsilon_i \quad (6)$$

Keterangan:

- Y_i (Dependent Variable): Variabel dependen atau variabel terikat, yaitu hasil atau keluaran yang ingin diprediksi oleh model berdasarkan variabel bebas.
- β_0 (Population Y Intercept): Intersep, yaitu nilai konstanta ketika variabel bebas X_i bernilai 0. Ini menunjukkan nilai awal dari variabel dependen.
- β_1 (Population Slope Coefficient): Koefisien kemiringan, yang menggambarkan seberapa besar perubahan pada Y_i setiap kali X_i bertambah satu unit. Koefisien ini menunjukkan kekuatan dan arah hubungan antara variabel bebas dan variabel terikat.
- X_i (Independent Variable): Variabel independen atau variabel bebas, yaitu faktor yang digunakan untuk memprediksi atau mempengaruhi nilai dari variabel dependen.
- ε_i (Random Error Term): Komponen galat acak yang menggambarkan perbedaan antara nilai yang diprediksi oleh model dan nilai aktual. Ini merepresentasikan faktor-faktor lain yang mempengaruhi Y_i tetapi tidak dimasukkan dalam model.

o **Random Forest**

Random Forest adalah algoritma yang membangun beberapa pohon untuk klasifikasi dan regresi, di mana pemisahan node dilakukan dengan menggunakan algoritma yang dioptimalkan untuk meminimalkan kehilangan squared-error. Algoritma ini bekerja dengan memilih atribut secara acak dan menerapkan metode CART untuk membuat pohon keputusan. Kumpulan pohon yang dihasilkan kemudian disebut sebagai "forest."

Berikut ini adalah rumus random forest :

$$F(x) = \frac{1}{j} \sum_{j=1}^{J} h_j(x) \quad (7)$$

Keterangan :

- $F(x)$ = Output dari Random Forest
- J = $jumlah\ pohon\ dalam\ kumpulan$
- h_j = $output\ dari\ pohon\ ke - j$

- Skenario eksperimen sederhana.

- Pelatihan Model (Training)
  - **Naive Bayes**: Melatih model Naive Bayes dengan data latih.
  - **Logistic Regression**: Melatih model regresi logistik menggunakan data latih.
  - **Random Forest**: Melatih model Random Forest dengan jumlah pohon tertentu
- Pengujian Model (Testing)
  - Gunakan data uji untuk memprediksi hasil menggunakan ketiga model.
  - Hitung **akurasi** dari setiap model:

$$Akurasi = \frac{Jumlah\ prediksi\ benar}{Total\ prediksi} \times 100\% \quad (8)$$

- Evaluasi Kinerja
  - Bandingkan akurasi dari ketiga algoritma.
  - Jika diperlukan, tampilkan **confusion matrix** dan metrik lain seperti **precision, recall, dan F1-score** untuk analisis lebih mendalam.
    - $Precision = \frac{TP}{TP + FP}$
    - $Recall = \frac{TP}{TP + FN}$
    - $f\text{-}1\ score = = 2 \times \frac{Recall \times Precision}{Recall + Precision}$

- Table 21. Hasil dan Analisis Algoritma Basic

	BOW				BOW Hyperparameter			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.42	0.49	0.42	0.44	0.42	0.49	0.42	0.44
Logistic Regression	0.61	0.59	0.61	0.60	0.60	0.59	0.60	0.59

Random Forest	0.62	0.61	0.62	0.61	0.59	0.59	0.59	0.59
	TF-IDF				**TF-IDF Hyperparameter**			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.42	0.48	0.42	0.44	0.43	0.50	0.43	0.45
Logistic Regression	0.63	0.62	0.63	0.61	0.62	0.61	0.62	0.61
Random Forest	0.62	0.63	0.62	0.60	0.58	0.60	0.58	0.58
	2-Grams				**2-Grams Hyperparameter**			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.44	0.49	0.44	0.44	0.44	0.49	0.44	0.44
Logistic Regression	0.58	0.57	0.58	0.56	0.55	0.54	0.55	0.55
Random Forest	0.55	0.54	0.55	0.52	0.50	0.53	0.50	0.51
	3-Grams				**3-Grams Hyperparameter**			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.37	0.49	0.37	0.39	0.49	0.29	0.49	0.33
Logistic Regression	0.58	0.57	0.58	0.56	0.59	0.58	0.59	0.57

Random Forest	0.55	0.54	0.55	0.52	0.59	0.58	0.59	0.57

- Table 22. Perbandingan akurasi Feature Extraction dengan algoritma klasifikasi

Perbandingan akurasi Feature Extraction dengan algoritma klasifikasi					
		BoW	**2-Grams**	**3-Grams**	**TF-IDF**
Algoritma	**Naive Bayes**	0.42	0.44	0.37	0.42
	Logistic Regression	0.61	**0.58**	**0.56**	**0.63**
	Random Forest	**0.62**	0.55	0.53	0.61

Logistic Regression dengan TF-IDF menghasilkan akurasi tertinggi sebesar **63%**, menunjukkan kombinasi ini paling optimal untuk klasifikasi teks. Random Forest terbaik dengan BoW (**62%**), sementara Logistic Regression mencapai **58%** dengan 2-Grams. TF-IDF secara keseluruhan unggul sebagai metode feature extraction

- Table 23. Akurasi Feature Extraction dengan Algoritma Klasifikasi Setelah Hyperparameter Tuning

Perbandingan akurasi Feature Extraction tiap algoritma dengan Hyperparameter					
		BoW	**2-Grams**	**3-Grams**	**TF-IDF**
Algoritma	**Naive Bayes**	0.42	0.44	0.49	0.43
	Logistic Regression	0.60	**0.55**	0.49	**0.62**
	Random Forest	**0.59**	0.50	**0.56**	0.58

Berdasarkan hasil perbandingan akurasi, pendekatan terbaik untuk klasifikasi pada dataset ini adalah dengan menggunakan algoritma **Logistic Regression** bersama feature extraction **3-Grams** atau **TF-IDF**, keduanya mencapai akurasi tertinggi sebesar **62%** Ini menunjukkan bahwa Logistic Regression lebih efektif dalam menangkap pola dari data teks dibandingkan dengan Naive Bayes dan Random Forest.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Naive Bayes dengan BOW :

- var_smoothing membantu model menangani data dengan varians yang rendah atau fitur bernilai kecil yang dapat menyebabkan ketidakstabilan.
- Nilai yang dipilih adalah [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3] menangani Varians Rendah seperti nilai kecil seperti 1e-9 atau 1e-8 cocok untuk dataset dengan varians rendah pada fitur dan meningkatkan Generalisasi seperti nilai lebih besar seperti 1e-4 atau 1e-3 membantu mengurangi sensitivitas model terhadap noise.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Logistic Regression dengan BOW :

- C=10.0 model diberikan fleksibilitas lebih besar untuk menyesuaikan data.
- solver='liblinear' untuk dataset kecil hingga menengah dan mendukung regularisasi.
- max_iter=2000 iterasi memastikan algoritme memiliki cukup waktu untuk mencapai konvergensi, terutama pada dataset kompleks.
- penalty='l2' membantu menghindari overfitting dengan menyusutkan parameter yang tidak signifikan.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Random Forest dengan BOW :

- n_estimators=100 dengan 100 pohon, model mampu menangkap kompleksitas data sambil tetap mempertahankan performa komputasi yang efisien.
- max_depth=50 untuk membatasi kedalaman pohon membantu menghindari overfitting, terutama pada data besar atau kompleks.
- min_samples_split=4 untuk mencegah pohon menjadi terlalu spesifik dan meningkatkan generalisasi.
- min_samples_leaf=2 memastikan bahwa setiap pohon tidak membuat daun dengan sampel terlalu kecil, yang dapat menyebabkan overfitting.
- max_features='sqrt' untuk pengaturan umum yang membantu meningkatkan diversitas antar pohon, meningkatkan generalisasi.
- bootstrap=True untuk membantu menciptakan pohon yang lebih bervariasi, meningkatkan kekuatan ensemble.
- class_weight='balanced' untuk mengatasi masalah data yang tidak seimbang dengan memberikan bobot lebih besar pada kelas minoritas.

- random_state=42 untuk memastikan bahwa eksperimen menghasilkan hasil yang sama di setiap eksekusi.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Naive Bayes dengan TF-IDF :

- var_smoothing membantu model menangani data dengan varians yang rendah atau fitur bernilai kecil yang dapat menyebabkan ketidakstabilan.
- Nilai yang dipilih adalah [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3] menangani Varians Rendah seperti nilai kecil seperti 1e-9 atau 1e-8 cocok untuk dataset dengan varians rendah pada fitur dan meningkatkan Generalisasi seperti nilai lebih besar seperti 1e-4 atau 1e-3 membantu mengurangi sensitivitas model terhadap noise.
- RandomizedSearchCV

  Parameter yang digunakan :

  - n_iter=10 untuk membatasi pencarian hingga 10 kombinasi hyperparameter secara acak.
  - cv=5 untuk melakukan validasi silang dengan 5 fold untuk memastikan generalisasi model.
  - scoring='accuracy' menggunakan akurasi sebagai metrik evaluasi untuk memilih hyperparameter terbaik.
  - random_state=42 untuk memastikan reproducibility hasil eksperimen.
  - n_jobs=-1: Memanfaatkan seluruh core CPU untuk mempercepat proses pencarian.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Logistic Regression dengan TF-IDF :

- C=10.0 model diberikan fleksibilitas lebih besar untuk menyesuaikan data.
- solver='liblinear' untuk dataset kecil hingga menengah dan mendukung regularisasi.
- max_iter=2000 iterasi memastikan algoritme memiliki cukup waktu untuk mencapai konvergensi, terutama pada dataset kompleks.
- penalty='l2' membantu menghindari overfitting dengan menyusutkan parameter yang tidak signifikan.
- solver='lbfgs' yaitu solver optimasi berbasis quasi-Newton, yang efisien untuk dataset ukuran menengah hingga besar.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Random Forest dengan TF-IDF :

- n_estimators=200 jumlah pohon biasanya akan meningkatkan akurasi model karena mengurangi risiko overfitting dan memberikan hasil yang lebih stabil.
- max_depth=20 untuk membatasi kedalaman pohon untuk menghindari overfitting.
- min_samples_split=10 minimal jumlah sampel yang diperlukan untuk memisahkan node internal.
- min_samples_leaf=5 Jumlah minimal sampel pada setiap daun untuk mencegah overfitting.
- class_weight='balanced' untuk menyeimbangkan pengaruh kelas yang tidak seimbang berdasarkan distribusi data.
- random_state=42 untuk memastikan hasil yang konsisten/reproducible.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Naive Bayes dengan 2-Grams :

- var_smoothing membantu model menangani data dengan varians yang rendah atau fitur bernilai kecil yang dapat menyebabkan ketidakstabilan.
- Nilai yang dipilih adalah [1e-9] menangani Varians Rendah seperti nilai kecil seperti 1e-9 cocok untuk dataset dengan varians rendah pada fitur.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Logistic Regression dengan 2-Grams :

- C=1.0 digunakan untuk keseimbangan regularisasi, mencegah overfitting namun tetap memungkinkan model untuk belajar dari data.
- penalty='l2' digunakan untuk stabilitas model dan menghindari penghapusan fitur yang berharga, serta mengatasi multikolinearitas.
- solver='lbfgs' dipilih untuk efisiensi dalam memproses dataset besar dengan banyak fitur.
- max_iter=300 memberikan cukup waktu bagi solver untuk mencapai konvergensi, penting untuk memastikan model belajar secara optimal.
- class_weight='balanced' mengatasi ketidakseimbangan kelas, yang sangat penting dalam dataset yang memiliki distribusi kelas yang tidak merata.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Random Forest dengan 2-Grams :

- n_estimators= 250 jumlah pohon biasanya akan meningkatkan akurasi model karena mengurangi risiko overfitting dan memberikan hasil yang lebih stabil.
- max_depth= 35 untuk membatasi kedalaman pohon untuk menghindari overfitting.
- min_samples_split=10 minimal jumlah sampel yang diperlukan untuk memisahkan node internal.
- max_features='sqrt' membantu mengurangi korelasi antara pohon-pohon di dalam hutan, yang meningkatkan keberagaman model dan mengurangi risiko overfitting.
- min_samples_leaf=5 Jumlah minimal sampel pada setiap daun untuk mencegah overfitting.
- class_weight='balanced' untuk menyeimbangkan pengaruh kelas yang tidak seimbang berdasarkan distribusi data.
- random_state=42 untuk memastikan hasil yang konsisten/reproducible.

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Naive Bayes dengan 3-Grams :

- var_smoothing membantu model menangani data dengan varians yang rendah atau fitur bernilai kecil yang dapat menyebabkan ketidakstabilan.
- np.logspace(-15, -1, num=10) menghasilkan nilai-nilai logaritmik untuk var_smoothing

➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Logistic Regression dengan 2-Grams :

- C=1.00 digunakan untuk keseimbangan regularisasi, mencegah overfitting namun tetap memungkinkan model untuk belajar dari data.
- penalty='l2' digunakan untuk stabilitas model dan menghindari penghapusan fitur yang berharga, serta mengatasi multikolinearitas.
- solver='lbfgs' dipilih untuk efisiensi dalam memproses dataset besar dengan banyak fitur.
- max_iter_value=3000 jumlah iterasi maksimum yang digunakan oleh solver untuk menemukan koefisien model yang optimal.
- solver_value = 'liblinear' menentukan algoritma optimasi yang digunakan untuk mencari parameter terbaik dalam model Logistic Regression.

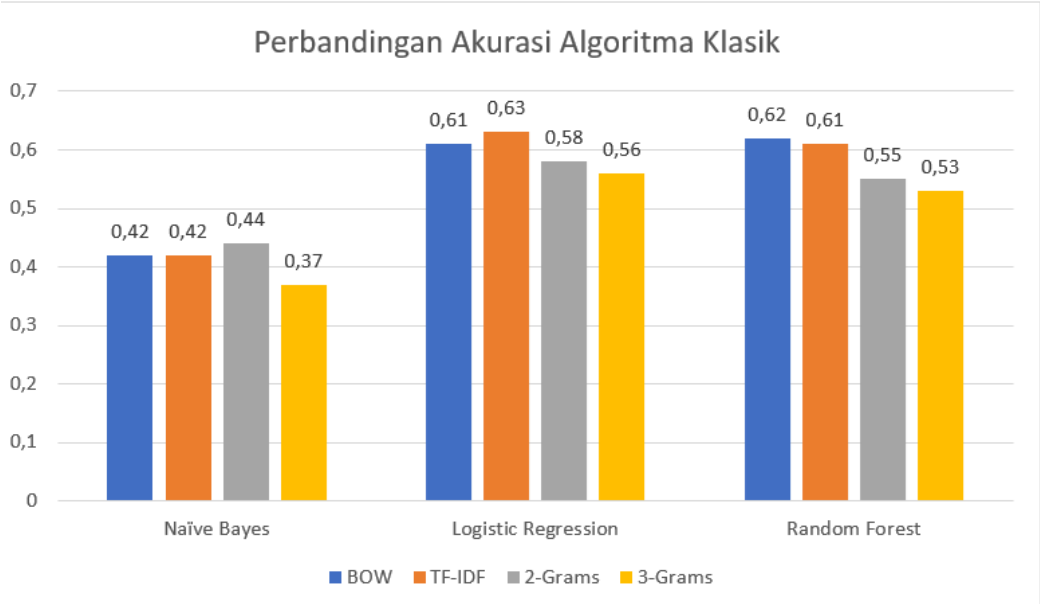
- penalty_value = 'l2' untuk menentukan jenis regularisasi yang digunakan dalam model.

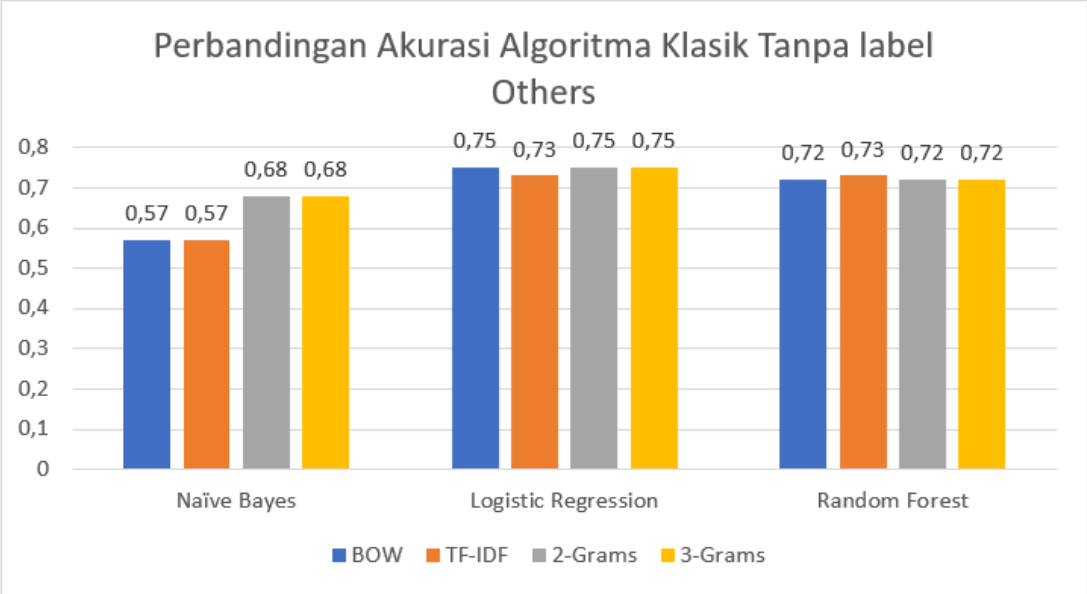
➢ Berikut adalah penjelasan tentang pemilihan nilai hyperparameter tuning pada algoritma Random Forest dengan 3-Grams :

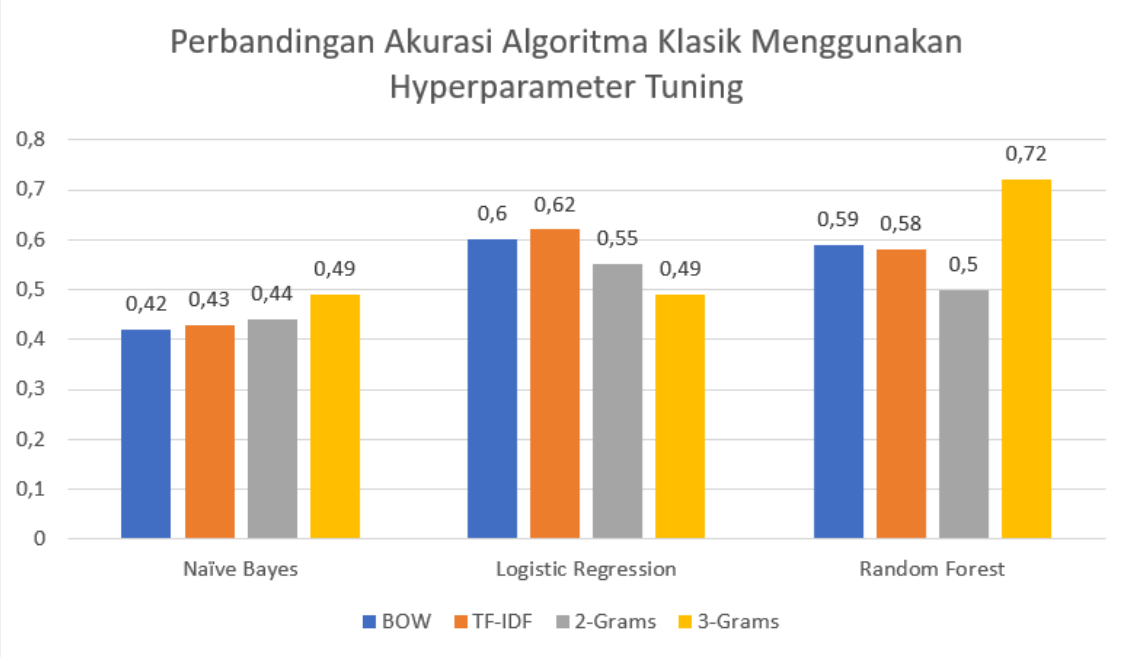
- param_grid

parameter yang digunakan :

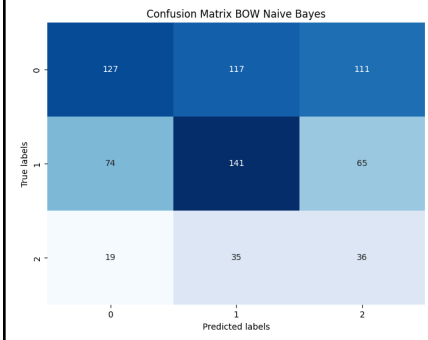
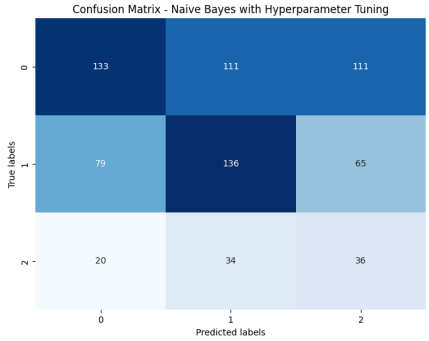
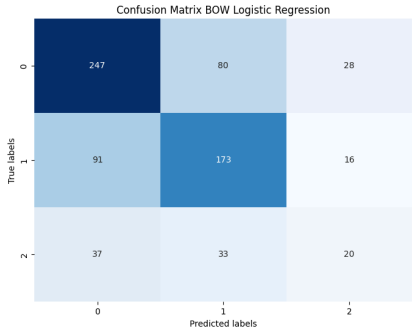
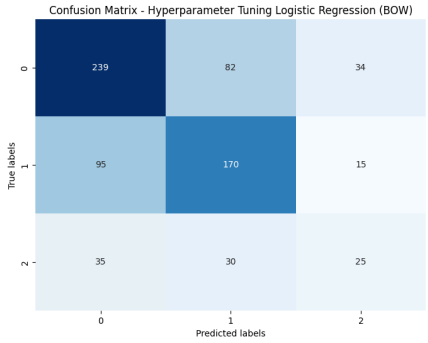
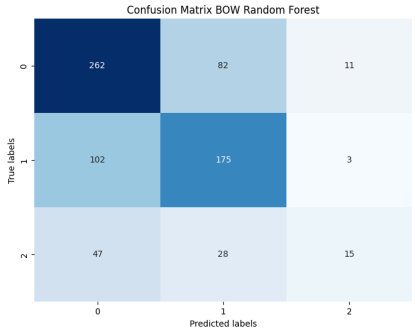
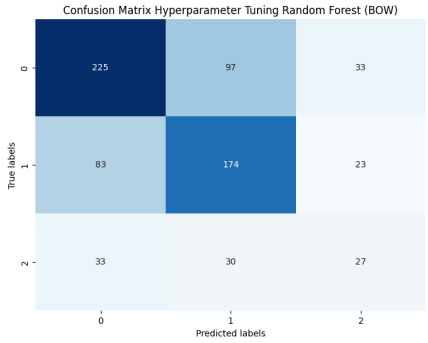
- 'C': [0.1, 1, 10] Dalam param_grid, ada tiga nilai yang diberikan untuk C yaitu 0.1, 1, dan 10. GridSearchCV akan mencoba semua kombinasi nilai ini untuk melihat mana yang memberikan kinerja terbaik.
- 'solver': ['newton-cg', 'liblinear'] GridSearchCV akan mencoba kedua solver ini (newton-cg dan liblinear) untuk mencari mana yang memberikan kinerja terbaik berdasarkan metrik evaluasi yang dipilih (seperti akurasi atau F1-score).
- 'penalty': ['l2'] regularisasi L2 menambahkan penalti terhadap nilai besar dari koefisien model, yang bertujuan untuk menghindari overfitting.

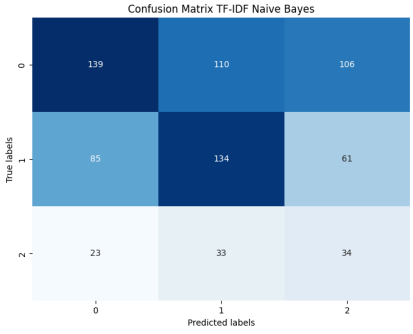
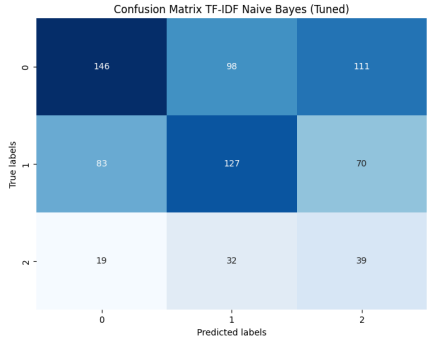
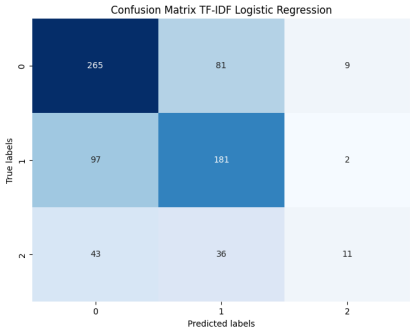
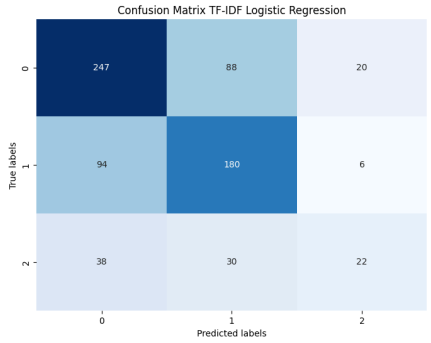
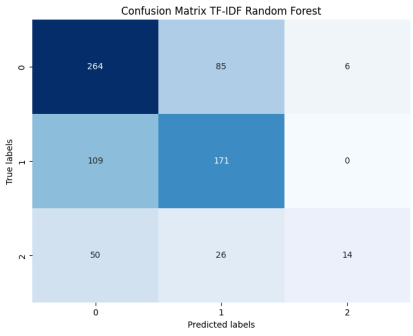
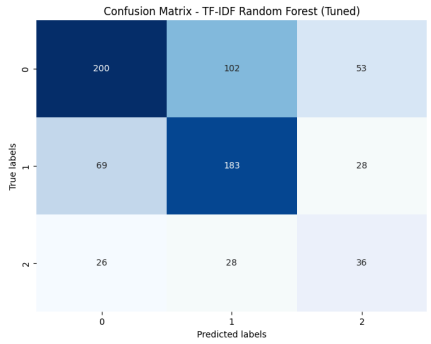


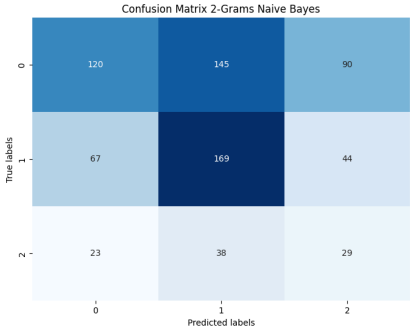
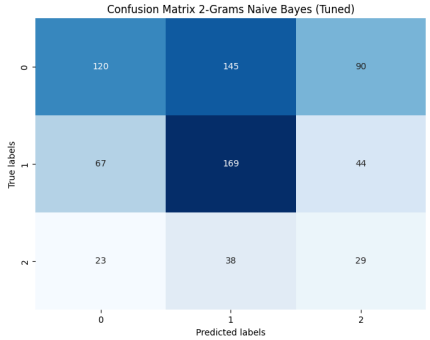
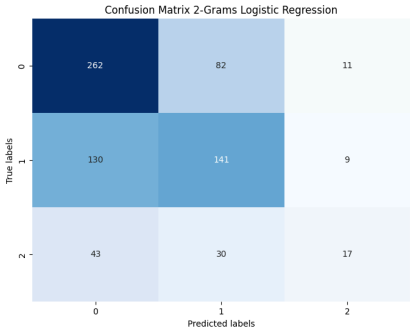
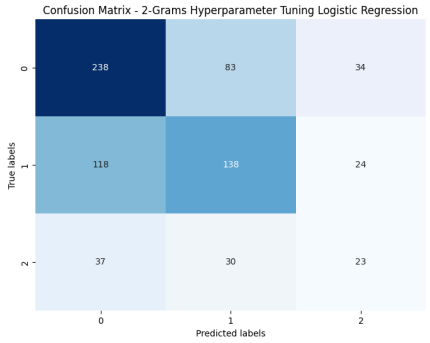
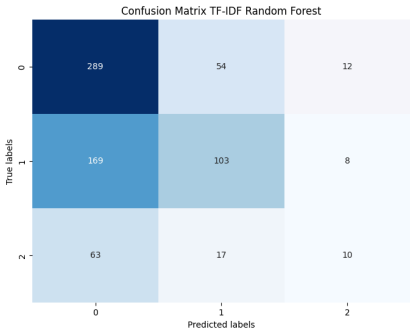
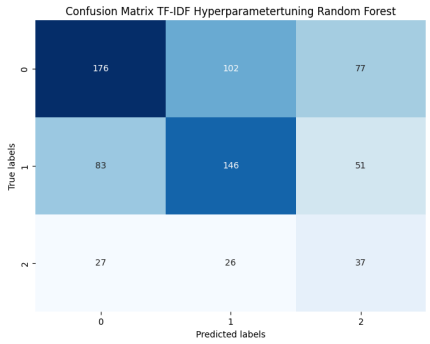

Perbandingan Akurasi Algoritma Klasik Tanpa label Others
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
0,57 0,57 0,68 0,68
0,75 0,73 0,75 0,75
0,72 0,73 0,72 0,72
Naïve Bayes
Logistic Regression
Random Forest
BOW
TF-IDF
2-Grams
3-Grams

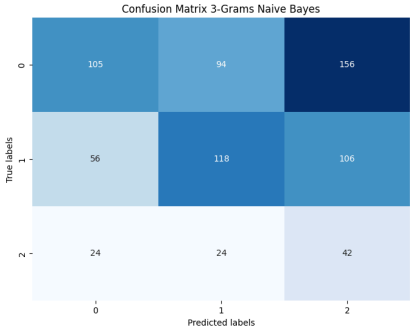
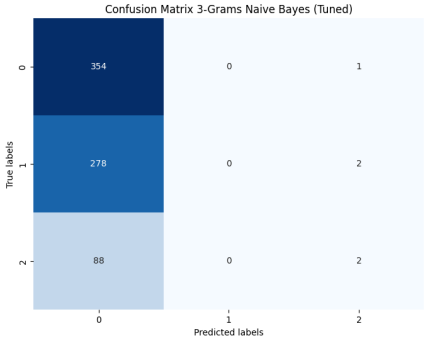
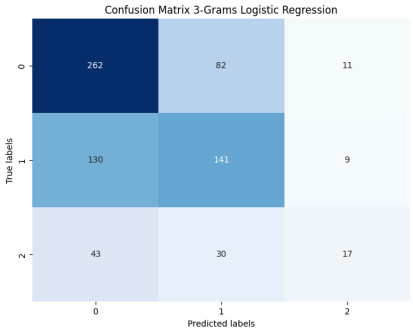
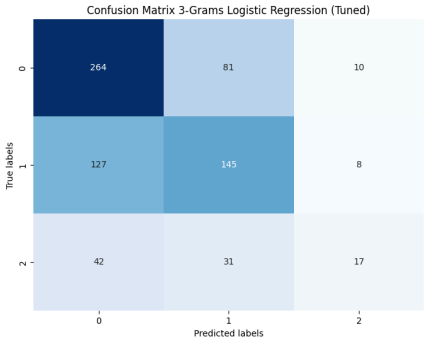
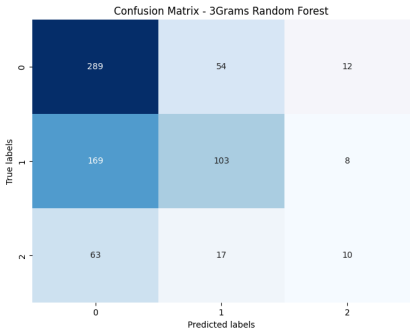
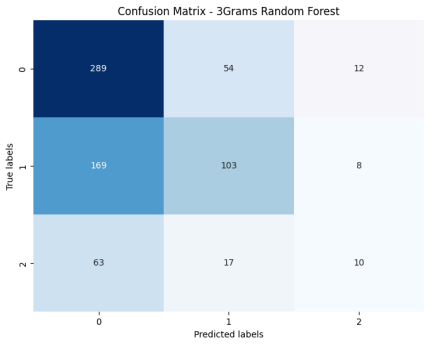

Perbandingan Akurasi Algoritma Klasik Menggunakan Hyperparameter Tuning
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
0,42 0,43 0,44 0,49
0,6 0,62 0,55 0,49
0,59 0,58 0,5 0,72
Naïve Bayes
Logistic Regression
Random Forest
BOW
TF-IDF
2-Grams
3-Grams

- Table 24. Visualisasi Confusion Matrix Feature Extraction Tanpa Label Others

Feature Extraction Dengan Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
BOW		
Naïve Bayers		
Logistic Regression		
Random Forest		

Feature Extraction Dengan Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
TF-IDF		
Naïve Bayers		
Logistic Regression		
Random Forest		

Feature Extraction Dengan Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
2-Grams		
Naïve Bayers		
Logistic Regression		
Random Forest		

Feature Extraction Dengan Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
3-Grams		
Naïve Bayers		
Logistic Regression		
Random Forest		

- Feature Extraction Tanpa Label Others
  - Perbandingan Hasil Classification Report Feature Extraction Tanpa Label Others

Table 25. Perbandingan Hasil tanpa Label Others

	BOW Tanpa Others				BOW Tanpa Others Hyperparameter			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.57	0.52	0.53	0.52	0.58	0.53	0.54	0.53
Logistic Regression	0.75	0.73	0.64	0.65	0.70	0.64	0.61	0.61
Random Forest	0.72	0.69	0.57	0.55	0.75	0.71	0.67	0.68
	TF-IDF				**TF-IDF Hyperparameter**			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.57	0.52	0.52	0.51	0.57	0.52	0.52	0.51
Logistic Regression	0.73	0.83	0.57	0.54	0.73	0.72	0.60	0.60
Random Forest	0.73	0.74	0.57	0.56	0.75	0.71	0.69	0.70
	2-Grams				**2-Grams Hyperparameter**			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.68	0.61	0.61	0.61	0.68	0.61	0.61	0.61

Logistic Regression	0.75	0.79	0.60	0.60	0.73	0.68	0.62	0.63
Random Forest	0.72	0.73	0.57	0.55	0.58	0.63	0.65	0.58
	3-Grams				**3-Grams Hyperparameter**			
Algoritma ML	**Acc.**	**Precision**	**Recall**	**F1**	**Acc.**	**Precision**	**Recall**	**F1**
Naïve Bayers	0.68	0.61	0.61	0.61	0.68	0.61	0.61	0.61
Logistic Regression	0.75	0.79	0.60	0.60	0.75	0.79	0.60	0.60
Random Forest	0.72	0.73	0.57	0.55	0.75	0.79	0.60	0.60

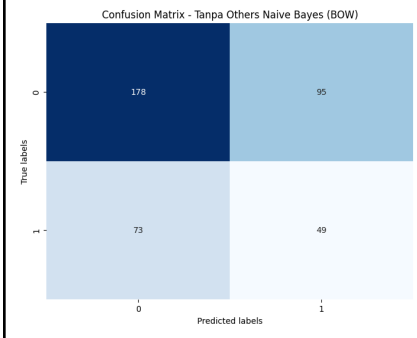
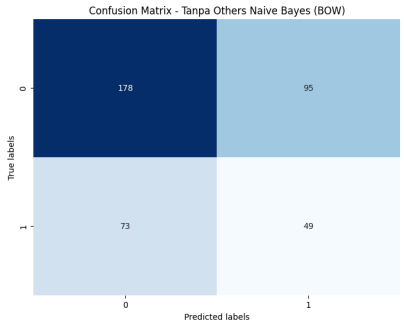
- Table 26. Perbandingan akurasi Feature Extraction dengan algoritma klasifikasi (Tanpa Others)

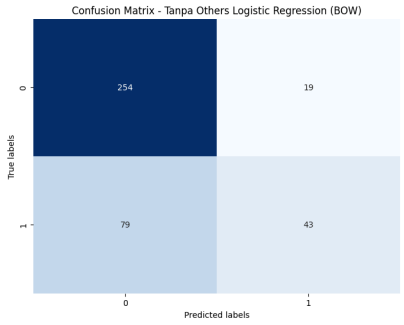
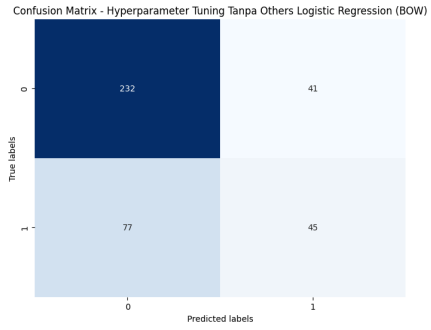
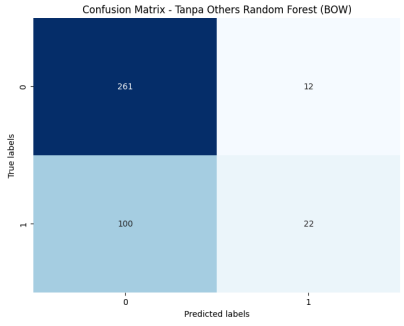
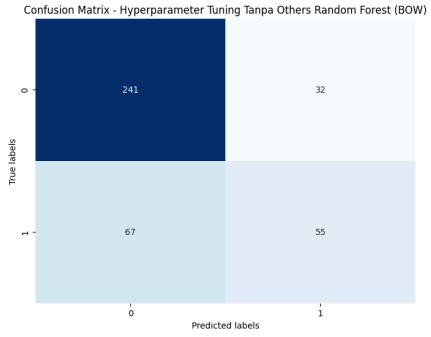
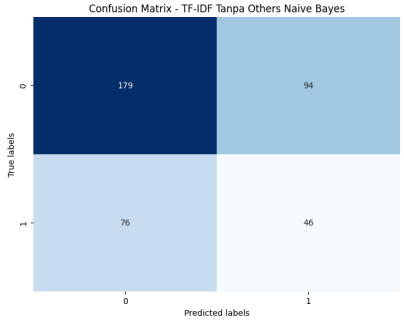
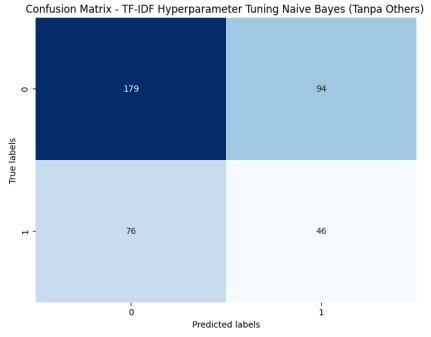
Perbandingan akurasi Feature Extraction dengan algoritma klasifikasi					
		BoW	**TF-IDF**	**2-Grams**	**3-Grams**
Algoritma	**Naive Bayes**	0.57	0.57	0.68	0.68
	Logistic Regression	**0.75**	**0.73**	**0.75**	**0.75**
	Random Forest	0.72	**0.73**	0.72	0.72

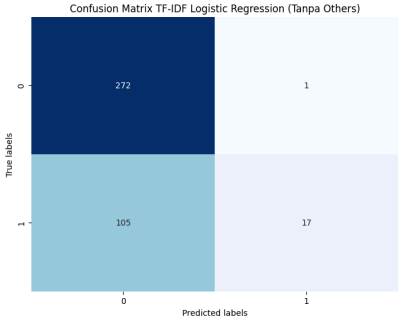
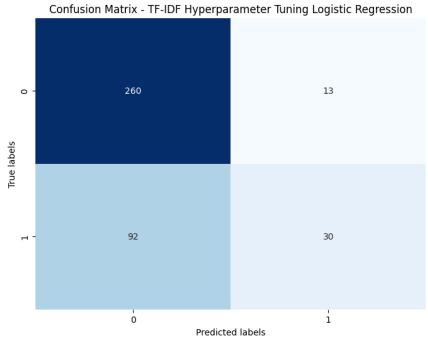
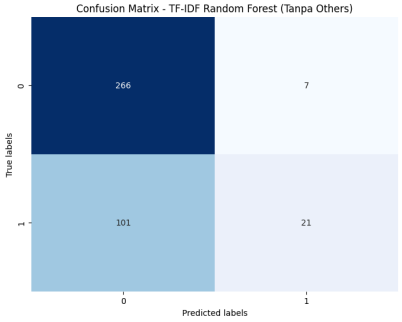
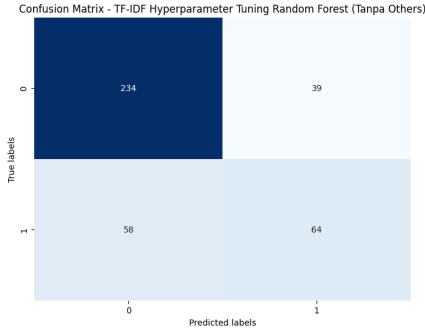
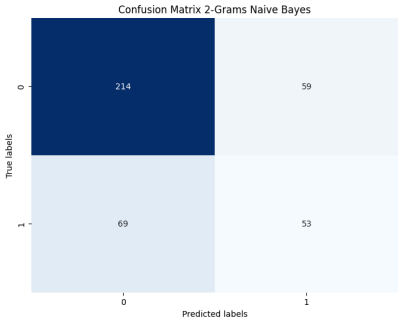
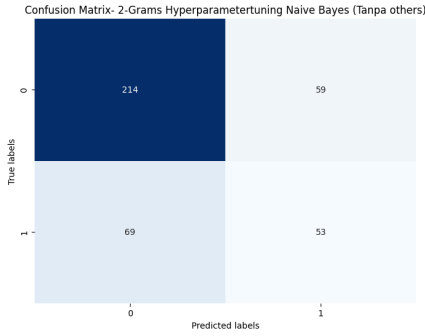
- Table 27. Perbandingan akurasi Feature Extraction dengan algoritma klasifikasi (Hyperparameter Tanpa Others)

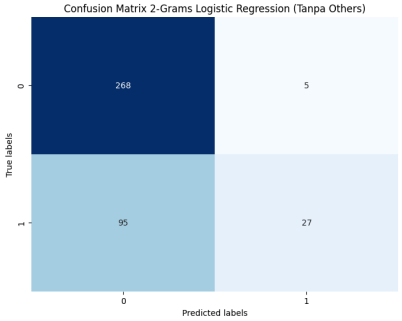
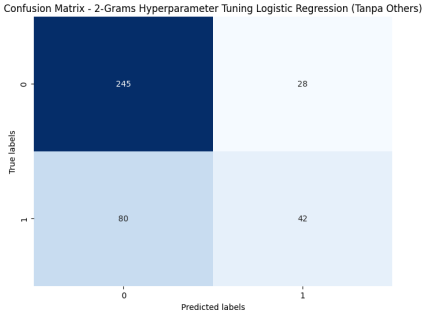
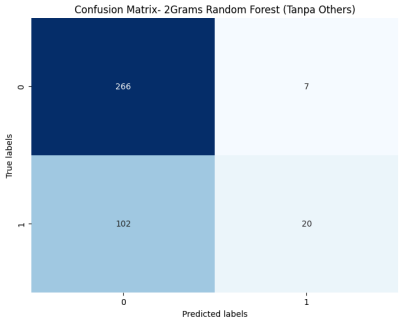
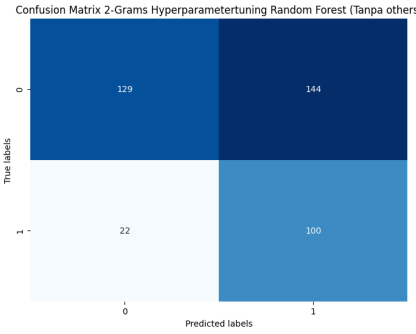
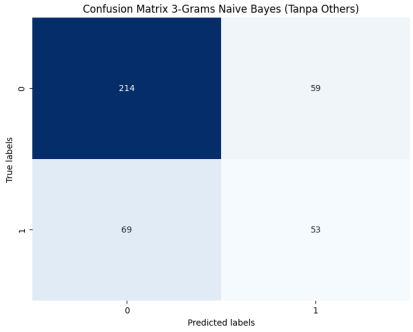
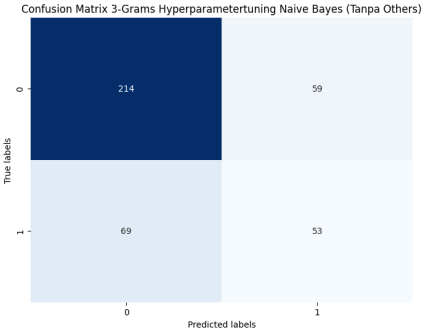
Perbandingan akurasi Feature Extraction dengan algoritma klasifikasi					
		BoW	**2-Grams**	**3-Grams**	**TF-IDF**
Algoritma	**Naive Bayes**	0.58	0.57	0.68	0.68
	Logistic Regression	0.70	0.73	**0.73**	0.73
	Random Forest	**0.75**	**0.75**	0.58	**0.75**

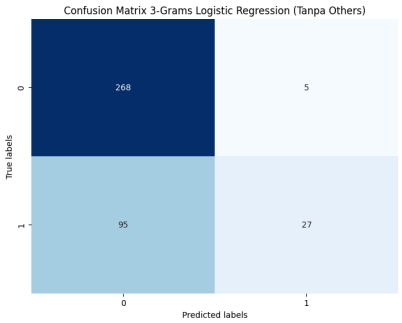
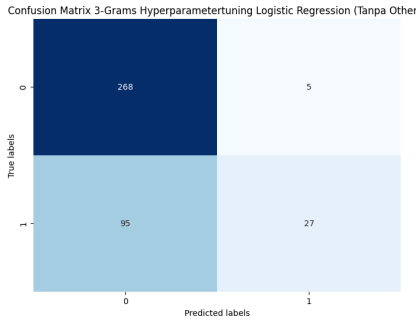
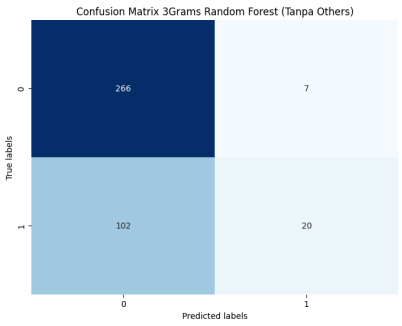
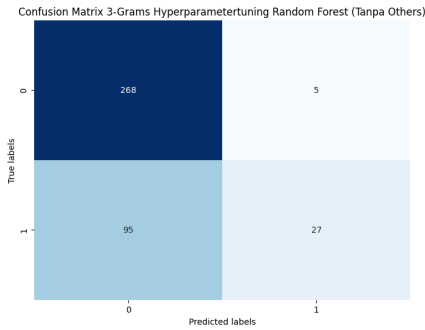
- Table 28. Visualisasi Confusion Matrix Feature Extraction Tanpa Label Others

Feature Extraction Tanpa Label Others	**Confusion Matrix**	**Hyperparameter Confusion Matrix**
BOW		
Naïve Bayers		

Feature Extraction Tanpa Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
Logistic Regression		
Random Forest		
TF-IDF		
Naïve Bayers		

Feature Extraction Tanpa Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
Logistic Regression		
Random Forest		
2-Grams		
Naïve Bayers		

Feature Extraction Tanpa Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
Logistic Regression		
Random Forest		
3-Grams		
Naïve Bayers		

Feature Extraction Tanpa Label Others	Confusion Matrix	Hyperparameter Confusion Matrix
Logistic Regression		
Random Forest		

- Error Analysis

  Untuk hasil analysis error yang lengkap, terdapat pada link berikut ini: [Analysis Error]

  - Naive Bayes menggunakan features extraction BOW

    Table 29. Error Analysis Naive Bayes BOW

full_text	Sentiment Prediksi	Sentiment Asli
sebagai sandwich generation saya takut banget jika nanti duit saya banyak kegocek di pertengahan jalan sumpah	Tidak Stress	Stress
roti yang pake sayur itu namanya apa sandwich generation	Tidak Stress	Others
kompetitif atau ambisnya dia saya menduganya	Others	Stress

karena dia adalah seorang sandwich generation takutnya dia juga bekerja keras bukan untuk dirinya sendiri memang setelah menikah istri bebas untuk membiayai keluarganya tetapi dikhawatirkan ada hal di keluarganya yang membebani dia		

Pada kalimat pertama, model memprediksi sentimen sebagai **Tidak Stress**, sedangkan sentimen asli adalah **Stress**. Hal ini terjadi karena fitur yang diekstraksi menggunakan Bag of Words (BoW) tidak mampu menangkap konteks emosional dalam kalimat tersebut, seperti ungkapan *"takut banget"* atau *"duit saya banyak kegocek"*. BoW hanya menghitung frekuensi kata tanpa mempertimbangkan hubungan antar kata, sehingga gagal memahami bahwa kalimat ini mencerminkan kekhawatiran yang mendalam. Pada kalimat kedua, model kembali memprediksi **Tidak Stress**, sedangkan sentimen asli adalah **Others**. Kalimat ini lebih berupa pertanyaan ringan, *"roti yang pake sayur itu namanya apa"*, yang sebenarnya tidak memiliki muatan emosi yang signifikan. Namun, model kesulitan membedakan konteks netral (Others) dari Tidak Stress karena BoW tidak dapat memahami nuansa konteks dalam kalimat sederhana seperti ini. Pada kalimat ketiga, model memprediksi **Others**, sementara sentimen asli adalah **Stress**. Kalimat ini cukup kompleks, membahas kekhawatiran mendalam tentang tanggung jawab sebagai sandwich generation. BoW cenderung gagal menangkap makna mendalam dari kalimat panjang ini karena banyaknya kata yang mungkin dianggap tidak signifikan, meskipun kata-kata tersebut sebenarnya memberikan konteks penting. Kesalahan ini menunjukkan keterbatasan BoW dalam menangani kalimat dengan kompleksitas tinggi atau ide yang beragam.

- Naive Bayes menggunakan features extraction TF-IDF

  Table 30. Error Analysis Naive Bayes TF-IDF

full_text	Sentiment Prediksi	Sentiment Asli
sebagai sandwich generation saya takut banget jika nanti duit saya banyak kegocek di pertengahan jalan sumpah	Others	Stress
roti yang pake sayur itu namanya apa sandwich generation	Tidak Stress	Others
kompetitif atau ambisnya dia saya menduganya karena dia adalah seorang sandwich generation takutnya dia juga bekerja keras bukan untuk	Others	Stress

dirinya sendiri memang setelah menikah istri bebas untuk membiayai keluarganya tetapi dikhawatirkan ada hal di keluarganya yang membebani dia		

Model Naive Bayes dengan fitur TF-IDF menunjukkan beberapa kesalahan dalam memprediksi sentimen. Pada kalimat pertama, yang sebenarnya mencerminkan stres melalui frasa seperti *"takut banget"* dan *"kegocek,"* model justru memprediksi kategori *Others*. Hal ini menunjukkan bahwa model tidak mampu menangkap konteks emosional dari kata-kata tersebut. Kesalahan serupa terjadi pada kalimat kedua, di mana model memprediksi kategori *Tidak Stress* untuk teks yang bersifat deskriptif, seperti *"roti yang pake sayur itu namanya apa sandwich generation,"* yang sebenarnya masuk ke kategori *Others*. Selain itu, pada kalimat ketiga, model memprediksi *Others* meskipun terdapat frasa seperti *"membebani"* yang mengindikasikan adanya tekanan emosional atau stress.

- Naive Bayes menggunakan features extraction 2-Grams

  Table 31. Error Analysis Naive Bayes 2-Grams

full_text	Sentiment Prediksi	Sentiment Asli
sebagai sandwich generation saya takut banget jika nanti duit saya banyak kegocek di pertengahan jalan sumpah	Tidak Stress	Stress
roti yang pake sayur itu namanya apa sandwich generation	Stress	Others
kompetitif atau ambisnya dia saya menduganya karena dia adalah seorang sandwich generation takutnya dia juga bekerja keras bukan untuk dirinya sendiri memang setelah menikah istri bebas untuk membiayai keluarganya tetapi dikhawatirkan ada hal di keluarganya yang membebani dia	Others	Stress

Berdasarkan tabel hasil prediksi menggunakan model Naive Bayes dengan fitur 2-grams, terdapat beberapa kesalahan dalam memprediksi sentimen sebenarnya. Pada kalimat pertama, model memprediksi sentimen sebagai *Tidak Stress*, padahal terdapat ungkapan seperti *"takut banget"* dan *"duit saya banyak kegocek,"* yang jelas menunjukkan adanya tekanan emosional atau stres. Kesalahan ini mengindikasikan bahwa model tidak sepenuhnya memahami konteks frasa yang mengindikasikan stres.

Pada kalimat kedua, prediksi model adalah *Stress*, meskipun teks tersebut lebih mengarah pada kategori *Others* karena hanya berupa pernyataan tentang istilah *"sandwich generation"* tanpa adanya indikasi emosional yang signifikan. Sedangkan pada kalimat ketiga, meskipun ada frasa seperti *"membebani dia,"* yang jelas mengindikasikan stres, model justru memprediksi kategori *Others*.

- Naive Bayes menggunakan features extraction 3-Grams

  Table 32. Error Analysis Naive Bayes 3=Grams

full_text	Sentiment Prediksi	Sentiment Asli
sebagai sandwich generation saya takut banget jika nanti duit saya banyak kegocek di pertengahan jalan sumpah	Others	Stress
roti yang pake sayur itu namanya apa sandwich generation	Stress	Others
enaknya jadi sandwich generation dirumah jadi punya power	Others	Tidak Stress

Model Naive Bayes dengan fitur 3-grams menunjukkan beberapa kesalahan dalam prediksi sentimen pada contoh teks yang diberikan. Pada kalimat "sebagai sandwich generation saya takut banget jika nanti duit saya banyak kegocek di pertengahan jalan sumpah", model memprediksi sentimen sebagai *Others*, padahal terdapat indikasi kuat dari kata "takut" yang seharusnya mengarah pada sentimen *Stress*. Ini menunjukkan bahwa meskipun model dapat menangkap konteks 3-grams, belum sepenuhnya efektif dalam memahami ketegangan emosional yang terkandung dalam teks. Selain itu, pada kalimat "roti yang pake sayur itu namanya apa sandwich generation", yang tidak mengandung konteks emosional signifikan, model salah memprediksi sentimen sebagai *Stress*, padahal kalimat ini lebih cocok dengan kategori *Others*. Terakhir, pada kalimat "enaknya jadi sandwich generation dirumah jadi punya power", yang menyatakan pandangan positif tentang menjadi bagian dari sandwich generation, model memprediksi sentimen sebagai *Others*, meskipun seharusnya prediksi yang lebih tepat adalah *Tidak Stress*, karena kalimat ini tidak mengandung indikasi perasaan tertekan atau stres. Kesalahan-kesalahan ini menunjukkan bahwa meskipun 3-grams dapat menangkap pola urutan kata, model Naive Bayes masih kesulitan dalam menangkap nuansa emosional atau perasaan kompleks dalam teks.

- Random Forest menggunakan features extraction BOW

Table 33. Error Analysis Random Forest BOW

full_text	Sentiment Prediksi	Sentiment Asli
bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu gua	Others	Stress
gini masyarakat kelas menengah tanggung sandwich generation subway turkey minumnya zionist	Stress	Others
berdasaekan pengalaman pribadi saya kita ini yang sandwich generation lebih aman berjodoh dengan yang sama sandwich generation juga sih jadi bisa saling ngertiin dan saling bantu hehehe	Others	Tidak Stress

Pada penggunaan model Random Forest dengan fitur ekstraksi BOW, prediksi sentimen menunjukkan perbedaan dengan sentimen asli yang ada. Pada kalimat pertama, *"bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu gua"*, model memprediksi sentimen sebagai *Others*, padahal kalimat ini mengandung kata-kata yang menunjukkan ketidaknyamanan dan stres, seperti "diinjek-injek" dan "dighibahin", yang seharusnya lebih cocok dengan kategori *Stress*. Pada kalimat kedua, *"gini masyarakat kelas menengah tanggung sandwich generation subway turkey minumnya zionist"*, meskipun tidak ada emosi yang eksplisit, model memprediksi *Stress*, yang mungkin tidak tepat karena kalimat ini terdengar lebih seperti opini umum. Pada kalimat ketiga, *"berdasaekan pengalaman pribadi saya kita ini yang sandwich generation lebih aman berjodoh dengan yang sama sandwich generation juga sih jadi bisa saling ngertiin dan saling bantu hehehe"*, model memprediksi *Others*, sementara ini lebih cenderung ke *Tidak Stress* karena menyebutkan hal positif tentang saling mengerti dan membantu antar sesama sandwich generation.

- Random Forest menggunakan features extraction TF-IDF

Table 34. Error Analysis Random Forest TF-IDF

full_text	Sentiment Prediksi	Sentiment Asli
bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu	Others	Stress

gua		
alhamdulillah banyak yang paham bahwa maksudnya hyunwoo bisa nabung juta won per bulan alias dia kelas menengah tidak punya tanggungan alias bukan sandwich generation kayak banyak orang di sini yang jangankan rp juta bulan nabung rp juta bulan aja belom tentu	Stress	Others
jadi tulang punggung keluarga sandwich generation tidak bisa ngarep sama orang lain	Others	Stress

Pada penggunaan model Random Forest dengan fitur ekstraksi TF-IDF, prediksi sentimen juga menunjukkan perbedaan dengan sentimen asli. Pada kalimat pertama, *"bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu gua"*, model memprediksi sentimen sebagai *Others*, meskipun kalimat ini jelas mengindikasikan tekanan emosional dan cocok dengan kategori *Stress*. Pada kalimat kedua, *"alhamdulillah banyak yang paham bahwa maksudnya hyunwoo bisa nabung juta won per bulan alias dia kelas menengah tidak punya tanggungan alias bukan sandwich generation kayak banyak orang di sini yang jangankan rp juta bulan nabung rp juta bulan aja belom tentu"*, model memprediksi sentimen sebagai *Stress*, tetapi ini tidak sesuai karena isi kalimat lebih bernuansa positif, sehingga seharusnya masuk dalam kategori *Others*. Pada kalimat ketiga, *"jadi tulang punggung keluarga sandwich generation tidak bisa ngarep sama orang lain"*, model memprediksi *Others*, padahal kalimat ini menunjukkan tekanan sebagai tulang punggung keluarga, sehingga lebih tepat masuk dalam kategori *Stress*.

- Random Forest menggunakan features extraction 2-Grams

  Table 35. Error Analysis Random Forest 2-Grams

full_text	**Sentiment Prediksi**	**Sentiment Asli**
bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu gua	Others	Stress
saya suka sandwich tapi nggak suka jadi sandwich generation	Stress	Others
gini rasanya jadi sandwich generation numpahin rasa sedih dikemudian hari disepelein dijadiin ledekan seolah olah hal sepele eh giliran nutup	Tidak Stress	Stress

nutupin masalah nunjukin seneng senengnya aja didepan semuanya malah disangka tidak prihatin sama keadaan alias bodoamatan sulittt		

Pada penggunaan model Random Forest dengan fitur ekstraksi 2-Grams, hasil prediksi sentimen menunjukkan adanya ketidaksesuaian antara sentimen prediksi dengan sentimen asli. Pada kalimat pertama, *"bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu gua"*, model memprediksi sentimen sebagai *Others*, padahal kalimat ini mencerminkan tekanan emosional yang kuat dan lebih tepat dikategorikan sebagai *Stress*. Pada kalimat kedua, *"saya suka sandwich tapi nggak suka jadi sandwich generation"*, model memprediksi sentimen sebagai *Stress*, meskipun kalimat ini lebih bernuansa ringan dan seharusnya masuk dalam kategori *Others*. Pada kalimat ketiga, *"gini rasanya jadi sandwich generation numpahin rasa sedih dikemudian hari disepelein dijadiin ledekan seolah olah hal sepele eh giliran nutup nutupin masalah nunjukin seneng senengnya aja didepan semuanya malah disangka tidak prihatin sama keadaan alias bodoamatan sulittt"*, model memprediksi sentimen sebagai *Tidak Stress*, padahal isi kalimat secara jelas mencerminkan tekanan emosional yang signifikan, sehingga lebih tepat dikategorikan sebagai *Stress*.

- Random Forest menggunakan features extraction 3-Grams

Table 36. Error Analysis Random Forest 3-Grams

full_text	Sentiment Prediksi	Sentiment Asli
bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu gua	Others	Stress
saya suka sandwich tapi nggak suka jadi sandwich generation	Stress	Others
tekun ulet mau belajar dan tidak gampang nyerah itu wajib punya sih ditengah gempuran para sandwich generation yang kena tekanan dikit langsung minta resign	Tidak Stress	Stress

Pada penggunaan model Random Forest dengan fitur ekstraksi 3-Grams, hasil prediksi menunjukkan adanya ketidaksesuaian yang cukup signifikan antara sentimen prediksi dan sentimen asli. Pada kalimat pertama, *"bagian dari sandwich generation tapi feedbacknya diinjek injek depan istri dan dighibahin ke sodara yang tidak tau apa apa itu*

gua", model memprediksi sentimen sebagai *Others*, padahal konteksnya mencerminkan tekanan emosional yang kuat, sehingga lebih tepat dikategorikan sebagai *Stress*. Pada kalimat kedua, *"saya suka sandwich tapi nggak suka jadi sandwich generation"*, model memprediksi sentimen sebagai *Stress*, meskipun konteks kalimatnya lebih ringan dan bernuansa humoris, sehingga lebih cocok masuk kategori *Others*. Kalimat ketiga, *"tekun ulet mau belajar dan tidak gampang nyerah itu wajib punya sih ditengah gempuran para sandwich generation yang kena tekanan dikit langsung minta resign"*, diprediksi sebagai *Tidak Stress*, meskipun terdapat elemen tekanan yang jelas dan seharusnya dikategorikan sebagai *Stress*.

- Logistic Regression menggunakan features extraction BOW

  Table 37. Error Analysis Logistic Regression BOW

full_text	Sentiment Prediksi	Sentiment Asli
penempatan istilah sandwich generation itu tidak salah saya salah satu darinya tapi keputusan memutus mata rantai itu tetap pada kita bagiku menyisihkan sebagian dari penghasilan bukanlah sebuah kewajiban melainkan bukti ketulusan karna sampe skarang cuma itu yang bisa kuberikan	Others	Tidak Stress
allah saya merasa desperate tolong dong dikasih pemimpin yang amanah saya kepikiran masa depan anak dan masa tua orangtua saya jika pemimpinnya nggak amanah sampai ngapa ngapain susah saya sebagai sandwich generation tidak tau mau ngapain lagi	Tidak Stress	Stress
pro sandwich generation itu berat apalagi kalau tidak dibarengi dengan keputusan childfree	Stress	Others

Kesalahan klasifikasi pada teks-teks ini menunjukkan kesulitan model dalam memahami konteks kompleks dan ekspresi emosional yang tidak eksplisit. Pada teks *"penempatan istilah sandwich generation itu tidak salah...,"* model memprediksi "Others" karena narasi cenderung deskriptif dan menggunakan bahasa netral. Namun, sentimen sebenarnya adalah "Tidak Stress," karena teks ini mencerminkan penerimaan dan pandangan positif terkait peran sebagai *sandwich generation.* Pada teks *"allah saya merasa desperate...,"* model memprediksi "Tidak Stress" karena kesulitan menangkap kata-kata yang mengindikasikan tekanan berat, seperti *"desperate"* dan *"kepikiran masa depan."* Sebaliknya, sentimen asli adalah "Stress," karena teks ini jelas menggambarkan

kekhawatiran mendalam terkait tanggung jawab finansial dan masa depan keluarga. Sementara itu, pada teks *"pro sandwich generation itu berat...,"* model memprediksi "Stress" karena menyebutkan istilah *"berat"* yang sering diasosiasikan dengan tekanan emosional. Namun, konteks sebenarnya lebih netral dan bersifat argumentatif, sehingga label yang benar adalah "Others." Kesalahan-kesalahan ini menunjukkan bahwa model mengalami kesulitan dalam menginterpretasikan nuansa konteks, ironi, atau sentimen yang tidak secara eksplisit diekspresikan dalam teks.

- Logistic Regression menggunakan features extraction TF-IDF

  Table 38. Error Analysis Logistic Regression TF-IDF

full_text	Sentiment Prediksi	Sentiment Asli
lancar terus rejeki sandwich generation	Others	Tidak Stress
saya iri ngeliat orang sebegitu dicintainya sama ayahnya dibeliin ini itu kebutuhan tercukupi kalau bapak kerja pasti saat ini kita bahagia sekarang bagaimana saya jadi generasi sandwich	Stress	Others
karna ak sdh jadi sandwich generation jdnya ak tu berdoa jika kd mau bisi laki yang sandwich generation sma kya saya alhamdulillahnya dipertemuin dengan lakian yang kaya sesuai doa saya	Stress	Tidak Stress

Kesalahan klasifikasi pada teks-teks ini menunjukkan tantangan model dalam menangkap konteks emosional dan maksud tersembunyi dalam narasi. Pada teks *"lancar terus rejeki sandwich generation,"* model memprediksi "Others" karena frasa ini terlihat netral dan tidak menunjukkan emosi eksplisit. Namun, konteks sebenarnya adalah ungkapan optimisme yang mencerminkan sikap positif, sehingga label yang benar adalah "Tidak Stress." Selanjutnya, pada teks *"saya iri ngeliat orang sebegitu dicintainya sama ayahnya dibeliin ini itu...,"* model memprediksi "Stress" karena kata-kata seperti *"iri"* dan *"bahagia"* bisa menunjukkan emosi negatif. Namun, teks ini sebenarnya menggambarkan refleksi atau perbandingan tanpa tekanan emosional berat, sehingga seharusnya dilabeli "Others." Pada teks *"karna ak sdh jadi sandwich generation jdnya ak tu berdoa...,"* model memprediksi "Stress" karena menyebut istilah *"sandwich generation"* dan tekanan finansial yang biasanya diasosiasikan dengannya. Namun, keseluruhan konteks

menunjukkan rasa syukur dan optimisme, sehingga label yang benar adalah "Tidak Stress." Kesalahan ini mencerminkan keterbatasan model dalam memahami konteks implisit dan mengintegrasikan emosi tersirat dalam narasi yang lebih panjang.

- Logistic Regression Menggunakan features extraction 2-Grams.

  Table 39. Error Analysis Logistic Regression 2-Grams

full_text	Sentiment Prediksi	Sentiment Asli
situasi ketika emak saya berpesan untuk nitip kakak dan adek saya sandwich generation moment	Stress	Others
saya sandwich generation dan pertama tama saya gapapa banget meskipun saya setiap malem masih nangis awal awal saya merasa terbebani karena uang yang seharusnya kutabung akhirnya dibuat bayar utang orang tua dan masih harus biyayain adek yang masih kuliah dan sekolah	Tidak Stress	Stress
udah cari kerja kemana mana lakuin ini itu update dan upgrade cv belajar hal baru tapi masih pengangguran dari semenjak lulus sumpah saya pengen kerja saya berdoa gapapa saya jadi sandwich generation asal keluarga saya serba cukup	Stress	Tidak Stress

Model mengalami kesalahan klasifikasi pada teks-teks ini karena kesulitan memahami konteks dan nuansa emosional secara mendalam. Pada teks *"situasi ketika emak saya berpesan untuk nitip kakak dan adek saya sandwich generation moment,"* model memprediksi "Stress" karena istilah *"sandwich generation"* sering dikaitkan dengan beban, tetapi teks tersebut sebenarnya hanya menggambarkan situasi umum tanpa tekanan emosional yang jelas, sehingga label yang tepat adalah "Others." Selanjutnya, dalam teks *"saya sandwich generation dan pertama tama saya gapapa banget meskipun saya setiap malem masih nangis...,"* model memprediksi "Tidak Stress" karena mungkin menekankan bagian awal teks yang tampak positif (*"gapapa banget"*), sementara konteks keseluruhan jelas menunjukkan tekanan emosional dan finansial yang signifikan, sehingga label yang benar adalah "Stress." Pada teks *"udah cari kerja kemana mana lakuin ini itu...,"* model memprediksi "Stress" karena adanya kata-kata seperti *"pengangguran"* dan *"saya pengen kerja,"* yang sering diasosiasikan dengan tekanan, meskipun teks menunjukkan tekad untuk menerima kondisi sebagai bagian dari tanggung jawab, yang seharusnya diberi label "Tidak Stress." Kesalahan ini mencerminkan

tantangan model dalam menangkap konteks yang kompleks, memahami narasi yang berlapis, dan membedakan antara deskripsi faktual, emosi tersirat, dan perasaan yang eksplisit.

- Logistic Regression Menggunakan features extraction 3-Grams.

  Table 40. Error Analysis Logistic Regression 3-Grams

full_text	Sentiment Prediksi	Sentiment Asli
generasi z di luar negeri banyak yang bergantung sama orang tua kok kalopun kerja duitnya bener buat foya doang kebetulan saya liat sendiri malah di indo temen saya yang generasi z lebih banyak yang rajin banyak yang jadi sandwich generation kudu double jobs	Stress	Tidak Stress
mau cerita sedikit sedih liat mami sama kakak tadi nangis mereka habid bahas masalah ekonomi dan posisi kakak yang mau gamau harus jadi sandwich generation gaperlu diceritain semua kali	Others	Stress
romance in the house drama ny ini ringan tetapi masih ada konflik sedikit yang tidak bikin bosen banyak nangis ny nonton ini karena related bnget sama saya yang generasi sandwich mirae dan ibunya yang bikin ini drama tidak bosenin tidak kuat mw nangis banget kl soal ibu dan anak	Stress	Others

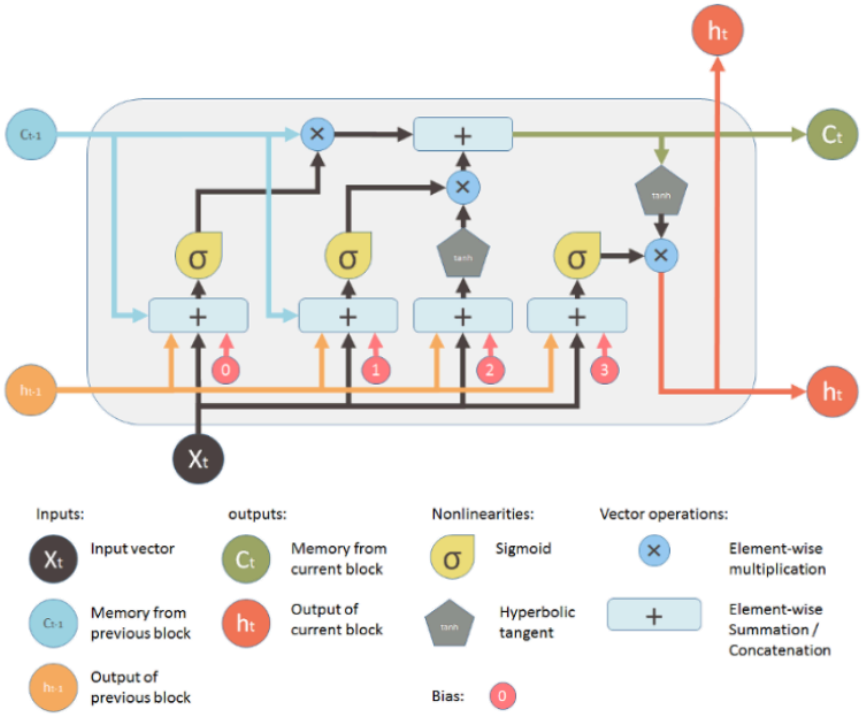
Model mengalami kesalahan klasifikasi pada beberapa teks karena ketidakmampuannya memahami konteks emosional secara mendalam dan membedakan nuansa narasi. Pada teks seperti *"generasi z di luar negeri banyak yang bergantung sama orang tua kok...,"* model memprediksi "Stress" karena adanya istilah seperti *"double jobs"* yang diasosiasikan dengan tekanan, meskipun konteks sebenarnya lebih netral dan menggambarkan observasi, sehingga label seharusnya "Tidak Stress." Dalam teks *"mau cerita sedikit sedih liat mami sama kakak tadi nangis...,"* model memprediksi "Others" karena fokus pada deskripsi emosional seperti *"nangis"* tanpa eksplisit menyebut tekanan spesifik, padahal konteks ekonomi dan tanggung jawab menunjukkan label seharusnya "Stress." Pada teks *"romance in the house drama ny ini ringan tetapi masih ada konflik...,"* model salah memprediksi "Stress" karena interpretasi kata-kata seperti *"nangis"* dan *"tidak kuat mw nangis banget,"* padahal konteks utama adalah refleksi pada drama yang tidak relevan secara emosional dengan tekanan nyata, sehingga label

seharusnya "Others." Kesalahan ini mencerminkan keterbatasan model dalam menangkap konteks yang lebih subtil dan interpretasi narasi yang bercampur dengan elemen deskriptif dan emosional.

- **Deep Learning**
  - **LSTM**

    Long Short-Term Memory (LSTM) adalah jenis khusus dari Recurrent Neural Network (RNN) yang dikembangkan untuk mengatasi masalah RNN tradisional dalam menangani urutan panjang. LSTM memiliki arsitektur khusus yang memungkinkan jaringan ini memiliki "memori," sehingga dapat mengatasi urutan yang memiliki ketergantungan jangka panjang dengan baik. Penemuan LSTM menjadi terobosan besar dalam bidang Deep Learning dan menjadi bagian penting dalam aplikasi Natural Language Processing (NLP). LSTM mampu mengingat informasi untuk jangka waktu yang lama, menjadikannya sangat efektif untuk tugas-tugas yang membutuhkan pemahaman konteks panjang.

- Rumus LSTM:



Rumus LSTM terdiri dari empat bagian, yang masing-masing dijelaskan dalam Persamaan 1, Persamaan 2, Persamaan 3, dan Persamaan 4.

- Forget Gate

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \quad (1)$$

- Input Gate

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (2)$$

- Memory Update

$$c_t = f_t \circ c_{t-1} + i_t \circ \phi(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (3)$$

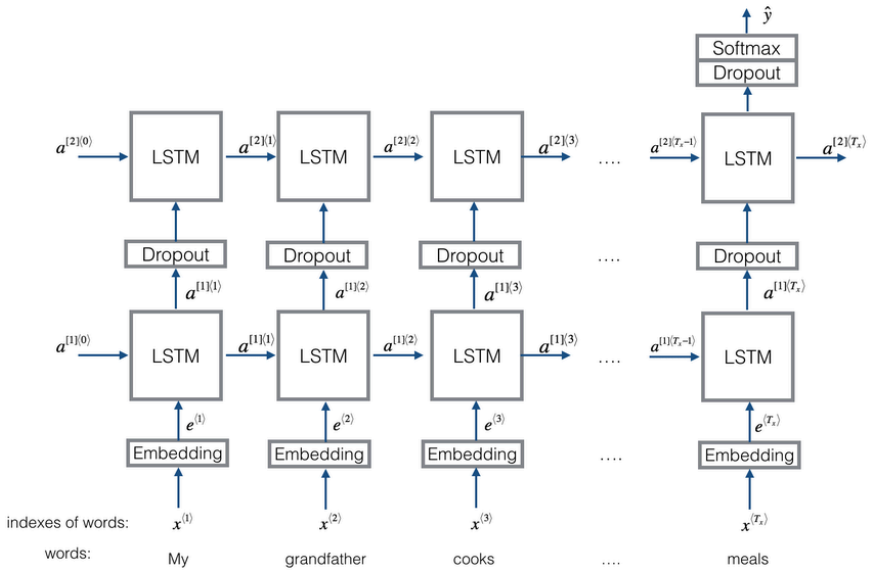
- Output Gate

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \circ \phi(c_t)$$

- LSTM pakai Glove

  LSTM (Long Short-Term Memory) dengan GloVe (Global Vectors for Word Representation) adalah kombinasi model dalam pengolahan bahasa alami. LSTM, jenis jaringan saraf yang menangani urutan data, efektif untuk memahami konteks jangka panjang dalam teks. GloVe, model embedding kata, mengubah kata-kata menjadi vektor numerik berdasarkan hubungan kata dalam korpus besar, sehingga kata dengan makna serupa memiliki representasi yang mirip. Dengan menggunakan vektor GloVe sebagai input, LSTM dapat memproses informasi semantik dan temporal secara lebih akurat, menjadikannya ideal untuk tugas seperti klasifikasi teks dan analisis sentimen.



A LSTM classifier based on GloVe word embeddings

- Berikut adalah penjelasan tentang pemilihan setiap layer pada model LSTM yang digunakan (Menggunakan Others) :
  1. Embedding Layer

     fungsinya adalah mengubah kata-kata (token) menjadi representasi vektor numerik yang kaya informasi semantik.. Parameter yang digunakan:
     - embedding_matrix.shape[0] memberikan jumlah kata unik dalam embedding matrix (baris matriks).
     - embedding_dim adalah dimensi vektor embedding yang biasanya sudah ditentukan dalam embedding pra-latih seperti GloVe
     - embedding_matrix adalah Menetapkan matriks embedding pra-latih (misalnya, GloVe) sebagai bobot awal untuk layer embedding.
  2. BiLSTM Layer

     Menggunakan Long Short-Term Memory (LSTM) untuk memproses data urut (seperti teks) dengan mempertimbangkan konteks dari kedua arah (maju dan mundur). Parameter yang digunakan :
     - $units = 8$: Jumlah unit memori dalam LSTM, yang menentukan kapasitas model untuk menyimpan informasi dari data urutan.
     - $kernel\_regularizer = 'l2'$ dan $recurrent\_regularizer = 'l2'$: Menambahkan regulasi L2 untuk mencegah overfitting.
  3. Dropout Layer (Layer Pertama)

     Fungsinya adalah Mencegah overfitting dengan cara secara acak

menonaktifkan (drop) 30% unit (neuron) selama pelatihan. Parameter yang digunakan :

- $rate = 0.3$: Persentase neuron yang dinonaktifkan.
  Membuat model lebih general dan tahan terhadap data baru.

4. Output Layer (Dense)
   Fungsinya adalah menghasilkan output berupa probabilitas setiap kelas dalam klasifikasi. Parameter yang digunakan :
   - $units = 3$: Jumlah kelas dalam data
   - $activation = 'softmax'$: Fungsi aktivasi softmax untuk mengkonversi nilai output menjadi probabilitas total 1.
     Hasilnya Probabilitas setiap data masuk ke salah satu dari 3 kelas yang ada.

- Berikut adalah penjelasan tentang pemilihan setiap layer pada model LSTM yang digunakan (Tanpa Others):
  1. Embedding Layer
     fungsinya adalah mengubah kata-kata (token) menjadi representasi vektor numerik yang kaya informasi semantik.. Parameter yang digunakan:
     - embedding_matrix.shape[0] memberikan jumlah kata unik dalam embedding matrix (baris matriks).
     - embedding_dim adalah dimensi vektor embedding yang biasanya sudah ditentukan dalam embedding pra-latih seperti GloVe
     - embedding_matrix adalah Menetapkan matriks embedding pra-latih (misalnya, GloVe) sebagai bobot awal untuk layer embedding.
  2. BiLSTM Layer
     Menggunakan Long Short-Term Memory (LSTM) untuk memproses data urut (seperti teks) dengan mempertimbangkan konteks dari kedua arah (maju dan mundur). Parameter yang digunakan :
     - $units = 8$: Jumlah unit memori dalam LSTM, yang menentukan kapasitas model untuk menyimpan informasi dari data urutan.
     - $kernel\_regularizer = 'l2'$ dan $recurrent\_regularizer = 'l2'$: Menambahkan regulasi L2 untuk mencegah overfitting.
  3. Dropout Layer (Layer Pertama)
     Fungsinya adalah Mencegah overfitting dengan cara secara acak menonaktifkan (drop) 30% unit (neuron) selama pelatihan. Parameter yang digunakan :

- $rate = 0.3$: Persentase neuron yang dinonaktifkan.
  Membuat model lebih general dan tahan terhadap data baru.

4. Output Layer (Dense)

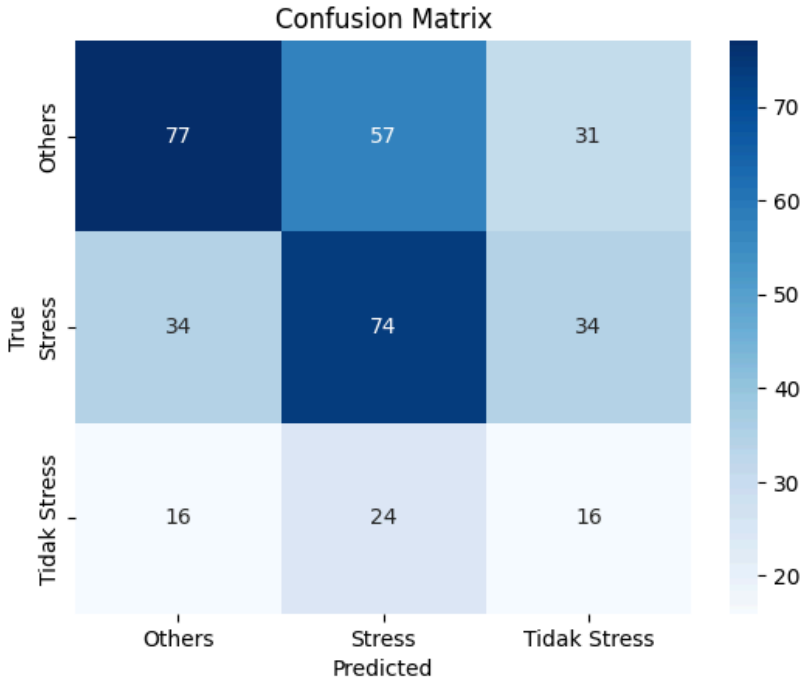
   Fungsinya adalah menghasilkan output berupa probabilitas setiap kelas dalam klasifikasi. Parameter yang digunakan :

   - $units = 2$: Jumlah kelas dalam data
   - $activation = 'softmax'$: Fungsi aktivasi softmax untuk mengkonversi nilai output menjadi probabilitas total 1.

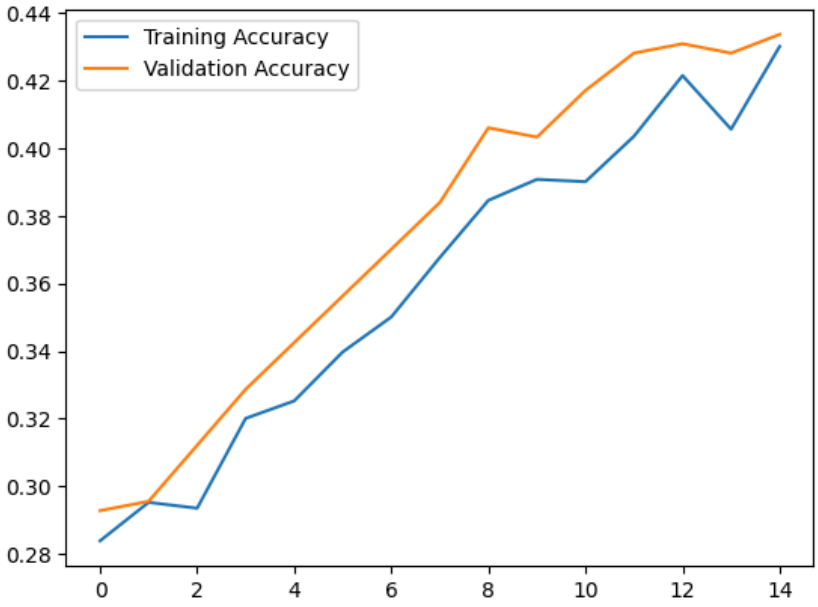
■ Table 41. *Classification Report LSTM Glove*

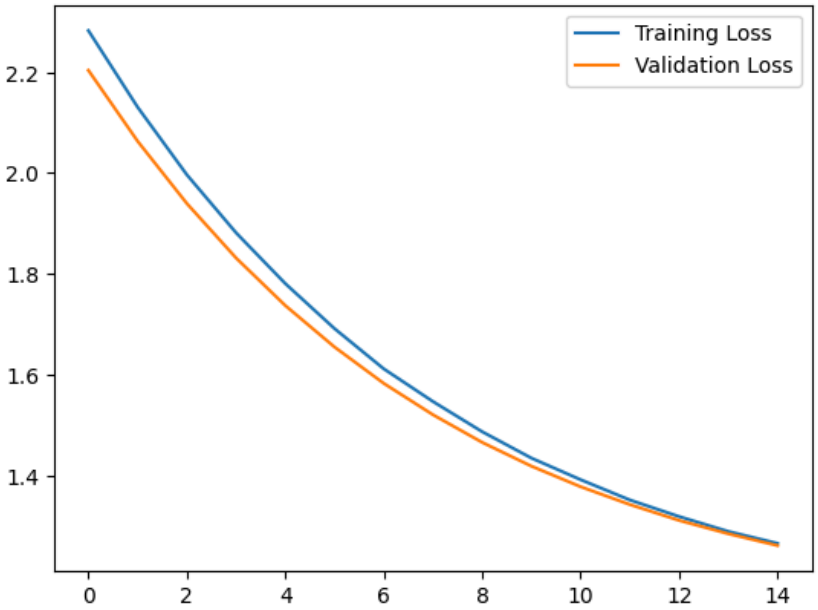
	Precision	Recall	F1	Support
Others	0.61	0.47	0.53	165
Stress	0.48	0.62	0.50	142
Tidak Stress	0.20	0.29	0.23	56
Accuracy			**0.46**	363
Macro Avg	0.43	0.42	0.42	363
Weighted Avg	0.49	0.46	0.47	363

■ Confusion Matrix LSTM pakai Others:



- Training dan Validation Acc dan Loss LSTM pakai Glove pakai Others

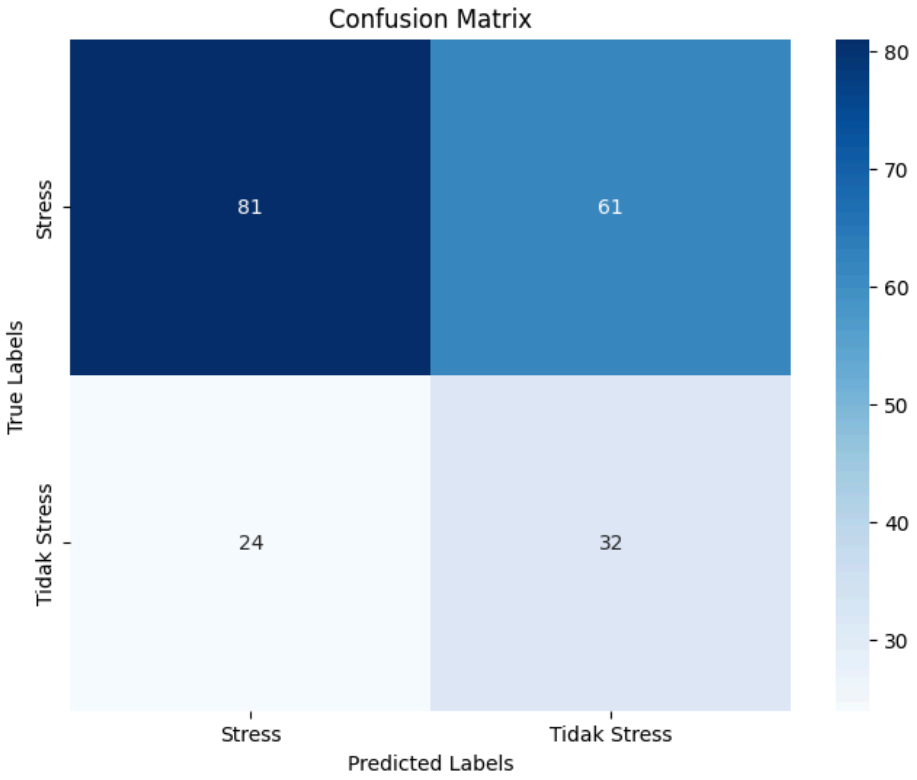




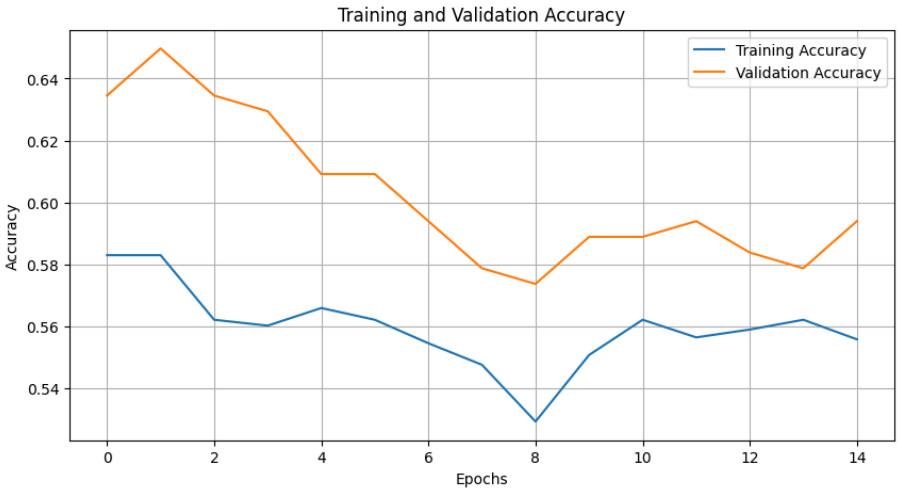
■ Table 42. *Classification Report LSTM Glove Tanpa Others*

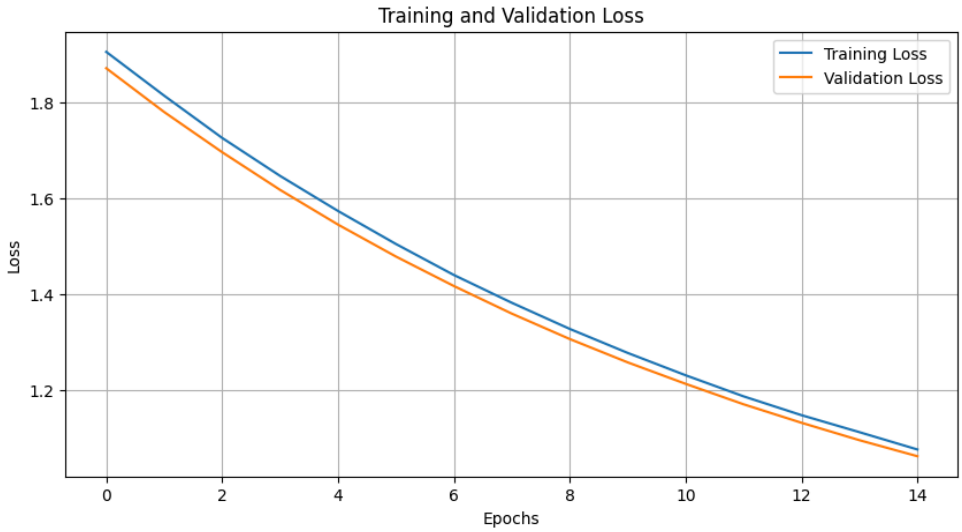
	Precision	Recall	F1	Support
Stress	0.77	0.57	0.66	142
Tidak Stress	0.34	0.57	0.43	56
Accuracy			**0.57**	198
Macro Avg	0.56	0.57	0.54	198
Weighted Avg	0.65	0.57	0.59	198

■ Confusion Matrix LSTM tanpa Others:



- Training dan Validation Acc dan Loss LSTM pakai Glove tanpa Others





- LSTM dengan Word2Vec

  LSTM (Long Short-Term Memory) dengan Word2Vec adalah kombinasi yang kuat dalam pemrosesan bahasa alami, terutama dalam analisis sentimen dan tugas klasifikasi teks lainnya. Word2Vec adalah metode yang mengubah kata-kata menjadi representasi vektor numerik dengan dua arsitektur utama yaitu Continuous Bag of Words (CBOW) dan Skip-Gram. CBOW memprediksi kata target berdasarkan kata-kata konteks di sekitarnya, sementara Skip-Gram memprediksi konteks berdasarkan kata target. Teknik ini memungkinkan untuk menangkap makna semantik kata,

di mana kata-kata dengan konteks serupa akan memiliki representasi vektor yang dekat satu sama lain dalam ruang vektor, yang sangat berguna dalam memahami konteks dalam analisis sentimen.

- Berikut adalah penjelasan tentang pemilihan setiap layer pada model LSTM yang digunakan :

1. Embedding Layer

   Fungsinya adalah mengubah kata-kata (token) menjadi representasi vektor numerik yang kaya informasi semantik.. Parameter yang digunakan:

   - embedding_matrix.shape[0] memberikan jumlah kata unik dalam embedding matrix (baris matriks).
   - output_dim=100 adalah menentukan dimensi dari vektor embedding untuk setiap kata.
   - weights=[embedding_matrix] menginisialisasi bobot layer embedding dengan matriks embedding yang sudah dilatih sebelumnya.
   - input_length=max_length menentukan panjang maksimum urutan input (jumlah kata dalam satu dokumen atau kalimat).
   - trainable=False menentukan apakah bobot embedding dapat dilatih ulang atau tidak selama pelatihan model.

2. BiLSTM Layer

   Menggunakan Long Short-Term Memory (LSTM) untuk memproses data urut (seperti teks) dengan mempertimbangkan konteks dari kedua arah (maju dan mundur). Parameter yang digunakan :

   - $units = 16$: Jumlah unit memori dalam LSTM, yang menentukan kapasitas model untuk menyimpan informasi dari data urutan.
   - $kernel\_regularizer = 'l2'$ teknik regularisasi untuk mencegah overfitting dengan menambahkan penalti ke bobot model. Regularisasi L2 membantu mengurangi bobot yang terlalu besar, sehingga model lebih sederhana dan tidak terlalu mengikuti fluktuasi kecil dalam data pelatihan.

3. Dropout Layer (Layer Pertama)

   Fungsinya adalah Mencegah overfitting dengan cara secara acak menonaktifkan (drop) 30% unit (neuron) selama pelatihan. Parameter yang digunakan :

   - $rate = 0.3$: Persentase neuron yang dinonaktifkan. Membuat model lebih general dan tahan terhadap data baru.

4. Output Layer (Dense)

   Fungsinya adalah menghasilkan output berupa probabilitas setiap kelas dalam klasifikasi. Parameter yang digunakan :

   - num_classes: Jumlah kelas dalam masalah klasifikasi, yang dalam hal ini adalah 3 kelas
   - $activation = 'softmax'$: Fungsi aktivasi softmax untuk mengkonversi nilai output menjadi probabilitas total 1.

- Berikut adalah penjelasan tentang pemilihan setiap layer pada model LSTM yang digunakan (Tanpa Others) :

  1. Embedding Layer

     Fungsinya adalah mengubah kata-kata (token) menjadi representasi vektor numerik yang kaya informasi semantik.. Parameter yang digunakan:

     - embedding_matrix.shape[0] memberikan jumlah kata unik dalam embedding matrix (baris matriks).
     - output_dim=100 adalah menentukan dimensi dari vektor embedding untuk setiap kata.
     - weights=[embedding_matrix] menginisialisasi bobot layer embedding dengan matriks embedding yang sudah dilatih sebelumnya.
     - input_length=max_length menentukan panjang maksimum urutan input (jumlah kata dalam satu dokumen atau kalimat).
     - trainable=False menentukan apakah bobot embedding dapat dilatih ulang atau tidak selama pelatihan model.

  2. BiLSTM Layer

     Menggunakan Long Short-Term Memory (LSTM) untuk memproses data urut (seperti teks) dengan mempertimbangkan konteks dari kedua arah (maju dan mundur). Parameter yang digunakan :

     - $units = 16$: Jumlah unit memori dalam LSTM, yang menentukan kapasitas model untuk menyimpan informasi dari data urutan.
     - $kernel\_regularizer = 'l2'$ teknik regularisasi untuk mencegah overfitting dengan menambahkan penalti ke bobot model. Regularisasi L2 membantu mengurangi bobot yang terlalu besar, sehingga model lebih sederhana dan tidak terlalu mengikuti fluktuasi kecil dalam data pelatihan.

3. Dropout Layer (Layer Pertama)
   Fungsinya adalah Mencegah overfitting dengan cara secara acak menonaktifkan (drop) 30% unit (neuron) selama pelatihan. Parameter yang digunakan :

- $rate = 0.3$: Persentase neuron yang dinonaktifkan.
  Membuat model lebih general dan tahan terhadap data baru.

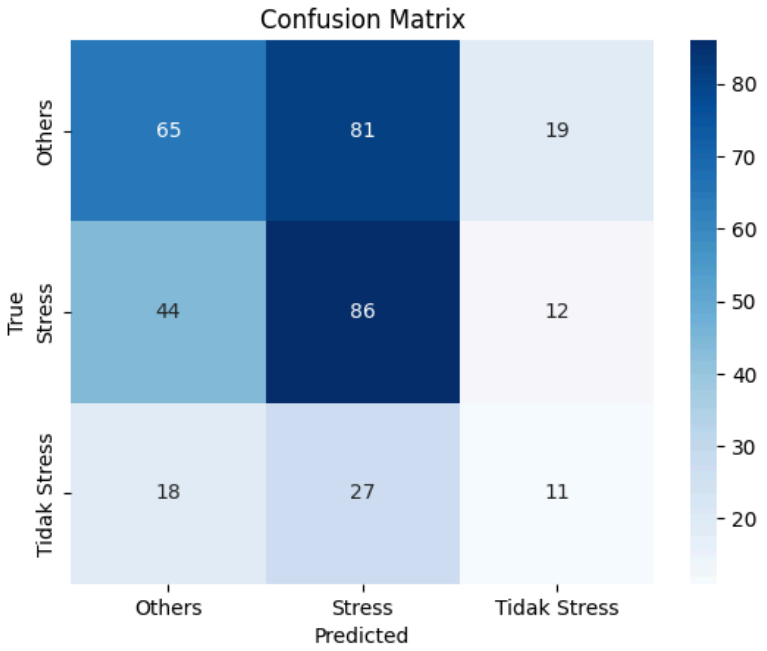
4. Output Layer (Dense)
   Fungsinya adalah menghasilkan output berupa probabilitas setiap kelas dalam klasifikasi. Parameter yang digunakan :

- $units = 2$ Jumlah kelas dalam masalah klasifikasi, yang dalam hal ini adalah 2 kelas
- $activation = 'softmax'$: Fungsi aktivasi softmax untuk mengkonversi nilai output menjadi probabilitas total 1.

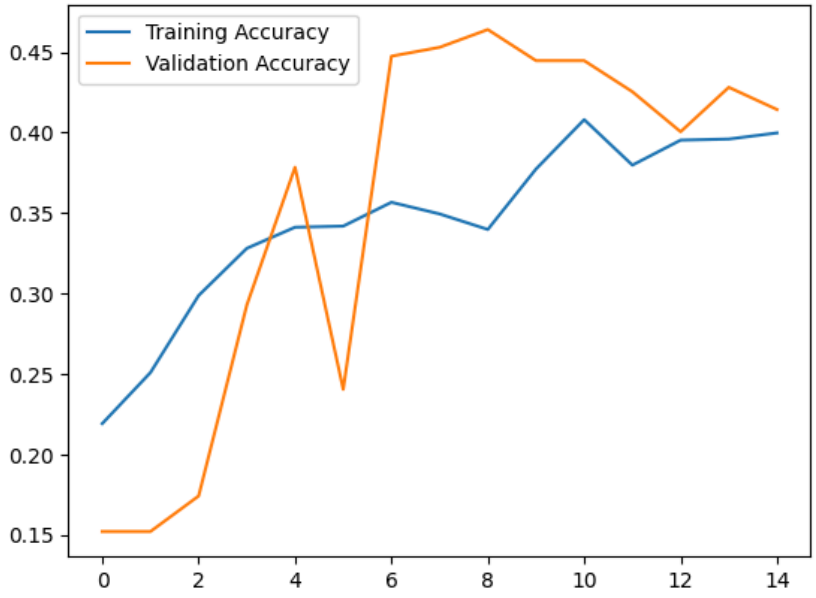
Table 43. *Classification Report LSTM Word2vec dengan Others*

	Precision	**Recall**	**F1**	**Support**
Others	0.51	0.39	0.45	165
Stress	0.44	0.61	0.51	142
Tidak Stress	0.26	0.20	0.22	56
Accuracy			**0.45**	363
Macro Avg	0.41	0.40	0.39	363
Weighted Avg	0.45	0.45	0.44	363

- Confusion Matrix LSTM pakai Word2Vec dengan Others:



  - Training dan Validation Acc dan Loss LSTM pakai Word2Vec dengan Others



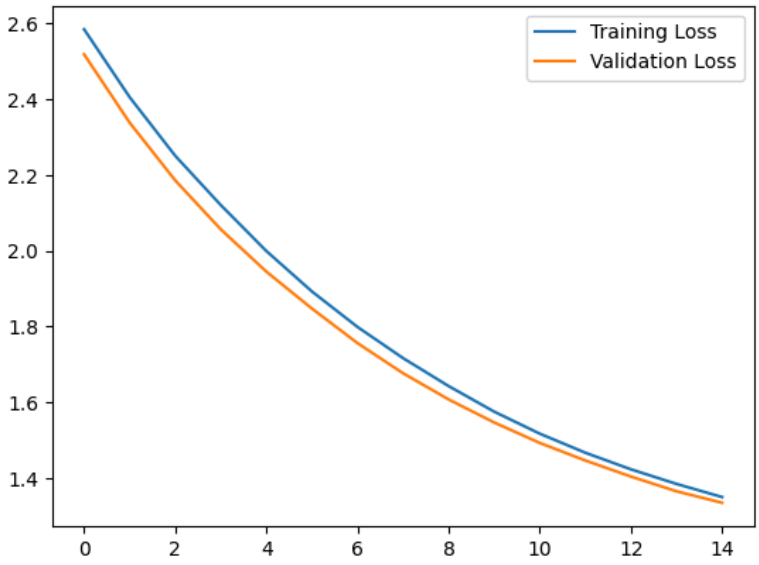


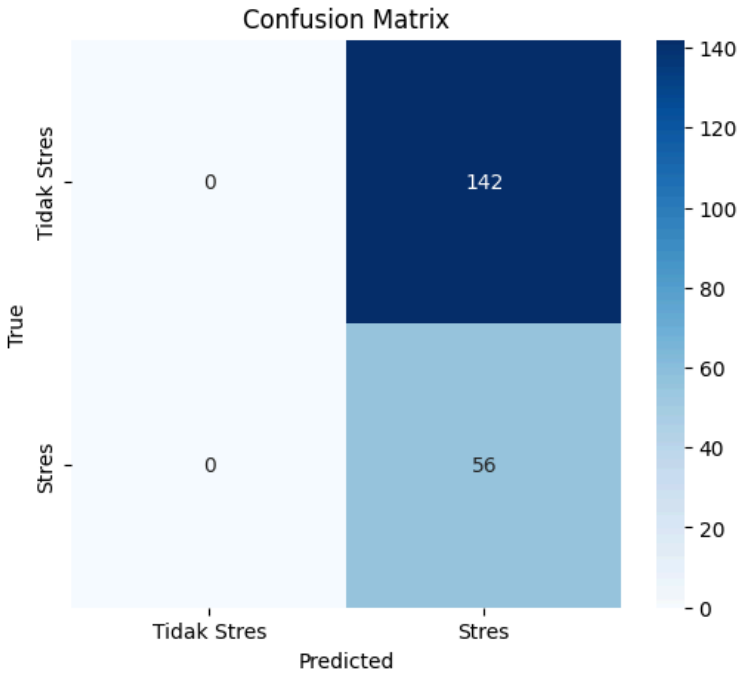
Table 44. *Classification Report LSTM Word2vec tanpa Others*

	Precision	**Recall**	**F1**	**Support**
Stress	1.00	0.00	0.00	142
Tidak Stress	0.28	1.00	0.44	56
Accuracy			**0.28**	198
Macro Avg	0.64	0.50	0.22	198
Weighted Avg	0.80	0.28	0.12	198

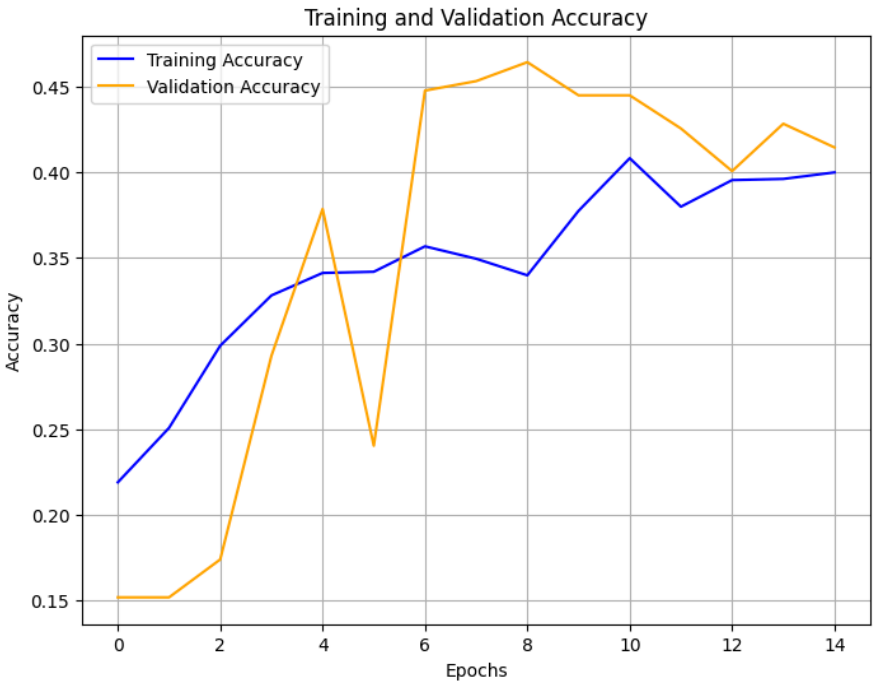
Hal diatas menunjukkan bahwa model LSTM dengan Word2Vec mengalami kesulitan dalam memprediksi salah satu kelas dengan benar, terlihat dari nilai 0.00 pada beberapa metrik evaluasi seperti precision, recall, atau F1-score. Hal ini biasanya terjadi karena data yang digunakan tidak seimbang (timpang), yaitu jumlah data pada satu kelas jauh lebih sedikit dibandingkan kelas lainnya.

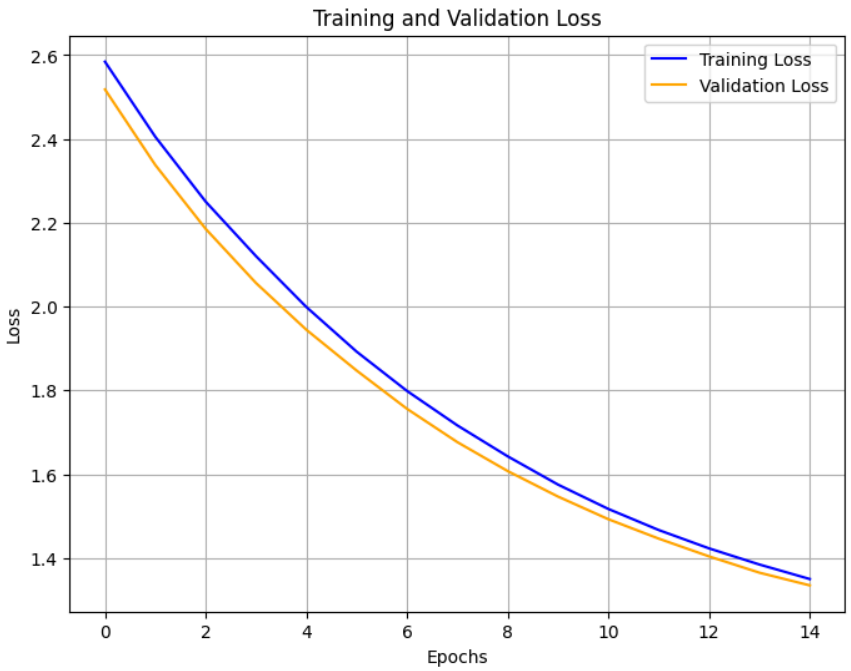
Ketimpangan data membuat model cenderung bias terhadap kelas yang lebih dominan. Untuk mengatasi hal tersebut, bisa menggunakan teknik seperti oversampling pada kelas minoritas, undersampling pada kelas mayoritas, atau menerapkan metode pengimbangan data seperti smote. Dengan cara ini, distribusi data menjadi lebih seimbang sehingga model memiliki peluang yang lebih baik untuk belajar dari semua kelas

- Confusion Matrix LSTM pakai Word2Vec tanpa Others:



- Training dan Validation Acc dan Loss LSTM pakai Word2Vec tanpa Others:





- Error Analysis LSTM
  - LSTM dengan Word2Vec

Table 45. *Error Analysis LSTM Word2vec*

full_text	Sentiment Prediksi	Sentiment Asli
welkom sudah hidup jadi sandwich generation padahal udah dari awal kerja tapiii dahlah sehat aje deh	Stress	Tidak Stress
takut salah pilih pasangan patriaki dan masalah terbesar finansial saya sandwich generation	Others	Stress
yang satunya anak papi saya yang gym membershipnya dimana dan satunya anak sandwich generation ignore time stamps	Stress	Others

Kesalahan klasifikasi terjadi karena model terlalu mengandalkan kata kunci seperti "sandwich generation" atau "finansial," yang sering diasosiasikan dengan stres, tanpa memahami konteks dan nada kalimat. Pada kalimat pertama, nada optimis dan penerimaan tidak ditangkap sehingga diklasifikasikan sebagai stres. Pada kalimat kedua, ekspresi

eksplisit tentang kekhawatiran dan tekanan finansial diabaikan, sehingga salah dianggap netral. Pada kalimat ketiga, model gagal mengenali nada bercanda, salah mengklasifikasikan sebagai stres hanya karena frasa "anak sandwich generation." Penyebabnya meliputi fokus berlebihan pada kata kunci, kurangnya sensitivitas terhadap nuansa emosional seperti humor atau optimisme, serta data pelatihan yang kurang representatif. Solusinya adalah memperbaiki dataset dengan lebih banyak contoh yang mencakup variasi nada, menggunakan model berbasis transformer seperti BERT, dan menambahkan fitur analisis nada untuk menangkap humor, sindiran, atau tekanan secara lebih akurat.

- LSTM dengan Glove

  Table 46. *Error Analysis LSTM Glove*

full_text	Sentiment Prediksi	Sentiment Asli
mungkin bisa jadi sandwich generasi juga saya tapi alhamdulillah tidak kerasa aja paling bayarin adek sekolah asuransi mama papa sama jajan adek atau misal orang rumah kepengen apa gt malah saya suka pengen banget ngasih duit jajan ke orang tua	Stress	Tidak Stress
lagian ngerawat orang tua tidak mudah dr sifat yang kadang ngeyel trus tidak bisa nemenin setiap hari karna harus kerja capek trus budaya indonesia kek gimana sandwich generation gitu	Others	Stress
tidak nanggung utang keluarga tidak jadi sandwich generation	Stress	Others

Pada penggunaan model analisis sentimen, prediksi menunjukkan perbedaan dengan sentimen asli yang ada. Pada kalimat pertama, "mungkin bisa jadi sandwich generasi juga saya tapi alhamdulillah tidak kerasa aja paling bayarin adek sekolah asuransi mama papa sama jajan adek atau misal orang rumah kepengen apa gt malah saya suka pengen banget ngasih duit jajan ke orang tua," model memprediksi Stress, padahal kalimat ini mencerminkan rasa syukur dan penerimaan, seperti pada frasa "alhamdulillah tidak kerasa aja," yang seharusnya lebih cocok dengan kategori Tidak Stress. Pada kalimat kedua, "lagian ngerawat orang tua tidak mudah dr sifat yang kadang ngeyel trus

tidak bisa nemenin setiap hari karna harus kerja capek trus budaya indonesia kek gimana sandwich generation gitu," model memprediksi Others, meskipun kalimat ini mengandung kata-kata seperti "tidak mudah," "capek," dan deskripsi tantangan yang jelas mencerminkan tekanan emosional, sehingga lebih tepat masuk kategori Stress. Pada kalimat ketiga, "tidak nanggung utang keluarga tidak jadi sandwich generation," model memprediksi Stress, padahal konteks sebenarnya adalah pernyataan netral dan menunjukkan ketiadaan beban, dengan frasa "tidak jadi sandwich generation" yang seharusnya lebih sesuai dengan kategori Others.

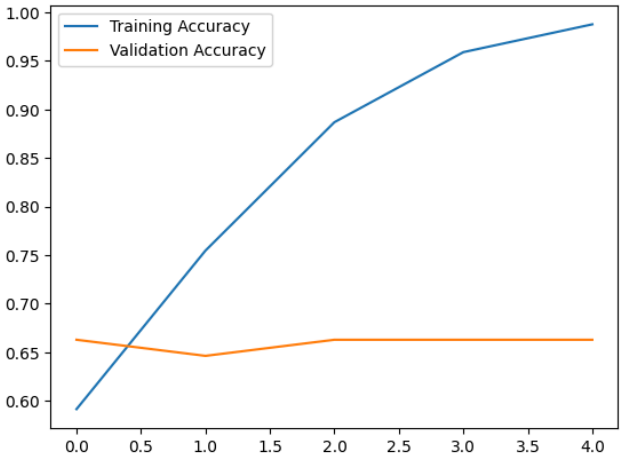
- **Pre-trained**
  - **IndoBertTweet**

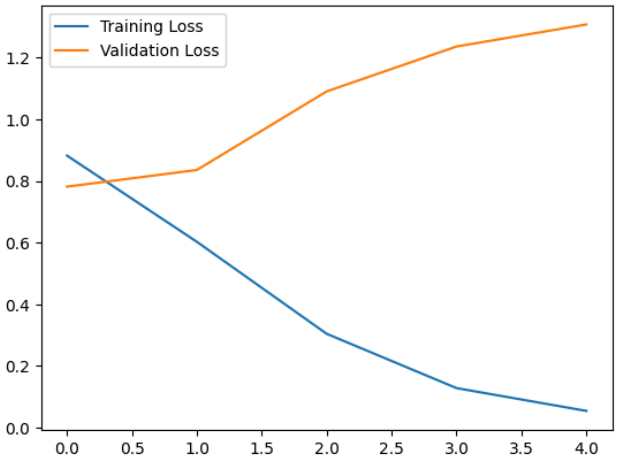
Analisis sentimen IndoBERTweet adalah metode yang digunakan untuk mengevaluasi dan mengklasifikasikan opini publik dari teks, khususnya yang berasal dari platform media sosial seperti Twitter. Metode ini memanfaatkan model IndoBERT, yang merupakan varian dari BERT (Bidirectional Encoder Representations from Transformers) yang dioptimalkan untuk bahasa Indonesia.

Table 47. *Classification Report BERT* BERT *pakai IndoBertTweet dengan Others*

	Precision	**Recall**	**F1**	**Support**
Others	0.69	0.65	0.67	165
Stress	0.72	0.74	0.72	142
Tidak Stress	0.35	0.34	0.34	56
Accuracy			**0.64**	363
Macro Avg	0.58	0.58	0.58	363
Weighted Avg	0.64	0.64	0.64	363

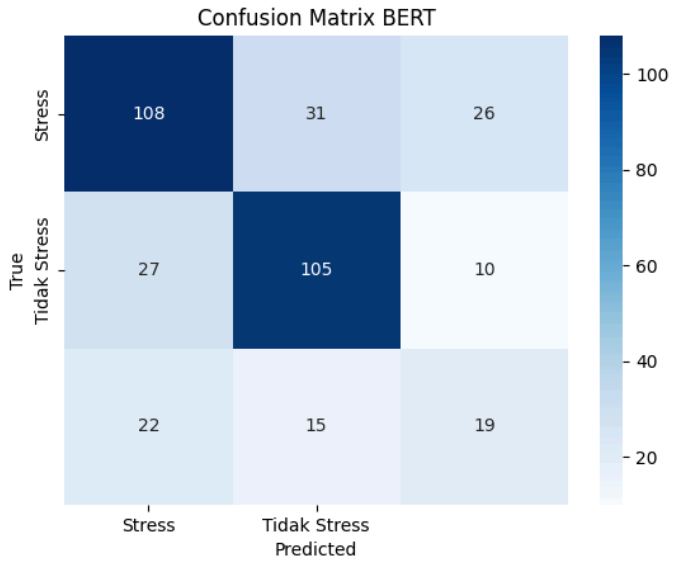
- Training dan Validation Acc dan Loss BERT pakai IndoBertTweet dengan Others





Kurva diatas mengalami overfitting, karena jumlah data latih tidak cukup banyak, model hanya bisa belajar dari contoh-contoh yang terbatas dan spesifik. Dengan demikian, model lebih rentan untuk menyesuaikan diri hanya dengan pola yang ada dalam data latih, alih-alih memahami pola umum yang berlaku di seluruh data.

- Confusion Matrix BERT pakai IndoBertTweet dengan Others:

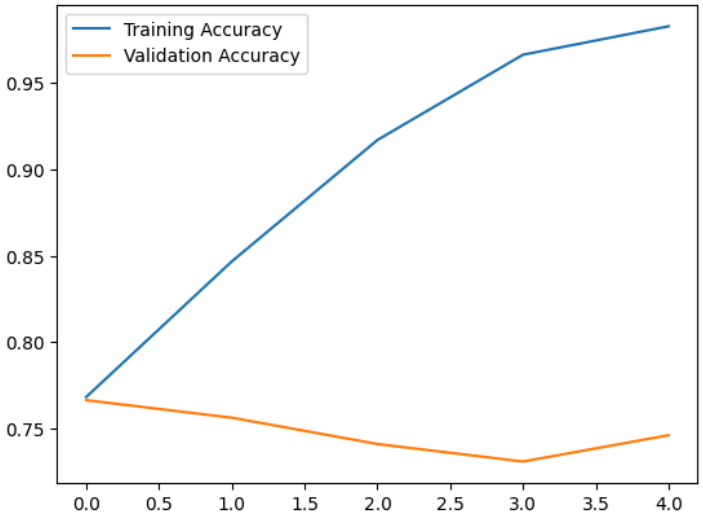


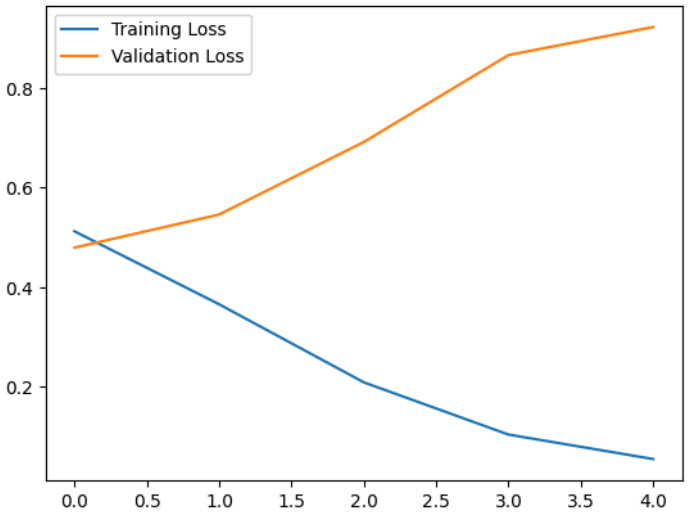
- Table 48. *Classification Report*BERT pakai IndoBertTweet *tanpa Others*

	Precision	**Recall**	**F1**	**Support**
Stress	0.88	0.86	0.87	142
Tidak Stress	0.67	0.71	0.69	56
Accuracy			**0.82**	198
Macro Avg	0.78	0.79	0.78	198
Weighted Avg	0.82	0.82	0.82	198

Karena kelas Stress lebih banyak dalam dataset, model cenderung mempelajari pola dari kelas tersebut secara berlebihan. Ini akan menyebabkan model lebih sering memprediksi kelas Stress, bahkan untuk data yang sebenarnya termasuk dalam kelas Tidak Stress. Ini mengarah pada akurasi yang tinggi untuk kelas dominan, namun rendah untuk kelas minoritas (Tidak Stress).

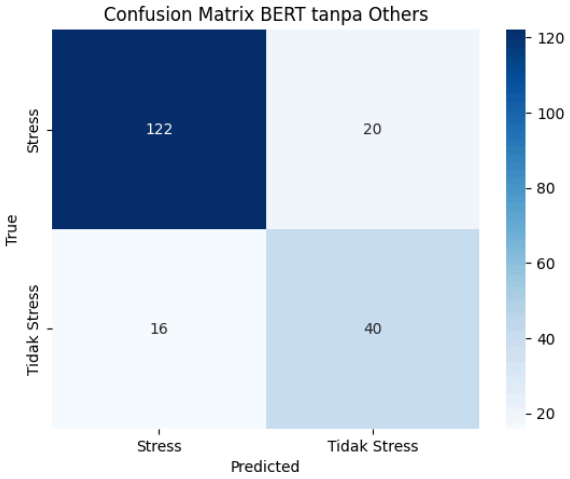
- Training dan Validation Acc dan Loss BERT pakai IndoBertTweet tanpa Others:





Kurva mengalami overfitting akibat data yang tidak seimbang antara kelas Stress dan Tidak Stres.

- Confusion Matrix BERT pakai IndoBertTweet tanpa Others:



- Error Analysis Transfer Learning

Table 49. *Error Analysis Indobertweet*

full_text	Sentiment Prediksi	Sentiment Asli
buat para sandwich generation jangan lupaa traktir diri kalian makaanan enak yang lain bisa diirit tapi perihal makan jangaan give ur body a treat	Tidak Stress	Others
stop denial jika lo emang sandwich generation saya	Others	Stress
saya sebagai kelas bawah mau childfree selama finansial belum memnuhi alasannya udah pengalaman sebagai anak yang tidak bisa kek orang mendapat pendidikan makanan bergizi dll belum lagi jadi sandwich generation dahlah nikah bukan prioritas bisa nikah syukur engga yaudah	Stress	Tidak Stress

Kesalahan klasifikasi dalam prediksi sentimen dapat terjadi karena beberapa faktor utama. Pertama, konteks sering kali tidak tertangkap sepenuhnya oleh model, terutama jika model lebih fokus pada kata-kata tertentu (keywords) tanpa memahami nada atau makna keseluruhan kalimat. Kedua, kalimat ambigu atau kompleks, seperti yang mencakup berbagai emosi atau informasi, dapat menimbulkan interpretasi berbeda.

Ketiga, ketidakseimbangan data pelatihan dapat menyebabkan bias prediksi, misalnya jika satu kategori lebih dominan daripada yang lain. Keempat, model mungkin kurang mampu menangkap hubungan kontekstual antara pengalaman pribadi, nada bicara, dan kategori sentimen. Terakhir, panjang dan kompleksitas kalimat, seperti yang memuat beberapa ide sekaligus, sering kali membuat model kesulitan menentukan sentimen utama. Untuk mengatasi ini, diperlukan dataset yang lebih representatif, model yang lebih canggih seperti berbasis transformer (misalnya BERT), serta preprocessing yang memecah kalimat panjang agar analisis lebih akurat.

Table 50. Tabel Perbandingan Akurasi dari Metode Machine Learning

		Accuracy Perbandingan							
		Dengan Others				**Tanpa Others**			
Classic Algorithm		**BoW**	**TF - IDF**	**2 - Grams**	**3 - Grams**	**BoW**	**TF - IDF**	**2 - Grams**	**3 - Grams**
	Naïve Bayers	42%	42%	44%	37%	57%	57%	68%	68%
	Logistic Regression	61%	**63%**	**58%**	**56%**	**75%**	73%	**75%**	**75%**
	Random Forest	**62%**	61%	55%	53%	72%	**73%**	72%	72%
Deep Learning		**Word2Vec**		**Glove**		**Word2Vec**		**Glove**	
	LSTM	45%		**46%**		28%		**57%**	
Transfer Learning		**IndoBERTweet**				**IndoBERTweet**			
	BERT	64%				**82%**			

- **Analisis Hasil Klasifikasi Sentimen Generasi Sandwich di Platform X**

Hasil evaluasi menunjukkan performa berbagai metode machine learning dalam klasifikasi sentimen Generasi Sandwich berdasarkan kategori tingkat stres. Analisis dilakukan pada dua skenario: dengan kategori tambahan "Others" dan tanpa kategori tersebut. Berikut analisis berdasarkan hasil yang diperoleh:

1. **Kinerja Classic Algorithm**
    - **Naïve Bayes** menunjukkan performa yang kurang optimal dibanding metode lain, terutama dalam skenario dengan kategori "Others". Namun, tanpa kategori "Others", akurasi meningkat signifikan hingga **68%**, menunjukkan bahwa model ini lebih sensitif terhadap penghapusan kategori "Others".
    - **Logistic Regression** memberikan performa yang konsisten baik dalam skenario dengan atau tanpa kategori "Others". Akurasi tertinggi **75%** dicapai pada skenario tanpa kategori "Others", menunjukkan kemampuan model ini untuk menangkap pola yang lebih kompleks dengan berbagai fitur, seperti **TF-IDF**, **2-Grams**, dan **3-Grams**.
    - **Random Forest** menunjukkan akurasi yang kompetitif, terutama dengan fitur **BoW** dan **TF-IDF**. Namun, akurasi sedikit menurun ketika menggunakan fitur n-grams, menunjukkan bahwa metode ini lebih cocok untuk fitur yang tidak terlalu kompleks.
2. **Kinerja Deep Learning**
    - **LSTM** memberikan hasil yang kurang kompetitif dibanding metode lainnya. Meskipun akurasi meningkat dari **46%** menjadi **57%** tanpa kategori "Others", model ini menunjukkan keterbatasan dalam menangkap pola teks yang lebih sederhana dibanding metode transfer learning.
3. **Kinerja Transfer Learning**
    - **IndoBERTweet** menunjukkan performa yang paling unggul di antara semua metode. Pada skenario dengan kategori "Others", akurasi mencapai **64%**, sedangkan tanpa kategori "Others", akurasi melonjak signifikan hingga **82%**. Hal ini menunjukkan bahwa IndoBERTweet mampu menangkap konteks yang lebih mendalam dalam data teks, terutama setelah kategori "Others" dihapus, yang mungkin mengurangi ambiguitas dalam data.

Kesimpulan Analisis

- Hasil analisis menunjukkan bahwa penghapusan kategori "Others" berkontribusi pada peningkatan akurasi hampir di semua metode, terutama pada model berbasis

klasik seperti Naïve Bayes dan Logistic Regression. Hal ini disebabkan oleh pengurangan ambiguitas kategori, sehingga model lebih fokus pada pola utama.

- Metode transfer learning dengan **IndoBERTweet** terbukti paling unggul, menunjukkan kekuatannya dalam memahami konteks dan representasi teks yang lebih kompleks. Hal ini menjadikan IndoBERTweet pilihan utama untuk klasifikasi sentimen tingkat lanjut.
- Algoritma klasik seperti Logistic Regression dan Random Forest tetap relevan untuk skenario dengan data yang lebih sederhana atau fitur yang kurang kompleks.

Dengan demikian, kombinasi fitur yang sesuai dan metode yang tepat sangat penting dalam meningkatkan performa klasifikasi sentimen pada data tingkat stress Generasi Sandwich.

- Checklist Laporan Untuk Bab 4

Tahap 6 (poin: 20): Knowledge Interpretation

- Pola-pola *useful* yang telah ditemukan.
    - **Penghapusan Kategori "Others" untuk Mengurangi Overfitting**
      Penghapusan kategori **Others** secara signifikan mengurangi risiko overfitting, terutama pada model klasik seperti Naïve Bayes dan Logistic Regression. Kategori **Others** sering menciptakan ambiguitas yang membuat model terlalu fokus pada pola yang tidak relevan.
      Contohnya Kalimat seperti *"Tidak tahu apakah ini stres atau bukan"* sebelumnya masuk kategori **Others**, namun setelah penghapusan, model lebih fokus membedakan kategori **Stress** atau **Tidak Stress**.

    - Contoh seperti *"roti yang pake sayur itu namanya apa sandwich generation"* sulit diklasifikasikan karena kalimat ini netral tanpa emosi eksplisit. Model sering memprediksi kategori *Tidak Stress* atau *Stress* karena BoW dan TF-IDF gagal membedakan konteks sederhana dari sentimen emosional. sedangkan Random Forest tampak bias terhadap pola kata tertentu tanpa memahami arti semantik.

    - Kesalahan prediksi pada frasa emosional seperti "takut banget" sering kali terjadi karena model tidak memiliki kemampuan untuk memahami konteks emosional yang lebih dalam. Misalnya, ketika seseorang menulis "takut banget jika nanti duit saya banyak kegocek," frasa ini jelas mencerminkan kekhawatiran finansial yang kuat dan dapat dikaitkan dengan tingkat stres tertentu. Namun, model mungkin hanya melihat kata "takut" secara terisolasi sebagai kata umum yang sering digunakan dalam konteks sehari-hari tanpa memperhatikan hubungan emosional antara kata "takut" dan situasi finansial yang disebutkan. Akibatnya, model cenderung salah memprediksi frasa ini sebagai "Tidak Stress" atau "Others," karena tidak mampu memahami bahwa keseluruhan kalimat tersebut mengandung nuansa kekhawatiran mendalam yang relevan dengan kategori stres. Keterbatasan ini mencerminkan bahwa model hanya bekerja berdasarkan pola statistik atau frekuensi kata, tanpa memahami semantik dan konteks kalimat secara menyeluruh.

    - **Konsistensi Logistic Regression dengan Penghapusan Kategori Ambigu**
      Logistic Regression mencapai akurasi tertinggi 75% tanpa kategori **Others**, menunjukkan model ini konsisten dalam menangkap pola pada fitur kaya seperti TF-IDF dan n-grams. Penghapusan kategori ambigu membantu model mengurangi bias pada data. Kalimat seperti *"Saya merasa tertekan dengan pekerjaan"* berhasil diprediksi sebagai **Stress** dengan lebih akurat.
    - **Keunggulan IndoBERTweet dalam Memahami Konteks Kategori**
      IndoBERTweet mencatat akurasi tertinggi 82% setelah kategori **Others** dihapus,

membuktikan kemampuannya dalam memahami konteks mendalam antara **Stress** dan **Tidak Stress**

- **Kelemahan LSTM dalam Menangkap Pola Sederhana**
  LSTM memiliki performa rendah dibandingkan model lainnya, terutama dalam pola sederhana. Penghapusan kategori **Others** membantu meningkatkan akurasi dari 46% menjadi 57%, meskipun masih kurang optimal. Hal ini menunjukkan model ini lebih rentan terhadap overfitting pada data ambigu.
  Kalimat seperti *"Saya cukup stres tetapi bisa ditangani"* sering salah diprediksi sebagai **Tidak Stress** karena pola teks sederhana.
- Penghapusan kategori **Others** mengurangi risiko overfitting dan meningkatkan akurasi hampir di semua model. Model modern seperti IndoBERTweet tetap unggul dengan pemahaman konteks mendalam, sementara model klasik lebih cocok untuk data sederhana tanpa kategori ambigu.
- Untuk penelitian selanjutnya, salah satu saran yang dapat diterapkan adalah penggunaan **SMOTE (Synthetic Minority Over-sampling Technique)** untuk menangani masalah ketidakseimbangan data. SMOTE bekerja dengan cara menghasilkan sampel sintetis untuk kelas minoritas, sehingga distribusi data menjadi lebih seimbang. Teknik ini sangat berguna ketika satu kelas "Stress", jauh lebih banyak dibandingkan kelas "Tidak Stress". Dengan menyeimbangkan jumlah data untuk masing-masing kelas, model dapat belajar lebih baik tentang pola pada kelas minoritas, mengurangi bias terhadap kelas mayoritas, dan meningkatkan generalisasi pada data baru.
- Memeriksa pelabelan data dengan lebih detail. Ketidaktepatan dalam pelabelan dapat mempengaruhi kinerja model secara signifikan.

Tahap 7 (poin: 15): Reporting

- Simple academic Poster

Klasifikasi Sentimen Generasi Sandwich di Platform X Berdasarkan Kategori Stress

Ardhika Yoga Pratama, Anissa Yulidha Rodiyah, Rahajeng Febri Shafiyah
rahajengfebri@webmail.umm.ac.id, anissa_yulidha_r@webmail.umm.ac.id, ardhikayp@webmail.umm.ac.id

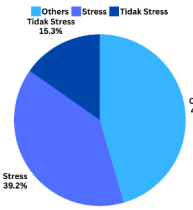
Setio Basuki, S.T, M.T, Ph.D
setio_basuki@umm.ac.id

Introduction

Generasi sandwich adalah mereka yang merawat orang tua dan anak-anak sekaligus. Dari komentar di platform X, terlihat tantangan dan tekanan yang mereka hadapi, membantu mencari solusi yang tepat.

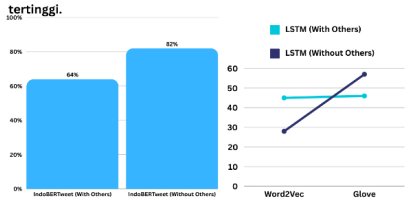
Dataset Overview

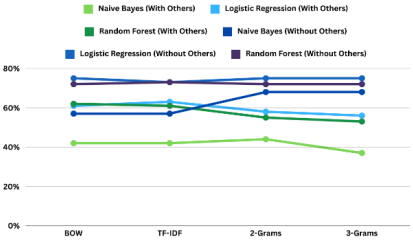
Dataset ini berisi 3.629 data yang diambil melalui crawling dari platform X, dengan tiga kategori label sentimen: 'Others,' 'Tidak Stress,' dan 'Stress,' di mana distribusi label tidak seimbang, dengan 'Others' mendominasi dan 'Tidak Stress' paling sedikit."



Pada skenario With Others, Logistic Regression dengan TF-IDF menunjukkan performa terbaik pada metode klasik, sementara LSTM dengan Glove mencapai akurasi tertinggi di metode deep learning. Selain itu, IndoBERTweet dalam transfer learning berhasil mencatatkan akurasi tertinggi di antara semua pendekatan.

Sebaliknya, pada skenario Without Others, hasil yang konsisten terlihat dengan dominasi Logistic Regression berbasis TF-IDF di metode klasik, LSTM dengan Glove di deep learning, dan IndoBERTweet sebagai metode dengan akurasi keseluruhan tertinggi.





Method

- Klasik:
  - Algoritma: Naïve Bayes, Logistic Regression, Random Forest.
  - Fitur: BoW, TF-IDF, 2-Grams, 3-Grams.
- Deep Learning:
  - Model: LSTM.
  - Representasi: Word2Vec, Glove.
- Transfer Learning:
  - Model: IndoBERTweet.
- Hyperparameter Tuning:
  - Dilakukan untuk semua pendekatan, baik dengan Others maupun tanpa Others.
- Hasil Terbaik:
  - Logistic Regression + TF-IDF (63%) dengan Others, IndoBERTweet (82%) tanpa Others.

Result

Perbandingan Akurasi dari Berbagai Metode Mechine Learning

Dengan Others

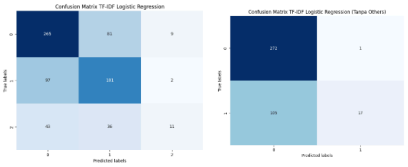
		BoW	TF-IDF	2-Grams	3-Grams
Classic Algorithm	Naive Bayes	42%	42%	44%	37%
	Logistic Regression	61%	**63%**	**58%**	**56%**
	Random Forest	**62%**	61%	55%	53%
Deep learning		Word2Vec		Glove	
	LSTM	45%		**46%**	
Transfer Learning		IndoBERTweet			
	BERT	**64%**			

Tanpa Others

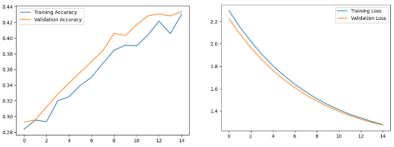
		BoW	TF-IDF	2-Grams	3-Grams
Classic Algorithm	Naive Bayes	57%	57%	68%	68%
	Logistic Regression	**75%**	**73%**	**75%**	**75%**
	Random Forest	72%	**73%**	72%	72%
Deep learning		Word2Vec		Glove	
	LSTM	28%		**57%**	
Transfer Learning		IndoBERTweet			
	BERT	**82%**			

Berikut ini adalah detail akurasi keseluruhan dari berbagai macam metode dalam mechine learning dengan fitur-fitur yang kami coba dan sari hasil ini didapatkan bahwa Logistic Regression + TF-IDF (63%) dengan Others, IndoBERTweet (82%) tanpa Others.

Visualisation



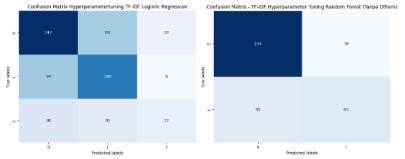
Visualisasi confusion matrix ini menampilkan evaluasi performa model dari semua metode, baik klasik, deep learning, maupun transfer learning, dengan akurasi terbaik sebesar 63% yang dicapai oleh Logistic Regression menggunakan TF-IDF with others.



Visualisasi validation epoch ini menunjukkan perbandingan akurasi dalam transfer learning dan deep learning, dengan menggunakan LSTM dan BERT. Meskipun IndoBERTwith mencapai akurasi tertinggi sekitar 82%, model LSTM dengan Glove with others menonjol dengan performa terbaik dalam hal akurasi dan stabilitas loss selama pelatihan.

Validation and Visualisation Hyperparameter Tuning with Classic Method

Hypermeter Tuning Akurasi Tertinggi (Without Others)					
		BoW	TF-IDF	2-Grams	3-Grams
Classic Algorithm	Naive Bayes	58%	57%	68%	68%
	Logistic Regression	70%	73%	**73%**	**75%**
	Random Forest	**75%**	**75%**	58%	**75%**



Hasil perbandingan menunjukkan bahwa akurasi terbaik dicapai oleh Logistic Regression dengan 3-grams dan Random Forest dengan fitur ekstraksi BoW, TF-IDF dan 3-grams sebesar 75%.

Conclusion

Analisis ini menunjukkan bahwa Logistic Regression dengan TF-IDF (kategori 'Others') mencatat akurasi 63%, dengan sisa 37% error yang disebabkan oleh kalimat tidak relevan dan sarkasme. IndoBERTweet tanpa kategori 'Others' mencapai akurasi 82%, sementara Random Forest dengan 2-grams dan TF-IDF (Tanpa Others) menunjukkan akurasi terbaik, dengan penghilangan 'Others' mengurangi bias terhadap istilah seperti 'generasi sandwich'.

- Jupyter Notebook (Python)
  - Algoritma Klasik & Deep Learning
  - IndoBERTweet